In silico drug discovery services in computing grid environments against neglected and emerging infectious diseases

N. Jacq

Numéro d'Ordre : D.U. 1715

UNIVERSITE BLAISE PASCAL

U.F.R. Sciences et Technologies

ECOLE DOCTORALE DES SCIENCES FONDAMENTALES
N°516

THESE

présentée pour obtenir le grade de
DOCTEUR D'UNIVERSITE
Spécialité : Bio-informatique

Par **JACQ Nicolas**
Ingénieur grade de master

In silico drug discovery services in computing grid environments against neglected and emerging infectious diseases

Recherche de médicaments *in silico* sur grilles de calcul contre des maladies négligées et émergentes

Soutenue publiquement le 12 décembre 2006 devant la commission d'examen.

M.	Alain BALDIT	Président
M.	Vincent BRETON	Directeur de thèse
M.	David HILL	Examinateur
M.	Martin HOFMANN-APITIUS	Rapporteur
M.	Doman KIM	Examinateur
M.	Steve LANGLOIS	Examinateur
M.	Eric MARECHAL	Rapporteur
M.	Luciano MILANESI	Examinateur

Remerciements

Je remercie chaleureusement le professeur Alain Baldit pour m'avoir accueilli au Laboratoire de Physique Corpusculaire de Clermont-Ferrand.

Je remercie vivement Monsieur Steve Langlois, directeur de projets de la société Communication & Systèmes, pour avoir accepté de soutenir cette thèse dans le cadre d'une Bourse Doctorale pour Ingénieur. Son aide, sa méthodologie et ses conseils amicaux ont été inestimables pour la réussite de la thèse.

J'adresse mes plus vifs remerciements à Messieurs Martin Hofmann-Apitius, professeur au Bonn-Aachen International Centre for Information Technology, Allemagne, David Hill, professeur à l'Université Blaise Pascal de Clermont-Ferrand, France, Eric Maréchal, directeur de recherche à l'Université Joseph Fourier de Grenoble, France, Doman Kim, professeur à la Chonnan National University, Corée du Sud, et Luciano Milanesi, chargé de recherche au CNR-Institute for Biomedical Technologies de Segrate, Italie pour l'honneur qu'ils m'ont fait en acceptant de juger cette thèse.

Je remercie du fond du cœur Monsieur Vincent Breton, chargé de recherche au CNRS, pour m'avoir encadré dans cette thèse et également accompagné dans mon travail depuis 6 ans. Je le remercie chaleureusement pour son soutien sans faille et ses conseils dans de nombreux domaines de la vie, sans lesquels cette thèse n'aurait pas été possible. Je rends hommage à son enthousiasme, sa rigueur scientifique, sa vision de l'avenir et à son amitié précieuse.

Je remercie toutes les personnes rencontrées lors de ma thèse et avec lesquelles j'ai collaboré et sympathisé. Au LPC: Pierre Reichstadt, Jean-Claude Chevaleyre. Dans RUGBI: Romain Nougarède, Jean-Philippe Roebuck, Nicolas Demesy, Jean-François Musso, Sylvain Reynaud, Fabio Hernandez, Alain Lecluse, Christophe Blanchet, Hervé Prévoteau. Dans WISDOM : Marc Zimmerman, Astrid Maass, Hurng-Chun Lee, Ying-Ta Wu, Giulio Rastelli. Dans EGEE: Johan Montagnat, Pierre Girard, Ignacio Blanquer, Geneviève Romier.

Comment ne pas remercier avec force toute l'équipe PCSV et les membres de l'association HealthGrid pour leur aide précieuse et leur amitié ! De l'équipe PCSV : Yannick Legré, Jean Salzemann, Vinod Kasam, Emmanuel Medernach, Lydia Maigne, Cheick Thiam, Ziad El Bitar, Denise Donnarieix. De l'association HealthGrid : Hélène Ruelle, Nicolas Spalinger, Nathanael Verhaegue, Pierre Bernat.

Enfin, je veux remercier ma famille pour m'avoir soutenu… Flo, pour qui je réserve mes mots… et mon Créateur.

Résumé

Les grilles de calcul sont une nouvelle Technologie de l'Information permettant la collecte et le partage de l'information, la mise en réseau d'experts et la mobilisation de ressources en routine ou en urgence. Elles ouvrent de nouvelles perspectives de réduction des coûts et d'accélération de la recherche *in silico* de médicaments contre les maladies négligées et émergentes. Dans ce contexte, la première partie de la thèse a porté sur la conception de services bio-informatiques sur grille. Ils facilitent le déploiement et la mise à jour sur la grille RUGBI de logiciels et de bases de données. La seconde partie a vu le déploiement d'expériences de criblage virtuel à haut débit sur l'infrastructure de grille EGEE. Les expériences ont démontré que les grilles collaboratives ont la capacité à mobiliser d'importantes ressources de calcul dans des buts bien définis pendant une période de temps significative, et qu'elles produisent des résultats biologiques pertinents.

Mots clefs: grille de calcul, recherche de medicaments in silico, maladies négligées, maladies émergentes, services bio-informatiques, deploiement de logiciels et de bases de données, mise à jour de bases de données, criblage virtuel à haut débit.

Summary

Computing grids are a new Information Technology offering unprecedented opportunities for collecting and sharing information, networking experts and mobilizing resources routinely or in an emergency. Grids open new perspectives to *in silico* drug discovery for the reduction of costs and the acceleration of research against neglected and emerging infectious diseases. In this context, the first part of the thesis focuses on the conception of bio-informatics services in the framework of the RUGBI grid which carry out the software and database deployment and update on grid resources. The second part focuses on the deployment of high throughput virtual screening by docking in the framework of the EGEE grid. The experiments demonstrated how collaborative grids have a tremendous capacity to mobilize very large CPU resources for well targeted goals during a significant period of time and that they can be used for producing relevant biological results in the drug discovery process.

Keywords: computing grids, in silico drug discovery, neglected diseases, emerging infectious diseases, bioinformatics services, software and database deployment, database update, high throughput virtual screening

Content

General introduction

Malaria is a major health problem for Burkina Faso's population. It is the leading cause of morbidity, hospitalization and mortality. Children under five years old and pregnant women are most at risk. In 2001, 45% of deaths for children under five years old in Burkina Faso were induced by malaria. Malaria is called a neglected disease because it affects the poor population of least developed countries and treatments are often not locally available for the population.

Avian influenza is a growing threat to world public health. Possible scenarios for the first influenza pandemic of the 21st century painted a grim picture for human health the world over, the survival of existing development projects, and the health of the global economy, with losses expected to reach around US$ 800 billion during the first year of a pandemic. Avian influenza is called an emerging infectious disease because it has existed previously but is rapidly increasing in incidence and in geographical range.

Both neglected and emerging infectious diseases are major public health concerns in the beginning of the 21st century. They demand international collaboration for early detection, epidemiological watch, prevention, and a constant search for drugs and vaccines. New drugs are constantly needed to anticipate the emergence of drug resistance as well as to cure new diseases. Actions are undertaken worldwide to fight these diseases.

Information Technologies increase the impact of these actions. Information Technology, or Information and Communication Technology, is the technology required for information managing and processing, especially in large organizations. In particular it deals with the use of electronic computers and computer software to convert, store, protect, process, transmit, and retrieve information from anywhere, anytime. Grids are a new Information Technology offering unprecedented opportunities for collecting and sharing information, networking experts and mobilizing resources routinely or in an emergency.

In silico drug discovery is one of the most promising strategies to speed-up the drug discovery process and to reduce its cost. Grids open new perspectives to *in silico* drug discovery for the reduction of costs for research and development against neglected diseases and the acceleration of research and development against emerging infectious diseases. The objective of this thesis was to explore the grid impact on the *in silico* drug discovery against these neglected and emerging infectious diseases.

In this context, two areas were investigated:

- The first area was the conception of services for grid-enabled protein structure prediction in the framework of the RUGBI grid.
- The second area was the deployment of high throughput virtual screening by virtual docking in the framework of the EGEE, AuverGrid and TWGrid grids.

Beginning in January 2003 and ending in December 2005, the goal of the French national RUGBI project was to design and deploy on the basis of free technologies and existing infrastructures a computing grid offering a set of services to analyze proteins. Today, these grid-enabled services are available through a web portal and aim to support academic biologists and small and medium life science enterprises. In this framework, two grid-enabled services to deploy and update protein structure prediction software and databases were developed to help life science scientists to access the grid. Services were developed for a broad range of bioinformatics applications but are particularly relevant for protein structure prediction in the perspective of *in silico* drug discovery.

The objective of the software and database deployment service is to allow registration, installation, modification and consultation of any biological software and database in a grid environment. The objective of the database update service is to make available the last version of a flat file database on the grid for the jobs launched. The result gives access to a ready-to-use software and updated database on the RUGBI grid.

I led the workpackage responsible for deploying bioinformatics services on the RUGBI grid infrastructure. Partners of the work package were Corpuscular Physics Laboratory of Clermont-Ferrand (CNRS-IN2P3), Communication and Systèmes company, Computing Center of IN2P3 (IN2P3) and Institut de Biologie et Chimie des Protéines (CNRS). After analyzing the requirements of the life science industrial partners involved in the project, I designed the grid-enabled bioinformatics services for academic and industrial end-users. I defined the use cases, the functional specifications and the information system. Developments were made in collaboration with the Communication & Systems company and research engineers from the Corpuscular Physics Laboratory of Clermont-Ferrand. For this reason, they are only summarized in this document.

Initiated in 2004, WISDOM is an international initiative to enable a virtual screening pipeline on a grid infrastructure against neglected and emerging infectious diseases. Virtual screening is about selecting and ranking *in silico* the best candidate drugs, i.e. the molecules which could impact the target biochemical activity. In this framework, two high throughput virtual docking experiments were deployed on public cluster grid infrastructures. They tested large chemicals libraries for their ability to interact with the target. Given the very large amount of data involved in the computation, such a large scale deployment is a stressful experiment for the grid infrastructure, which is called a data challenge.

The first large scale docking experiment ran on the EGEE grid production service in July and August 2005 against targets relevant to research on malaria. The objective was the deployment of a CPU consuming virtual docking application generating large data flows to test the grid operation and services. The second large scale docking experiment ran on the EGEE, Auvergrid and TWGrid grid production services in April and May 2006 against targets relevant to research on avian influenza. The objective was to improve the performance of the *in silico* high throughput screening environment in less than three months and to test another environment, the DIANE production system, which enables users to have efficient

and interactive control of the massive molecular dockings on the grid. These achievements demonstrated the relevance of grids for the drug discovery process against neglected and emerging infectious diseases and in enabling world-wide and multidisciplinary collaboration.

I led the grid deployment of the two WISDOM experiments. Docking experiment preparation and output analysis were leaded by Fraunhofer Institute SCAI for the first experiment against malaria, and by Genomics Research Center (Academia Sinica of Taiwan) and Institute for Biomedical Technologies (CNR) for the second experiment against avian influenza. The first step of the deployment was the design of the WISDOM experiment from the docking application requirements. Then I developed the WISDOM production system for installing, testing and deploying the application on large scale grids. The third step was the deployment: it required to manage an international grid expert staff (the Biomedical Task Force) for controlling the deployment and to interact with the different actors of the grid infrastructures (user support, grid system administrators, middleware developers, etc.). The last step was the deployment statistics analysis to report the deployment and the faults, and to improve the WISDOM production system for the next experiments. In the second experiment against avian influenza, the DIANE production system was deployed and used in parallel to the WISDOM production system by Computing Center of Academia Sinica and CERN. They analyzed the DIANE deployment statistics and the tools comparison was done jointly. For this reason, the DIANE development and deployment are only summarized in this document.

Chapter 1 introduces the present challenges raised by neglected and emerging infectious diseases. It focuses specifically on malaria and on the influenza A virus subtype H5N1. Then it discusses how Information Technologies can help to develop new strategies against them. It will focus on one of these strategies, *in silico* drug discovery, and it will highlight its relevance for neglected and emerging infectious diseases. Finally, it describes how one new Information Technology, the grid, can greatly improve the *in silico* drug discovery process.

Chapter 2 proposes several definitions for a grid. Then it focuses on the most widespread grids, the desktop and cluster grids in order to compare them. The aim is to identify the best grid environments to be used to develop *in silico* drug discovery services against neglected and emerging infectious diseases. Finally the infrastructures and the technologies of the grids used will be justified and described. They are the EGEE, Auvergrid, TWGrid and RUGBI grids.

After a summary of related works, chapter 3 presents the two grid-enabled services to deploy and update protein structure prediction software and databases in the RUGBI framework. The objectives, architecture, results, limitations and perspectives are presented. Services were made for a broad range of bioinformatics applications but are useful for protein structure prediction in the perspective of *in silico* drug discovery.

Chapter 4 introduces the requirements for a large scale deployment on the public cluster grid infrastructure EGEE. Then different grid-enabled initiatives to deploy large scale virtual screening or particle physics experiments are reported. The chapter continues with a description of the bioinformatics components of a docking experiment in order to understand how to build an efficient WISDOM production system. Finally the design of the WISDOM production system is presented. Specific issues, preparatory steps on the AuverGrid infrastructure, production system design, license management and accounting data management are detailed.

Chapter 5 describes in detail the three objectives of the first data challenge having used the WISDOM production system described in the previous chapter. Then an analysis of the large scale deployment is proposed, followed by a description of achievements in terms of scale. It reports issues related to the deployment and the monitoring of the *in silico* docking experiment as well as experience with grid operation and services. Perspectives about the biological results from the deployment, the grid-enabled virtual screening against malaria and a new data challenge against neglected diseases are finally presented.

Chapter 6 describes in detail the three objectives of the second data challenge. Then deployment strategy improvements for the WISDOM production system and the DIANE framework are presented. An analysis of the large scale deployment is proposed, followed by a description of achievements in terms of scale, focused on the comparisons of the WISDOM and DIANE production systems. It reports the effects of the deployment and the monitoring of improvements of the WISDOM production system concerning deployment stability. Perspectives stemming from the biological results from the deployment and a grid-enabled high throughput virtual screening service are presented.

Chapter 1. *In silico* drug discovery against neglected and emerging infectious diseases

1.1. Introduction

Neglected and emerging infectious diseases concern mainly the poor population. The number of deaths and clinical incidences impact world-wide opinion. Beyond the dramatic individual circumstances, these diseases affect significantly the development of the least developed countries.

Drug discovery is a major piece of the global puzzle to reduce the influence of disease and consequently the human and economic costs. New drugs are constantly needed to anticipate the emergence of drug resistance as well as to cure new diseases. *In silico* drug discovery offers a new alternative to reduce the cost of drug development and to speed-up the discovery process.

Information Technologies also open perspectives for the sharing of information between all the actors of the drug development process. A knowledge environment integrating and sharing tools, data, computing power and storage space is necessary to better exploit the different steps of the *in silico* pipeline, like protein structure prediction, or the virtual screening of large databases of molecules.

Grid technology can contribute to building such environment sharing resources and providing real-time services for urgent needs. A grid is a new Information Technology proposing a collaborative environment, based on an infrastructure of interconnected resources.

The aim of this chapter is to introduce the present challenges raised by the neglected and emerging infectious diseases, and to discuss how Information Technologies can help to develop new strategies again them. We will then focus on one of these strategies, *in silico* drug discovery and on one new Information Technology, the grid.

The chapter is organized as follows:

- After the introduction, the second section introduces the issues related to neglected and emerging infectious diseases. We will specifically focus on malaria and on the influenza A virus subtype H5N1.
- The third section highlights how Information Technologies could significantly improve the fight against these diseases.
- The fourth section focuses on a specific piece of the puzzle, drug discovery and introduces the concept of *in silico* drug discovery to highlight its relevance for neglected and emerging infectious diseases.

- The last section describes how grids can greatly improve the *in silico* drug discovery process.

1.2. Issues related to neglected diseases and emerging infectious diseases

1.2.1. Overview of neglected disease

In order to fight neglected diseases, the issues surrounding them must first be understood. In the following section, there will therefore be an overview of the issues concerning neglected diseases and a focus on progress with malaria.

Introduction

Neglected diseases are here defined as: diseases without existing treatment; diseases with old, ineffective or difficult to administer treatment; or diseases with effective treatment but which is not locally available for the population. They are often referred to as tropical diseases, endemic diseases or orphan diseases [1,2,3].

They affect the poor population of developed and least developed countries. Infants and children are particularly concerned. They are one of the main mortality and morbidity causes in the world. For instance, HIV/AIDS, malaria and tuberculosis account for 5.6 million deaths; leishmaniasis is endemic in about one hundred countries; the clinical incidence number by year for the Chagas disease is 17 million [4].

The economic impact of repeated episodes of illness and long-term disability is a major cause of underdevelopment in many countries today. The economic burden of malaria alone has cost Africa billions of dollars this decade. In addition to the cost of lost working days, the cost of treatment for repeated bouts of malaria can also be a huge burden for the poorest families [4,5].

Solutions are needed to fight these diseases that deeply impact both the countries and their population.

The search for vaccines and drugs

It is estimated that only 10% of the world's medical research is devoted to conditions that account for 90% of the global disease burden. There is thus an urgent need to develop better drugs and vaccines for diseases that are largely confined to least developed countries.

Today, no vaccines are available for malaria and AIDS and there are no prospects for vaccines becoming available soon. Drugs are the main support for disease control. But the vast majority of drugs available to date have at least one of the following drawbacks: insufficient efficacy or increasing loss of effectiveness, high level of toxicity, inaccessibility, and/or high costs [6]. Only 1% of the new drugs in the last century concern tropical diseases. This rate illustrates the lack of interest from pharmaceutical industries [7,8] and the lack of public-private partnership, with the exception of a few recent initiatives [9] presented in the paragraph below. Several factors depress the market for innovative new drugs for these diseases, including poor regulatory infrastructure in many countries, competing counterfeit

drugs [8,10,11] and the risk relating to intellectual property rights, innovation and access to essential medicines [1,11,12].

New drugs are needed however for neglected diseases like malaria where parasites keep developing resistance to the existing drugs or sleeping sickness for which no new drug has been produced for years. New drugs against tuberculosis are also needed as the treatment now takes several months and is therefore hard to manage in least developed countries.

Status

A recent increase in funding opportunities through national governments and philanthropic institutions has allowed million dollar budget projects within the framework of public-private partnerships [13,14]. Medicines for Malaria Venture [15], Global Alliance for Tuberculosis Drug development, Drugs for Neglected Diseases Initiatives and Institute for One World Health regularly succeed in identifying drug candidates, called leads, followed by clinical development. Leading international pharmaceutical groups such as Pfizer, Sanofi-Aventis and Novartis are involved in the activities of the United Nations and the WHO to combat the diseases of the poor. There are also important individual industry initiatives promoting Research and Development (R&D) for neglected diseases, such as a malaria initiative from Sanofi-Aventis, the Diseases of the Developing World Initiative from GlaxoSmithKline and the Novartis Institute for Tropical Diseases in Singapore by Novartis.

Fighting neglected diseases means not only new drug or vaccine development but also providing help to medical infrastructures in least developed countries. Information, education and communication are the key fields to promote. For instance, the access to HIV/AIDS treatment has considerably improved in Sub-Saharan Africa: 10% of the people infected with HIV/AIDS are treated with antiretrovirals today [16]; but the treatments are complex and life-long, and the drugs used for tri-therapy must be kept in a cool environment. Patient prevention activities and drug supply, storage and distribution infrastructures are being progressively deployed by national governments, international agencies or non-governmental organizations.

1.2.2. Focus on malaria

After this brief introduction to neglected diseases, this section will focus on malaria to describe in more detail the different issues.

Disease status

The number of cases and deaths due to malaria is increasing in many parts of the world. There are about 300 to 500 million new infections, 1 to 3 million new deaths [17] and a 1 to 4% loss of gross domestic product (at least 12 billion dollars) annually in Africa caused by malaria [18]. The main causes for the resurgence of malaria are the following:

- the 2 most widely used drugs against malaria, chloroquine and sulphadoxine-pyrimethamine, have been rendered useless by drug resistance in a large part of the world [19-21];

- anopheles mosquitoes, the disease vector, have become resistant to some of the insecticides used to control the mosquito population.

Figure 1 presents the world-wide distribution of malaria resistance [22,23].

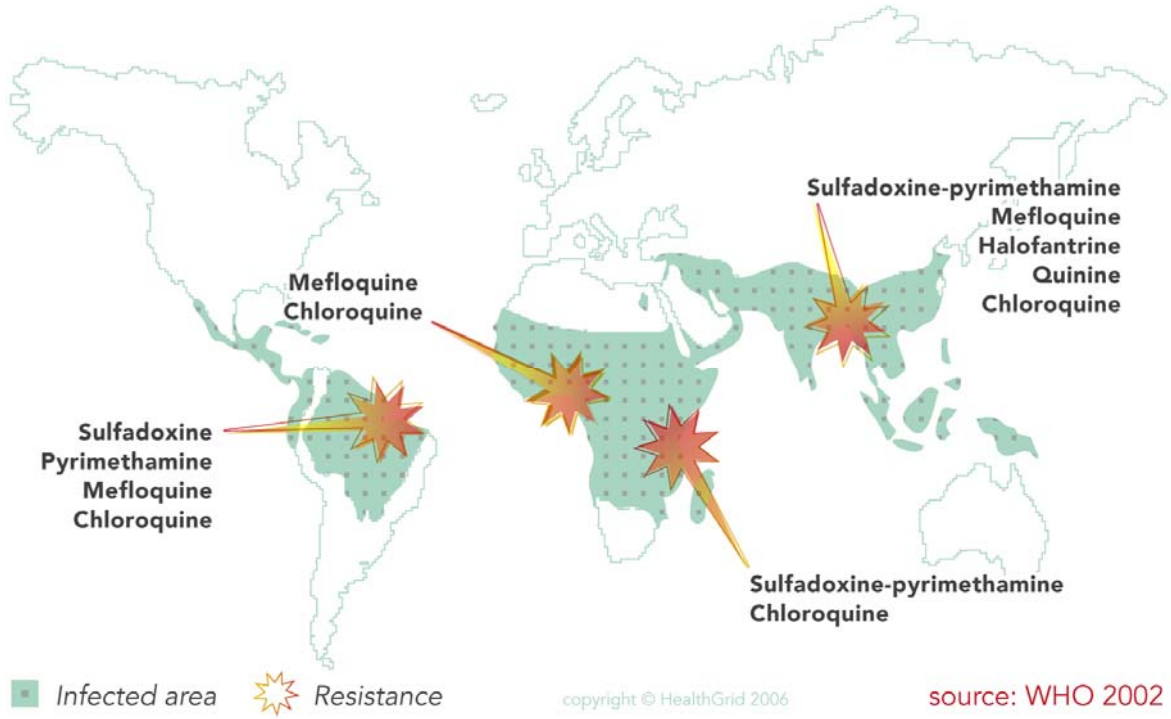


Figure 1: World-wide distribution of malaria resistance [22,23]

Current efforts focus on chemoprophylaxis using artemisinin, an antiplasmodial molecule from *Artemisia annua*, which can be produced efficiently and cheaply. However the scientific community is worried about the massive use of artemisinin. Its efficiency might be ruined by the emergence of the parasitic resistance it could trigger [24,25].

Given the small number of available drugs and the resistance they induce, discovery of new targets, the molecules on which drugs usually act, and of new drugs remains a key priority.

The search for vaccines and drugs

Understanding the disease is essential in the drug and vaccine discovery process. Malaria is caused by a protozoan parasite, plasmodium. There are several species of plasmodium infecting cattle, birds and humans. Species infecting humans are *Plasmodium falciparum*, which causes the worst clinical condition, *Plasmodium vivax*, *Plasmodium malariae* and *Plasmodium ovale*. Malaria has a complex lifecycle, presented in figure 2, involving multiple stages in mosquitoes and humans [26].

- During a blood meal, a malaria-infected female anopheles mosquito inoculates sporozoites into the human host (1).
- Sporozoites infect liver cells (2) and mature.

- Liver cells release merozoites (3). After this initial replication in the liver, the parasites undergo asexual multiplication in the blood.
- Merozoites infect red blood cells (4) and mature.
- Blood cells release merozoites (5).
- Some parasites differentiate into gametocytes, the sexual stages (6). Blood stage parasites are responsible for the clinical manifestations of the disease.
- The gametocytes, ingested by an anopheles mosquito during a blood meal (7), fuse to form zygotes, which give rise to sporozoites.
- Sporozoites make their way to the mosquito's salivary glands (8).
- Inoculation of the sporozoites (1) into a new human host perpetuates the malaria life cycle.

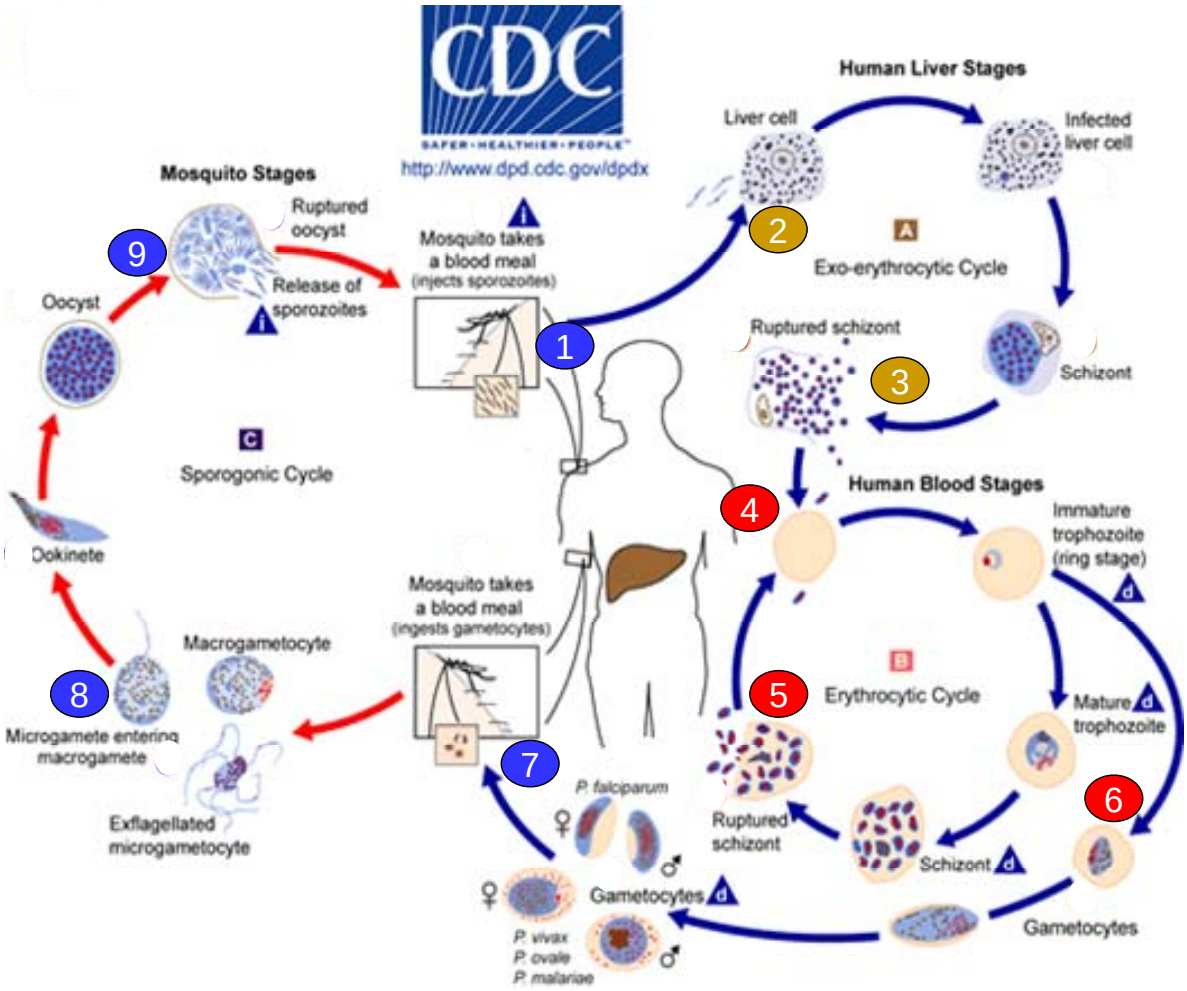


Figure 2: Diagram of the life cycle of malaria [26]

Several target proteins play a pivotal role in the life cycle of *Plasmodium falciparum*. These targets are well studied and have been validated for their significance for parasite survival as well as their "drugability" [26bis]. However, there is considerable concern in malaria so far as the available drugs focus on a limited number of biological targets.

Therefore, there is unanimity that substantial scientific effort should be devoted to the development of additional novel targets, in the hope that subsequently developed drugs would not demonstrate cross-resistance with presently known antimalarials. These potential antimalarial drug targets can be broadly classified into three categories, and each category has many individual targets. The three categories are i) targets involved in hemoglobin degradation (proteases like plasmepsins, falcipains), ii) targets involved in metabolism and iii) targets engaged in membrane transport and signaling [26ter].

For instance, plasmepsin, the aspartic protease of Plasmodium, is responsible for the initial cleavage of human haemoglobin inside the food vacuole of the parasite inside the erythrocytes (see figure 2) [463]. The significance of plasmepsins in the Plasmodium life cycle and the presence of X-ray crystal structure data make plasmepsins ideal targets for anti-malaria therapy and new approaches in rational drug design, respectively. But though several peptidic and non-peptidic inhibitors have been described as inhibitors for plasmepsins, none of them were effective in killing the parasite in cell culture. This is due to the fact that large size compounds cannot easily penetrate the food vacuole where hemoglobin degradation occurs.

To increase the chances of developing new and better drugs and vaccines, it is very important to know the sequence of the genomes of the parasites that cause malaria. The knowledge of the gene and protein expression at different stages in the life cycle and under pressure from different drugs is also essential. The availability of genomic information for humans [27,28], *Anopheles gambiae* [29], the major vector in Africa, and *Plasmodium falciparum* parasite [30] now enables the development of new approaches to interfere with the mechanism of the development of infectious sporozoites in anopheles mosquitoes and to reduce contact between infectious mosquitoes and humans. All Plasmodium molecular data have been collected and organized in the PlasmoDB public database as early as sequencing outputs have been made available [31-34]. Much effort is also being devoted to biochemical and molecular studies of resistance mechanisms and to the mapping of resistance genes. For both vector control and chemotherapy, knowing the gene sequences of anopheles and plasmodium species should lead to the discovery of targets against which new insecticides or anti-malarial drugs can be produced [35].

At the same time, at least 35 malaria vaccine candidates have undergone phase 1 clinical trials in humans [36], and 13 have moved into more advanced clinical development. Preclinical development of more than a dozen other candidates is being supported [37].

Issues

The drugs and vaccines under development are likely to be patented and only developed at prices unaffordable to governments or villagers in tropical countries. Genomics research is crucial to fight malaria, but this research must be strongly linked to ground studies in order to control the disease, especially in countries with annual health budgets of less than $10 per person [38].

Evaluation of the impact of new drugs and new vaccines requires careful monitoring of clinical tests, especially in areas of high malaria transmission where it is important to distinguish recurrence of parasites, due to recrudescence of incompletely cured infections, from re-infection due to new mosquito bites.

Disease Controllers are also faced with the necessity of monitoring goal-oriented field work on a long term basis. Monitoring tools to control malaria in plagued areas include reducing contacts between infected mosquitoes and humans by filling in breeding sites, larviciding, spraying houses with insecticides, insecticide-impregnated bed nets, and house screening. Effective application of these measures would lead to a reduction in malaria morbidity and mortality [22]. However, given the extremely high transmission rate of *Plasmodium falciparum*, especially where *Anopheles gambiae* mosquitoes predominate, the impact of these tools is limited by the capacity of mosquitoes to develop resistance and by the necessity to maintain the interventions for many years.

1.2.3. Overview of emerging infectious diseases

In order to fight emerging infectious diseases, the issues surrounding them must first be understood. In the following section, there will thus be an overview of the issues concerning emerging infectious diseases and a focus on progress with influenza A virus subtype H5N1.

Introduction

Emerging infectious diseases are here defined as infections that have newly appeared in a population or have existed previously but are rapidly increasing in incidence or in geographical or human host range [39,40]. Diseases like HIV/AIDS, SARS, Nipah virus encephalitis and variant Creutzfeldt-Jakob are newly emerging diseases. Diseases like West Nile virus in the Western hemisphere and Dengue in South America are reemerging, or resurging, diseases. Agents of bioterror, like Anthrax, are deliberately emerging diseases. Figure 3 gives examples of emerging and reemerging infectious diseases throughout the world [39].

Several factors, which frequently differ for the different diseases, are involved in the emergence of infectious diseases such as:

- economic development and land use,
- human demographics and behavior,
- international travel and commerce,
- poverty and social inequality,
- microbial adaptation and change,
- or human susceptibility to infection.

These factors ease disease dissemination, enhance infectivity and amplify pathogenicity [41]. About 75% of emerging pathogens are zoonotic [42], which means communicated by animals to humans. Avian influenza emergence is the result of interaction with animals.

Consequently there is a need for a global strategy against emerging infectious diseases that can affect many countries in the world in only a few months. The search for vaccines and drugs is difficult here, because it is a very long process.

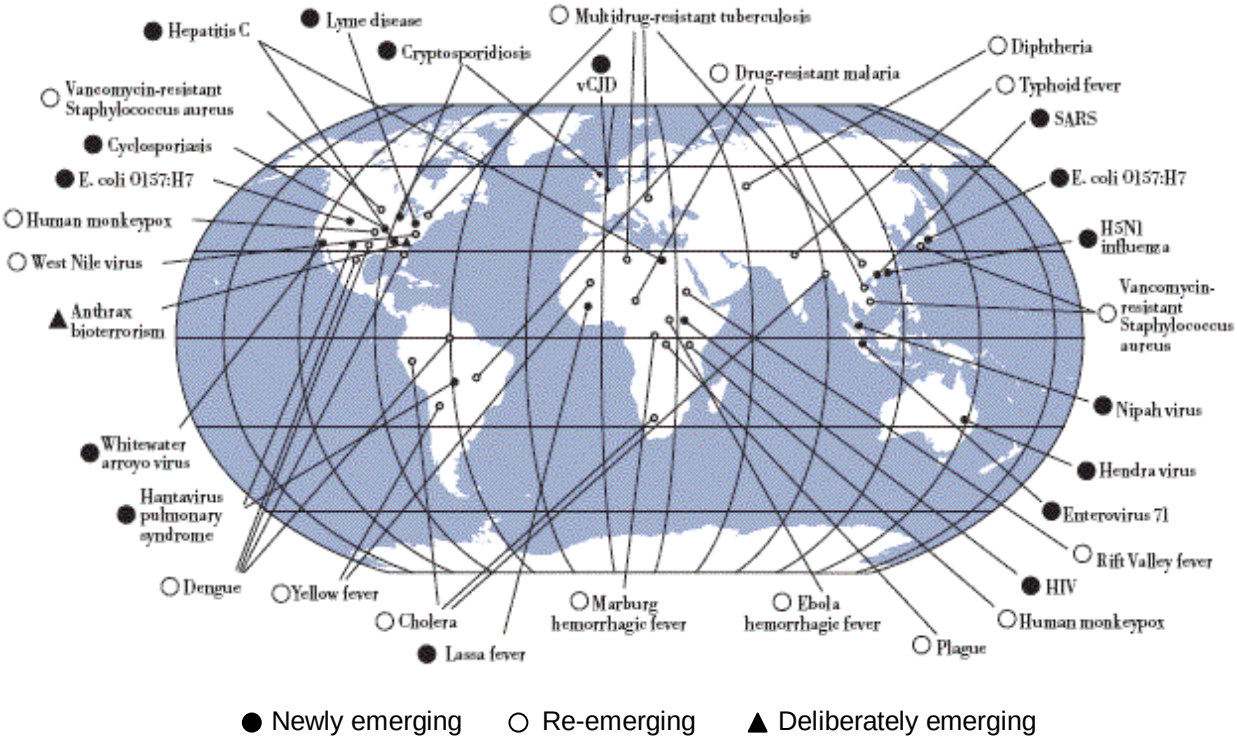


Figure 3: Examples of emerging and reemerging infectious diseases throughout the world [39]

The search for vaccines and drugs

Although vaccination is the primary strategy for the prevention of infectious diseases, there are a number of likely scenarios for which vaccination is inadequate and effective drugs would be of the utmost importance. In the case of emerging infectious diseases, time is a critical factor in the sense that there is a risk of a world-wide pandemic. However, drugs take years of development and vaccines specific to newly arising strains require several months of preparation.

Issues

Several recent health events underscore the need for a public health system ready to address whatever disease problems which might arise. For example, the emergence of SARS in Asia in 2002, and the speed with which it was characterized and contained, underscores the importance of cooperation between researchers and public health officials [43]. In view of the pandemic threat, many strategies are studied and applied around the world: surveillance and disease control in animals, surveillance and disease control in humans, development of rapid diagnostic kits, drugs stockpiling [4], public relations and education, disaster management response, etc. Moreover, in the context of a pandemic, government and health care leaders need to be prepared to make difficult decisions based on ethical values [44].

1.2.4. Focus on influenza A virus subtype H5N1

After this brief presentation of emerging infectious diseases, this section will focus on avian influenza to highlight the most relevant issues.

Motivation

Influenza A viruses have 16 H subtypes and 9 N subtypes. Only viruses of the H5 and H7 subtypes are known to cause the highly pathogenic form of the disease. Influenza A virus subtype H5N1, also known as H5N1, is capable of causing illness in many animal species, including humans [45]. A bird-adapted strain of H5N1 is the causative agent of H5N1 flu, commonly known as avian influenza, and is endemic in many bird populations. H5N1 flu killed tens of millions of birds and spurred the culling of hundreds of millions of other birds in an attempt to control its spread [46].

H5N1 causes severe disease in humans and poses an unprecedented pandemic threat [47]. Figure 4 presents the occurrence of influenza A viruses infecting humans from 1918 to 2005 [40]. The length of the arrows indicates the duration of the virus presence. The influenza A virus transmission to humans has been observed since 1997, but there have been experiences of the subtype N1 at least since 1918, when it killed about 50 million people during the so-called "Spanish flu" pandemic [48]. A risk exists for human-to-human transmission of the H5N1 virus [46,49]. Over the past few years, many more influenza strains have emerged with the capability of infecting humans.

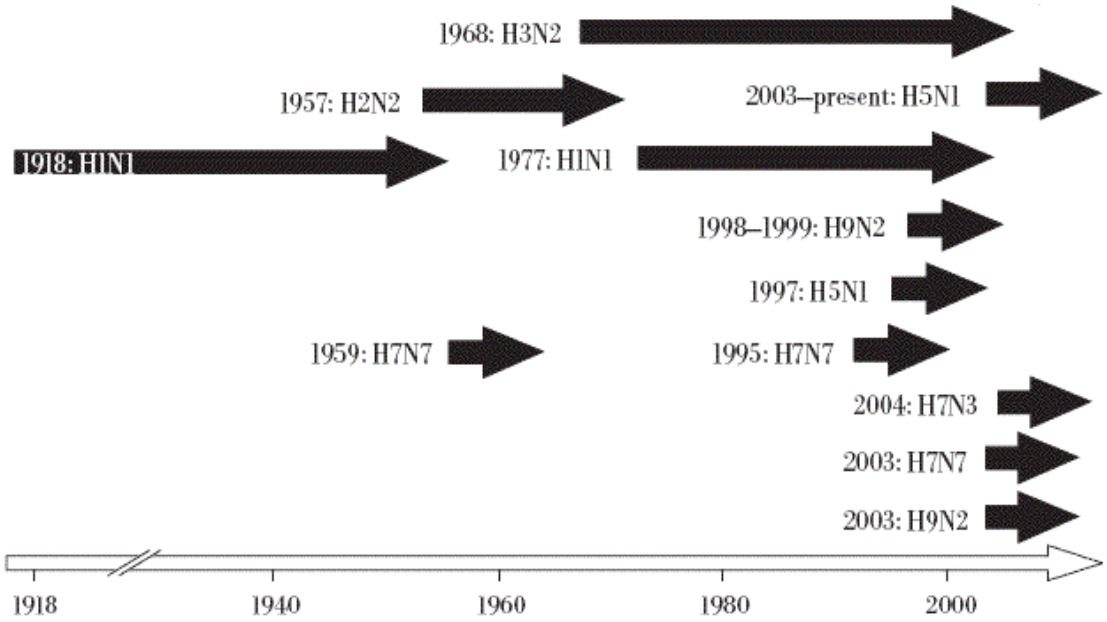


Figure 4: Influenza virus time line. Occurrence of influenza viruses infecting humans, from 1918 to 2005 [40]

There is a continuous research effort around the world to search for vaccines and drugs against all the known flu viruses.

The search for vaccines and drugs

HA is the abbreviation for hemagglutinin, an antigenic glycoprotein found on the surface of the influenza viruses. It is responsible for binding the virus to the cell that is being infected. NA is the abbreviation for neuraminidase, an antigenic glycosylated enzyme found on the surface of the influenza viruses. It facilitates the release of viruses from infected cells [50]. The hemagglutinin (HA) and neuraminidase (NA) are most medically relevant as targets for antiviral drugs and antibodies. HA and NA are also used as the basis for the naming of the different subtypes of influenza A viruses. So H5N1 comes from the H and the N.

No vaccine against the H5N1 influenza is yet available for humans or animals [51]. Even if a vaccine was available, vaccine supplies would be inadequate during a pandemic, particularly for the population of least developed countries [52,53]. For instance, the global demand for vaccines in the event of an H5N1 pandemic will be at least of 4 billion doses. But the vaccine production capacity of the world pharmaceutical companies would allow the production of no more than 100 million doses in 6 months.

Figure 5 [54] presents the life cycle of the influenza virus:

- First the individual virus enters the cell lining of the respiratory tract (1).
- The cell is induced to take up the virus because hemagglutinin on the virus binds to the sialic acid (2 and 3). The main role of the M2 ion channel is to acidify the membrane-bound virus particle while it is contained within the acidic endosome [55].
- The virus then dispatches its genetic material (made up of RNA) and its internal proteins to the nucleus of the cell (4 and 5).
- Messenger RNA is produced when some of the internal proteins duplicate the RNA (6).
- This messenger RNA is used by the cell as a template for making viral proteins (7 and 8) and genes which become new viral particles and leave the cell covered with sialic acid. This sialic acid needs to be removed so that the hemagglutinin molecules on one particle don't attach to the sialic acid on others, thus causing the new viruses to clump together and stick to the cell.
- The sialic acid is removed from the surface of the new viral particle by neuraminidase (9) and the new viral particles are able to travel and invade other cells (10).

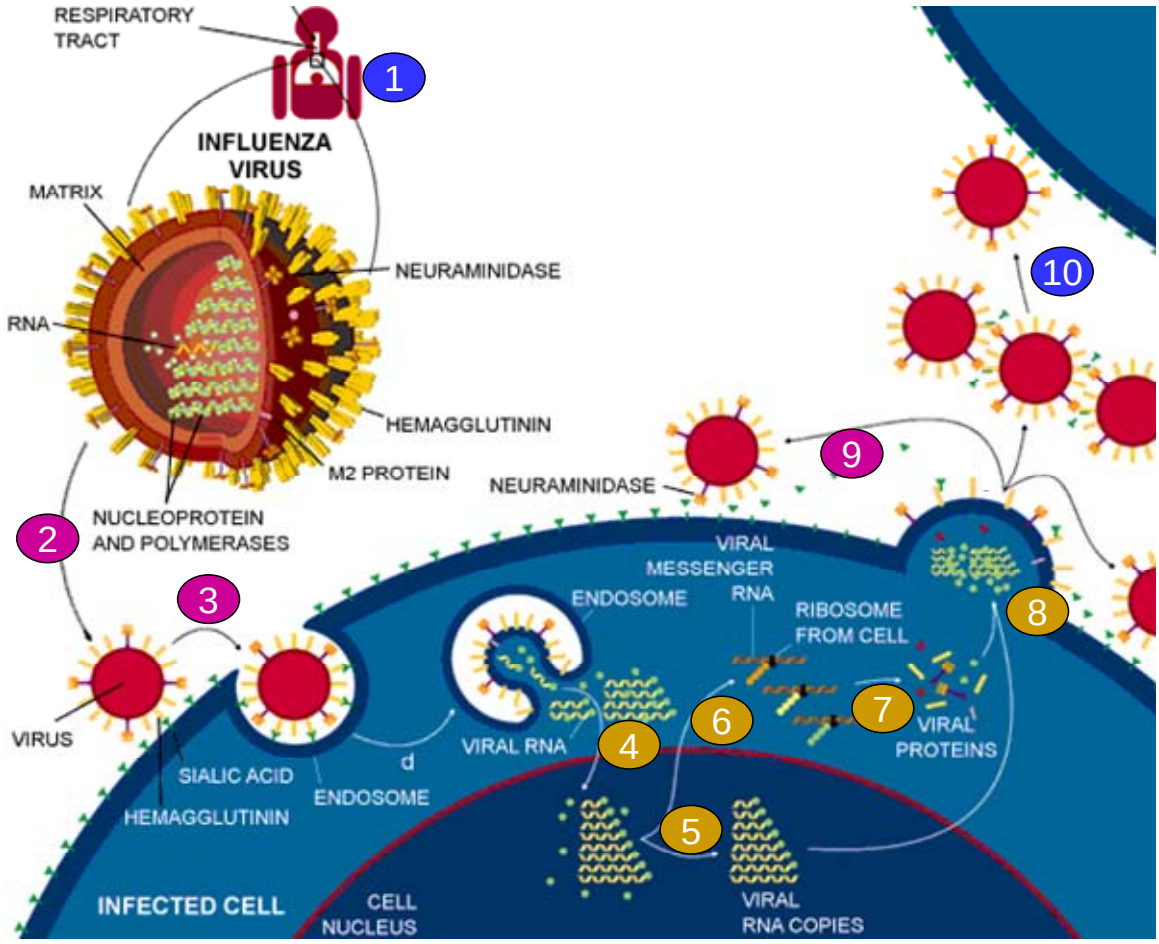


Figure 5: Schema of the life cycle of the influenza virus [54].

Two classes of antiviral agents are used to treat influenza: the M2 ion channel inhibitors and the NA inhibitors. The M2 inhibitors amantadine and rimantadine inhibit influenza A viruses but drug-resistant mutants are largely present in treated patients [56,57]. The action of NA is essential for virus proliferation and infectivity; therefore, blocking its activity generates antivirus effects. Two of the present drugs, oseltamivir (Tamiflu®) and zanamivir (Relenza®) [58] were discovered through structure-based drug design targeting NA.

Whereas zanamivir seems preferable in terms of development of resistance, the route of delivery could be problematic for some populations as it has to be administrated through oral inhalation instead of capsules like oseltamivir. Oseltamivir constitutes an important treatment option and stockpiling of this drug is part of pandemic-preparedness plans [59].

However, data on the development of drug resistance in human influenza A virus is scarce. Scientists showed that the N1 and N2 subtypes could evolve into variants under drug stress [60]. Furthermore, the report of development of drug resistance variants [61] is another potential concern. The adaptive mutation or re-assortment of the virulent avian influenza virus and the human influenza virus may lead to a more efficient form of transmission of the virus which could lead to human-to-human transmission and thus trigger an influenza pandemic.

The full support of the international scientific community is urgently needed to understand better the spread and evolution of the virus, and the factors determining its

transmissibility and pathogenicity in humans [62]. This in turn demands that scientists with different fields of expertise have full access to comprehensive genetic-sequence, clinical and epidemiological data from both animal and human virus samples. Several countries and international agencies have recently taken steps to improve sharing of influenza data [63-66], following the initiative of leading veterinary virologists in the field of avian influenza.

The search for vaccines and drugs is long and difficult so that other strategies need to be explored to avoid or prevent a pandemic.

Issues

Influenza experts and health officials throughout the world are extremely concerned about the global spread of avian H5N1 influenza and the possibility that it could lead to the next pandemic [46,67]. Initiatives have been started to establish national, regional [68] and world-wide collaborations. All countries worldwide should be able to prevent, detect early and control the virus quickly [69]. To minimize the threat from animal sources, governance, legislation, policies and resources try to be in compliance with the World Organisation for Animal Health (OIE) international standards. The veterinary services are a key actor in rapid disease control.

1.3. Potential impact of Information Technologies

In the previous section, we have described briefly the status of neglected and emerging infectious diseases in the world. Among the actions needed to better monitor and fight these diseases, many of them would benefit from a more systematic use of Information Technologies. In this section, we discuss how these technologies could have a significant impact on both neglected and emerging infectious diseases.

1.3.1. Information Technologies impacting neglected diseases

Fighting neglected diseases suffers from a lack of means. Information Technologies opens new perspectives:

- Genomics research opened new ways to find new drugs to cure malaria, vaccines to prevent malaria, insecticides to kill infectious mosquitoes and strategies to prevent development of infectious sporozoites in the mosquito [70]. Information Technologies impact these studies with tools to search and analyze huge amounts of data produced by genome sequencing. *In silico* biology supplies for instance the first steps of gene annotation via target identification or the modeling of pathways and the identification of proteins mediating the pathogenic potential of the parasite.
- Information Technologies contribute to the development and deployment of new drugs and vaccines thanks to the collection of epidemiological data for research (modeling, molecular biology) and the monitoring of ground studies to control malaria and to control the clinical tests in plagued areas. Efficient *in silico* tools are required to

manage data in geographically distributed healthcare centers. These tools need to be adapted, according to local conditions such as existing infrastructure.

- Information Technologies impact disease monitoring by the control of the policy impact and programs, but also thanks to monitoring and warning systems to detect and prevent abnormal disease evolution. Thus information from health centers, such as hospitals or dispensaries, could be collected and analyzed possibly in real time. Information about drug delivery and vector control could be monitored at the same time.
- Helping the medical development of least developed countries involves improving the ability of southern countries to undertake health innovation. Information technologies strengthen the integration of their life science research laboratories in the world community by interconnecting southern and northern health centers. Sharing knowledge and expertise are possible with teleconsulting, telediagnosis, patient follow-up and e-learning, providing access to resources, including computing and storage resources [71]. High speed Internet network access in health care centers and laboratories is the first requirement to access international resources.

Thus, Information technologies impact international collaborations through *in silico* support, data collection, monitoring system and resource sharing. This list of contributions addressing neglected diseases would greatly benefit from computing and data management in an adapted collaborative environment.

1.3.2. Information Technologies impacting emerging infectious diseases

Time can be a critical factor in fighting emerging infectious diseases. Information Technologies can help with the following actions:

- Early detection enables surveillance and disease control in domestic and wild animals, surveillance and disease control in humans, and rapid and transparent notification. Information Technologies contributes to these actions by data collection and data analysis tools used in a distributed model. Local, national and international chains of command involving health care centers need to be appropriate to identify and respond to infectious disease threats. Subjacent infrastructure and Information Technologies tools can support rapid information transfer in the different nodes.
- After a possible urgent intervention, information technologies support expert and efficient collaboration in the world and computing resources required by trend analysis and prediction made by epidemiology. These studies impact on evaluation, capacity building and strengthening, and effective countermeasures.
- Information Technologies impact anticipation and the preparation needed for essential medical supplies and equipment, and for pandemic response. Training and guidelines are required for health-care professionals and are easily available on web sites or by email. Information Technologies can contribute to the prevention by education required for people concerned by pandemic risk.

- Emerging infectious diseases have evolved into a crisis situation partly because of inadequate knowledge and understanding about the diseases. As soon as a new disease emerges, partnerships must be developed based on Information Technologies to conduct basic, applied and clinical research, to manufacture vaccines and drugs to prevent and treat disease, and to deliver these therapies to the patients who need them. *In silico* drug discovery with effective computing power contributes to speeding up the research, to fostering the necessary collaborations and to improving the quality of the output.

In case of a pandemic risk, international collaboration among clinicians, researchers, government and industry is highly and quickly required for early detection, epidemiological watch, prevention, and a fast search for new drugs and vaccines. This list of contributions addressing emerging infectious diseases would greatly benefit from an efficient collaborative infrastructure accelerating actions and quickly discovering solutions.

1.4. Developing *in silico* drug discovery

In this section, we are going to focus on one of the most promising contributions of Information Technologies to drug development, namely *in silico* drug discovery.

1.4.1. Introduction to drug discovery

Drug discovery is defined here as process by which drugs are discovered and/or designed. Drug candidates are inputs to the drug development process. Drug development manages preclinical safety studies and clinical phases, with clinical trials. Registration and delivery are the last steps of the full process. Drug discovery and development represent a complex multi-phase (12-15 years) and multi million-dollar process (at least $800 million) [72-74].

There are currently more than 200 major pharmaceutical companies. As in some other industries, economic pressures are forcing pharmaceutical companies toward greater efficiency [75]. Only 1 New Chemical Entity (drug candidate obtained by the drug discovery process) in 10,000 becomes a product [76] after preclinical evaluations and clinical trials. Biopharmaceutical properties such as oral bioavailability and formulation issues are responsible for about 39% of failures, whereas toxicity constitutes about 21%. These factors are as important as lack of efficacy, which is responsible for about 29% of failures [77-79]. Consequently biopharmaceutical properties and toxicity factors must be taken into account as soon as possible in the drug discovery process.

Reducing the research timeline in the discovery stage and having enhanced information about the leads are key priorities for pharmaceutical companies worldwide. Collaborations with academic laboratories and small biotechnology or pharmaceutical companies are crucial, mainly in exploratory research, then in the chemistry stage and progressively less during the drug development phases [9].

The diversity and complexity of the information required to arrive at well-founded decisions based on both scientific and business criteria is remarkable and well-recognized in

the industry. Thus the pharmaceutical R&D enterprise presents unique challenges for Information Technologists and Computer Scientists [80]. All aspects of managing, sharing and understanding this information by multidisciplinary project teams is critical to the R&D process and subject to substantial investment and exploration of new Information Technology approaches [81].

This section will describe *in silico* drug discovery, or computer-aided drug discovery. We will particularly focus on two important steps of the process: protein structure prediction and virtual screening.

1.4.2. *In silico* drug discovery overview

Motivation

The drug discovery goal is to find new molecules that bind with specific macromolecules, known to play a key role in the disease evolution, in a manner that changes their function for the benefit of life.

Recent progress in genomics, transcriptomics, proteomics, high throughput screening, combinatorial chemistry, molecular biology and pharmacogenomics has radically changed the traditional physiology-based approach to drug discovery where the organism is seen as a black box. The approach is now to understand how disease and infection are controlled at the molecular and physiological level and to target specific entities based on this knowledge.

In silico drug discovery is one of the most promising strategies to speed-up the drug discovery process. It is important to know and control the *in silico* process, that is described below.

Process

Figure 6 shows the different phases of a drug discovery process [82-86] with their approximate duration [87], their success rate [77-79] and the corresponding *in silico* contributions.

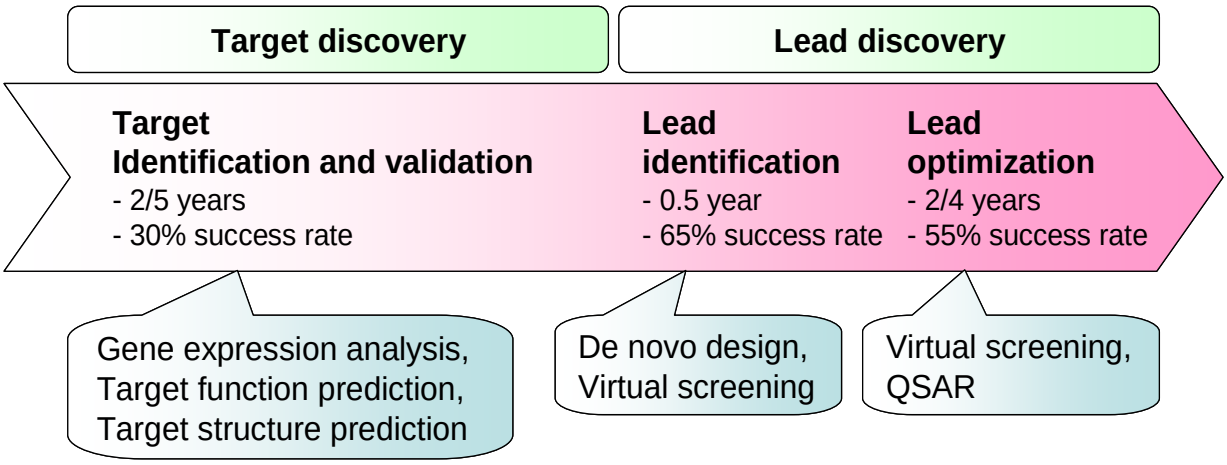


Figure 6: Representation of the different phases of the drug discovery process with their duration, their success rate and the corresponding *in silico* contributions.

A target is a cellular or genetic molecule which is believed to be associated with a desired change in the behavior of diseased cells and on which drugs usually act. The target identification and validation aims to isolate and select it. *In silico* drug discovery contributes to the target discovery by gene expression analysis, target function prediction and target three-dimensional (3D) structure prediction for post-processing.

To identify a lead compound, a substance affecting the target selected in a drug-like way, two different *in silico* pipelines can be used which speed up the process and reduce costs avoiding useless *in vitro* tests [88]: the *de novo* design and virtual screening. *De novo* design [89] builds iteratively a compound from the structure of a protein active site. Virtual screening selects *in silico* the best compound from a molecule database.

Lead optimization addresses the development from the most promising lead compounds to a safe and effective drug. Instead of expensive and longer *in vitro* and *in vivo* tests, evaluation of the basic chemical properties can be achieved by virtual screening and Quantitative Structure Activity Relationship (QSAR). QSAR is a quantitative correlation process of chemical structure with well-defined methods, such as optimization for pharmaceutical properties (Absorption, Distribution, Metabolism, Excretion and Toxicity (ADMET)) or efficacy against the target organism [78,90].

Status

In silico drug discovery contributes to increasing biological system knowledge [91], to managing data in a collaboration space, to speeding up analysis and consequently increasing the low success rate of the traditional "wet" approach. The efficiency gains of such an integrated knowledge system could correspond to a saving of 35% costs, or about US$300 million, and 15% time, or two years of development time per drug [92].

Nevertheless, in spite of increasing levels of investment in *in silico* techniques, there is a steady decline in the number of new molecules that enter clinical development and reach the market [93-95]. Many factors have changed over the past 10 years [96-99], particularly the domination of the target-based drug discovery paradigm [100-101], favoring screening and rational drug discovery programs. A new approach aims to integrate rational drug discovery with a strong physiology and disease focus [102].

Protein structure prediction and high throughput structure-based virtual screening are two important sets of methods for the *in silico* process. Improving the quality and the quantity of analyzed data during these steps will benefit the next steps in drug development. These methods are described below in order to understand their requirements.

1.4.3. Focus on the protein structure prediction

The target in the drug discovery process is frequently a protein. Understanding the function and the physiological role of this target is fundamental for the discovery of novel drugs and protein-based products with medical, industrial or commodity applications [103]. Protein structure modeling is thus crucial for the next lead discovery step: virtual screening. Protein structure prediction is introduced below.

Motivation

Time-consuming and relatively expensive X-ray crystallography and nuclear magnetic resonance (NMR) spectroscopy [104] determine experimentally physical molecular structure, but these methods are not widely applicable and sometimes inaccurate [105]. Only 1 in 20 proteins form useful crystals for structure studies [106]. Furthermore, massive amounts of protein sequence data may be derived from large-scale DNA sequencing efforts such as the Human Genome Project [27,28]. Currently, the sequences of over a million proteins have been discovered using molecular biology, but only approximately 30,000 experimentally studied structures are known [107]. Protein structure prediction attempts to bridge this growing gap [108].

Protein structure prediction aims to determine the 3D structure of a protein from its amino acid sequence. In relation with experimentally determined structures [109,110], this complex process allows the identification of the structure-function relationships of protein families, the function of an individual protein, or provides disease target structures for drug design [111].

The pipeline is detailed in the next section.

Process

A wide range of approaches are routinely applied for such predictions: *ab initio* prediction, fold recognition, and homology modeling [112-115]. These methods are primarily based on sequence alignment and also possibly on secondary structure prediction methods.

Ab initio [116], or *de novo*, protein modeling methods seek to build 3D protein models "from scratch", i.e. based entirely on physics and chemistry laws rather than on previously solved structures [117]. Solved structures are generally stored in the Protein Data Bank [118] (PDB). Local sequence, structural relationships and secondary structure prediction are incorporated into the prediction process [119].

Fold recognition [120], or protein threading, aims to find the best set of structures, or templates, in the PDB which are similar to a given target sequence. The most accurate models are obtained when a single template can be found in the PDB that has a high sequence similarity to the target protein.

Homology or comparative modeling [112] refers to the process of building a model from one such unique template. Because a protein's fold is better conserved in the evolution than its amino acid sequence [121], a target sequence can be modeled with reasonable accuracy from a very distantly related template.

Dynamic analyze of protein structure appears as a new computational method. It is increasingly common to model structure variation with time, otherwise known as 4D structure modeling. Structure-driven screenings have resulted in successful hit rates of about 10%, in stark contrast with hit rates of about 0.01% with conventional high throughput screening techniques [121].

Figure 7 [122] presents the modeling pipeline from a protein sequence to the 3D protein model (in yellow). Each stage (in pink) described above involves many different databases

(Uniprot, PDB, Prosite etc.) and tools (Blast, Clustalw, PHD, Modeler, Scope, Rosetta etc.). Bioinformatics techniques are used to analyze sequence, structure and physical-chemical studies of the conformational energies and dynamics of proteins [113,114]. The stage outputs, manually checked, determine the next steps (in green).

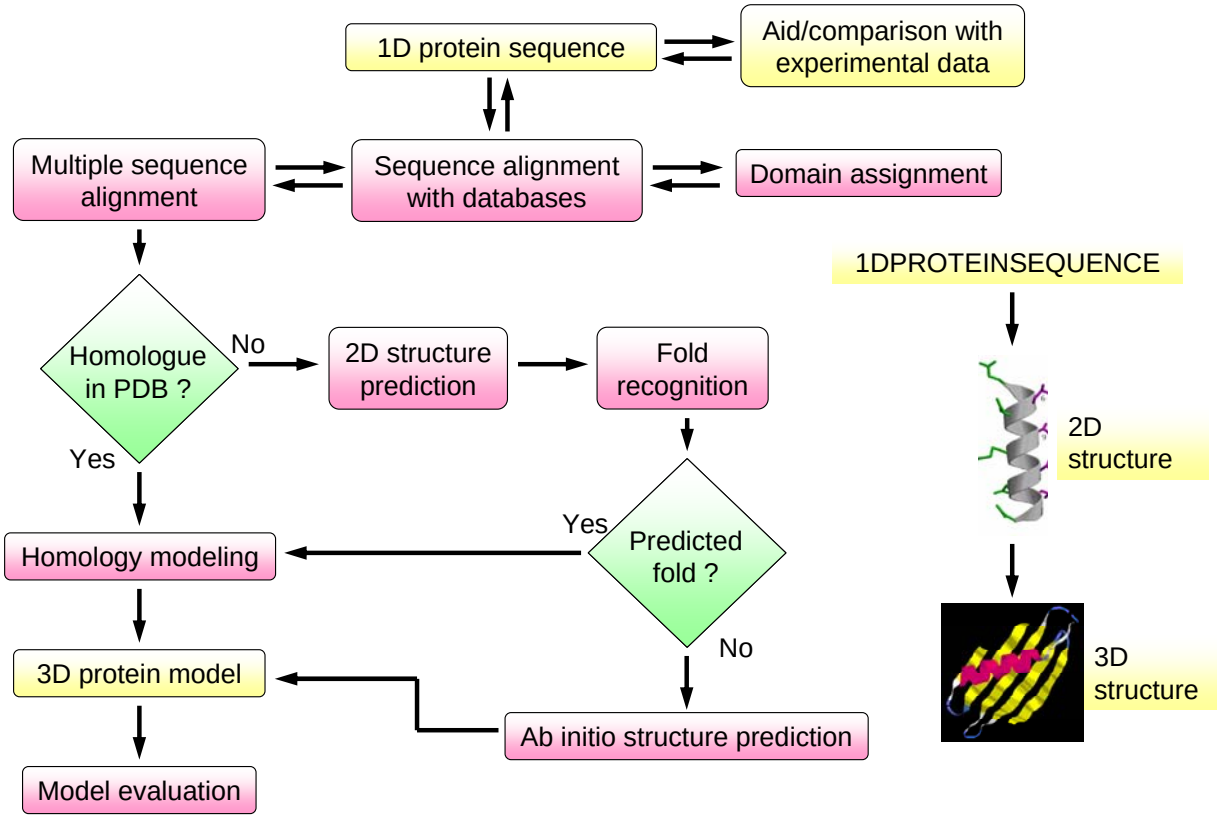


Figure 7: Protein structure prediction pipeline

This pipeline shows the difficulties in obtaining a good protein structure prediction. A progress report about research difficulties and advances follows.

Status

Several factors make protein structure prediction a difficult task, including:

- the protein size,
- the large number of possible structures for one protein,
- the multiple possible protein conformations depending on its environment,
- the fact that the biologically active conformation may not be the most thermodynamically favorable.

Generally low accuracy comparative models are based on less than 30% sequence identity [110,124,125]. Complex structures are difficult to obtain because of several technical problems that will probably take years to overcome [126]. But the great structural successes now almost invariably involve large macromolecular complexes [127-130]. The wealth of structural data from structure-genomics initiatives [131-134] gives greater applicability to homology modeling [135,136]. In some cases, structural bioinformatics using computational

technology in combination with experimental technologies allows the modeling of a protein's structure and the production of drug targets in as little as 60 days [137].

There is a wide range of applications for protein structure models for drug discovery [138]. Protein structure is used for target identification and selection, for the identification of hits by virtual screening, for the screening of fragments and for lead optimization [139]. Several examples are listed here:

- Accuracy comparative models are frequently helpful in refining functional predictions that have been based on a sequence match alone because compound binding is more directly determined by the structure of the binding site than by its sequence.
- The size of a compound may be predicted from the volume of the binding site cleft, and the location of a binding site for a charged compound can be predicted from a cluster of charged residues on the protein. Fortunately, errors in the functionally important regions in comparative models are often relatively small because functional regions such as active sites tend to be better conserved in the evolution than the rest of the fold.
- Accuracy comparative models may be useful for assigning the fold of a protein. Fold assignment can be very helpful in drug discovery, because it can cut short the search for leads by pointing to compounds that have been previously developed for other members of the same family [140,141].

Many tools are commercialized or available online [142]. Automation makes comparative modeling accessible to both experts and non-specialists alike, but technical aspects, such as the delay in waiting for the server response limits the number of components that can be included and increases the number of software developed in-house [113]. Many of the servers are tested at the bi-annual CAFASP meetings and continually by the LiveBench and EVA [143-147] web servers for assessment of automated structure prediction methods. However, in spite of automation, manual intervention is generally still needed to maximize the accuracy of the models. The modeling methods tend to require vast computational resources, and thus have only been carried out for tiny proteins. There are distributed computing projects that attempt to solve the protein prediction problems [148-150].

1.4.4. Focus on high throughput structure-based virtual screening

Once a molecular target has been identified, high throughput screening can be applied to experimentally identify interesting compounds, called ligands. A complementary method is to apply ligand-based or structure-based virtual screening to screen *in silico* a database of molecules. The next section will focus on the high throughput virtual structure-based screening that is an important step in the discovery of leads for drug development.

Motivation

In the first step of high throughput screening, several bioassays are set up and several hundreds to thousands of compounds are tested. Then the active compounds are identified and

their potency is estimated. In the final step their chemical structure and mechanism of actions are determined. Screening has become a highly sophisticated, automated modern technique [151-153] that screens hundreds to thousands of compounds against a target in a few weeks and subsequently analyzes the data to identify novel leads. Active compounds are characterized by IC_{50} values lower than 10 μM [139] where IC50 is the *in vitro* concentration of an inhibitor that is required for 50% inhibition of a target. Lead optimization is needed to lower this value below 10 nM.

There are now millions of chemicals that can be synthesized. But it is very expensive to screen such a high number of compounds in the experimental laboratories using high throughput screening techniques. Besides the heavy costs and the *in vitro* limitations (for instance, the low solubility of certain molecules), the hit rate in high throughput screening is quite low, in the range of 1 per 100,000 compounds when screening is done with targets such as enzymes [154,155]. In addition, screening does not tell why and how the detected hits act upon the target [156]. Rather than performing a large, costly, high throughput screen, a more focused screening campaign is often more appropriate for the most tractable targets for which there is information about compounds and protein structure [139]. Compounds for such a campaign can be processed through a virtual screening pipeline.

Virtual screening is about selecting and ranking *in silico* the best candidate drugs, i.e. the molecules which could impact the target biochemical activity. It is used for instance as a filter for removing the toxic compounds which are likely to fail in the final stages of the drug discovery process. High throughput virtual screening tests large chemicals libraries for their ability to interact with the target. Millions of chemical compounds available in the laboratories or companies are also recorded in 2D or 3D electronic databases. If high throughput screening was carried out *in silico* in a reliable way, one could reduce the number of molecules requiring *in vitro* and then *in vivo* testing from a few million to a few hundred.

The tools available for performing the computational analyses can be categorized as ligand-based or structure-based. The ligand-based, or similarity-based, strategy is to use information provided by a molecule or a set of molecules that are known to bind to the desired target and to use this to identify other compounds in a database with similar properties [157-161]. When the target structure is known, structure-based, or receptor-based, methods are preferred [162,163]. This second method is detailed below.

Process

Once a target structure is available, the identification of chemical starting points for lead optimization can be achieved through virtual docking, [164-168]. These approaches require a detailed scoring of the interactions [169] between the target and the ligands. Then the use of molecular dynamics and molecular mechanics computations improves the quality of *in silico* scoring [170]. All these methods are explained below.

Figure 8 describes a typical workflow of a structure-based virtual screening run against a specific target and a compound database (in yellow) [171,172]. It consists of many steps (in pink), possibly in repeated automated cycles, or with visual check points (in green).

The first step consists in filtering and preparing a compound database for the docking step. Libraries of compound 3D structures are made openly available by academic laboratories or by chemistry companies which can produce them. In the random approach [173], compounds for docking are selected from a compound database with a large variety and according to their drug likeness [77,174] or pharmacokinetic properties [175,176]. In the focused, or directed, approach, the selection of compounds is made using a set of criteria, e.g. by pharmacophore or docking models [165,166,177-179].

At the same time as the first step, the active site of the protein needs also to be prepared for the docking tool (adding charges, hydrogens, water molecules…) [171]. All available target structures can be used to generate only one composite structure, or the screening process can be conducted on each individual structure requiring more computing time.

The next step is molecular docking. Protein-ligand docking is about computing the binding energy of a protein target to a library of potential drugs using a scoring algorithm. The target is typically a protein which plays a pivotal role in a pathological process, e.g. the biological cycles of a given pathogen (parasite, virus, bacteria…). The goal is to identify which molecules could dock on the protein active sites in order to inhibit its action and therefore interfere with the molecular processes. Many docking software codes [180-187] are available under a variety of licensing policies. They differ in the sampling algorithms used, the handling of compound and protein flexibility (Many docking algorithms make the simplifying, but potentially quite inaccurate, assumption that the protein receptor is a rigid object and attempt to dock the ligand to it), the scoring functions they employ, and the CPU time required to dock a molecule against a given target [171]. Performance of the docking code depends on its speed and its ability to accurately predict the binding mode, i.e. the orientation and conformation of the compound inside the active site.

The different available docking tools use different scoring methods [188]. Scoring is the prediction of binding affinity, which is used in ranking the solutions. There is a wide choice of scoring functions available [189], and they can be categorized as being physical-based (force-field), empirical or knowledge-based. Even if some progress has recently been made [190], further analyses are required to refine the docking score, to estimate the participation of water molecules, to minimize the number of false negative or positive hits or to propagate the true hits to the top of the list [172,191,192].

Post-analysis, the last step, is then required before final visual inspection. Consensus scoring [193-195], incorporation of solvation energies, better description of electrostatic interactions [196-197] and geometrical analysis of the compound-protein surface area can be used. Molecular dynamics [163,198,199] enables a flexible treatment of the target-ligand complexes at room temperature for a given simulation time and therefore is able to refine ligand orientations by finding more stable complexes. Molecular dynamics partially solves potential deficiencies in conformation and orientation search which might arise from docking. It also allows the re-ranking of molecules based on more accurate scoring functions [200]. This post-analysis method is time and memory-demanding but more accurate than docking methods.

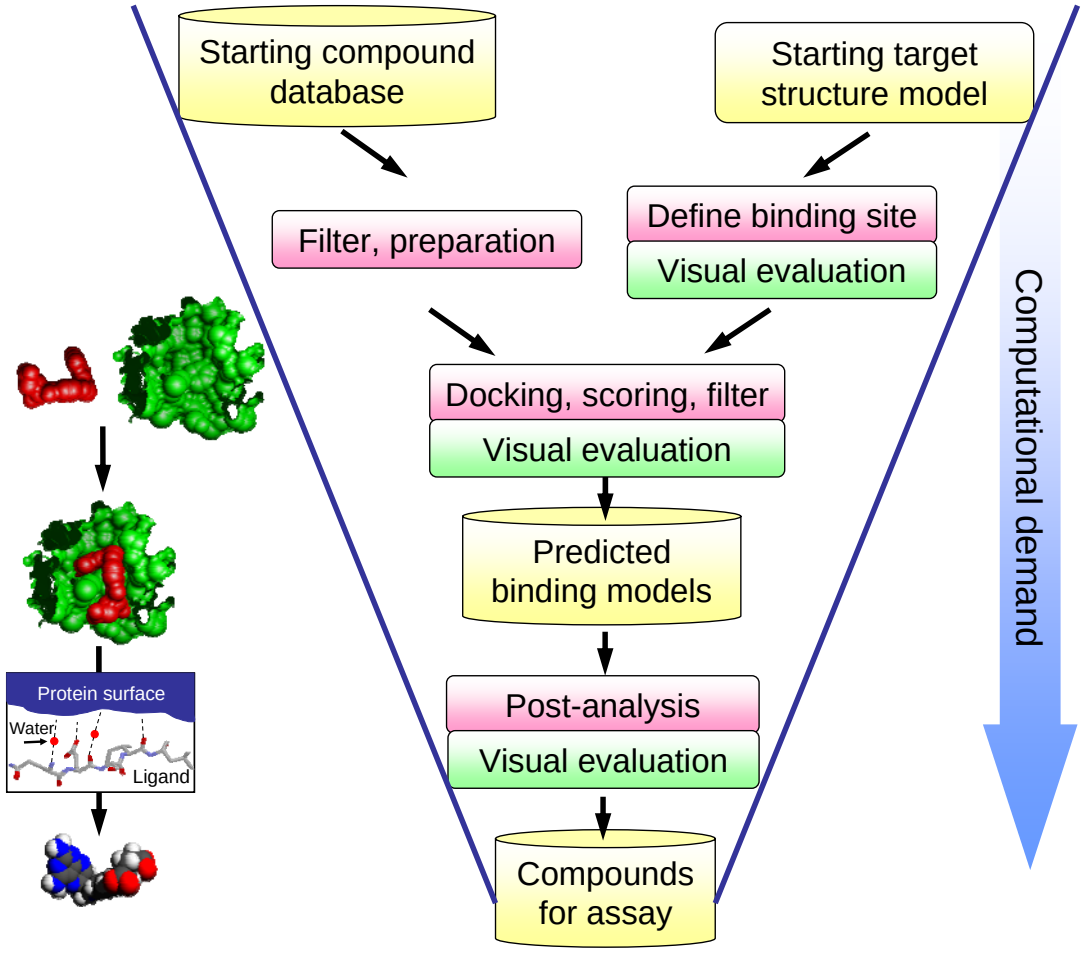


Figure 8: Structure-based virtual screening workflow

As explained before, the virtual screening pipeline requires significant computing resources. As a consequence, a progressive integration of the computational methods in traditional drug discovery is being observed.

Status

Current efforts within the pharmaceutical industry are directed at reducing the time and costs for drug development. High throughput virtual screening seems to be a well accepted, rapid, efficient and cost effective technology used by several pharmaceutical companies and academic institutes for screening chemical compounds in the drug discovery process [174,201]. But within many groups, virtual screening remains marginal [202]. The reason for this is that they have invested in facilities for high throughput screening. Sophisticated logistics and dedicated manpower allow 1 million compounds to be screened in less than two months. Consequently, the use of *in silico* screening to reduce the number of compounds docked *in vitro* is still limited [203]. The limited adoption of virtual screening comes also from the wrong structures of some protein models, the inaccuracy of scoring functions, the lack of flexibility of the target protein in docking methods [92,139,167], the lack of an

automated virtual screening pipeline and the high computation time required to screen millions of compounds [161,171,204].

Despite these difficulties, applying *in silico* techniques before, after and in parallel with high throughput screening improves the overall process quality in terms of information content and success rate [203]. Successful examples of novel hits identification and significant enrichment have been described in the literature [205-215]. For instance, Doman [205] compared the traditional and virtual screening performances for the protein tyrosin phosphatase 1B. The virtual screening hit rate reached 34.8% while the hit rate of high throughput screening was only 0.021%. Selected molecules had better IC_{50} and drug-like quality.

Virtual screening can definitively improve the quality of the drug discovery output [Davies] as it is much more a technique to reduce costs induced by experimental screening. Current drug discovery is driven by data [216]. Successful lead discovery needs high quality target and compound data [72]. These data need to be reusable, shareable, suitable and finally used to generate reliable predictions [217]. Data integration in the *in silico* but also traditional drug discovery process is thus a challenge [218].

1.4.5. Challenges in the development of *in silico* drug discovery

The previous sections explained that reducing the research time and cost in the discovery stage and enhancing information about the leads are key priorities for pharmaceutical companies worldwide. To achieve this goal, *in silico* drug discovery must be able to square up to the challenges described below:

- The *in silico* drug discovery process includes the management of a large variety and quantity of scientific data. For example: images, sequences, models, databases. Data integration is thus a challenge to increase knowledge discovery but also to ease the complex workflow. This implies data format standardization, dataflow definition in a distributed system, infrastructure and software providers for data storage, services for data and meta-data registration, data manipulation and database updates.
- The *in silico* drug discovery process also includes the management of a large variety and quantity of software. Software integration is another challenge to build efficient and complex workflows and to ease data management and data mining. Software can be provided in a distributed environment such as a web server on the Internet. Different experts are absolutely necessary to maintain and update software and workflows to propose new methods or pipelines, to use remote services, exploit outputs, and finally to propose compounds for assay. A software workflow will assist the scientist and the decision-maker in organizing their work in a flexible manner, and in delivering the information and knowledge to the organization.
- Deploying intensive computing is a challenge for *in silico* drug discovery. For instance, computing 1 million docking probabilities or modeling 1,000 compounds on one target protein requires in the order of a few TFlops during one day. Very large

computing resources are also needed to describe accurately protein structure models by computational methods based on all-atom physics-based force fields including implicit solvation. Computing power is also required for bioinformatics resource centers where server access is saturated by the large number of short tasks requested by users.

- Joining the new Information Technologies with life science to enable *in silico* drug discovery requires strong remote collaboration between different public and private experts when addressing neglected and emerging infectious diseases. It also involves strong sharing of resources: data and knowledge, software and workflow, and infrastructures such as computing, storage and networks. The collaboration space needs experts to maintain the resources. Having tools and data accessible to everyone in collaboration requires intuitive interfaces that need to be maintained. These interfaces reduce the development time of new methods. They also help the integration of data and software from *in silico* drug discovery but also from experimental processes.
- Security is a key challenge for pharmaceutical industries but also for academic institutes in most cases. Effective protection of intellectual properties and sensitive information requires, for instance, authentication of users from different institutions, mechanisms for management of user accounts and privileges and support for resource owners to implement and enforce access control policies.

Challenges to develop *in silico* drug discovery are data and software integration, intensive computing deployment, remote collaboration and resources sharing, and of course security. Thus, there is a need for a powerful and secured environment sharing and integrating remote resources such as tools, data, computing and storage.

1.5. Grid added value for *in silico* drug discovery

Developing *in silico* drug discovery for neglected and emerging infectious diseases requires a robust infrastructure and relevant services to support distributed resource sharing. Resources are here defined as: computing, storage and network; as well as data, knowledge, software and workflow; but also instruments and sensors; and finally people and organization.

The grid is a new Information Technology that can provide these resources. A grid is the combination of networked resources and the corresponding management software, which provides services for the user. Grid technology provides the collaborative Information Technology environment to enable the combination between life science research, field work, health systems, pharmaceutical industries and infrastructure. It proposes a new paradigm for the collection and analysis of distributed information where data are no longer centralized in one single repository. On a grid, data can be stored anywhere and still be transparently accessed by any authorized user. The computing resources of a grid are also shared and can be mobilized on demand. The grid will be further defined in chapter 2.

The grid added value in the development of *in silico* drug discovery for neglected and emerging infectious diseases has multiple dimensions:

- grids offer unprecedented opportunities for resource sharing and collaboration;
- grids open exciting perspectives for handling information flows;
- grids provide the resources to speed up the execution of time-consuming software.

Grids offer unprecedented opportunities for resource sharing and collaboration.

Grid allows

- the sharing of resources in a cross-organizational collaboration space between the pharmaceutical industry and academic research institutions, and between developed and least developed countries,
- the creation of as a virtual laboratory for the different actors, increasing cooperation and communication between partners,
- the mobilizing of resources routinely or in an emergency,
- the sharing of diverse, complex, large and distributed information for collaborative exploration and mutual benefit,
- the use of new Information Technology such as large databases or time-consuming software,
- the optimal exploitation of resources by taking advantage of spare computing cycles or by maximizing the use of high performance computing platforms usage,
- the reduction of hardware costs.

Grids open exciting perspectives for handling information flows.

Grid allows

- the deployment of services for healthcare and research centers in endemic regions,
- the deployment of infrastructures to collect data and improve disease surveillance and monitoring,
- the building of knowledge space with genomics and medical information (epidemiology, status of clinical tests, drug resistances, etc.),
- access to relevant data, periodically updated data bases and publications,
- the federation of regional or international databases for disease study and monitoring of vector control, clinical trials and drug delivery,
- the provision of transparent and secure access to storage and the archiving of large amounts of data in an automated and self-organized fashion,
- connection, analysis and structuring of data and information in a transparent mode according to pre-defined rules (science or business process based).

Grids provide the resources to speed up the execution of time-consuming software.

Grid allows

- access to large computing resources for *in silico* drug discovery, data analysis and mathematical modeling,
- the application of high performance computing to new areas,
- the production of additional or more accurate analyses,

- the facilitation of the exchange of tools and workflows between scientists,
- the performance of computing intense tasks in a transparent way by means of an automated job submission and distribution facility,
- access to services and resources 24 hours a day,
- the running of the same job on many platforms across different sites,
- access to computing resources by a single efficient path.

Grids are unique tools for collecting and sharing information, networking experts, mobilizing resources routinely or in an emergency. Grid is thus an appropriate environment to develop *in silico* drug discovery for neglected and emerging infectious diseases.

1.6. Conclusion

In this first chapter, the challenges raised by neglected and emerging infectious diseases have been described. We have demonstrated how Information Technologies can contribute to address some of these challenges in the fight against neglected diseases by offering support to *in silico* pipelines, data collection, monitoring systems and international resource sharing. Fighting emerging infectious diseases requires also early detection, epidemiological watch, prevention, fast search for new drugs and vaccines thanks to international collaborations. We have also described in this chapter the interest of *in silico* drug discovery in reducing costs and time to market of new drugs. Developing *in silico* drug discovery requires data integration, software integration, computing resources, remote collaboration and resource sharing, with strong security constraints.

Grids are unique tools for collecting and sharing information, networking experts, mobilizing resources routinely or in an emergency. Grid is a support to share resources and collaboration, to handle information flows, and to provide the resources to speed up the execution of time-consuming software. Consequently grid technology can contribute to lowering the barrier between the pharmaceutical industry and academic research institutions, and between developed and least developed countries. The philosophy of sharing resources in projects through the grid opens exciting perspectives for such topics as neglected and emerging infectious diseases as it fosters international collaboration.

In the next chapter we are going to enter into the details of what a grid is and more precisely what are the best grid environments to be used to develop *in silico* drug discovery.

1.7. References

[1] Kettler, H. E. and Modi, R. Building local research and development capacity for prevention and cure of neglected diseases: the case for India. Bull. World Health Organization 79, 742–747 (2001).
[2] Medecins Sans Frontieres, Access to Essential Medicines Campaign. Fatal imbalance, the crises in research and development for drugs for neglected diseases (2001).
[3] Reich, M. R. The global drug gap. Science 287, 1979–1981 (2000).
[4] World Health Organization, The world health report 2004 – changing history, (2004).

[5] World Health Organization, Infectious diseases report – Removing obstacles to healthy development, (1999).
[6] Remme, J.H., et al., Strategic emphases for tropical diseases research: a TDR perspective, Trends in Parasitology. Vol.18 No.10 October 2002.
[7] Trouiller, P. et al., Drugs development for neglected diseases: a deficient market and a public health policy failure, Lancet 359, 2188–2194 (2002).
[8] Mrazek, M. F. and Mossialos, E., Stimulating pharmaceutical research and development for neglected diseases, Health Policy 64, 75–88 (2003).
[9] Nwaka, S. and Ridley, R. G., Virtual drug discovery and development for neglected diseases through public-private partnerships, Nature Reviews Drug Discovery (2003).
[10] Bruneton, C. et al., The drug trade between European countries and developing countries, Med. Trop. 57, 375–379 (1997).
[11] Widdus, R., Public–private partnerships for health: their main targets, their diversity, and their future directions, Bull. World Health Organization 79, 728–734 (2001).
[12] World Health Organisation, Report of the Commission on Macroeconomics for Health - Macroeconomics and Health: Investing in Health for Economic Development, (2001).
[13] Kettler, H. and Towse, A., Public–private partnerships for research and development: medicines and vaccines for diseases of poverty, Office of Health Economics, (2002).
[14] Wheeler, C. and Berkley, S., Initial lessons from public–private partnerships in drug and vaccine development, Bull. World Health Organization 79, 728–734 (2001).
[15] TDR News, MMV: New Medicines for Malaria Venture, (1999).
[16] Joint United Nations Programme on HIV/AIDS, 2006 Report on the global AIDS epidemic, (2006)
[17] Unicef and World Health organization, World Malaria report, (2005).
[18] Hoffman, S., U.S. medicine, (2000).
[19] Weisner, J., et al., Angew. New Antimalarial drugs, Chem. Int. 42 5274-529 (2003).
[20] Rosenthal, P. J., Antimalarial Chemotherapy: Mechanisms of Action, Resistance, and New Directions in Drug Discovery, Humana Press, (2001).
[21] Greenwood, B. et al., Malaria in 2002, Nature 415, 670-672 (2002).
[22] World Health Organization, The world health report 2002 - reducing risks, promoting healthy life, (2002).
[23] Maréchal, E., private communication, (2006).
[24] Towie, N., Malaria breakthrough raises spectre of drug resistance, Nature 440:852-853 (2006).
[25] Afonso, A., et al, Malaria parasites can develop stable resistance to Artemisinin but Lack Mutations in Candidate Genes atp6 (Encoding the Sarcoplasmic and Endoplasmic Reticulum Ca2+ ATPase), tctp, mdr1, and cg10, Antimicrobial Agents and Chemotherapy 50 480-489 (2006).
[26] National Center for Infectious Diseases, Division of Parasitic Diseases, (2006) and http://www.cdc.gov.
[26bis] Mehlin, C. Structure based drug discovery for Plasmodium falciparum. Combinatorial Chemistry & High Throughput Screening, 8, 5-14 (2005).
[26ter] Pattanaik, P., et al., Prospectives in drug design against malaria, Current Topics in Medicinal Chemistry, 2, 483-505 (2002).
[27] Lander, E.S., et al., Initial sequencing and analysis of the human genome, Nature 409, 860–921 (2001).
[28] Venter, J.C., et al., The sequence of the human genome, Science 291, 1304–1351 (2001).
[29] Holt, R., et al., The genome sequence of the malaria mosquito Anopheles gambiae. Science 298:129–49 (2002).
[30] Gardner, M.J., et al., Genome sequence of the human malaria parasite Plasmodium falciparum. Nature 419:498–511 (2002).
[31] Coppel, R.L., Bioinformatics and the malaria genome: facilitating access and exploitation of sequence information. Mol Biochem Parasitol. 118:139-145 (2001).
[32] Kissinger, J.C., et al., The Plasmodium genome database, Nature 419:490-492 (2002).
[33] Bahl, A., et al., PlasmoDB: the Plasmodium genome resource. A database integrating experimental and computational data, Nucleic Acids Res. 31:212-215 (2003).

[34] Carucci, D.J., Advances in malaria genomics since MIM Arusha, 2002, Acta Tropica 95:260-264 (2005).
[35] Fidock, D.A., et al., Antimalarial drug discovery: efficacy models for compound screening. Nat Rev Drug Discov. 3:509–20 (2004).
[36] Alonso, P.L., et al., Efficacy of the RTS,S/AS02A vaccine against Plasmodium falciparum infection and disease in young African children: randomized controlled trial, Lancet 364:1411–20 (2004).
[37] World Health Organization, Portfolio of malaria vaccines currently in development, (2004) and http://www.who.int/vaccine_research/documents/en/malaria_table.pdf
[38] United Nations Joint Programme on HIV/AIDS, UNAIDS executive director calls for action to protect youth from HIV/AIDS, (2002).
[39] Morens, D.M., et al., The challenge of emerging and re-emerging infectious diseases, Nature 430:242–9 (2004).
[40] Fauci, A.S., Infectious diseases: considerations for the 21st century, Clin Infect Dis. 32:675–85 (2001).
[41] Morse, S.S., Factors in the Emergence of Infectious Diseases. Emerging Infectious Diseases 1:7–15 (1995).
[42] Taylor, L.H., et al., Risk Factors for Human Disease, Emergence, Philosophical Transactions of the Royal Society B: Biological Sciences 356:983–89 (2001).
[43] National Institute of Allergy and Infectious Diseases, research on severe acute respiratory syndrome, (2005) and http://www.niaid.nih.gov/factfiles/sars.htm
[44] University of Toronto Joint Centre for Bioethics, Stand on guard for thee: Ethical considerations in preparedness planning for pandemic influenza, a report of the University of Toronto Joint Centre for Bioethics Panedemic influenza Working Group, (2005).
[45] International Committee on Taxonomy of Viruses 46.0.1. Influenza virus A, (2002) and http://www.ncbi.nlm.nih.gov/ICTVdb/ICTVdB/46010000.htm
[46] Li, K. S., et al., Genesis of a highly pathogenic and potentially pandemic H5N1 influenza virus in eastern Asia, Nature 430:209-213, (2004).
[47] The Writing Committee of the World Health Organization, Consultation on Human Influenza A/H5. Avian Influenza A (H5N1) infection in humans, N Engl J Med 353:1374-85 (2005).
[48] Johnson N.P.A.S. and Mueller J., Updating the accounts: global mortality of the 1918–1920 "Spanish" influenza pandemic, Bull Hist Med 76:105–15 (2002).
[49] K. Ungchusak, et al., Probable Person-to-Person Transmission of Avian Influenza A (H5N1), Thailand. N Engl J Med. 352(4):333-40, (2005).
[50] Couch, R., Chapter 58. Orthomyxoviruses Multiplication, Baron, S. (ed.) Medical Microbiology. Galveston, (1996).
[51] Fedson, D.S., Preparing for pandemic vaccination: an international policy agenda for vaccine development, J Public Health Policy 26:4–29 (2005).
[52] World Health Organization, Global Programme on Influenza Vaccine research and development, current status, (2005).
[53] Fedson, D.S., Vaccine development for an imminent pandemic; why we should worry, what we must do. Human Vaccines 2:38–42 (2006).
[54] http://www.ch.ic.ac.uk/local/projects/sanderson/
[55] Lamb, R.A., et al., The influenza A virus M2 ion channel protein and its role in the influenza virus life cycle, Wimmer, E. (ed.), Receptor-Mediated Virus Entry into Cells., Cold Spring Harbor Press 303-321 (1994).
[56] Hayden, F.G., Perspectives on antiviral use during pandemic influenza, Phil Trans R Soc London 356: 1877–84 (2001).
[57] Shiraishi, K., et al., High frequency of resistant viruses harboring different mutations in amantadine-treated children with influenza, J Infect Dis 188: 57–61 (2003).
[58] Gubareva, L.V., et al., Influenza virus neuraminidase inhibitors, The Lancet 355:827-835 (2000).
[59] Moscona, A., Neuraminidase inhibitors for Influenza, N Engl J Med 353:1363-73 (2005).
[60] Kiso, M., et al., Resistant influenza A viruses in children treated with oseltamivir: descriptive study, Lancet 364(9436):759-65 (2004).

[61] de Jong, M. D., et al., Oseltamivir Resistance during Treatment of influenza A (H5N1) Infection, N. Engl. J. Med., 353(25):2667-72, (2005).
[62] Bogner, P., A global initiative on sharing avian flu data, Nature 442, 981 (2006).
[63] Editorial, Nature 441, 1028 (2006).
[64] OIE/FAO Network of Expertise on Avian Influenza, OFFLU Keeps its Pace on Global Sharing Virus Samples Press Release, (2006) and http://www.offlu.net/portals/0/pdf/Press.pdf
[65] Rukmantara, Tb. , A. Bird Flu Data Now Open to All, The Jakarta Post (2006)
[66] CDC and APHL, Make Influenza Virus Sequence Data Publicly Accessible, Press Release (2006) and http://www.cdc.gov/od/oc/media/pressrel/r060822.htm
[67] Lipatov, A.S., et al., Influenza: emergence and control, J Virol 78:8951–9 (2004).
[68] Thailand country brief review on research and development on avian influenza, 2005
[69] Stöhr, K., Avian influenza and pandemics – research needs and opportunities, N Engl J Med 352:405-7 (2005).
[70] C.F. Curtis, S.L. Hoffman, Science 290, (2000) 1508-1509.
[71] Breton, V. et al, Grid Technology for Biomedical Applications, Lecture Notes in Computer Science 3402 204–218 (2005).
[72] Bleicher, K. H., et al., Hit and Lead Generation: Beyond High-Throughput Screening, Nat. Rev. Drug. Discov. 2, 369-378, (2003).
[73] DiMasi, J.A. et al., J Health Econ, 22, 151-185 (2003).
[74] Myers, S. and Baker, A., Drug discovery – an operating model for a new era, Nat. Biotechnol. 19 727–730 (2001).
[75] Annamalai, T. R., The Life Sciences Challenge; an industry under pressure to innovate, SETLabs briefings 2:1 (2004).
[76] Heilman, R.D., Drug development history, overview, and what are GCPs?, Qual Assur 4(1) 75-9 (1995).
[77] Lipinski, C., et al., Experimental and computational approaches to estimate solubility in drug discovery and development settings, Adv. Drug Deliv. Rev. 23, 3–25 (1997).
[78] Venkatesh, S. and Lipper, R., Role of the development scientist in compound lead selection and optimization, J. Pharm. Sci. 89, 145–154 (2000).
[79] Kennedy, T., Managing the drug discovery/development interface, Drug Discovery Today, 2, (1997).
[80] Augen, J., Information technology to the rescue, Nat. Biotechnol. 19 BE39–BE40 (2001).
[81] Bilofsky, H., et al., Chapter 5: Grid enabled pharmaceutical R&D: Pharmagrids, the Healthgrid White Paper: From Grid to HealthGrid, IOS Press (2005).
[82] Hodgman, C., An information-flow model of the pharmaceutical industry. Drug Discov. Today 6 1256–1258 (2001).
[83] Ridley, R., Antimalarial drug discovery and development: an industrial perspective, Exp. Parasitol. 87, 293–304 (1997).
[84] Roberts, S.A., Drug metabolism and pharmacokinetics in drug discovery, Curr. Opin. Drug Discov. Devel. 6, 66–80 (2003).
[85] Frantz, S., Screening the right candidate, Nature Rev. Drug Discov. 2, 331 (2003)
[86] Di, L. and Kerns, E.H., Profiling drug-like properties in discovery research, Curr. Opin. Chem. Biol. 7, 402–408 (2003).
[87] Ernst & Young, Tufts CSDD, and Boston Consulting Group, Life Science Insights, (2004).
[88] Anderson, A.C., The process of structure-based drug design, Chem. Biol. 10, 787–797 (2003),
[89] Schneider, G. and Fechner, U., Computer-based de novo design of drug-like molecules, Nat. Rev. Drug Discov. 4, 649–663 (2005).
[90] Ridley, R. G., Medical need, scientific opportunity and the drive for antimalarials, Nature 415, 686–693 (2002).
[91] Lipinski, C., and Hopkins, A., Navigating chemical space for biology and medicine, Nature 432, 855-861 (2004).
[92] Boston Consulting Group, BCG Estimate – A Revolution in R&D – The Impact of Genomics, (2001).
[93] Ashburn, T.T., and Thor, K.B., Drug repositioning: identifying and developing new uses for existing drugs, Nature Rev. Drug Discov. 3, 673-683 (2004).

[94] Haak Van den, et al., Industry Success Rates 2004, CMR Report 04-234R (2004).
[95] FDA, Innovation and Stagnation: Challenge and Opportunity on the Critical Path to New Medical Products, FDA White Paper (2004).
[96] Handen, J.S., et al., The industrialization of drug discovery, Drug Discov. Today 7 83–85 (2002).
[97] Drews, J., et al., Drug Discovery: a historical perspective, Science 287 1960–1964(2000).
[98] Drews, J., Strategic trends in the drug industry, Drug Discov. Today 8 411–420 (2003).
[99] Chanda, S.K., et al., Fulfilling the promise: drug discovery in the post-genomic era, Drug Discov. Today 8 168–174 (2003).
[100] Drews, J., et al., Innovation deficits in the pharmaceutical industry, Drug Inf. J. 30 97–108 (1996).
[101] Weisbach, J.A., et al., Diagnosing the decline of major pharmaceutical research laboratories; a prescription for drug companies, Drug Dev. Res. 34 243–259 (1995).
[102] Sams-Dodd, F., Target-based drug discovery: is something wrong?, Drug Discovery Today 10:2 139-147 (2005).
[103] Peitsch, M., About the use of protein models, Bioinformatics 18 (2002) 934-938
[104] Wishart, D., NMR spectroscopy and protein structure determination: applications to drug discovery and development, Curr. Pharm. Biotechnol. 6 105-120 (2005).
[105] DePristo, M.A., et al. Heterogeneity and inaccuracy in protein structures solved by X-ray crystallography, Structure 12 831-838 (2004).
[106] Maggio, E.T., and Ramnarayan, K., Recent Developments in Computational Proteomics, Drug Discovery Today, (2001).
[107] Westbrook, J., et al., The Protein Data Bank and structural genomics, Nucleic Acids Res 31:489-491 (2003).
[108] Baker, D. and Sali, A., Protein structure prediction and structural genomics, Science 294:93-96 (2001).
[109] Burley, S.K., et al., Structural genomics: beyond the human genome project, Nat. Genet. 23 151 (1999).
[110] Vitkup, D., et al., Completeness in structural genomics, Nat. Struct. Biol. 8 559 (2001).
[111] Fetrow, J.S., et al., Functional analysis of the Escherichia coli genome using the sequence-to-structure-to-function paradigm: identification of proteins exhibiting the glutaredoxin/ thioredoxin disulfide oxidoreductase activity, J. Mol. Biol. 282 703–711 (1998).
[112] Marti-Renom, M.A., et al., Comparative protein structure modeling of genes and genomes, Annu. Rev. Biophys. Biomol. Struct. 29 291–325 (2000).
[113] Ginalski, K., et al., Practical lessons from protein structure prediction, Nucleic Acids Res. 33, 1874–1891 (2005).
[114] Petrey, D. and Honig, B., Protein structure prediction: inroads to biology, Mol Cell 20(6):811-819 (2005).
[115] Moult, J., et al., Critical assessment of methods of protein structure prediction (CASP)-round V, Proteins Suppl. 53 334–339 (2003).
[116] Bonneau, R. and Baker, D., Ab Initio Protein Structure Prediction: Progress and Prospects, Annu. Rev. Biophys. Biomol. Struct. 30 173-89 Review (2001).
[117] Nanias, M., et al., Protein structure prediction with the UNRES force-field using Replica-Exchange Monte Carlo-with-Minimization; comparison with MCM, CSA, and CFMC, J. Comput. Chem. 26 1472–1486 (2005).
[118] Berman, H.M., et al., the Protein Data Bank, Nucleic Acids Res. 28, 235-242 (2000).
[119] Bradley, P., et al., Toward high-resolution de novo structure prediction for small proteins. Science 309 1868–1871 (2005).
[120] Bowie, J.U., et al., A method to identify protein sequences that fold into a known three-dimensional structure, Science 253(5016):164-70 (1991).
[121] Kolodny, R., et al., Comprehensive evaluation of protein structure alignment methods: scoring by geometric measures, J. Mol. Biol. 346 1173–1188 (2005).
[122] Augen, J., The Evolving Role of Information Technology in the Drug Discovery Process, Drug Discovery Today (2002).
[123] Aloy, P. et al., Protein complexes structure prediction challenges for the 21st century, Current Opinion in Structural Biology, 15:15-22 (2005).

[124] Brenner, S. E., Target selection for structural genomics, Nat. Struct. Biol. 7 (Suppl.) 967 (2000).
[125] Sali, A., 100,000 protein structures for the biologist, Nat. Struct. Biol. 5 1029 (1998).
[126] Russell, R.B., et al., A structural perspective on protein-protein interactions, Curr Opin Struct Biol 14:313-324 (2004).
[127] Zhang, G., et al., Crystal structure of Thermus aquaticus core RNA polymerase at 3.3 A ° resolution, Cell 98:811-824 (1999).
[128] Cramer, P., et al., Architecture of RNA polymerase II and implications for the transcription mechanism. Science 288:640-649 (2000).
[129] Ban, N., et al., The complete atomic structure of the large ribosomal subunit at 2.4 A ° resolution, Science 289:905-920 (2000).
[130] Yusupov, M.M., et al., Crystal structure of the ribosome at 5.5A ° resolution. Science 292:883-896 (2001).
[131] Heinemann, U., et al., Facilities and methods for the high-throughput crystal structural analysis of human proteins, Acc. Chem. Res. 36 157–163 (2003).
[132] Service, R.F., Structural genomics. Tapping DNA for structures produces a trickle, Science 298 948–950 (2002).
[133] Rupp, B., High-throughput crystallography at an affordable cost: the TB Structural Genomics consortium crystallization facility, Acc. Chem. Res. 36 173–181 (2003).
[134] Lesley, S.A., et al., Structural genomics of the Thermotoga maritima proteome implemented in a high-throughput structure determination pipeline, Proc. Natl. Acad. Sci. U.S.A. 99 11664–11669 (2002).
[135] Kopp, J. and Schwede, T., Automated protein structure homology modeling: a progress report, Pharmacogenomics 5 405–416 (2004).
[136] Takeda-Shitaka, M., et al. Protein structure prediction in structure based drug design, Curr. Med. Chem. 11 551–558 (2004).
[137] Structural Bioinformatics Inc. and AtheroGenics Announce Drug Discovery Collaboration, Business Wire, (2002).
[138] Jacobson, M., and Sali, A., Comparative protein Structure Modelling and its applications to drug discovery, Annu. Rep. Med. Chem. 39, 259–274 (2004).
[139] Congreve, M., et al., Structural biology and drug discovery, Drug Discov Today 10(13):895-907 (2005).
[140] Gordon, R.K., et al., Anti-HIV-1 activity of 3-deaza-adenosine analogs. Inhibition of S-adenosylhomocysteine hydrolase and nucleotide congeners, Eur. J. Biochem. 270 3507 (2003).
[141] von Grotthuss, M., et al., mRNA cap-1 methyltransferase in the SARS genome, Cell. 113, 701 (2003).
[142] Fox, J.A., et al., The bioinformatics links directory: a compilation of molecular biology web servers, Nucleic Acids Res. 33, W3–W24 (2005).
[143] Fischer, D., et al., CAFASP2: The Second Critical Assessment of Fully Automated Structure Prediction Methods, Proteins, 45 (Suppl. 5), 171 (2001).
[144] Bourne, P. E., CASP and CAFASP experiments and their findings, Methods Biochem. Anal., 44, 501 (2003).
[145] Bujnicki, J. M., et al., LiveBench-2: The Second Large-Scale Evaluation of Protein Structure Prediction Servers, Prot. Sci., 10, 352 (2001).
[146] Eyrich, V. A., et al., EVA: continuous automatic evaluation of protein structure prediction servers, Bioinformatics, 17, 1242 (2001).
[147] Koh, I.-Y.Y., et al., EVA: evaluation of protein structure prediction servers, Nucleic Acids Res., (2003).
[148] http://boinc.bakerlab.org/rosetta/
[149] Taufer, M., et al, Predictor@home: A "Protein Structure Prediction Supercomputer" Based on Public-Resource Computing, IEEE Transactions on Parallel and Distributed Systems 17 8 786-796 (2006).
[150] Stefan, M., et al., Folding@Home and Genome@Home: Using distributed computing to tackle previously intractable problems in computational biology, Computational Genomics (2002).
[151] Seneci, P. and Miertus, S., Combinatorial chemistry and high-throughput screening in drug discovery: different strategies and formats, Mol. Divers. 5 75–89 (2000).

[152] Bailey, N. et al., Solution-phase combinatorial chemistry in lead discovery, Chimia 51, 832–837 (1997).
[153] Dolle, R.E., Comprehensive survey of combinatorial library synthesis: 2003. J. Comb. Chem. 6, 623–679 (2004).
[154] Spencer, R.W., High throughput virtual screening of historic collections on the file size, biological targets, and file diversity, Biotechnol. Bioeng 61, 61-67 (1998).
[155] Lahana, R., How many leads from HTS?, Drug Discov. today 4, 447-448 (1999).
[156] Hopkins, A.L. and Groom, C.R., Opinion: the druggable genome, Nat. Rev. Drug Discov. 1, 727–730 (2002).
[157] Mestres, J., and Knegtel, R.M.A., Similarity versus docking in 3D virtual screening, Perspect. Drug Des. Discovery 20 191–207 (2000).
[158] Mason, J.S., et al., 3-D pharmacophores in drug discovery, Curr. Pharm. Des. 7 567–597 (2001).
[159] Srinivasan, J., et al., Evaluation of a novel shape-based computational filter for lead evolution: Application to thrombin inhibitors. J. Med. Chem. 45 2494–2500 (2002).
[160] Bajorath J., Integration of virtual and high-throughput screening, Nat. Rev. Drug. Discov. 1, 882-894 (2002).
[161] Lengauer, T., et al., Novel technologies for virtual screening, Drug Discov. Today 9, 27–34 (2004).
[162] Halperin, I., et al., Principles of docking: an overview of search algorithms and a guide to scoring functions, Proteins Struct. Funct. Genet. 47 409–443 (2002).
[163] Gohlke, H. and Klebe, G., Approaches to the description and prediction of the binding affinity of smallmolecule ligands to macromolecular receptors, Angew Chem Int Ed Engl 41:2644–2676 (2002).
[164] Anderson, A.C. and Wright, D.L. The design and docking of virtual compound libraries to structures of drug targets, Curr. Comp. Aided Drug Des. 1 103–127 (2005).
[165] Shoichet, B.K., Virtual screening of chemical libraries, Nature 432, 862–865 (2004).
[166] Chin, D.N., et al., Integration of virtual screening into the drug discovery process. Mini Rev. Med. Chem. 4, 1053–1065 (2004).
[167] Kitchen, D.B., et al., Docking and scoring in virtual screening for drug discovery: methods and applications. Nat. Rev. Drug Discov. 3, 935–949 (2004).
[168] Barril, X., Virtual screening in structure-based drug discovery. Mini Rev. Med. Chem. 4, 779–791, (2004).
[169] Raha, K. and Merz, K.M., Large-scale validation of a quantum mechanics based scoring function: predicting the binding affinity and the binding mode of a diverse set of protein-ligand complexes, J. Med. Chem. 48, 4558–4575 (2005).
[170] Kuhn, B., et al., Validation and use of the MM-PBSA approach for drug discovery, J. Med. Chem. 48, 4040–4048 (2005)
[171] Lyne, P.D., Structure-based virtual screening: an overview, Drug Discov. Today 7 1047–1055 (2002).
[172] Ghosh, S., et al., Structure-based virtual screening of chemical libraries for drug discovery. Curr Opin Chem Biol 10 3 194-202 (June 2006).
[173] Waszkowycz, B., et al., Large-scale virtual screening for discovering leads in the postgenomic era, IBM systems journal, 40 360-376 (2001).
[174] Walters, W.P., and Murcko, M.A., Prediction of 'drug-likeness', Adv. Drug Deliv. Rev. 54 255–271 (2002).
[175] Palm, K., et al., Correlation of drug absorption with molecular surface properties, J. Pharm. Sci. 85 32–39 (1996).
[176] Walters, W.P., et al., Virtual screening – an overview. Drug Discov. Today 3 160–178 (1998).
[177] Schnecke, V. and Bostrom, J. Computational chemistry-driven decision making in lead generation, Drug Discov Today 11:43-50 (2006).
[178] Young, S.S., et al. Initial compound selection for sequential screening, Curr Opin Drug Discov Devel 5:422-427 (2002).
[179] Baringhaus, K.H. and Hessler, G., Fast similarity searching and screening hit analysis, Drug Discov Today Technol 1:197-202 (2004).

[180] Rarey, M., et al., A fast flexible docking method using an incremental construction algorithm. J Mol Biol 261:470-489 (1996).
[181] Jones, G., et al., Development and validation of a genetic algorithm for flexible docking. J Mol Biol 267:727-748 (1997).
[182] Friesner, R.A., et al., Glide: a new approach for rapid, accurate docking and scoring. 1. Method and assessment of docking accuracy. J Med Chem 47:1739-1749 (2004).
[183] Abagyan, R., et al., ICM-A new method for protein modeling and design: applications to docking and structure prediction from the distorted native conformation. J Comput Chem 15:488-506 (1994).
[184] Makino, S. and Kuntz, I.D., Automated flexible ligand docking method and its application for database search. J Comput Chem 18:1812-1825 (1997).
[185] Ewing, T.J., et al., DOCK 4.0: search strategies for automated molecular docking of flexible molecule databases, J Comput Aided Mol Des 15:411-428 (2001).
[186] McGann, M.R., et al., Gaussian docking functions. Biopolymers 68:76-90 (2003).
[187] Goodsell, D.S., et al., Automated docking in crystallography: analysis of the substrates of aconitase. Proteins 17:1-10 (1993).
[188] Chen, H., et al., On evaluating molecular-docking methods for pose prediction and enrichment factors, J Chem Inf Model 46:401-415 (2006).
[189] Ajay, A. and Murcko, M.A., Computational methods to predict binding free energy in ligand-receptor complexes, J. Med. Chem. 38 4953–4967 (1995).
[190] Klon, A.E., et al., Finding more needles in the haystack: a simple and efficient method for improving high-throughput docking results, J Med Chem 47:2743-2749 (2004).
[191] Pearlman, D.A. and Charifson, P.S., Are free energy calculations useful in practice? A comparison with rapid scoring functions for the p38 MAP kinase protein system, J. Med. Chem. 44 3417–3423 (2001).
[192] Stahl, M. and Schulz-Gasch, T., Scoring functions for protein ligand interactions: a critical perspective, Drug Discov. Today. Technol. 1, 231–239 (2004).
[193] Charifson, P.S., et al., Consensus scoring: a method for obtaining improved hit rates from docking databases of three-dimensional structures into proteins. J. Med. Chem. 42 5100–5109 (1999).
[194] Bissantz, C., et al., Protein-based virtual screening of chemical databases. 1. Evaluation of different docking/scoring combinations. J. Med. Chem. 43 4759–4767 (2000).
[195] Stahl, M., and Rarey, M., Detailed analysis of scoring functions for virtual screening. J. Med. Chem. 44 1035–1042 (2001).
[196] Honig, B. and Nicholls, A., Classical electrostatics in biology and chemistry, Science 268 1144–1149 (1995).
[197] Grant, J.A., et al., A smooth permitivity function for Poisson-Boltzman solvation methods. J. Comp. Chem. 22 608–640 (2001).
[198] Lamb, M.L. and Jorgensen, W.L., Computational approaches to molecular recognition. Curr. Opin. Chem. Biol. 1, 449 (1997).
[199] Kollman, P.A., Free energy calculations: applications to chemical and biochemical phenomena, Chem. Rev. 93, 2395 (1993).
[200] Huang, D. and Caflisch, A., Efficient evaluation of binding free energy using continuum electrostatics solvation; J. Med. Chem., 47, 5791-5797 (2004).
[201] Jorgensen, W.L., The many roles of computation in drug discovery. Science 303:1813–1818 (2004)
[202] Gershell, L.J. and Atkins, J.H., A brief history of novel drug discovery technologies. Nat. Rev. Drug Discov. 2, 321–327 (2003)
[203] Davies, J.W., et al., Streamlining lead discovery by aligning *in silico* and high-throughput screening, Current Opinion in Chemical Biology, 10-4 343-351 (2006).
[204] Claussen, H., et al. The FlexX database docking environment – rational extraction of receptor based pharmacophores. Current Drug Discovery Technologies 1, 49–60 (2004).
[205] Doman, T.N., et al., Molecular docking and high throughput screening for novel inhibitors of protein tyrosine phosphatase 1B. J. Med. Chem. 45 2213–2221 (2002).

[206] Baxter, C.A., et al., New approach to molecular docking and its application to virtual screening of chemical databases. J. Chem. Inf. Comput. Sci. 40 254–262 (2000).
[207] Tondi, D., et al., Structure-based discovery and in-parallel optimization of novel competitive inhibitors of thymidylate synthase. Chem. Biol. 6 (1999), pp. 319–331.
[208] Schapira, M., et al., Rational discovery of novel nuclear hormone receptor antagonists. Proc. Natl. Acad. Sci. U.S.A. 97 1008–1013 (2000).
[209] Schapira, M., et al., Nuclear hormone receptor targeted virtual screening. J Med Chem 46:3045–3059 (2003).
[210] Perola, E., et al., Successful virtual screening of a chemical database for farnesyltransferase inhibitor leads. J. Med. Chem. 43 401–408 (2000).
[211] Osterberg, F., et al., Automated docking to multiple target structures: incorporation of protein mobility and structural water heterogeneity in AutoDock. Proteins 46:34–40 (2002).
[212] Carlson, HA., Protein flexibility and drug design: how to hit a moving target. Curr Opin Chem Biol 6:447–452 (2002).
[213] Cavasotto, C.N. and Abagyan, R.A., Protein flexibility in ligand docking and virtual screening to protein kinases. J Mol Biol 337:209–225 (2004).
[214] Shoichet, B.K., et al., Lead discovery using molecular docking, Curr Opin Chem Biol 6:439–446 (2002).
[215] Vangrevelinghe, E., et al., Discovery of a potent and selective protein kinase CK2 inhibitor by high-throughput docking, J. Med. Chem. 46, 2656–2662 (2003).
[216] Oprea, T.I., Virtual screening in Lead discovery: a viewpoint. Molecules 7, 51-62 (2002).
[217] Stahl, M., et al., Integrating molecular design resources within modern drug discovery research: the Roche experience. Drug Discov Today. 11:326-33 (2006).
[218] Searls, D.B., Data integration: challenges for drug discovery, Nat. Rev. Drug Discov. 4 45–58 (2005).

Chapter 2. Computing grids

2.1. Introduction

In the first chapter, challenges to fight neglected and emerging infectious diseases and to develop *in silico* drug discovery were presented. Grid technology is a solution to collect and share information, network experts, mobilize resources routinely or in an emergency.

Computing grids are still a relatively new notion even if the distributed computing concept is older. The first aim was the desire to share computing resources. But today many complex services have been conceived to build a full user-oriented environment. There are many grids but a common requirement is the need to have efficient resource management through a multi-layer set of software, called middleware.

There are two main grid hardware types. The desktop grid (inspired from peer-to-peer) concentrates on sharing systems such as desktop PCs connected to the Internet. The cluster grid aggregates distributed machines such as clusters. Desktop PCs networks can amass computing power, as does the SETI@home project. As well as computing resources, computing grids deploy advanced services like data management. The two technologies are complementary, but a cluster grid seems better suited to perennially sustain infrastructure.

There are also different cluster grid infrastructures and technologies in the world. There are many criteria to choose the best one for a given objective. For instance, the European project EGEE is the largest production cluster grid in the world and is adapted for computing power and data intensive applications. The French national grid RUGBI, based on existing grid technologies, provides a more flexible and secure environment for companies and academic institutions in life sciences.

The aim of this chapter is to define what a grid is and more precisely what are the best grid environments to be used to develop *in silico* drug discovery services against neglected and emerging infectious diseases.

The chapter is introduced as follows:

- After the introduction, the second section introduces some definitions.
- The third section will focus on the most widespread grids, the desktop and cluster grids in order to compare them.
- Then the infrastructures and the technologies of the grids used will be justified and described in the fourth and fifth sections.

2.2. Defining the grid

This first section gives several ideas to help better understand what a grid is. Following the exploration of several different proposals for a definition, there will be a grid taxonomy, and then an explanation of the general architecture of a grid.

2.2.1. Some definitions

Research in a grid environment and research about grid technology has greatly increased in the last 10 years [219]. One common reason is explained by Moore's law, illustrated by figure 9 [220]. The rapidly decreasing cost of broadband networks allows data, applications and hardware access from everywhere. Workflows can now be built with remote resources. At the same time, the large deployment of computing resources on the Internet and the World Wide Web has stimulated resource mutualization.

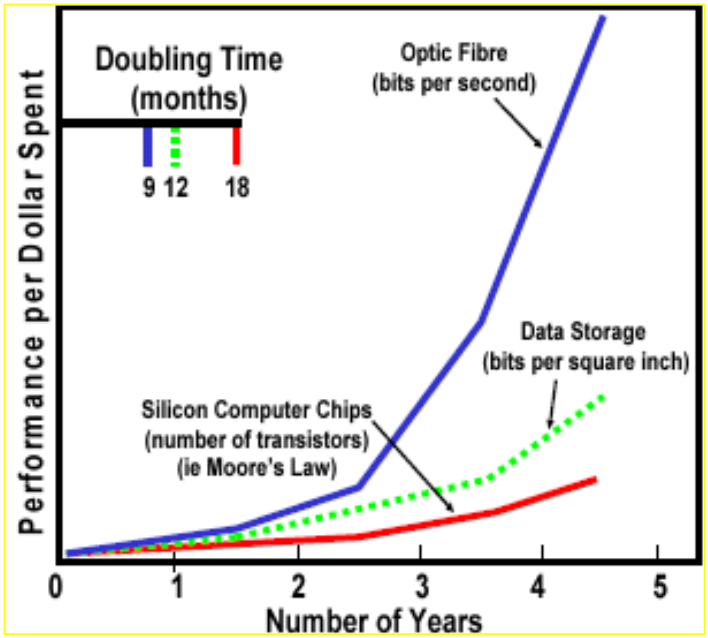


Figure 9: Moore's law vs. storage improvements vs. optical improvements [220]

Ian Foster and Carl Kesselman, pioneers of the grid, proposed a definition in 1998: "*A computational grid is a hardware and software infrastructure that provides dependable, consistent, pervasive, and inexpensive access to high-end computational capabilities*" [221]. In 2000, with Steve Tuecke, they added to the original statement: grid computing is concerned "*with coordinated resource sharing and problem solving in dynamic, multi-institutional virtual organizations*" [222]. Then in 2002, Ian Foster modified again his definition arguing: the grid is "*a system that coordinates resources that are not subject to centralized control, uses standard, open, general purpose protocols and interfaces, delivers non-trivial qualities of service*" [223].

But several scientists differ [224,225] or propose other definitions [226-234]. For instance, a general definition is given by Andrew Grimshaw: "*A grid is all about gathering*

together resources and making them accessible to users and applications". He proposes also a more detailed definition: "*A grid enables users to collaborate securely by sharing processing, applications, work flows and processes, and data across heterogeneous systems and administrative domains for collaboration, faster application execution, and easier access to data*" [235].

The grid promises to share resources for scientific collaborations on an unprecedented scale [222,236]. The computing grid name comes from the well-known analogy with an electrical power grid [237]. The computing power would be delivered just like electricity from an outlet, without knowing where the power came from or its complexity and reliability. An obvious similarity between computational and electrical grids is that both aggregate heterogeneous power sources (thermal, hydro or nuclear power and workstations, clusters or supercomputers). But the comparison is limited in many areas probably because the grid field is much younger than the electrical power grid. For instance, a computing grid still needs an operational model, coordinated systems operation to ensure network stability, and ease of use [238].

Today the grid concept can be applied in all public or private sectors: science, business, engineering, medicine, culture etc. [236,237,239]. For instance, a scientific collaboration can pool computers and storage across participating institutions to obtain instantaneous access to more resources than available locally [235]. The resources may be administered by different organizations, distributed and heterogeneous. Gathering together resources (e.g. computing power, data, applications…) and making them accessible in a secure manner to users and organizations involves dealing with the physical characteristics of the Grid. This differs from the Internet where the user has to choose to which machine he wants to connect and which information he wants to retrieve out of the tremendous amount of data available [240]. The end-user does not need to know where the resource is physically located, the type of machine it is on, that it may have failed and recovered, etc. It requires virtualization of the shared resources and high-level interfaces [241]. Of course, grid computing power is no substitute for developing intelligent software: the grid is a tool to enhance research and collaboration.

2.2.2. A grid taxonomy

Instead of concluding on a universal definition of the grid, this section presents briefly 3 different grid architectures to reach the end-user goal. Most grids fall into one or several of these categories: computational-oriented grid, data-oriented grid and knowledge-oriented grid [237].

Computational grid

A computational grid is defined here as an environment providing access to a network of distributed processors. Processors are located in workstations (e.g. Seti@home [242]), clusters (e.g. EGEE [243] or OSG [244]) or/and supercomputers (e.g. DEISA [245] or TeraGrid [246]) [247,248] across multiple organizations. Computational grids can support distributed and parallel computing [221,249], high throughput computing [221] or on-demand

computing [250,251]. Different services are available to submit jobs, monitor them and collect the results. Moreover, due to the processor heterogeneity (speed, architecture, software platform, memory, storage, connectivity etc.), information management is needed to select the best node for such applications and to use it efficiently. Furthermore, it requires basic data management like data transfer.

This approach is adapted to application, which can be easily divided into jobs that are solved independently. Such a task is called an embarrassingly parallel job. Many existing resource-intensive applications are using grids for computing in science, engineering and commerce. These include Monte Carlo simulations and parameter sweep applications, such as ionization chamber calibration [252] or drug design [253]. The advantages are numerous: allowing on-demand aggregation of resources at multiple sites, processing data applications, reducing execution time, providing access to remote databases and software, taking advantage of time zone and random diversity (in peak hours, users can access resources in off-peak zones), providing the flexibility to meet unforeseen emergency demands by renting external resources for a required period [238].

Data grid

A data grid [254-256], or information grid, is defined here as an environment providing access to distributed data [257]. Another definition is an environment that provides a dynamic logical name space for coordinated sharing of heterogeneous distributed storage resources based on local and global policies across administrative domains. Data are public or private databases (e.g. BIRN [258]), files (e.g. Gnutella [259]) or metadata. The metadata represents information about the data itself (files instance, origin, relationships between data…). Different services are available to collect, query, move, replicate, integrate and analyze the distributed data [260]. Moreover, computational power is required at least to organize the data.

This approach is adapted to data access and sharing application such as medical data sharing. Few applications are using grids only for data management [254,261]. Data grids need to support the access to a large amount of data [262] in dedicated data storage devices or flowing from scientific instruments and sensors [263]. Secure remote access to databases can enable cross-correlations among databases maintained by different groups or within a single company. Because of the cheaper cost of disk space, memory space and computing cycles in comparison with bandwidth, it is better to avoid moving data in large quantities [250]. This implies computing close to the data.

Knowledge grid

A knowledge grid [264-266] is defined here as an environment providing access to distributed knowledge understanding and manipulation. Knowledge can be defined as the sum of all types of data and information within the scope of interest and is composed of relevant databases, information sources, document/knowledge bases, metadata and a knowledge map [267]. Knowledge needs a semantic ontology [268,269]. Ontologies are a formal way of representing knowledge in which concepts are described by their meaning and their

relationship to each other [270]. Different services are needed like the distributed mining and extraction of knowledge from data repositories available on the GRID.

This approach is adapted to respond to high-level questions and finds the appropriate processes to deliver answers in the required form. Today, there is no knowledge grid operational on grid infrastructure, even if several initiatives exist [271,264]. It requires computational power and data management to manipulate concepts, to automate the process of knowledge discovery and to run complex *in silico* experiments [272,273]. Knowledge grid should support virtual laboratories (e.g. myGrid [274], Virtual lab [275]) and is particularly important for pharmaceutical research [276].

2.2.3. Grid architecture

From this taxonomy, it can be concluded that the complexity of the underlying hardware resources needs to be managed and hidden by a software layer providing services for users [235].

Thus, figure 10 presents a common grid architecture schema [233,235,277,278].

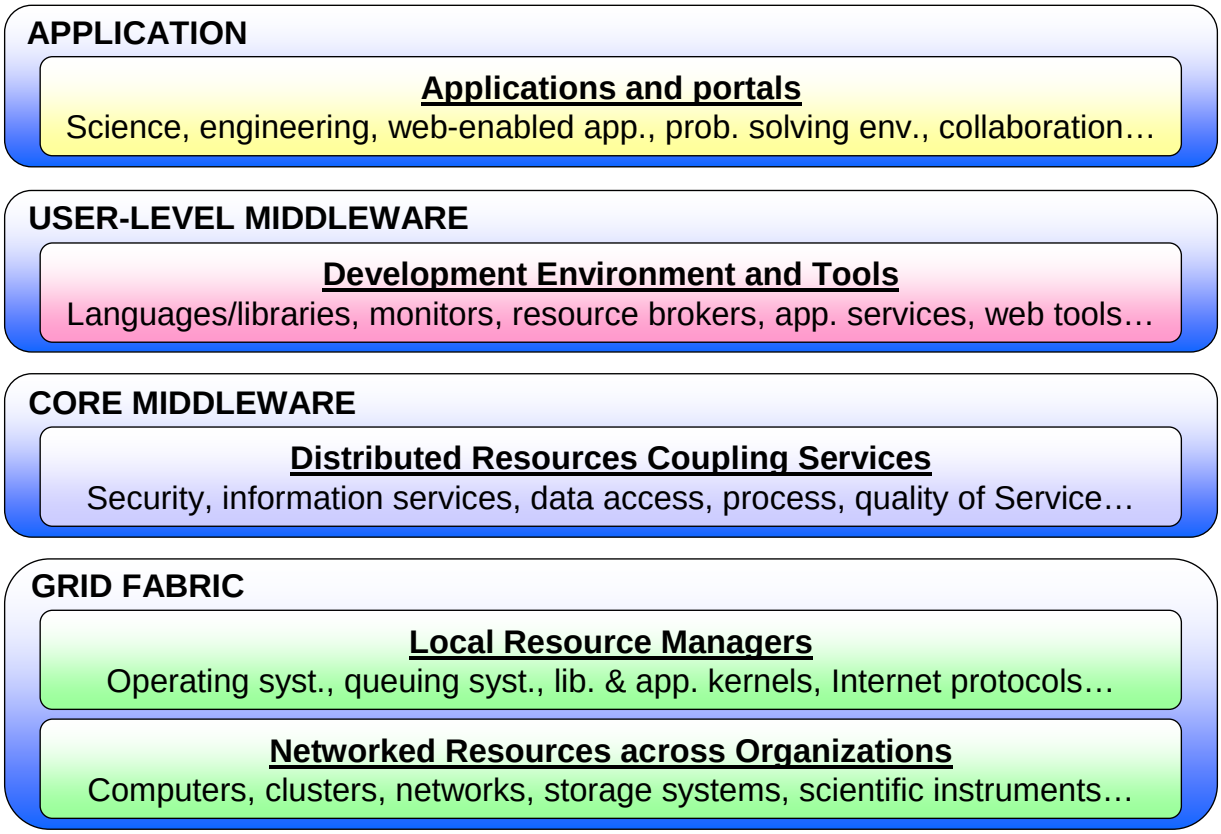


Figure 10: Example of common grid architecture

The grid fabric layer consists of distributed and various resources across organizations controlled by local resource managers. The core middleware layer offers services to abstract the complexity and the heterogeneity of the fabric level. The user-level middleware layer provides higher level abstractions and services thanks to the core middleware interfaces. The

applications and portals are developed using grid-enabled programming environment and interfaces and the services provided by the user-level middleware.

This architecture is a model, but there are multiple types of grids, according to their goals, their technologies, their ownership or their hardware for instance.

2.3. Desktop and cluster grid computing

Hardware resource is another way to classify grids. Those resources may include supercomputers, clusters, instruments such as telescopes and microscopes, computer-controlled factory floor tools, servers, desktop machines, laptops, PDAs etc. [235]. This section presents and compares the 2 most common grids: the desktop grids and the cluster grids. They are specially adapted for CPU (Central Processing Unit) intensive and embarrassingly parallel applications, which are involved in the *in silico* drug discovery process.

2.3.1. Desktop grid overview

Motivation

In the last decade, observers have reported low CPU usage in desktop PCs around the world [279-282]. Many of these cheap PCs are connected to the Internet. At the same time, the demand for CPU power, in science for instance, has exploded [283]. Consequently, several initiatives were proposed to take advantage of this available computing power. For instance, Seti@home initiative [242] sustained a processing rate of about 60 Teraflops for several years [284].

A desktop, or scavenging, grid [285] is defined here as the sharing of idle desktop workstations, or PCs, cycles to solve scientific problems. This computing is also called global or internet computing [286]. The computing resources can be available internally to an organization like a company. When the computing resources are freely available from the public through the Internet, the desktop grid is called a volunteer grid [287,288]. This definition requires voluntary participation as opposed to parasitic computing that can force computers on the Internet to produce results [289].

A desktop grid needs to be potentially deployed on millions of machines. The next paragraph presents an example of deployment.

Deployment

Desktop grids were inspired from peer-to-peer computing [290,291]. In true peer-to-peer computing, there is no central authority; thus each node not only works, but also can schedule new tasks. But for a desktop grid, one or several central authorities control computation of the work units, automatically upload input and collect outputs through the Internet, check output integrity and facilitate analysis. The computing task is sent to the client by the server upon client request [283].

Figure 11 presents an example of a project deployed on a desktop grid using the BOINC framework. BOINC (Berkeley Open Infrastructure for Network Computing) is one of the main software platforms for public-resource computing that provides built-in support for distributed computing on heterogeneous PCs connected to the Internet or to Intranet networks [284,292].

The initial step is the project preparation: the framework is installed on different project servers (pink components) and the project-specific components are developed and integrated into the framework (green components). Then, the desktop computers of participants are registered in the BOINC database thanks to the web server (1). The client software is then automatically installed on them. The BOINC database contains information about project applications, participants, work and results (2). First, the client computer connects to the server and asks for work (3) (this is called pull-mode). Then the scheduling server downloads the instructions, the applications and input files for the core client (4). Once the computation is done, the results are uploaded onto the scheduling server and the output files are stored thanks to the data servers (4). Validation and post-processing are then executed (5).

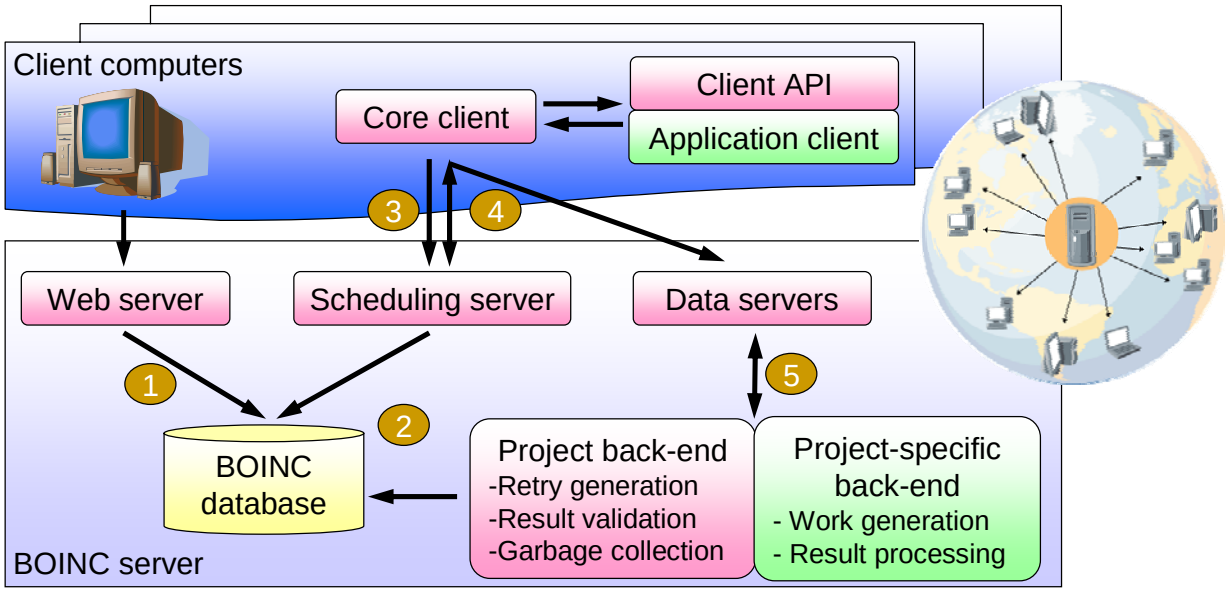


Figure 11: Example of a project deployed on a desktop grid using the BOINC framework

There are many criteria required for the deployment of an application on a desktop grid [283,293-296]:

- The application must be divisible into independent work units of an appropriate size. Depending on the network, the amount of data transferred should be small and rare.
- The software must run on multiple operating systems. Download must be fast and not require a lot of disk space. The process to download and install the software must be executed automatically. The code upgrade must be automated.
- The process to submit jobs and to retrieve the output must be executed automatically. The software has the lowest work priority on the PC. The work is only carried out

when the screen-saver is active for less interference. Virtual memory should not be used to avoid blocking the rest of the system. The intermediate results may be saved to minimize losses in case of system crash.

- It should be noted that network bandwidth and Moore's law impose lower and upper computing time limits for tasks that are worth distributing [297]. The resulting range from seconds to months is reflected in current global computing projects, with several hours as a typical value.
- The process must meet the security standards. The execution environment protects the participant's PC from potentially malicious code from a computing application. Different methods are used to control the output integrity against hardware malfunctions, incorrect software modifications, or malicious attacks. Encryption and user keys can also be used.

Furthermore, several features are required for an application in a volunteer grid :

- The application must be readily available.
- The output must not be lost if the participant wishes to stop his participation or stop the automated online connections.
- The application and the application web site must have good graphical user interface and great content about the project to attract and keep participants. Progress and statistics should be displayed.

Thus only certain types of applications are useful for a desktop grid. But there are many initiatives to build frameworks with different technologies and to use them in applicative projects.

Initiatives

Many projects have used or are using BOINC, including Climateprediction.net [298], LHC@home [299] or Einstein@Home [300]. Other projects develop their own software framework, like Condor [301], XtremWeb [302] or Folding@home [150,303]. United Devices [304], Entropia [305] and Platform Computing [306] are the main commercial providers of distributed computing platforms. They have the same server functions as BOINC and use relational databases to store task participant data [307].

The best known initiative is SETI@home [279,308], for which the framework is BOINC. Millions of participants processed a database of large pulsar signals in a search for extraterrestrial intelligence. Radio data signals are distributed through the internet by a central server to PCs, where they are analyzed by a screensaver program to assess the presence of a non-random signal that runs while the computer would otherwise be idle. The notion of virtual organizations [222,309] and virtual enterprises [310] emerged from this success. The understanding that 10,000 desktop PCs with an average performance of 500 megaFLOPS and appropriate software are equivalent to a 5 teraFLOPS supercomputer developed a computational economy for sharing and aggregating resources to solve problems [283,311].

As a consequence, there are many different desktop grid initiatives in life science [283]. Predictor@home [149] is using the BOINC framework to accommodate both protein structure

conformational sampling and protein refinement. World community grid [312] is using United Devices and BOINC platforms to support the Help Defeat Cancer project, fightAIDS@home and the Human Proteome Folding project. This last project is being carried out in collaboration with grid.org [313]. Grid.org is using United Devices platform to support the Smallpox Research project, the Anthrax Research project and the Cancer Research project. French Decrypthon [314], is using the United Devices platform to accelerate the proteomic and genomic research in human diseases. D2OL [315] is using its own platform to screen large compound databases against targets of different diseases such as malaria and avian influenza. Many pharmaceutical companies, such as Bristol-Myers Squibb and Novartis, are using idle time of thousands of desktop computers. They acquire teraflops of cheap computing power for their drug discovery research.

Desktop grids are highly used by different projects, but they are mainly only for computational projects. A cluster grid offers a larger variety of services.

2.3.2. Cluster grid Overview

Motivation

A cluster is a set of computing units physically gathered in the same place and coordinated to improve computer capacity and storage. They are used in many different areas such as science organizations and companies. Clusters can have different sizes and can be composed of heterogeneous PCs. The basic required services are load balancing and back-up mechanisms. Two of the advantages of this approach are the scalability and the cost: a cluster can grow simply by adding new cheap PCs to it. But this growth will be limited by the need to communicate between computers. An answer to the increasing requirements in computing and storage is the building of cluster grids inside or between organizations [221]. For instance, the need to manage the future huge data production from the Large Hadron Collider of CERN in 2007 [316] has motivated the high energy physics community to interconnect all clusters of their European computing centers and research laboratories with large bandwidth network connections.

A cluster grid [317] is defined here as the sharing of geographically distributed clusters to solve problems. They allow selection and aggregation of distributed resources, such as instruments across multiple organizations. This enables exploration of large problems with huge data sets, but also small but daily needs. High-level services for complex applications can also be built on cluster grid [276]. They can be the infrastructure for data and knowledge grids, supporting collaborations and expertise.

A cluster grid may interconnect hundreds of clusters supporting each day thousands of jobs. An example of deployment is presented below.

Deployment

A cluster grid is the combination of networked resources operated by common or interoperable middleware(s), which provides services for communities structured usually in virtual organizations [243]. The middleware is a set of software components that sits between

the user application and the underlying resources. It provides services to access and to use resources. A virtual organization is a grid-wide identification and authorization unit representing a community of users sharing some grid resources. Inside this virtual organization, there is some degree of trust, accountability, and opportunities for sanctions in response to inappropriate behavior [278].

Figure 12 presents an example of project deployment on a cluster grid [233] using the Globus Toolkit. The Globus Project is developing open-source and standard-based technologies to build computational grids. It is one of the main world-wide middleware for cluster grids [318].

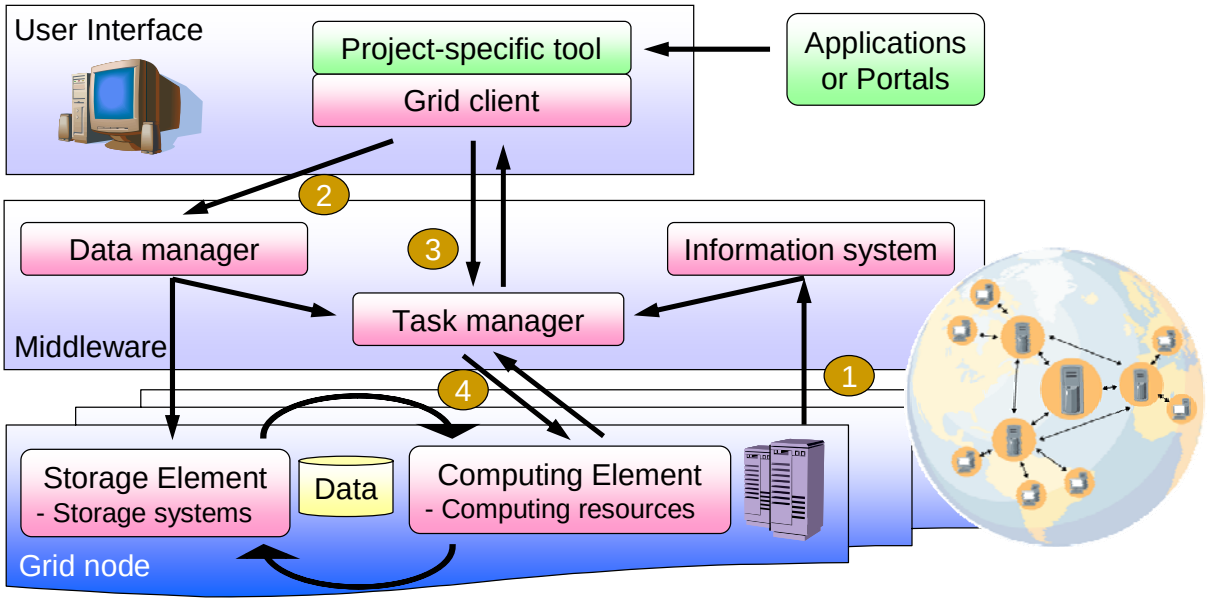


Figure 12: Example of project deployment in a cluster grid

The initial step is the project preparation. The management services and the user interface, which is the grid entry point for the user, are installed on workstations (in pink). Only three services are figured here, but there are many other possible services in a middleware, such as security, user management, accounting, etc. The computing element is the service representing the computing resources which here is a cluster. The storage element is the service representing the storage resources which here is a disk. The project-specific tool on the user interface is developed to interface the applications with the grid tools to access the grid (in green). Then, available elements in the grid nodes are published in the information system (1). The next possible step is the upload of the data on the storage elements and their registration thanks to the data manager (2). After that the application tasks are submitted by the user from the user interface on the task manager with the application software and optionally other data (3) (this is called push-mode because the client submits work). The task manager asks the information system and possibly the data manager to select the most adequate resources according to the task description. Next, the tasks and their data are submitted to the computing elements to be executed possibly with copied data from the

storage elements (4). At the end of the task, the output can be stored on a storage element and registered in the data manager. The task report and the data outputs are transferred on the user interface through the task manager, to be checked and processed by the project-specific tool.

Many different applications and uses are possible on a cluster grid notably thanks to the infrastructure flexibility, its permanent availability and the services developed in the different projects [319].

- The application might be divided into independent work units of an appropriate size. Parallel jobs are also appropriate when using only one cluster. Depending on the network, small or large data can be manipulated easily.
- The software may be installed on one or multiple operating systems, depending on the cluster grid middleware. The installation process is more or less easy depending on the proposed services.
- The applications are managed by the task manager and are executed on resources. Resource specifications of the cluster work units are registered in the information system and exposed to the task manager.
- The process has security features thanks to the services and the existing trust between the different partners in the virtual organization. Several tools are available such as certificates, public key infrastructure for authentication and integrity, encryption, usage/access policies, etc. The control of the job output is under the responsibility of the application specific tool.

Many different applications can be deployed on a cluster grid. The number of cluster grid projects and technologies has vastly increased in the last decade. A few of them are described in the next paragraph.

Initiatives

There are an increasing number of applications that employ cluster grid technologies [320-327]. Some of them integrate sensor networks with grid computing to enable real-time modeling and data collection [328]. The Grid is laying the foundation of e-Science, a new way to carry out research through distributed global collaborations enabled by the Internet [283]. E-science can be defined as digitally enhanced, or electronic, science in global collaboration. Many large collaborative e-science projects worldwide are now applying these technologies in different disciplines [329-333]. Promising grid-enabled virtual laboratories are in progress [276]. In the long term, the grid is going to impact the organization, the resource management, the network, the security and the administration of the community concerned.

Several grid projects orientated towards the life sciences are underway, such as the North Carolina BioGrid [334], the Canadian BioGrid [335], the EUROGRID project [336], the cancer biomedical informatics Grid [337-339], the Simdat project [340] and the Biomedical Informatics Research Network [258]. All these projects involve the sharing of computational resources and the large-scale movement of data. One example is OpenMolGrid [341], an extensible, grid-enabled environment using UNICORE for the molecular science and engineering system that is effectively used in molecular design [342,343]. Other projects

focus on grid-enabled frameworks for developers or users, like P-grade [344] which provides a visual environment for application development, Triana [345] for workflow formulation, GridSphere [346] for creating web portal environments or Genius [347] for job submission and data access.

There are high-quality grid middleware such as Globus, Nordugrid ARC [348], LCG-2 (LHC computing Grid release 2) [349] and gLite [350], Legion [351], Gridbus [352], UNICORE [353], e-Toile [354] or Grid'5000 [355]. These software act as middleware interlinking resources of multiple computers within or between institutions using open and general purpose protocols for secure high performance distributed computing [356]. For instance, UNICORE is an open-source grid middleware, which offers a fully integrated solution consisting of a powerful and easy-to-use Java-based graphical user interface and server side components. Several chemistry- and biochemistry-related applications have already been successfully integrated [357-359].

In addition, several commercial organizations, such as IBM, Sun, HP, Oracle and Nice are actively involved in the development of enterprise and global utility grid technologies [239]. Some companies host their own private cluster grids, although deployment is still complex [360]. Sun recently proposed Internet-based access to a utility computing resource at the affordable price of $1 for 1 hour of computing. Many economic models are studied [361,362].

A need has emerged for communities to have standards and a standards-based architecture that would facilitate better interoperability among various grid middleware systems and Grid-enabled applications [338,363]. The Open Grid Services Architecture (OGSA) [364,365] based on web Services Resource Framework [366] aims at unifying web and grid service frameworks. A web service is a software system designed to allow inter-computer interaction over a network. Web services allow grid developers to take advantage of standard message formats and communications mechanisms for communicating between heterogeneous components and architectures. OGSA standards integrate Globus Toolkit technologies with web services mechanisms [367]. To facilitate standardization of fast-evolving Grid technologies, the Global Grid Forum [368] was founded. The OGSA-DAI [369,370] is an implementation of the OGSA Data Access and Integration Standards developed within the Global Grid Forum. It provides tools and runtime support for development and deployment of data services in the grid.

2.3.3. Comparing desktop and cluster grids

Desktop and cluster grids are the main widespread grids. It is legitimate to compare them in order to choose the best infrastructure to deploy *in silico* drug discovery for neglected and emerging infectious diseases. Table 1 compares desktop and cluster grids [257,283,371,372].

The main added values of desktop grids are their deployment simplicity, their infrastructure and deployment costs and the computing power gain. This computing power gain is potentially huge for volunteer grids. Moreover, desktop grid technology has proved its maturity, even if the platforms are still under development: they can generate production

discontinuities on the grid management server because of software update or hardware failures [149].

Features	Desktop grid	Cluster grid
Deployment simplicity	Yes	No
Infrastructure and deployment costs	Cheap	Expensive
Computing power gain	Variable, potentially huge	Variable
Applications scope	Limited	Extended
Data management	Very limited	yes
Network bandwidth	Expensive, scarce	Abundant
Availability and reproducibility	Variable	Yes
Additional services	No	Yes
Resources management	No	Yes
Secured resources	No	Yes
Standards	No	In progress
Enabling collaboration	Very limited	Yes
Code stability	Good	Many updates
Fault tolerance	Limited	Limited
Security	Limited	Limited

Table 1: Comparison of desktop and cluster grid features

The main disadvantages of desktop grids are the limitations on applications deployment: only embarrassingly parallel applications with limited data management requiring low network bandwidth can be deployed. Another limitation is the intermittent participation time of the participating PCs: the average connection time of the PC can be only 28% [373]. Global security management is a critical challenge and still an issue for the user who provides data and application, but also for the participant that installs the software at home [149,283,374]. Fault tolerance is a major criterion to deploy a production grid [330,375].

Computational errors management is difficult for a desktop grid [149,259], even if methods like computing replication contribute to solve this problem [295].

An important added value of cluster grids is that they accept a larger number of applications from all areas thanks to high network bandwidth, large data management and better availability of resources. Furthermore, beyond the 24 hours a day availability of resources, the cluster infrastructure guarantees a stable environment for the scientist for storing results and rerunning experiments. Many complex services can be implemented, such as data replication, metadata management, workflow management or service discovery [319]. Standard development for infrastructure and service interoperability is in progress [278]. Moreover, cluster grid resources are powerful, reliable, flexible and secure. Many security features exist in a cluster grid: authentication [221], authorization [376,377], data encryption etc. There is trust between users of a community (a virtual organization) and the resources administrators in the project. For instance, it is hard to become an authorized user. But the process is not yet fully secure, particularly in the area of data management (data safety, legacy…). For all these reasons, a cluster grid environment enables international collaboration and expertise sharing.

But cluster grids are still complex to deploy (technically, but also organizationally between partners), clusters are more expensive to maintain, and their potential computing power is lower than a world-wide volunteer grid. Moreover, even if computational grids can now produce research results, the code stability of a cluster grid is not yet reached. Another limitation is the fault tolerance. Resubmission systems exist in a cluster grid, but the success rate is still low.

Thus desktop and cluster grids seem complementary and could be used simultaneously in a project or in an organization. However, the advantages of cluster grids are more important for a large number of applications and users. This environment is starting to give a stable environment for building complex experiments and knowledge spaces although there remain a few challenges [257]. Poor reliability is not acceptable for a production environment, and particularly for users who pay to access the grid. The security aspect is vital for the involvement of companies or sensitive public research (nuclear power stations, drug discovery…). The grid needs to be hidden to the user. Easy-to-program interfaces are necessary for application developers.

To conclude this comparison, a cluster grid seems more relevant to develop an *in silico* drug discovery process to fight neglected and emerging infectious diseases. Thus, the applications developed during this thesis were deployed on cluster grid environments. These grid infrastructures and technologies are now described.

2.4. Some cluster grid infrastructures: EGEE, AuverGrid, TWGrid and RUGBI

In the previous sections, the advantages of cluster grids were established. The drug discovery process is typically composed of a set of CPU consuming applications with large

data sets to analyze. Beyond the computational aspects, cluster grids open perspectives for building a collaboration space enabling the discovery of drugs against neglected and emerging infectious diseases.

The performance of a cluster grid infrastructure depends on many criteria [92]: maturity (robust, powerful), value (impact on cost, time and success probability), competitive advantage (sustainable), organization (drive change), deployment (ease of use, ease of configuration, non-intrusiveness at participating sites), success (new users, results). Today no grid projects meet all these quality requirements.

The European EGEE grid infrastructure is one of the most advanced production grids, satisfying several of these criteria. The French regional AuverGrid and the national Taiwan TWGrid are 2 daughter infrastructures of EGEE. The RUGBI infrastructure was built specifically for life science industrials and academic institutions to meet some specific requirements. Thus these 4 infrastructures, on which bioinformatics services were deployed, are now briefly described.

2.4.1. The EGEE infrastructure

The EGEE (Enabling Grid for E-sciencE) [243] project brings together experts from over 27 countries with the common aim of building on recent advances in Grid technology and developing a production Grid infrastructure which is available to scientists 24 hours-a-day. The project aims to provide researchers in academia and industry with access to major computing resources, independent of their geographic location. The project was built on the successes of the European DataGrid Project [378]. The EGEE project is at this moment in its second phase (EGEE-II).

The EGEE infrastructure is now the largest production scientific grid in the world with a large number of applications installed and used on the available resources [379]. Expanding from originally two scientific fields, high energy physics and life science, EGEE now integrates applications from many other scientific fields, ranging from geology to computational chemistry. The infrastructure, whose map is presented in figure 13, involves more than 200 sites spread across Europe, America and Asia, 20,000 CPUs, 10 petabytes of disk space, 10,000 concurrent jobs and 60 Virtual Organizations. The production service is based today on the mixed LCG-2/gLite services. The grid is organized hierarchically, with resource centers that are under the responsibility of Regional Operation Centers (one per federation) which themselves are coordinated by the Operations Management Center (OMC). The goal of this hierarchy is to offer an efficient, responsive and scalable grid service to the users.

Applications are deployed within the framework of a Virtual Organization. A Virtual Organization is a grid-wide identification and authorization unit representing a community of users sharing some grid resources. For instance, the biomedical Virtual Organization scaled up to about 3,000 CPUs and 21 TB disk space in the summer of 2005.

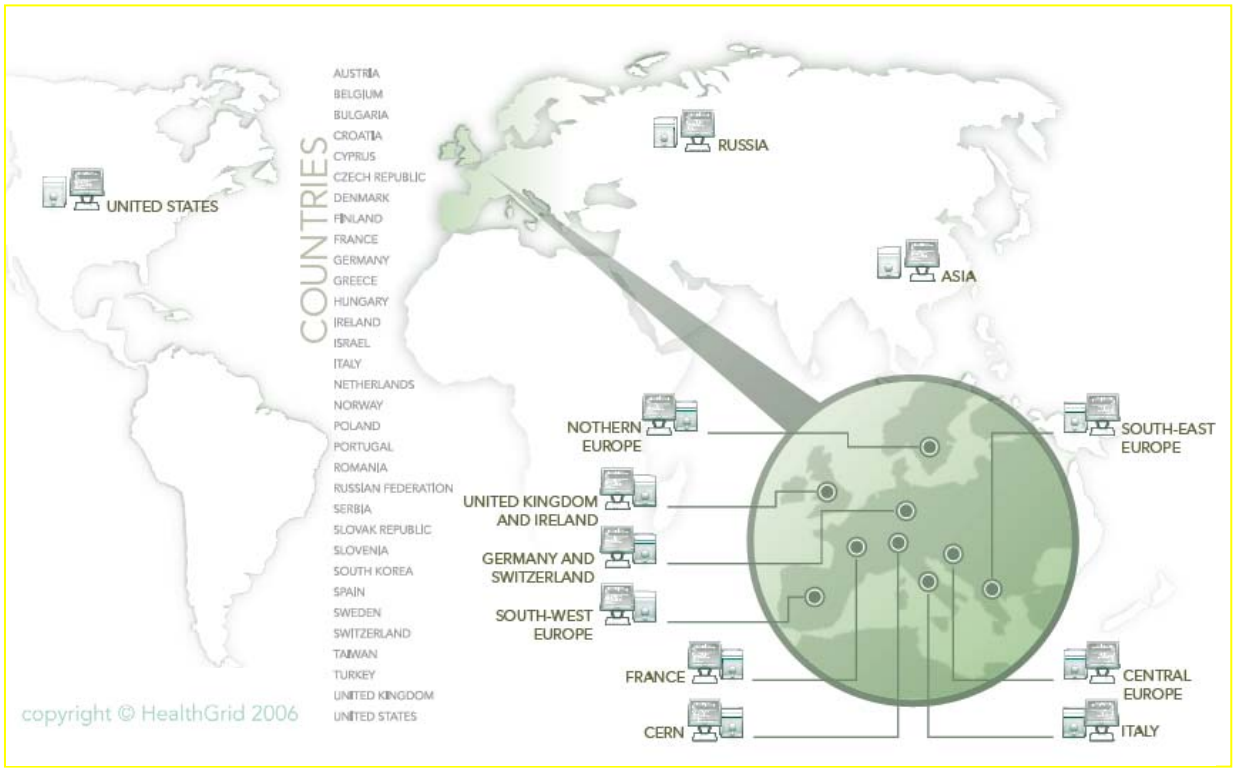


Figure 13: Map of the world-wide infrastructure EGEE with Operating Centers

EGEE is connected with several infrastructure projects, like NAREGI [380] in Japan and OSG [243] in the USA. It supports other infrastructure projects like SEEGRID [381] in South-Eastern Europe and EELA [382] in South America. Other related projects address specific high-level services such as DILIGENT (digital library) [383] or specific application areas such as BIOINFOGRID (bioinformatics) [384]. Furthermore, the EGEE project wants to support and promote the development of an African Grid infrastructure to benefit the continent and help reduce the digital divide. To achieve this, it promotes the extension of recent connections to the European Grid infrastructure, and the leveraging of this infrastructure for the development of a sister network in Africa [385].

The EGEE infrastructure demonstrates its success by the number of active applications and the new projects that want to collaborate with it. Among the infrastructures based on the same technology as EGEE, we are going to describe the AuverGrid and the TWGrid infrastructures.

2.4.2. The AuverGrid infrastructure

AuverGrid [386] is the first French regional grid, deployed in the Auvergne region. Its goal is to explore how a grid can provide the resources needed for public and private research at a regional level. AuverGrid aims to share technological expertise and to transfer grid competences from European projects to public and private regional actors. With more than 800 CPUs available at 12 sites, AuverGrid hosts a variety of scientific applications from particle physics to life science, environment and chemistry. The infrastructure enables pluridisciplinary collaboration between institutes and industries. The production service is

based on LCG-2 services. Some application services created in the RUGBI project are used now in AuverGrid. Figure 14 illustrates the transfer of expertise from the EGEE and RUGBI networks to the AuverGrid project [387]. CC-IN2P3, LPC and the Biopôle of Clermont-Limagne are stations of three lines, the EGEE, RUGBI and AuverGrid infrastructures. They allow the transfer of expertise from Europe and France to many institutes of the French region Auvergne.

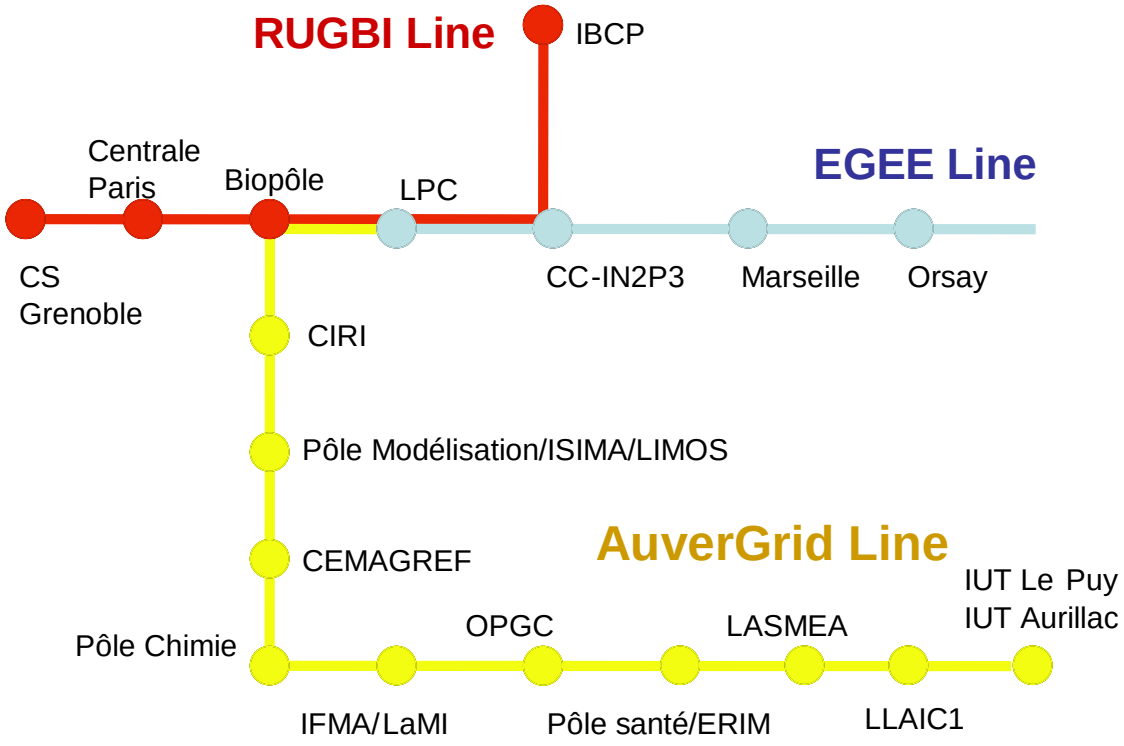


Figure 14: Map of the infrastructure AuverGrid [387]

AuverGrid is a successful example of an infrastructure integrating new institutes and new applications. The deployment of the thesis applications proved its reliability.

2.4.3. The TWGrid infrastructure

The Taiwan Grid, TWGrid [388], is responsible for operating a Grid Operation Center in the Asia-Pacific region. Apart from supporting a world-wide Grid collaboration in high-energy physics, TWGrid is also in charge of federating and coordinating regional Grid resources to promote the Grid technology to e-Science activities (e.g. life science, atmospheric science, digital archive, etc.) in Asia. The production service is based on the mixed LCG-2/gLite services.

TWGrid aims first to support the high energy physics community. But it has also integrated new applications, like life science. Their central role in the Asia-Pacific region proves the reliability of their infrastructure.

2.4.4. The RUGBI infrastructure

The goal of the RUGBI project [389] is to design and deploy on the basis of free technologies and existing infrastructures a computing grid offering a set of services to analyze proteins. These services aim to support academic biologists and small and medium life science enterprises. They were identified through an analysis of the needs of three French biotech companies located at Biopôle Clermont-Limagne. Today the RUGBI grid offers direct, secure and high-performance access to different software and databases, listed in table 2. The RUGBI grid is able to be enriched easily by new free or commercial tools thanks to the deployment service described in chapter 3.

Category	Resources
Alignment tools	Blast (MPI), Clustal, Fasta
Tools for protein secondary structure prediction	Gor, Simpa, Predator, Sopm, Dsc, Phd
Metabolic pathways identification tool in network	DDTool
Virtual screening tool	Autodock
Genomics and proteomics database	EMBL, UniProtKB
Metabolic pathways database	KEGG
Structural database	PDB, NCI Diversity set

Table 2: Tools and databases available on the RUGBI grid.

There are two portals to access the RUGBI grid. SecProt [390] gives an anonymous but limited access to protein secondary structure prediction methods. The general RUGBI portal [391] provides all available services but requires personal registration to control the access to logical and physical resources. It is hosted on several servers to avoid saturation by a unique entry point. The user can manage his applications, his data and his task flows; he can also manage his profile, rights and even his group (like a company or a team).

Figure 15 presents the RUGBI architecture. The RUGBI grid currently has 5 sites in Clermont-Ferrand, Lyon and Grenoble interconnected by a high speed network. They can have different components, such as a web portal to access the grid, a Controller server to manage the grid, an exploitation server for grid administration, a computing cluster and a storage bay. Users from academic institutions and Biopôle Clermont-Limagne access the grid with a client and a certificate.

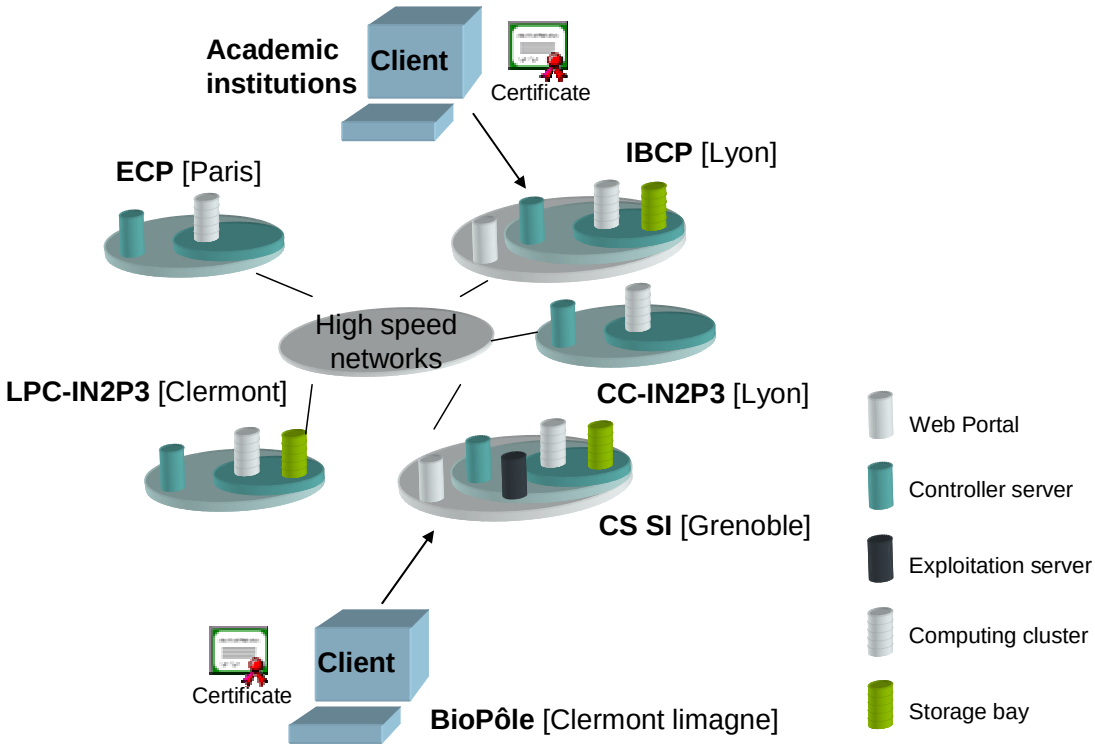


Figure 15: RUGBI architecture with the node composition

The RUGBI infrastructure is young, but it is based on web services technology. This project was chosen to explore the creation of new bioinformatics services during this thesis, because it offers a Service Oriented Architecture.

2.5. Some cluster grid technologies: EGEE and RUGBI

An infrastructure is based on the middleware, which is the software layer which interconnects resources and services. During the thesis, the middleware of the EGEE, AuverGrid and TWGrid infrastructures was LCG-2. The middleware of the RUGBI project was built on web services. This section gives the main concepts of the grid technologies used in EGEE and RUGBI.

2.5.1. The EGEE grid technology

The grid relies on advanced software, called middleware, which interfaces resources and applications. The LCG-2 middleware deployed on the EGEE infrastructure provides many components: a User Interface, a Computing Element, a Storage Element, a Workload Management System relying on resource broker machines, a Data Management System, an Information System, an Authorization and Authentication System, an Accounting System, and several monitoring and installation services. All the grid activities and resource sharing within EGEE are operated and coordinated within the scope of Virtual Organizations, the virtual communities across laboratories and institutes around the world. Middleware services find convenient places for the application to be run, optimize use of resources, organize efficient

access to data, deal with authentication to the different sites that are used, run the job and monitor progress, recover from problems and transfer the result back to the user.

The main contributors to the LCG middleware are the Virtual Data Toolkit [392], the European DataGrid Project [378], the DataTAG Project [393] and the LHC Computing Grid [349]. The middleware LCG is based upon the Globus toolkit release 2. The main middleware components are briefly described [394] below. See [395] for a complete and detailed overview.

The User Interface

The User Interface is the gateway to the EGEE grid components. This machine hosts the personal user accounts and user's certificates. From the User Interface, a user can be authenticated and authorized to use the EGEE grid resources. It provides the command line interface and APIs to perform the grid operations.

The Computing Element

A Computing Element is built usually on a homogeneous cluster of computing nodes called Worker Nodes and a node acting as a grid gate or front-end to the rest of the grid. The Computing Element is responsible for accepting jobs and dispatching them for execution to the Worker Nodes through batch schedulers. On the Worker Nodes, all commands and Application Programming Interfaces for performing actions on the grid resources and grid data are available. Each EGEE site runs at least one Computing Element.

The Storage Element

A Storage Element provides uniform access to large storage space. The Storage Element may control large disk arrays or mass storage systems. Almost all EGEE sites provide one or more Storage Elements. The main file access protocol is GSIFTP, which is used for the file transfer. The GSIFTP protocol offers the functionality of FTP enhanced with the Grid Security Infrastructure security based on a X509 certificate (public/private key) infrastructure.

The Workload Management System

The Workload Management System is responsible for the management and monitoring of jobs submitted from a User Interface. The Job Description Language [396] is used for describing computation tasks. It specifies executable files, data dependencies and specific application requirements.

An example of Job Description Language for a basic "echo" job is presented in figure 16. The concatenated attributes "executable" and "arguments" give the full command line: "/bin/echo Hello reader!". The unix standard and error outputs are redirected automatically to the files stdout.log and stderr.log. The files will be downloaded on the User Interface at the end of the job thanks to the OutputSandbox attribute.

```
Executable = “/bin/echo”;
Arguments = “Hello reader!”;
StdError = “stderr.log”;
StdOutput = “stdout.log”;
OutputSandbox = {“stderr.log”, “stdout.log”};
```

Figure 16: Basic job attributes of a Job Description Language file

A set of services running on the Resource Broker machine helps to match job requirements in the Job Description Language file to the available resources (as gathered from the Information System), schedules the job for execution to an appropriate Computing Element, tracks the job status, and allows users to retrieve their job output. Figure 12 from the section 2.3.2 summarizes the process.

A sandbox is a service to pack and unpack a collection of files and to transfer it. The Input Sandbox service uploads the files from the User Interface to the Computing Element via the Resource Broker. The Output Sandbox service downloads the files from the Computing Element to the User Interface via the Resource Broker. The Logging and Bookkeeping service keeps information on a job status and allows the user to query its status. A status corresponds to a step in the job submission process. Current statuses are:

- “Submitted” corresponds to jobs submitted by the user through the User Interface and not yet handled by the Resource Broker. It corresponds also to jobs failed and automatically resubmitted by the Resource Broker.
- “Waiting” corresponds to jobs accepted by the Resource Broker but which are not yet allocated to a Computing Element.
- “Ready” corresponds to jobs for which the matching resources are found and which are submitted to a Computing Element.
- “Scheduled” corresponds to jobs accepted by a Computing Element and which are queuing for execution.
- “Running” corresponds to jobs executed on a Worker Node.
- “Done” corresponds to jobs for which the execution is finished.

User credentials for a limited lifetime (Proxies) can be automatically renewed through a Proxy Service. A proxy service offers a computer network service to allow clients to make indirect network connections to other network services.

The Data Management System

The Data Management System allows users to move files in and out of the grid, to replicate them among different Storage Elements, and to locate them. A number of available and supported protocols are employed for data transfer. Globus GridFTP protocol [397] is the most commonly used. GridFTP is a ftp protocol implementing secured and reliable multichanneled transfers, optimized for high bandwidths over wide-area networks using the Globus Security Infrastructure as security layer. A central file catalogue, the Replica Location Service, keeps information about file location and about some file metadata. A Logical File Name, given by the user, is associated to a file and its replica. It can be used to designate data

to be registered and retrieved independently of their physical storage location. The LCG File Catalogue replaced the Replica Location Service of the EGEE biomedical Virtual Organization during the spring of 2006. The LCG File Catalogue has the same functionalities as the Replica Location Service.

The Information System

The Information System provides information about the grid resources and their status. The information is published by the Grid Resource Information Service running on each individual resource (Computing Element and Storage Element). Then the information is propagated into a hierarchical database structure: firstly in the Grid Information Index Services at every site and then in the Berkeley Database Information Index as a central collector. The Grid Resource Information Service is based on the Globus Monitoring and Directory Service. Information is published following a specific schema that goes under the name of Glue [398].

Limitations of the EGEE technology

The EGEE project builds an efficient production middleware and infrastructure to provide scientists with large computing and storage needs. It is for instance appropriate for high throughput structure-based virtual screening, which requires huge computing power to analyze large databases. But there are some limitations for other applications such as protein structure prediction, which requires many different short tasks using frequently updated databases. This complex resource management increases the grid overhead time. The data and meta-data management is limited to some basic functions (transfer, registration, replication). Moreover, EGEE middleware is still complex to use and hardly accessible to users without skills in computer science. There is a need to build high-level services adapted to the specific needs of bioinformatics and offering a user-friendly and secure environment.

The RUGBI grid technology was built to meet these requirements.

2.5.2. The RUGBI grid technology

The RUGBI grid technology is based on web services. The first part of this section gives some definitions about the web services used by the RUGBI grid. Then the components of the architecture are presented.

Focus on Web services technology

This section is mainly based on [399]. The World Wide Web Consortium (W3C) [400] develops interoperable technologies (specifications, guidelines, software, and tools) to lead the Web to its full potential. The initial idea behind web services [401] was to enable the World Wide Web to become the support for real applications and to be a means of communication between them.

The web services specifications, recommended by the W3C, propose a set of standards and protocols allowing interaction between distant machines over a network. These interactions are made possible through the use of standardized interfaces which describe basically what are the available operations in a service, which messages are exchanged

(requests and responses), and where the service is physically located on the network and through which support. This interface, which is just a conceptual representation of an application written in a given programming language, is written in WSDL (Web Service Description Language) [402]. The typical file extension is ".wsdl".

The WSDL describes the structure of the data sent to and received from the service, the different operations supported on the service, how to communicate with the service, and finally where the service is located.

The glue between the services, or between a server hosting a web service and a client (any piece of software that will communicate with the web services), which enable them to communicate are these request and response messages. They can be described in a standardized way on the network and be exchanged with a standard protocol over basic http or SMTP or any common Internet protocol. All the messages and description languages are based on XML (Extensible Markup Language).

The XML is a way of describing data. An XML file can contain the data too, as in a database. Its primary purpose is to facilitate the sharing of data across different systems, particularly systems connected via the Internet. The advantages of the XML format are: the XML document transfer through the network is simple; the XML is understood by the Relational Data Base Management Systems or XML native Data Base Management Systems; the XML file is flat; the XML is extensible.

The XML native Data Base Management Systems is a database designed from the ground up to store XML data, like Apache Xindice. A known query language is XPath. The DTD (Document Type Definition) is a XML schema language used to describe the structure constraints of a XML document. It is like a grammar, defined by the XML user, allowing the checking of XML document validity.

The main language used to make web services communicate with each other is SOAP (Simple Object Application Protocol). SOAP is a protocol for exchanging XML messages over a network. It defines a certain structure for the XML messages (the SOAP envelope), and a framework that defines how these messages should be processed by the software. SOAP has the advantage of being implemented in several languages and toolkits.

Web services provide a convenient service oriented framework and this architecture is well suited for distributed system integration. They offer great interoperability (mainly because of standardized specifications). They enable the communication of processes and transfer of data independently of the programming language of the underlying applications. Therefore, by extension, virtually almost any piece of software can be exposed as a web service. They can be considered as firewall-friendly, because they are based on standard Internet protocols.

But web services are semantically poor. They are not adapted for transferring huge quantities of data. The performance can be worsened due to the overhead of sending XML messages.

Thus, to build an infrastructure, it is necessary to implement grid components, such as grid wide security standards, resource scheduling and other components. The RUGBI grid technology combines the web and the grid components.

Architecture

Figure 17 presents the different layers of the grid. The different components of the RUGBI grid are:

- The Computing Elements and the Storage Elements, composed of the operating systems, the batch manager, the grid components of Globus and the grid components developed by the RUGBI project associated to the web services interfacing to the Controller.
- The Controllers hosting the grid components and the web services, and associated to the database servers supporting the Information System.
- The grid portals interfacing to the Controllers.
- The exploitation server for logging, bookkeeping and grid administration.

All the elements, except the exploitation server, are distributed on the different nodes of the grid.

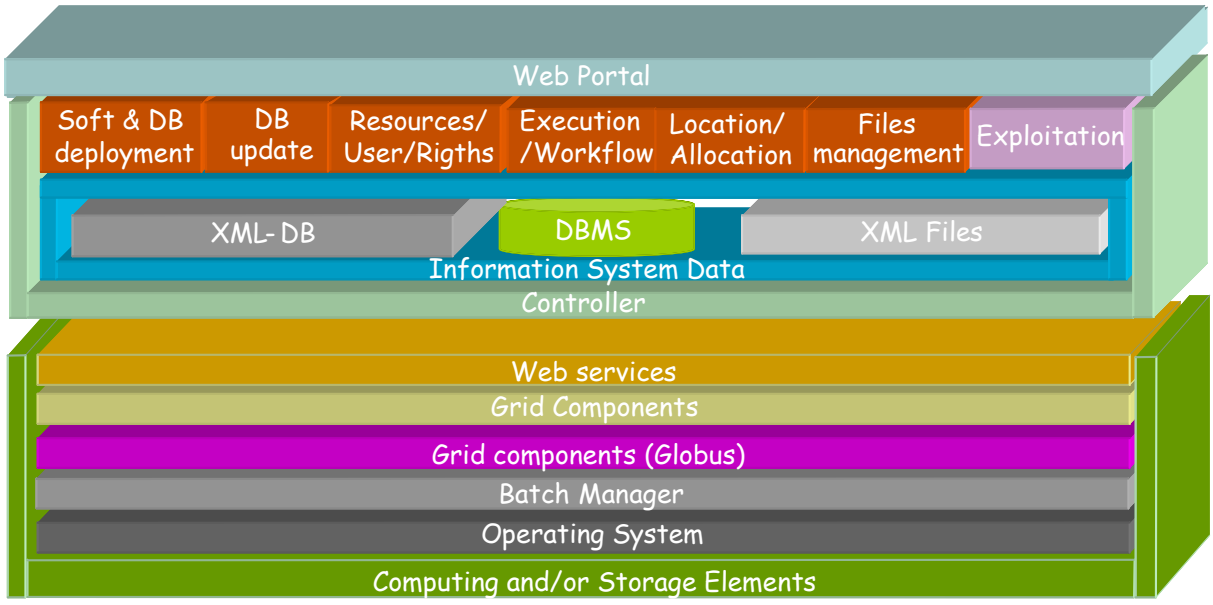


Figure 17: Multi-layer architecture of the RUGBI grid

The grid components provide the tools for the job submission (job submission, node selection), the tools for the job execution (logging, database management…), and the common tools to all components (Virtual Organization management on the nodes, user session management, cluster management…). For instance, the WorkflowEngine module [403] manages the different subtasks for an execution (job, data transfer, grid component invocation, fault-tolerance…). Most bioinformatics jobs require only a few minutes of CPU; contrary to other grid middlewares adapted to long tasks, grid components allow an efficient submission of short tasks. Another example is the DatabaseFinder module used mainly to find

the best location of a given database on the grid for a given user. This module ensures also that a database used by a job will not be modified or deleted during the job execution. This module is used by the database update service explained in chapter 3. The grid components are based on the Globus Toolkit (2 and 4). GridFTP is used for grid data transfer. This choice was based on the fact that GridFTP is quite standard, and used in several grid middlewares.

The Controller, thanks to web services, manages grid users and groups, manages grid components and logical resources (applications and databases), defines the rights to the resources, defines resource configuration, locates the resources (storage spaces and logical resources), executes the job work flows, allocates the jobs to the best sites, updates and synchronizes the Information Systems, manages data, exploits the grid information to produce the work and connexion history and manages user and group subscription. Each service is described by a WSDL file and each method is described by its name, its arguments, its results and its description. The storage format is XML, with a DTD for each resource type, a Relational Data Base Management System or a XML native Data Base Management System.

The communication protocol between the portals and the Controller server is based on SOAP for its flexibility and its exchange security.

Combining grid components and web services eases the creation of efficient user services.

2.6. Conclusion

In this second chapter, grid technologies were defined and described in order to address the requirements of the *in silico* drug discovery process against neglected and emerging infectious diseases. Many grid definitions exist, and there are many ways to classify them, depending on their goals, their technologies, their ownership or their hardware for instance. The complexity of the underlying hardware resources needs to be managed and hidden by software layers providing services for users.

The middleware, or grid operating system, differs for the desktop and cluster grids. They are the most widespread grids, and their objectives are different. A desktop grid provides a powerful and easy to deploy computing power environment whereas a cluster grid is service-oriented and provides a durable environment with various resources. The best type of grid to address the requirements of *in silico* drug discovery against neglected and emerging infectious diseases is the cluster grid.

There are different cluster grid infrastructures and technologies in the world. The choice of a grid infrastructure depends on many criteria. The European grid EGEE, the French regional grid AuverGrid and the Taiwan national grid TWGrid are large production infrastructures, based on the EGEE technologies, providing a powerful environment for computing and data intensive applications. The French national grid for bioinformatics RUGBI is an infrastructure, based on web services, providing a useful environment for industrials and academic institutions in life sciences. The work described in this document was deployed on these different infrastructures.

The aim of the next chapter is to propose services for grid-enabled protein structure prediction on the RUGBI grid. We are going to describe a software and database deployment service and a database update service.

2.7. References

[219] Trunnell, M., Computing power: drawing power from the people, Scientific Computing World Nov./Dec. 14–16 (2001).
[220] Stix, G. and Writer, S., The Triumph of the Light, Scientific American, (2001).
[221] Foster, I. and Kesselman, C., The Grid: Blueprint for a Future Computing Infrastructure, Morgan Kaufmann, (1998).
[222] Foster, I., et al., The Anatomy of the Grid: Enabling Scalable Virtual Organizations, International Journal of High Performance Computing Applications, 15(3) 200-222 (2001).
[223] Foster, I., What Is the Grid? A Three Point Checklist, Grid Today 1:6 (2002).
[224] Gentzsch, W., Response to Ian Forster's "What is the Grid?", Grid today, (2002).
[225] Grimshaw, A., What is a Grid ?, Grid today, (2002).
[226] Crook, C., Computers and the Collaborative Experience of Learning, Routeledge (1994).
[227] Stevens, R., et al., From the I-WAY to the National Technology Grid. Comm. of the ACM, 40(11):50–60 (1997).
[228] Dillenbourg, P., Collaborative Learning: Cognitive and Computational Approaches. Elsevier Science (1999).
[229] IBM Solutions Grid for Business Partners: Helping IBM Business Partners to Grid-enable applications for the next phase of e-business on demand, (2002).
[230] Amin, K., et al., Open Collaborative Grid Services Architecture (OCGSA). In Proc. Euroweb'02 101–107 (2002).
[231] Asensio, J.I., et al., From collaborative learning patterns to component-based CSCL application. In Proc. ECSCW'03 workshop From Good Practices to Patterns (2003).
[232] Bote-Lorenzo, M.L., Grid Characteristics and Uses: a Grid Definition, Postproc. of the First European Across Grids Conference, Springer-Verlag LNCS 2970 291-298 (2004).
[233] Buyya, R. and Venugopal, S., A Gentle Introduction to Grid Computing and Technologies, CSI Communications, (2005).
[234] The Grid Café - What is Grid?, CERN (2005) and http://gridcafe.web.cern.ch/
[235] Grimshaw, A.S. and Natrajan, A., Legion: lessons learned building a grid operating system, Proceedings of the IEEE 93(3):589-603 (2005).
[236] Teasley, S. and Wolinsky, S., Scientific collaborations at a distance, Science 292 2254 (2001).
[237] DG Information Society, Building grids for Europe, (2004).
[238] Chetty, M. and Buyya, R., Weaving Computational Grids: How Analogous Are They with Electrical Grids ?, Computing in Science and Engineering July/August 4(4):61-71, (2002).
[239] Gentzsch, W., Grid computing in Industry, CSI communications 2005
[240] Breton, V., et al., DataGrid, Prototype of a Biomedical Grid, Methods of Information in Medicine 42(2)143-148 (2003).
[241] Camarinha-Matos, L. and Afsarmanesh, H., (eds), Infrastructure for virtual enterprises: networking industrial enterprises, Kluwer academic press (1999).
[242] Kopela, E., et al., Seti@home - massively distributed computing for seti, Computing in Science & Engineering January/February, 3(1):78 – 83 (2001).
[243] Gagliardi, F., et al. Building an infrastructure for scientific Grid computing: status and goals of the EGEE project, Philosophical Transactions: Mathematical, Physical and Engineering Sciences, 363 1729-1742 (2005) and http://www.eu-egee.org/
[244] http://www.opensciencegrid.org/index.php
[245] http://www.deisa.org/
[246] Pennington, R., Terascale Clusters and the TeraGrid, Invited talk, Proceedings for HPC Asia, 407-413 (2002).

[247] Raman, A., et al., PARDISC: A Cost-Effective Model for Parallel and Distributed Computing, Proc. Third Int'l Conf. High-Performance Computing, IEEE CS Press, (1996).
[248] Buyya, R., ed., High Performance Cluster Computing: Architectures and Systems, (1999).
[249] Krauter, K., et al., A taxonomy and survey of grid resource management systems for distributed computing. Int. J. of Software Practice and Experience, 32(2):135–164, (2002).
[250] Gray, J., Distributed computing economics, Technical report, (2003).
[251] Dongarra, J. and Casanova, H., Netsolve: A network server for solving computational science problems, Technical report, (1995).
[252] Abramson, D., et al., High-Performance Parametric Modeling with Nimrod-G: Killer Application for the Global Grid? Proc. Int'l Parallel and Distributed Processing Symp., IEEE CS Press, (2000).
[253] Buyya, R., et al., The Virtual Laboratory: A Toolset to Enable Distributed Molecular Modelling for Drug Design on the World-Wide Grid, J. Concurrency and Computation: Practice and Experience, (2002).
[254] Venugopal, S., et al., A taxonomy of Data Grids for distributed data sharing, management, and processing, ACM Comput. Surv. 38 1 (2006).
[255] Hoschek, W., et al., Data management in an international data Grid project. In Proceedings of the 1st IEEE/ACM International Workshop on Grid Computing, Springer-Verlag, (2000).
[256] Dullmann, D., et al., Models for Replica Synchronisation and Consistency in a Data Grid, 10th IEEE Symposium on High Performance and Distributed Computing, (2001).
[257] Kesselman, C., et al, The data grid: Towards an architecture for the distributed management and analysis of large scientific datasets, Journal of Network and Computer Applications, 187 (2001).
[258] Grethe, J.S., et al., Biomedical Informatics Research Network: Building a National Collaboratory to Hasten the Derivation of New Understanding and Treatment of Disease, Stud. Health Technol. Inform., 112 100–109 (2005).
[259] Saroiu, S., et al., A Measurement Study of Peer-to-Peer File Sharing Systems, Proceedings of Multimedia Computing and Networking (2002).
[260] Kunszt, P., et al., Advanced Replica Management with Reptor, In 5th Int. Conf. on Parallel Processing and Applied Mathematics, (2003).
[261] Muan Hong, N., et al., BioSimGrid: grid-enabled biomolecular simulation data storage and analysis, Future Gener. Comput. Syst. 22:657–664 (2006).
[262] Moore, R., et al., Data-intensive computing and digital libraries, Commun. ACM 41 11 56.62 (1998).
[263] Chervenak, A., et al. Towards an architecture for the distributed management and analysis of large scientific datasets, J. Netw. Comput. Appl. 23 187–200 (2000).
[264] Cannataro, M. and Talia, D., The knowledge grid, Commun. ACM 46, 89–93 (2003).
[265] Cannataro, M. and Talia, D., Semantics and Knowledge Grids: Building the Next-Generation Grid, IEEE Intelligent Systems 19(1) 56-63 (2004).
[266] Blythe, J., et al., Transparent Grid Computing: A Knowledge-Based Approach, Proc. 15th Innovative Applications of Artificial Intelligence Conference AAAI Press, (2003).
[267] Peitsch, M.C., et al. Informatics and knowledge management at the Novartis Institutes for BioMedical Research, Scip.online 46 1–4 (2004).
[268] Hug, H., et al., Ontology-based knowledge management of troglitazone-induced hepatotoxicity, Drug Discov. Today 9 948–954 (2004).
[269] De Roure, D. and Hendler, J.A., e-Science: the grid and the semantic web, IEEE 1094, 65–71 (2004).
[270] Bard, J.B. and Rhee, S.Y., Ontologies in biology: design, applications and future challenges. Nat. Rev. Genet, 5 213–222 (2004).
[271] Sun, X., et al., A Scalable P2P Platform for the Knowledge Grid, IEEE Transactions on Knowledge and Data Engineering 17 12 1721-1736 (2005).
[272] Hey, T. and Trefethen, A., e-Science and its implications, Philos. Transact. A. Math. Phys. Eng. Sci. 361, 1809–1825 (2003).
[273] Falk Hoffman, H., Statement from CERN and the scientific Community, European Organization for Nuclear Research (2004).

[274] Stevens, R.D., et al., myGrid: personalised bioinformatics on the information grid. Bioinformatics 191(1) i302–i304 (2003).
[275] Rauwerda, H., et al, The Promise of a virtual lab, Drug Discov Today. 11(5-6):228-36 (2006).
[276] Hewett, M., et al., PharmGKB: the pharmacogenetics knowledge base, Nucleic Acids Res. 30, 163–165 (2002).
[277] Baker, M., et al., Grids and Grid Technologies for Wide-Area Distributed Computing, International Journal of Software: Practice and Experience 32(15):1437-1466 (2002).
[278] Foster, I., The Grid: a new infrastructure for the 21st century science, Physics Today 42-47 (2002).
[279] Anderson, D.P. and Kubiatowicz, J., The worldwide computer, Sci.Am. 3, 40–47 (2002).
[280] Livny, M., High-throughput resource management, (Foster, Kesselman eds) In The Grid: Blueprint for a New Computing Infrastructure 311–337 (1999).
[281] Mutka, M. and Livny, M., The available capacity of a privately owned workstation environment. Performance Evaluation 12, 269–284 (1991).
[282] Ryu, K.D. and Hollingsworth, J., Exploiting fine grained idle periods in networks of workstations, IEEE Transactions on Parallel and Distributed Systems 11 683–698 (2000).
[283] Loewe, L., Global computing for bioinformatics, Briefings in Bioinformatics 3:377-388 (2002).
[284] Anderson, D.P., et al., SETI@home: An Experiment in Public-Resource Computing, Communications of the ACM, 45(11) 56-61 (2002).
[285] Oram, A., ed., Peer-to-Peer: Harnessing the Power of Disruptive Technologies, O'Reilly Press, (2001).
[286] Fedak, G., et al., XtremWeb: a generic global computing system, Proceedings of the First International Workshop on Global and Peer-to-Peer Computing on Large Scale Distributed Systems at the 1st IEEE/ACM International Symposium on Cluster Computing and the Grid, IEEE Press, 582–588 (2001).
[287] Anderson, D.P. and Fedak, G., The Computational and Storage Potential of Volunteer Computing, IEEE/ACM International Symposium on Cluster Computing and the Grid (2006).
[288] Shavitt, Y. and Shir, E., DIMES: Let The Internet Measure Itself. Computer Communication Review 35(5) 71-74 (2005).
[289] Barabasi, A. L., et al., Parasitic computing, Nature 412 894–897 (2001).
[290] Kant, K., et al., A framework for classifying peer-to-peer technologies, Proceedings of the 2nd IEEE/ACM International Symposium on Cluster Computing and the Grid, IEEE Computer Society 368–375 (2002).
[291] Skillicorn, D. B., Motivating computational grids, Proceedings of the 2nd IEEE/ACM International Symposium on Cluster Computing and the Grid, IEEE Computer Society 401–406 (2002).
[292] Anderson, D.P., BOINC: A System for Public-Resource Computing and Storage, Proceedings of the 5th IEEE/ACM International Workshop on Grid Computing, (2004) and http://boinc.berkeley.edu/projects.php
[293] Acharya, A. and Setia, S., Availability and Utility of Idle Memory in Workstation Clusters, SIGMETRICS 99 (1999).
[294] Kondo, D., Characterizing and Evaluating Desktop Grids: An Empirical Study, Proceedings of the International Parallel and Distributed Processing Symposium (2004).
[295] Taufer, M., et al., III. Homogeneous Redundancy: a Technique to Ensure Integrity of Molecular Simulation Results Using Public Computing, Proceedings of the 14th Heterogeneous Computing Workshop HCW (2005).
[296] Richards, W.G., Virtual screening using grid computing: the screensaver project, Nature Reviews Drug Discovery 1 551-555 (2002).
[297] Loewe, L., evolution@home: experiences with work units that span more than 7 orders of magnitude in computational complexity, in Proceedings of the 2nd IEEE/ACM International Symposium on Cluster Computing and the Grid, IEEE Computer Society, 425–431 (2002).
[298] Stainforth, D.A., et al., Uncertainty in the predictions of the climate response to rising levels of greenhouse gases, Nature 433 (2005) and Climateprediction.net, http://climateprediction.net/
[299] http://athome.web.cern.ch/athome/
[300] http://einstein.phys.uwm.edu/

[301] Litzkow, M.J., et al., Condor - A Hunter of Idle Workstations, Proceedings of the 8th International Conference of Distributed Computing Systems 104-111 (1988).
[302] Germain, C., et al, XtremWeb: Building an Experimental Platform for Global Computing, First IEEE/ACM International Workshop on Grid Computing (2000).
[303] Pande, V., et al., Atomistic Protein Folding Simulations on the Submillisecond Time Scale Using Worldwide Distributed Computing, Biopolymers, 68:91-109, (2003).
[304] Wolski, R., et al., Models and Modeling Infrastructures for Global Computational Platforms, Workshop on Next Generation Software, IPDPS, (2005).
[305] Chien, A., et al., Entropia: architecture and performance of an enterprise desktop grid system. J. Parallel Distrib. Comput. 63 597-610 (2003).
[306] http://www.platform.com/
[307] Anderson, D.P., High-Performance Task Distribution for Volunteer Computing, First IEEE International Conference on e-Science and Grid Technologies, (2005).
[308] Sullivan,W., et al., A new major SETI project based on project SERENDIP data and 100,000 personal computers, In Astronomical and Biochemical Origins and the Search for the Life in the Universe (1997).
[309] Barnatt, C., Office space, cyberspace & virtual organization, Journal of General Management 20 78–91 (1990).
[310] Buyya, R., et al., Economic Models for Management of Resources in Peer-to-Peer and Grid Computing, Proc. SPIE Int'l Conf. on Commercial Applications for High-Performance Computing, (2001).
[311] Gray, J., Distributed Computing Economics, Microsoft Research Technical Report MSR-TR-2003- 24 (2003).
[312] http://www.worldcommunitygrid.org/
[313] http://www.grid.org/home.htm
[314] http://www.decrypthon.fr/
[315] http://www.d2ol.com/
[316] LHC design report, CERN-2004-003, (2004) and http://ab-div.web.cern.ch/ab-div/Publications/LHC-DesignReport.html
[317] Baker, M., et al., The Grid: International Efforts in Global Computing, Proc. Int'l Conf. Advances in Infrastructure for Electronic Business, Science, and Education on the Internet, (2000).
[318] Foster, I., Globus Toolkit Version 4: Software for Service-Oriented Systems, International Conference on Network and Parallel Computing, Springer-Verlag LNCS 3779 2-13 (2005).
[319] Clarke, I., et al., Freenet: A Distributed Anonymous Information Storage and Retrieval System, International Workshop on Designing Privacy Enhancing Technologies, Springer-Verlag (2000).
[320] Casanova, H., et al., Heuristics for scheduling parameter sweep applications in Grid environments, Proceedings of the 9th Heterogeneous Computing Workshop, IEEE, 349-363 (2000).
[321] Goble, C., The grid: from concept to reality in distributed computing, Scientific Computing World May/June, Bioinformatics World 2 8–11 (2002).
[322] Berman, F., et al., Grid Computing: Making the Global Infrastructure a Reality, John Wiley & Sons (2003).
[323] Solomonides, A., et al., MammoGrid and eDiamond: grids applications in mammogram analysis, Proceedings of the IADIS International Conference: e-Society 2003, 1032-1033 (2003).
[324] Tweed, T. and Miguet, S., Medical Image Database on the grid: strategies for data distribution. Proceedings of HealthGrid'03, 152–162. (2003).
[325] Martone, M.E., et al., E-neuroscience: challenges and triumphs in integrating distributed data from molecules to brains, Nature Neuroscience 7:467-472 (2004).
[326] Parashar, M., et al., Application of grid-enabled technologies for solving optimization problems in data-driven reservoir studies, Proceedings of the International Conference on Computational Science, Pt 3. Springer-Verlag, 3038 805–812 (2004).

[327] Sulakhe, D., et al., GNARE: An Environment for Grid-Based High-Throughput Genome Analysis, In CCGrid 2005 BioGrid Workshop, (2005).
[328] Tham, C.-K. and Buyya, R., SensorGrid: Integrating Sensor Networks and Grid Computing, CSI communications, (2005).
[329] Stevens, R., et al., From the I-WAY to the National Technology Grid, Commun.ACM 40 50-61 (1997).
[330] Johnston, W.E., et al., Grids as production computing environments: the engineering aspects of NASA's information power grid, Proc. 8th IEEE Symposium on High Performance Distributed Computing, IEEE Press (1999).
[331] Goble, C., Review: the low down on e-science and grids for biology, Comp. Funct. Genomics 2 365–370 (2001).
[332] Szalay, A. and Gray, J., The world-wide telescope. Science 293, 2037–2040 (2001).
[333] Harris, F., Lamanna, M. (Eds), EGEE user forum Book of Abstracts, (2006).
[334] http://www.ncbiogrid.org
[335] http://www.cbr.nrc.ca
[336] http://www.eurogrid.org
[337] Covitz, P.A., et al., caCORE: a common infrastructure for cancer informatics, Bioinformatics 19, 2404–2412 (2003).
[338] Sanchez, W., et al., caGRID White Paper National Cancer Institute Center for Bioinformatics (2004).
[339] Buetow, K.H., Cyberinfrastructure: empowering a "third way" in biomedical research, Science 308 821–824 (2005).
[340] http://www.simdat.org
[341] Sild, S., et al., Open computing grid for molecular science and engineering, J. Chem. Inf. Model. 46 953–959 (2006).
[342] Dubitzky, W., et al., Grid-enabled data warehousing for molecular engineering, Parallel Comput. 30 1019–1035 (2004).
[343] Mazzatorta, P., et al., OpenMolGRID: molecular science and engineering in a grid context, Proceedings of the International Conference on Parallel and Distributed Processing Techniques and Applications, 775–779 (2004).
[344] http://www.lpds.sztaki.hu/pgrade/
[345] Taylor, I., et al., Visual Grid Workflow in Triana, Journal of Grid Computing 3(3-4):153-169 (2005).
[346] www.gridsphere.org
[347] https://genius.ct.infn.it/
[348] Pajchel, K., et al., Usage statistics and usage patterns on the NorduGrid: Analyzing the logging information collected on one of the largest production Grids of the world, Proceedings of CHEP 2004 2 711 (2005).
[349] The LCG Editorial Board, LHC Computing Grid Technical Design Report, CERN-LHCC-2005-024 (2005).
[350] http://glite.web.cern.ch/glite
[351] Grimshaw, A.S. and Wulf, W.A., The Legion Vision of a Worldwide Virtual Computer, Commun. ACM 40(1):39-45 (1997).
[352] Buyya, R., and Venugopal, S., The Gridbus Toolkit for Service Oriented Grid and Utility Computing: An Overview and Status Report, Proceedings of the First IEEE International Workshop on Grid Economics and Business Models 19-36 (2004).
[353] Romberg, M., The UNICORE grid infrastructure, Sci. Program. 10 149–157 (2002).
[354] Cappello, F., et al., Grid'5000: A Large Scale, Reconfigurable, Controlable and Monitorable Grid Platform, Proceedings of the 6th IEEE/ACM International Workshop on Grid Computing, (2005).
[355] Vicat-Blanc Primet, P., et al., e-Toile : High Performance Grid Middleware. Proceedings of Cluster'2003 (2003).
[356] Asadzadeh, P., et al., Global Grids and Software Toolkits: A Study of Four Grid Middleware Technologies, CoRR Computer Science 0407001 (2004).

[357] Huber, V., Supporting Car-Parrinello molecular dynamics with UNICORE, International Conference on Computational Science, Springer-Verlag 560–567 (2001).
[358] Pytlinski, J., et al., BioGRID - uniform platform for biomolecular applications, Proceedings of the 8th International Euro-Par Conference on Parallel Processing, Springer-Verlag 881–884 (2002).
[359] Maran, U., et al., Mining of the chemical information in GRID environment, FGCS in press (2006).
[360] Stockinger, H., Grid Computing: A Critical Discussion on Business Applicability, IEEE Distributed Systems Online, 7(6) (2006).
[361] Kubiatowicz, J., et al., OceanStore: An Architecture for Global-Scale Persistent Storage, 9th Intl. Conf. on Architectural Support for Programming Languages and Operating Systems, (2000).
[362] Buyya, R., Economic-based Distributed Resource Management and Scheduling for Grid Computing, PhD Thesis, (2002).
[363] Gronager, M., et al., LCG and ARC middleware interoperability, Proceedings of CHEP (2006).
[364] Foster, I., et al., Grid services for distributed system integration, Computer 35 37–46 (2002).
[365] Foster, I., et al., The Physiology of the Grid: An Open Grid Services Architecture for Distributed Systems Integration, Open Grid Service Infrastructure Working Group Technical Report, Global Grid Forum (2002).
[366] http://www.globus.org/wsrf/
[367] Graham, S., et al., Building Web Services with Java: Making Sense of XML, SOAP,WSDL, and UDDI, Sams (2001).
[368] http://www.globalgridforum.org/
[369] Antonioletti, M., et al., The Design and Implementation of Grid Database Services in OGSA-DAI, Concurrency and Computation: Practice and Experience 17(2-4) 357-376 (2005).
[370] Karasavvas, K., et al., Introduction to OGSA-DAI Services, Lecture Notes in Computer Science 3458 1-12 (2005)
[371] Foster, I. and Iamnitchi, A., On Death, Taxes and the Convergence of Pear-to-peer and Grid computing, 2nd International Workshop on Peer-to-Peer Systems (2003).
[372] Soberman, M., Les grilles informatiques : état de l'art et déploiement, JRES 2005, (2005).
[373] Wilcox-O'Hearn, B., Experiences Deploying A Large-Scale Emergent Network, 1st International Workshop on Peer-to-Peer Systems, Springer-Verlag (2002).
[374] Sarmenta, L.F.G., Sabotage-tolerance mechanisms for volunteer computing systems, Future Generation Computer Systems 18(4) 561-572 (2002).
[375] Anstreicher, K., et al., Solving Large Quadratic Assignment Problems on Computational Grids. Mathematical Programming 91(3):563-588 (2002).
[376] Thompson, M., et al., Certificate-based Access Control for Widely Distributed Resources, 8th Usenix Security Symposium (1999).
[377] Pearlman, L., et al., A Community Authorization Service for Group Collaboration. IEEE 3rd International Workshop on Policies for Distributed Systems and Networks, (2002).
[378] http://eu-datagrid.web.cern.ch/eu-datagrid/
[379] Gagliardi, F., et al., How to Build an International Grid: Infrastructure, Applications and Community, CTWatch Quarterly 1(4) (2005).
[380] http://www.naregi.org/index_e.html
[381] http://www.see-grid.org/
[382] http://www.eu-eela.org/
[383] http://www.diligentproject.org/
[384] http://www.bioinfogrid.eu/
[385] Jones R., et al., Lessons from Europe's International Grid Initiatives: Grid Technology in Africa, Proceedings of IST-Africa 2006 conference, (2006).
[386] www.auvergrid.org/
[387] Breton, V., private communication, (2005).
[388] http://twgrid.org/
[389] http://rugbi.in2p3.fr/
[390] http://rugbi.ibcp.fr/

[391] http://agena.c-s.fr:8081/rugbiportal/
[392] http://www.cs.wisc.edu/vdt/
[393] http://datatag.web.cern.ch/datatag/
[394] Campana, S., et al., Analysis of the ATLAS Rome Production Experience on the LHC Computing Grid, IEEE International Conference on e-Science and Grid Computing, (2005).
[395] https://edms.cern.ch/file/498079/0.1/LCG-mw.pdf/
[396] Raman, R., et al., Matchmaking: Distributed Resource Management for High Throughput Computing, Proceedings of the Twelfth IEEE International Symposium on High-Performance Distributed Computing, (2003).
[397] Allcock, W., et al., GridFTP: Protocol extensions to ftp for the grid. Tech. rep., Argonne National Laboratory, (2001).
[398] Andreozzi, S., et al., Sharing a conceptual model of Grid resources and services, Conference on Computing in High Energy and Nuclear Physics, (2003).
[399] Salzemann, J., et al., EMBRACE reports on Technology Survey D3.1.1, (2006).
[400] http://www.w3.org/
[401] Kreger, H., Web Services Conceptual Architecture, IBM, (2001).
[402] http://www.w3.org/TR/wsdl
[403] Reynaud, S., and Hernandez, F., A XML-based Description Language and Execution Environment for Orchestrating Grid Jobs, IEEE International Conference on Services Computing 2 192-199 (2005).

Chapter 3. Services for protein structure prediction in a grid environment

3.1. Introduction

In chapter 1, we introduced the concept of *in silico* drug discovery. One step of the process is protein structure prediction, which aims to determine the 3D structure of a protein from its amino acid sequence. Understanding the function and the physiological role of a protein target is fundamental for the discovery of new drugs. This complex process contributes for instance to the identification of the function of an individual protein or the provision of target structures for drug design.

Scientists need access to many public or private software and databases to analyze sequences and structures as well as physical and chemical properties. But technical aspects, such as the delays in waiting for the server response due to multiple accesses, limit the use of the protein structure prediction servers. Chapter 2 explained that a cluster grid can address these requirements by providing a secure environment to share resources (software, database, computing power, storage…).

The RUGBI grid was developed to offer a direct, secure and high performance access to different software and databases to analyze proteins. The services which are going to be described in this chapter were developed to enable protein analysis in a grid environment. They are not specific to protein structure prediction. Grid-enabled protein analysis requires a deployment service to allow registration, modification and consultation of bioinformatics software and databases. Once registered, the software or the database is automatically installed on determined grid nodes in grid private space and is ready to use thanks to automatically generated interfaces.

But scientists have to be able to access easily the last update of the databases in order to apply the software. The frequent and regular update of the databases is a recurrent issue for all host or mirror centers. A service to update and distribute biological databases makes available the last version of a flat file database on the grid for the jobs launched by a RUGBI user. It must be noted that, in the context of bioinformatics, the term database refers to a large set of catalogued sequences (for instance: protein sequences), and does not include standard data management systems [404].

The aim of this chapter is to present two grid-enabled services to deploy and update protein structure prediction software and databases. Services were developed for a broad range of bioinformatics applications but are particularly relevant for protein structure prediction in the perspective of *in silico* drug discovery.

The chapter is introduced as follows:

- After the introduction, the second section gives a brief summary of related works on this topic.
- The third section is about the service used to deploy software and databases in the RUGBI framework.
- The fourth section concerns the database update and distribution service in the RUGBI grid. The objectives, architecture, results, limitations and perspectives are presented in the third and fourth sections.

3.2. Related works

The grid-enabled bioinformatics services of the RUGBI grid aim to offer a personalized and secured working environment on grid. The software and database deployment service, based on web services, helps the user to install and use their own software and databases in a computing grid environment. The database update and distribution service downloads and spreads automatically and periodically the database on determined grid nodes. Some projects propose bioinformatics services in a grid environment to ease the use of distributed and shared resources between many actors.

The word resource is used for logical resources (data, software, licenses) and physical resources (Computing and Storage Elements).

Grid-enabled software and database deployment

Several projects like North Carolina BioGrid [334], EuroGrid BioGrid [405], Asia Pacific BioGrid [406] and GénoGrid [407], share computing and storage resources for a community with the aim of deploying data analysis tools. Dedicated web interfaces give access to public databases which are integrated in the environment. But there is no common bioinformatics service to register and install a new software or database.

A Problem Solving Environment is an integrated computing environment for composing, compiling, and running applications in a specific area [408]. Projects like MyGrid [274], Proteus [409] and ProGenGrid [410] develop Problem Solving Environments [411] bioinformatics grids.

MyGrid is a project dedicated to the integration of distributed data, workflow management and environment personalization. The knowledge, the services and the bioinformatics resources are categorized with ontologies. An ontology defines the terms used to describe and represent an area of knowledge. Taverna [412], a component of this user toolkit, is a powerful tool to realize *in silico* experiments. But MyGrid does not provide computing or storage resources. A scientist can register his software (their service) or his database, but they are installed on his own computing or storage resource. Issues about access saturation or execution delay due to computing limitation for instance are not resolved.

Proteus and ProGenGrid build service oriented architecture to gather tools and databases, encapsulated in web services and to execute them transparently on grid using ontologies and

workflows. This model aims to bridge the gap between Information Technology and the life sciences using both web and grid components. They are between the two previous project types, using ontologies, workflows, grid execution and result presentation. However, Proteus is proteomics-oriented only. Furthermore, the user in both projects cannot install and use their own software and databases.

In comparison with these related works, the services we developed allow an identified user to deploy any public software and database in their private storage space. If the user has the corresponding license, a commercial tool can also be deployed with the help of node administrators if necessary. The services described here provide software components to ease biologists' activities with metadata, ensure the interface with the exploitation service of the grid infrastructure, guarantee interoperability between databases and software and manage program execution and public or private data manipulation. All services are integrated in a secure and robust environment.

Database update and distribution in a grid environment

The biological database providers [413,414] make them publicly available in ftp servers. Each bioinformatics center or laboratory [415] needs to download the updated database to give access to its users. There are update services, but they can be integrated in a specific commercial tool, such as the SRS-Prisma module of SRS [416], a web-based retrieval system for biological data. There is an update server development [327] but the aim of this package is to download the update of several public databases in an unique local database, instead of distributing the database on many nodes. Scientists working on a grid infrastructure such as EGEE need to have the ability to deploy the database they are interested in on the grid nodes they select.

A large number of research papers and development projects focused on file replication. But there is not much literature about database update services in a grid environment. A review is carried out in [417].

In this paper, the authors proposed a Replica Consistency Service conceived for data grids, allowing asynchronously replica updates in a single-master scenario. This method is close to the method described in this document, but their service is designed for relational databases whereas the RUGBI service is flat file-oriented.

xNDT redistributes databases from the Swedish EMBnet node to a number of national EMBnet subnodes [418]. It is being adapted for a grid environment but it is limited for the moment to MySQL relational database and is still in development.

The service proposed here is based on the same process as [417], but it is specialized for flat files.

3.3. Service for the deployment of software and databases

3.3.1. Service objective

The software and database deployment service allows registration, modification and consultation of bioinformatics software and databases. Once registered, the software or the database is installed on determined grid nodes and ready to use thanks to automatically generated interfaces. Each registered RUGBI user is authorized to deploy software or databases on a private grid space. The user is considered as the administrator of the resource. He also decides the access rights associated to the resource (public, group, individual). The deployment process can be achieved through three steps presented in figure 18: Resource information registration, Resource download and resource installation using the Information System.

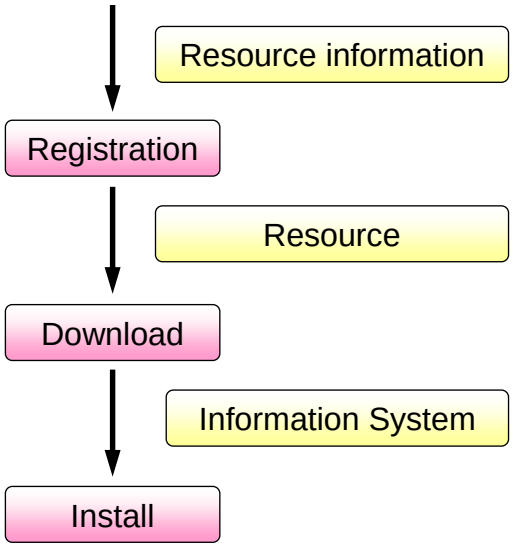


Figure 18: Resource deployment process

After this first service description, the next section presents the service components.

3.3.2. Service components

Some of the components needed for the service were detailed in chapter 2: the Storage Element where the resource is stored, the Computing Element where the software is installed and the Controller managing the Information System and the deployment service. They will not be described again here.

The next paragraph defines bioinformatics software and databases to understand how to deploy them on the grid. Then the Information System is presented. The last paragraphs describe briefly the database update service and the execution service: service functionalities are used by the deployment service. The database update service will be detailed in chapter 3.4.

Software and databases to be deployed

A bioinformatics software command line requires basically the pathways of input file(s), parameter file(s), database file(s) and/or output file(s). The execution modes of bioinformatics software are various and difficult to model. For instance, one of these command line components can be a repository instead of being the name of the file. A component can also be implicit without appearing in the command line (see Blast execution mode [419]). The installation process can also be varied. One example of this diversity is the download localization (ftp, sftp, http, shttp...) of the software. Finally, some complementary information is available like software publication and licensing.

A biological database records data generally split into textual entries composed of a datum with associated metadata. In a protein sequence database, like Swissprot [420], the datum is the sequence and metadata are items such as the identifier of the protein or the bibliographic references. For the RUGBI project, only the sequence flat files will be used by the software, and thus installed on the grid. But the process is the same for a full database. These worldwide databases are available on the Internet through for example HTTP or FTP transfers for a desktop or site usage. Their structure is varied (multiple files and repositories). Complementary information is available like database publication and licensing.

The Information System

The Information System manages the users, the computing and storage resources, the software, the databases, etc. There are four types of information by resource: the administration information for the installation, the configuration and the deployment; the information used by the portals to define the user interface; the configuration and localization information, for the allocation and localization services; and the execution information for the execution services. The Information System is hosted by the Controller. Each Controller has an updated Information System.

The Database Update Service

The Database Update Service can be used by the deployment service to download a resource from an external location on a Storage Element and to distribute it on other determined grid sites. The Database Update Service is hosted on a Storage Element. It uses the resource information from the Information System.

The Execution Service

The Execution Service can be used by the deployment service to install a resource on a working repository of a Storage Element or a Computing Element. Its main role is to manage the job execution on a node, including for instance data or software transfers from the working repository to the temporary repository of a Worker Node.

The next section presents the Deployment Service architecture.

3.3.3. Service architecture

Protocols were designed to help administrators to deploy resources. Protocols describe the different steps required to install the resources and ensure the communication between the

actors of these protocols thanks to the Information System. This section provides the Unified Modelling Language (UML) class diagram of the Deployment Service, the DTD model of software and database resources rendered as a graph diagram and the UML sequence diagram for a resource deployment.

The UML class diagram of the Deployment Service

Figure 19 is a UML class diagram for the Deployment service. The service is a part of a global Resource Service. Software and database resources and deployment methods are developed.

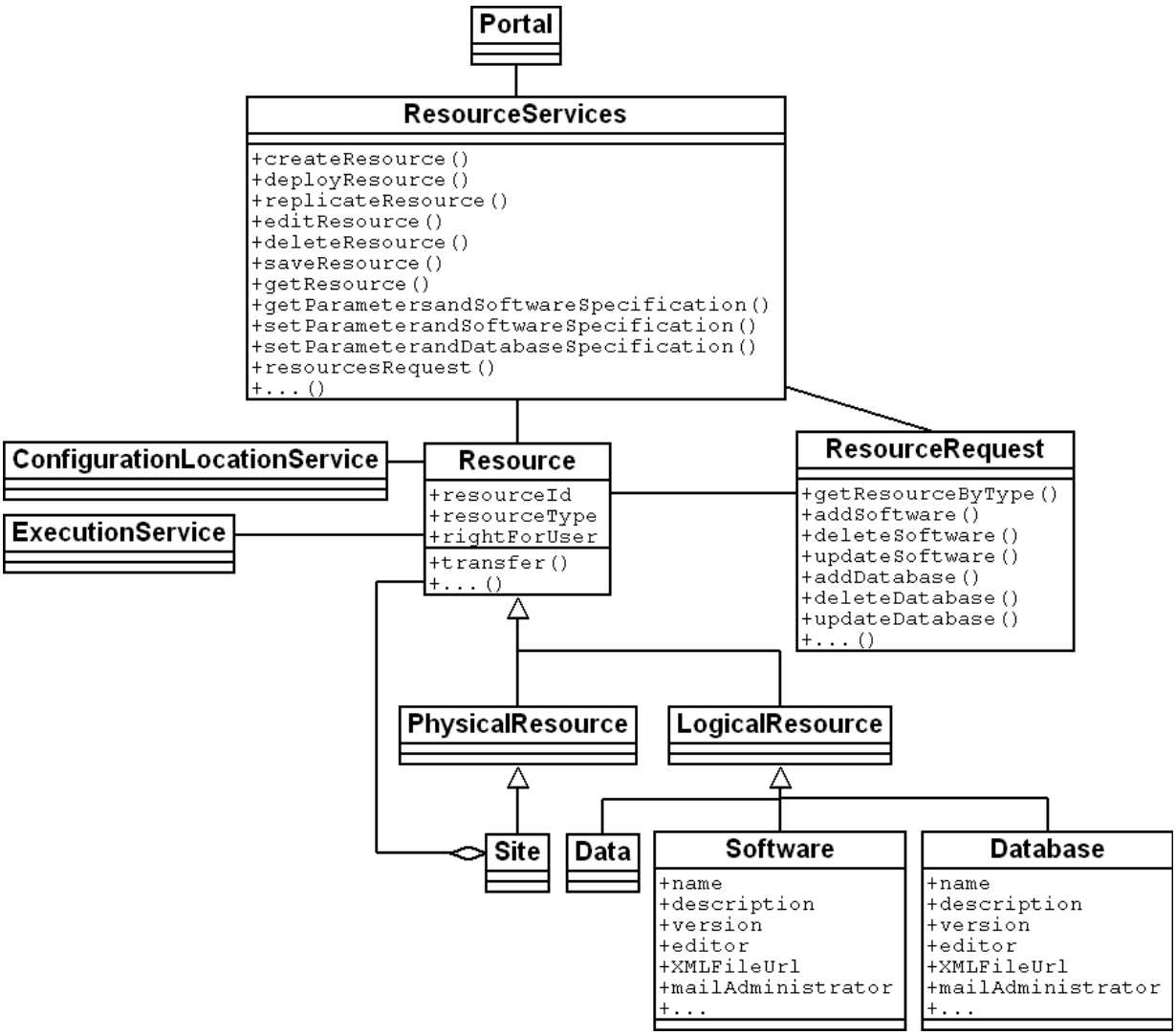


Figure 19: UML class diagram of the Deployment service

A resource can be physical (a grid site) or logical. Logical resources are software, database or data (the data resource is used by the Data Manager of the RUGBI project). Software and database attributes are for instance their name, their description, their version, their XML file URL and their administrator mail. Software and databases are described in the next section.

The resource class is used by several services. Here are shown the ExecutionService for the job execution and the ConfigurationLocationService to configure and locate a physical resource.

The ResourceServices methods allow the resource deployment through the portal. The createResource method defines a resource (its parameters, its install mode…) thanks to the resource DTD model (described below) and transfers the resource from an external location to a dedicated repository in an internal Storage Element. The deployResource method transfers a resource from the dedicated repository to a working repository and installs it (compilation, parameters…). The replicateResource method replicates the resource on different grid nodes. Other methods, such as editResource, deleteResource, etc., allow resource manipulation.

Methods of the ResourceRequest table allow a search of resources by type, and the addition, deletion or updating of a resource (from version n to version n+1).

The DTD models of the resources

Information about software and databases is dynamic and textual. It is easily described in XML files. These files have their own DTD for each resource type. The DTDs of the software and database XML files are rendered as graph diagrams in figures 20 and 21.

The DTD graph diagrams are composed of:

- The mandatory element Software or Database which contains unique attributes like Identifying and resource Name.
- The mandatory element Characteristics contains the common information about the resource (attributes Version, Date, Confidentiality…). It is subdivided into a mandatory element Copyright (attributes commercial or free Category, Weburl…) and other informative elements (Author, Bibliography…).
- The element Deployment manages the install on a grid site and the composition of the resource. For the database resource, this element possesses a few attributes such as deployment Type or Number of Files.
    - The mandatory element Install possesses several attributes for the resource install such as Required Space or Path; Install and Uninstall Scripts or Required Architecture are specific to the software resource. The Download element is necessary with attributes like Protocole or Url.
    - The element Structure describes the structure of the set of Files (attributes Name, Size, Format, mandatory…) and Directories (attribute Path) of the resource.
    - The element Environment for the software resource is still not used.
- For the database resource, the element Use indicates software authorized to use the database.
- For the software resource, the element Execution describes all software parameters necessary for the execution and for the saving of the generated output. The element Execution contains for instance the attributes Input Usage and Output Usage which

explain how the input or the output is recognized by the software (file, command line parameter, location, location of implicit input or output…).

- The elements Processparamfile and Globalpreexe specify if there is a pre-process on the parameter file or globally before the execution (like a format conversion).
- The element Processresultfile and Globalpostexe specify if there is a post-process on the result file or globally after the execution (like a format conversion).
- The element Parameter is used for the command line creation (attributes Type of the argument like file or value; argument Rank in the command line; Optional or mandatory in the command line, simple or advanced Category for the submission interface of the software…). Examples of the subdivided elements from the element Parameter are the element Storagepath, which indicates where an output is stored or should be stored, or the element Ihm, used to generate the submission interface of the software and composed of a description and a type (list, file, text, value…).
- Finally the elements Inputstructure and Outputstructure describe the structure of the input or output environment.

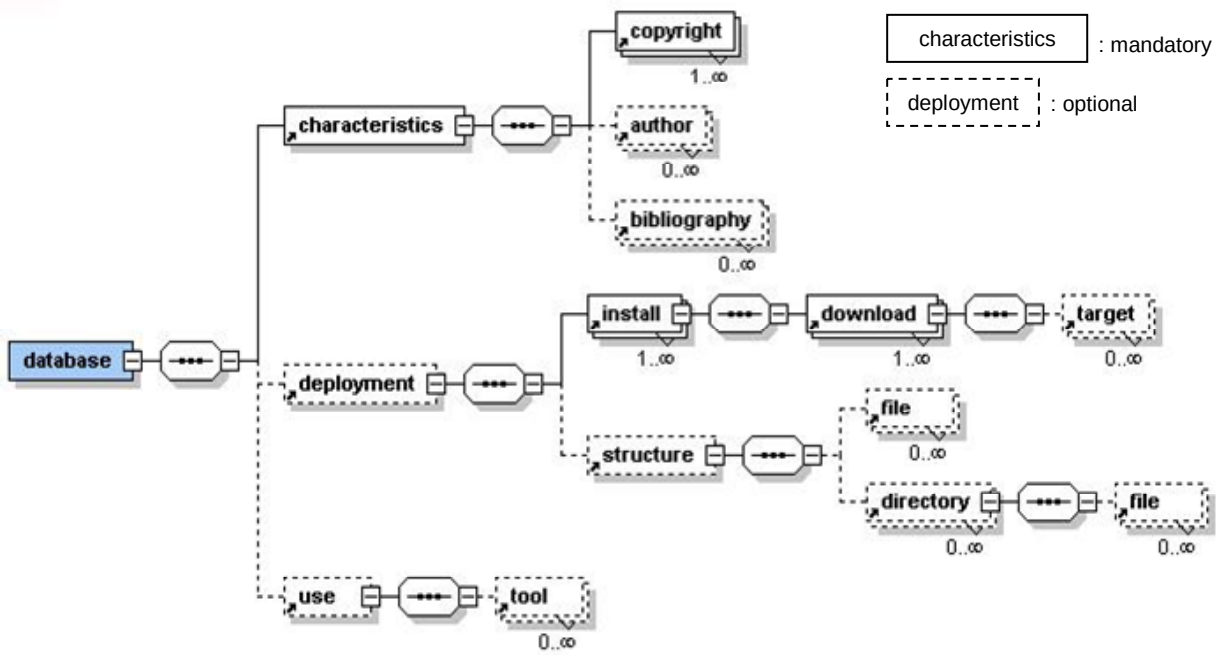


Figure 20: DTD model rendered as a graph diagram of the database resource.

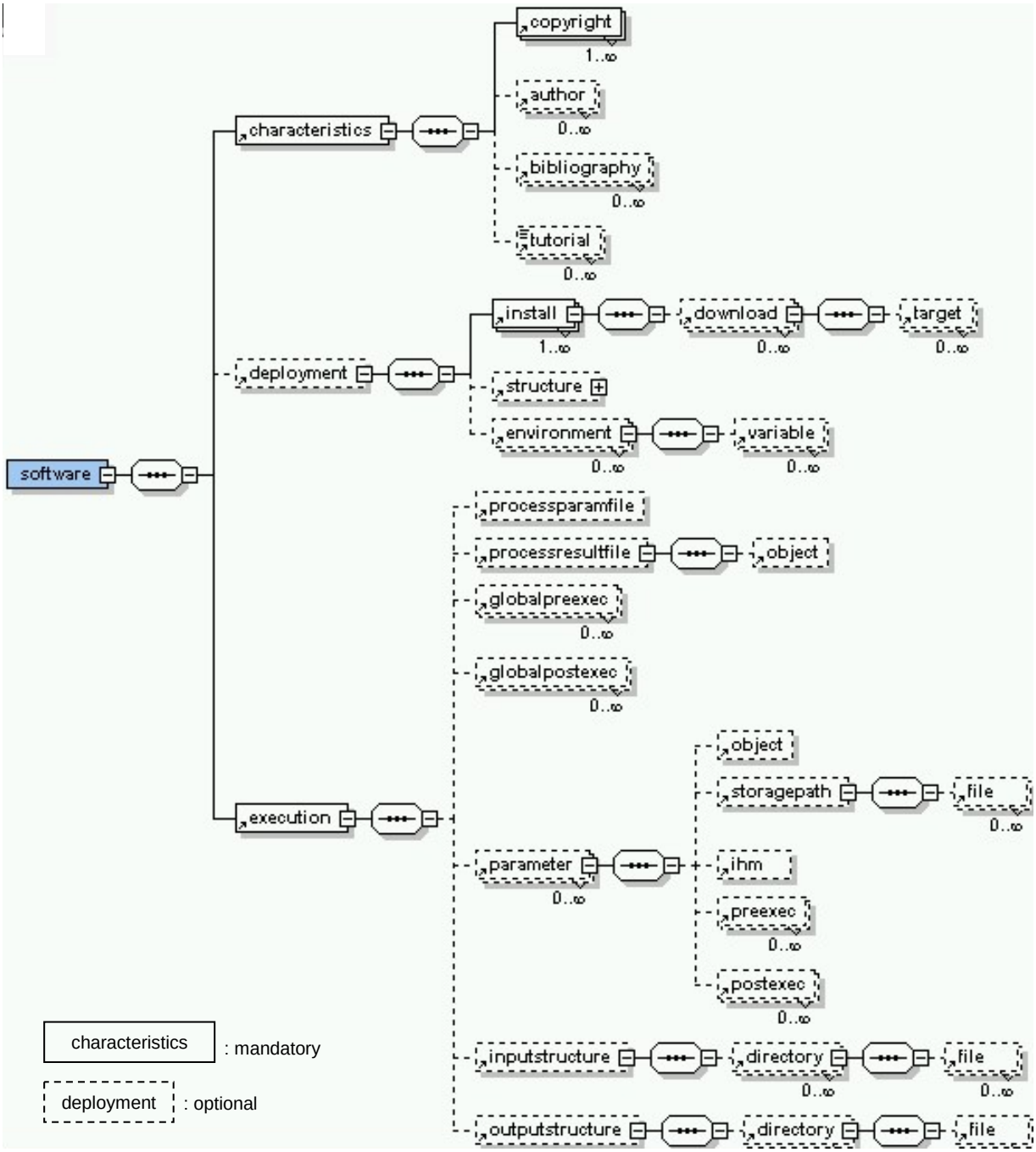


Figure 21: DTD model rendered as a graph diagram of the software resource.

Information about software and databases is used for:

- The resource deployment and localization by the end-users and the site administrators
- The resource information description for the end-users
- The resource choice by the end-users
- The software interface generation by the portals
- The software execution (task flow creation: generation of the command line, transfer of the resource on the execution site, output storage)
- The update service

Each resource file can be easily managed by its administrator with the methods described in the UML class diagram.

The UML sequence diagram for resource deployment.

Figure 22 presents the UML sequence diagram for software deployment. The ResourceService manages the deployment. The UpdateService and the ExecutionService execute the resource transfer and install.

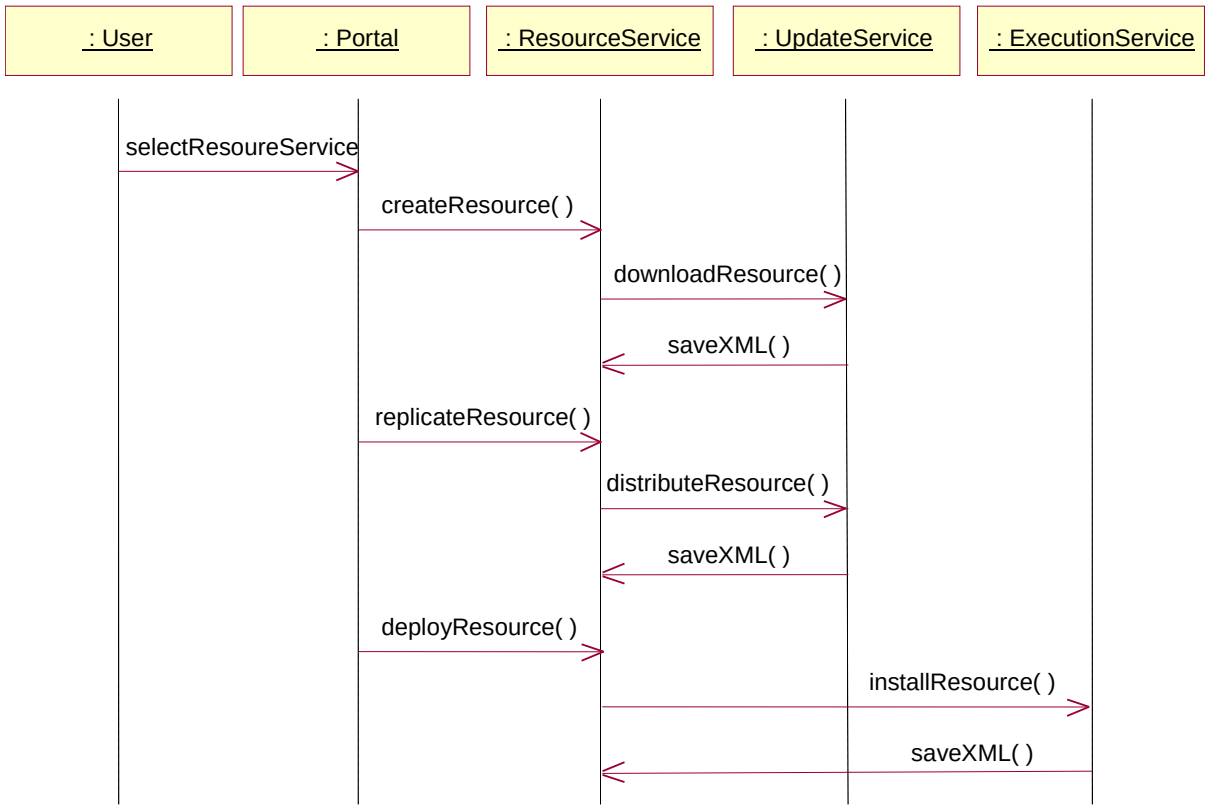


Figure 22: UML sequence diagram for resource deployment

Service implementation

The Resource Service and the Information System are hosted on the Controller server. The server is in Java for its portability, as well as for its integration with other technologies (All the Globus Toolkit APIs are available in Java). The storage format for information is XML, with a DTD for each resource type, a Relational Data Base Management System or a XML native Data Base Management System. Queries between the Resource Service and the Information System are carried out thanks to XPath with the Java API Jaxen [421]. Communications are made with SOAP (for instance communications between the portal and the Controller). Technologies used for data transfer are reported in the database update service.

The next section describes a case study of a software deployment.

3.3.4. Result: Case study of a software deployment

Using the deployment service, diverse databases and software were integrated (see table 2 in chapter 2.4.4). A case study for demonstration is the deployment of Gor [422], a tool for protein secondary structure prediction.

Protein secondary structure consists of local inter-residue interactions mediated by hydrogen bonds. The most common secondary structures are alpha helices and beta sheets. The random coil indicates an absence of regular secondary structure. The GOR method predicts protein secondary structures of an amino-acids sequence using all possible pair frequencies within a window of 17 amino acid residues. After cross-validation on a data base of 267 proteins, the version IV of GOR has a mean accuracy of 64.4% for a three state prediction (Q3: percentage of residues predicted correctly as helix, strand, coil or for all three conformational states). The program gives two outputs, one eye-friendly giving the sequence and the predicted secondary structure in rows. The second gives the probability values for each secondary structure at each amino acid position. The predicted secondary structure is the one of highest probability compatible with a predicted helix segment of at least four residues and a predicted extended segment of at least two residues. Gor is simple open-source command line software. It has few parameters, calls only two small databases and produces two small outputs. In this case study, the aim is to predict secondary structures of the aspartic hemoglobinase II, or Plasmepsin 2, of *Plasmodium falciparum* (Primary accession number of Swissprot: P46925) thanks to Gor IV.

Once connected to the RUGBI portal with its personal certificate, the scientist accesses a view of his environment. He can see for instance the list of software he is authorized to use. Figure 23 shows the Administration interface for software management.

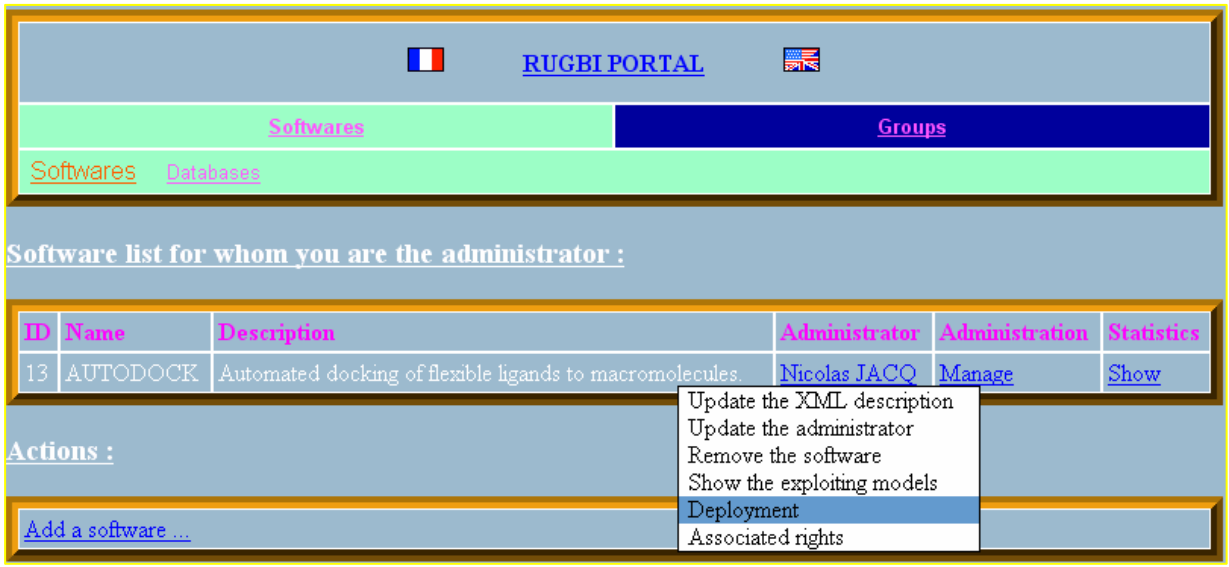


Figure 23: Software administration interface

The first stage of the deployment of Gor on the grid is the choice of the grid site where the software will be deployed. Once selected, the registration interface appears to describe the

software. Figure 24 presents the registration interface with the attribute Characteristics to fill in, conceived from the DTD model of the resource, and the XML file generated automatically after validation.

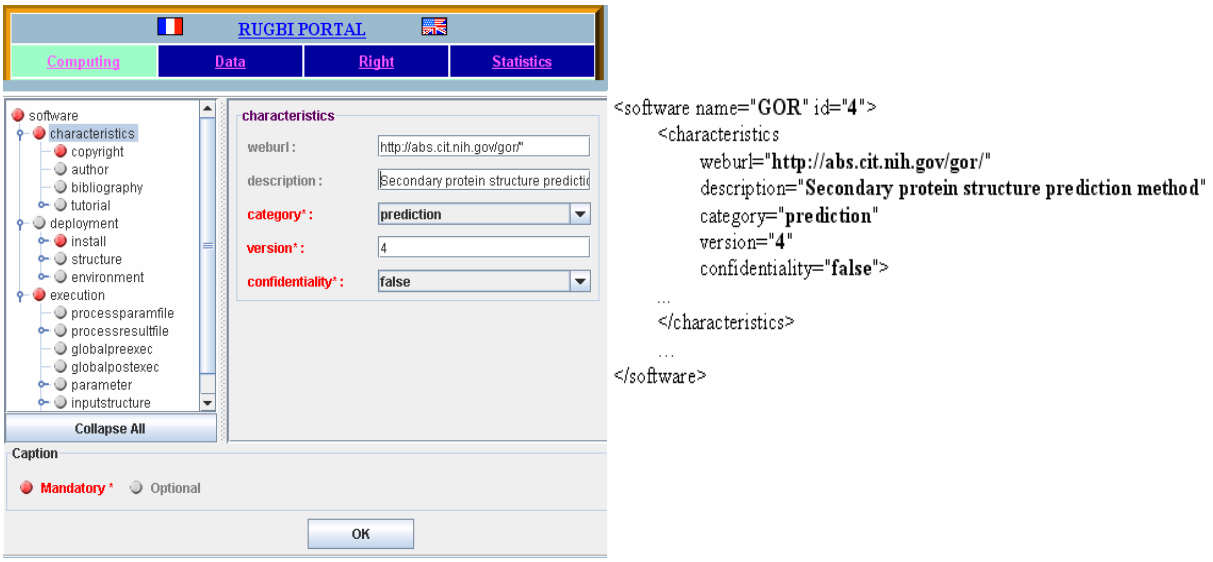


Figure 24: Registration interface of the Gor software generated by a DTD model and XML file generated after registration.

The user must fill in the fields with the resource information. After validation, the software administrator can see this new resource in the list of his available resources (Software utilization interface). The user will then deploy the software on his private space (see figure 23) thanks to the information from the elements Deployment, Install and Download of the software resource (see figure 21). A task flow is generated to download the resource from its external storage source or from the user local machine, to transfer the resource in the storage space, to install it (if compilation is required, only the site administrator is authorized to install it), then to update the site's configuration. The XML file is registered in the Information System of the Controller and replicated on the other Controllers of the grid. The software will be accessible through a portal only for users authorized by the software administrator (Group menu). Different versions of the same software can be installed. At the end of this protocol, the resource is ready to be used on all portals. The deployment is completed.

An example of use of the XML file is the Gor submission with the amino-acids sequence Plasmepsin 2 by a classical submission interface. Figure 25 presents the XML file of the Gor software and the automatically generated access interface. The simple mode requires only the amino-acids sequence in fasta format. The registration format is thus a file (the text which can be written in the access interface, such as in figure 25, is automatically converted into a file). The corresponding Parameter attributes are Category, with the value "simple", and Type, with the value "file". The Object attribute is Format with the value "fasta". The expert, but optional, mode is partially visible in the figure. Default values are registered in the XML file.

For instance, the Parameter attribute value of the Ihm attribute “Description Number of sequence” is 267.

```
<software name="GOR" id="4">
    <execution ... >
        <parameter value""
                optional="false"
                type="file"
                rank="4"
                io="in"
                category="simple">
            <object name="sequence"
                format="fasta"/>
            <storagepath path="input"/>
            <ihm mmiobject="sequence"
                description="sequence to use"/>
        </parameter>
        <parameter optional="false"
                value="267"
                type="value"
                rank="1"
                io="in"
                category="advanced">
            <ihm mmiobject="value"
                description="Number of sequence"/>
        </parameter>
        ...
    </execution>
</software>
```

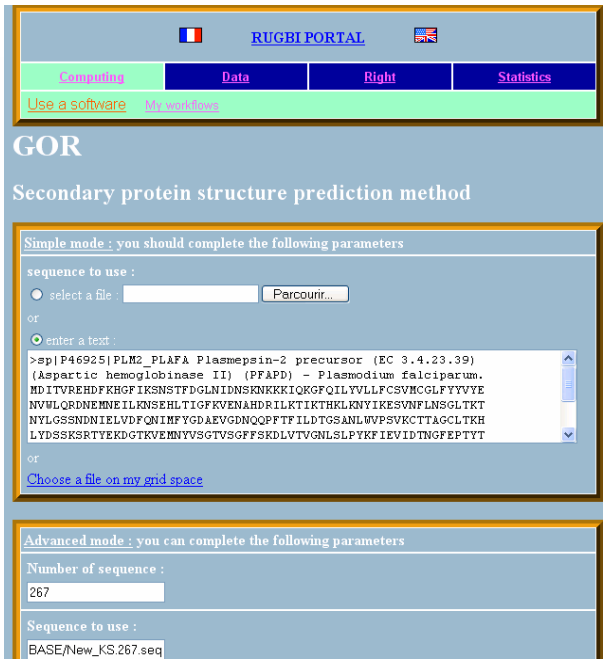


Figure 25: XML file of the Gor software and the access interface automatically generated.

Once the interface has been filled and validated, the task execution and the job monitoring are processed independently. The full command line is recomposed from the XML file and the user information. The interoperability between software and databases is checked thanks to the attributes on format, directory and use restriction. An email is automatically sent to the user once the process is finished. Figure 26 presents on the left the execution report of the task run on CSSI-Grenoble-Cluster-4.0 node. On the right is presented the predicted state for each amino-acid of the Plasmepin 2 sequence. The H is a helix, the E is an extended or beta strand and the C is a coil.

The user can register the result on their local disk or on the grid in order to save it and eventually use it again with other software. The installed software can also be used again with other data. The environment is thus personalized and saved for a specific user in the RUGBI environment. This deployment service is paramount to make grid use transparent for the scientist.

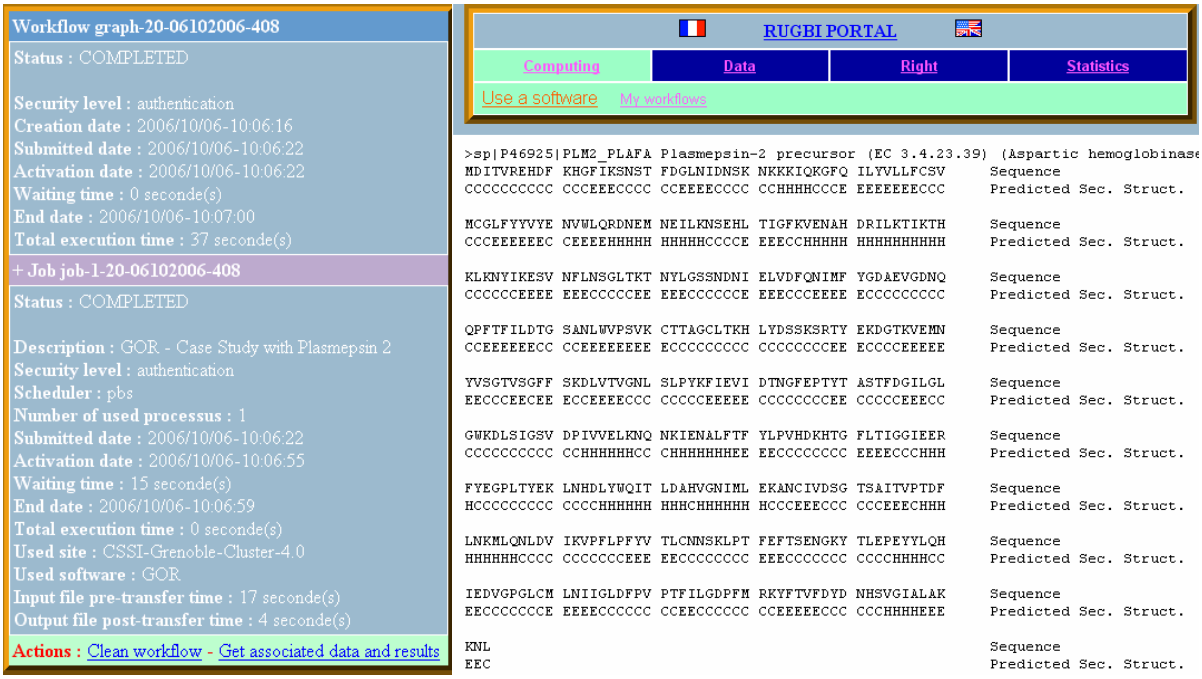


Figure 26: Prediction visualization in text format of Plasmepsin 2 with Gor IV

3.3.5. Service limitations and perspectives

However, the service has some limitations.

Limitations

As this service aims to integrate a large number of software and databases, the resource data models presented in figure 24 and 25 are complex. The user that wants to register their resource needs to know it very well. One example is the number of elements and attributes to describe the software command line. The registration step can be long, even if many elements are optional. The user administrator can decide to register only the software simple mode without advanced options for instance.

A second limitation is the management of licenses. Registering and using in-house private software is easy and secure in the RUGBI grid. But there are license modes that cannot be automatically managed with the registration interface.

A third limitation is the risk of accumulating many different versions of the same software confusing the view for the end-users of a group. This problem can appear if the group is large or badly managed. The problem will not appear for the installation of software or databases in public space because this installation needs be validated by the RUGBI grid administrators.

Perspectives

Despite these limitations, the deployment service is a key for the future of the RUGBI grid. New software and databases are regularly emerging. Thanks to the grid flexibility and automation, a scientist can build his personalized environment with his favorite resource. Solving the limitation issues presented above could increase the number of users.

The registered information on resources is useful for all other RUGBI services. It could also be used by a workflow manager, a project manager or an annotation manager. The RUGBI grid can be easily extended with these new services thanks to web services. A bioinformatics experiment manager is under preparation by a RUGBI partner. Moreover, the Embrace network of excellence will promote the web services, and in the future, services developed in the Embrace framework could be integrated in the RUGBI framework.

The database deployment service would be extremely limited without a database update and distribution service. This is the topic of the next section.

3.4. Database update service

3.4.1. Service objective

The RUGBI grid aims to offer its services to a community of scientists, for instance in protein structure prediction. Some of the applications call third party databases. As the main criterion of the grid is sharing out, there will be several spaces storing the databases for the jobs, from which they will directly pick up the information. Most of the required databases are stored and updated on ftp repositories. For instance, the Swissprot database is updated biweekly as a new complete database file including all the new entries added since the last release. It is a goal in itself to give the users the most up to date version of each database for their jobs to run on. It must be noted that only the last version of the database will be available on the RUGBI grid. Previous versions of the database (n-1, n-2 etc.) will not be kept for storage space reasons: for instance, it takes only 6 versions of the EMBL [423] sequence database to reach one terabyte of data.

Hence a service that will update each database site through the grid according to the last modification on their repository is needed. The system must be light and transparent enough to not disrupt job execution and to automate the procedure thus requiring a minimum of human intervention. At the end the service should be a black box delivering up to date databases on the grid that should prevent users from wondering which is the version deployed.

The update process can be achieved through the three steps presented in figure 27. First, given that all necessary information on the databases deployed and their respective repository locations are available, the comparison step informs the system if an update is needed. It is quite obvious no process is going to be started if the databases on the grid are already up to date, it would just consume unnecessary computing power and network resources. The comparison process indicates which parts of the database have been subject to modification and updates the grid database information. Those parts will be downloaded during the second step from the repositories. In the worst case, the entire database will be downloaded. The third step updates all the databases distributed on the grid thanks to the grid Information System.

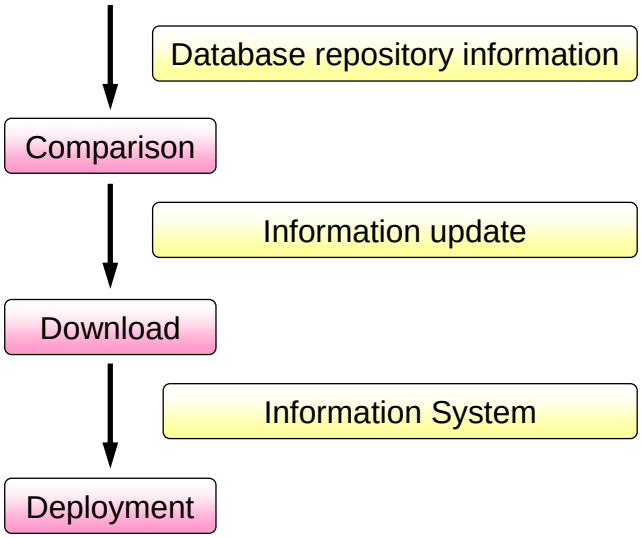


Figure 27: Database update process

After defining the components and the architecture of the service, we will describe its deployment on the RUGBI grid.

3.4.2. Service components

The service was developed in the framework of the RUGBI project. Several components needed for the service were detailed in chapter 2 and previously in this chapter: the Storage Element where the database is stored, the grid components for the storage process, the Controller managing the Information System and the DTD model of a database resource. They will not be described again here.

The next paragraph defines databases to understand how to update and distribute them on the grid. Then the concept of Storage Element of Reference is explained. Finally the DatabaseFinder service is described.

Databases to be updated and distributed

The databases required in the RUGBI project for *in silico* drug discovery are UniProtKB/Swissprot [420], UniProtKB/TrEMBL [420], EMBL [423], PDB [118], KEGG [424] and NCI [425]. The genomics and proteomics sequence databases EMBL and UniProtKB are used in the sequence alignment step of the protein structure prediction pipeline. The protein structure database PDB is used in the homology modeling step. The metabolic pathways database KEGG is used to identify molecular networks. The compound structure database NCI is used in the virtual screening pipeline.

They are constituted of several flat files, organized in directories. They are just handled as file systems: files can be added, removed or modified and are available directly on ftp servers anonymously.

The Information System of the RUGBI grid is based on XML description files of each element of the grid. There is an XML file for each database to specify for example its URL on the ftp server, the update frequency, etc.

The external database repository described in this chapter is an ftp server but the database update service will be able to download information from other servers, like http.

The Storage Element of Reference

There are several Storage Elements of Reference, one per database, which are repositories of the databases on the grid. A repository space is a persistent space used to store information (jobs cannot use this space). These grid repositories should be synchronized with the ftp servers.

Per Storage Element of Reference, there are several Storage Elements where the working repositories for the jobs can be found. A working repository is a space used to store information currently used by running jobs (as jobs are using these spaces, they are protected as critical sections).

The DatabaseFinder service

The RUGBI architecture also includes a service called the DatabaseFinder, used mainly to find the best location of a given database on the grid for a given user. Furthermore, the service manages locks put on the databases by the running jobs to prevent their modification while the job is being executed. Moreover it can forbid a job to use a database in a given location, hence allowing the update service to perform its operations without disrupting job execution. Locks are needed to ensure the safety of jobs and the integrity of the replication of the databases. For security issues, locks are automatically removed after a given period of time and jobs must renew their locks to protect the critical sections on which they run. If a job is stopped during its execution or if there is a failure preventing it from removing its lock, the lock will be limited in time.

3.4.3. Service architecture

The service is intended to be a direct interface between the database ftp servers and the Storage Elements of the grid. Our main developments will focus on the RUGBI grid infrastructure but the whole service is designed to work as a standalone tool that we can adapt easily to any middleware architecture and with any database.

The database update service as a client/server application

The update service as a client/server application is conceived in two parts communicating with each other. Figure 28 presents the update service architecture.

Firstly, the server, or master service, deployed on the Storage Element of Reference regularly checks external sites, typically ftp servers, for updates. It performs a comparison of the version available on the ftp with the one deployed on the grid using the XML description file of the databases given by the Information System.

If there is a difference, based on the date or a new file or repository, the update service replaces or rebuilds the databases by downloading the necessary files or directory on to repository spaces on the Storage Element of Reference. Other optional processes like format change or index computation are possible and can be described in the install attribute of the file.

After that first step, the master service queries the Information System to know the Storage Elements that host the database and notifies the clients deployed on them to update the database. The Configuration and Location service is used here. When an update notification is received by a client, it extracts the data from the Storage Element of Reference. To do that, it simply downloads the differences on the repository of the Storage Element of Reference in a new local working space, closed for jobs, using the specified transfer protocol.

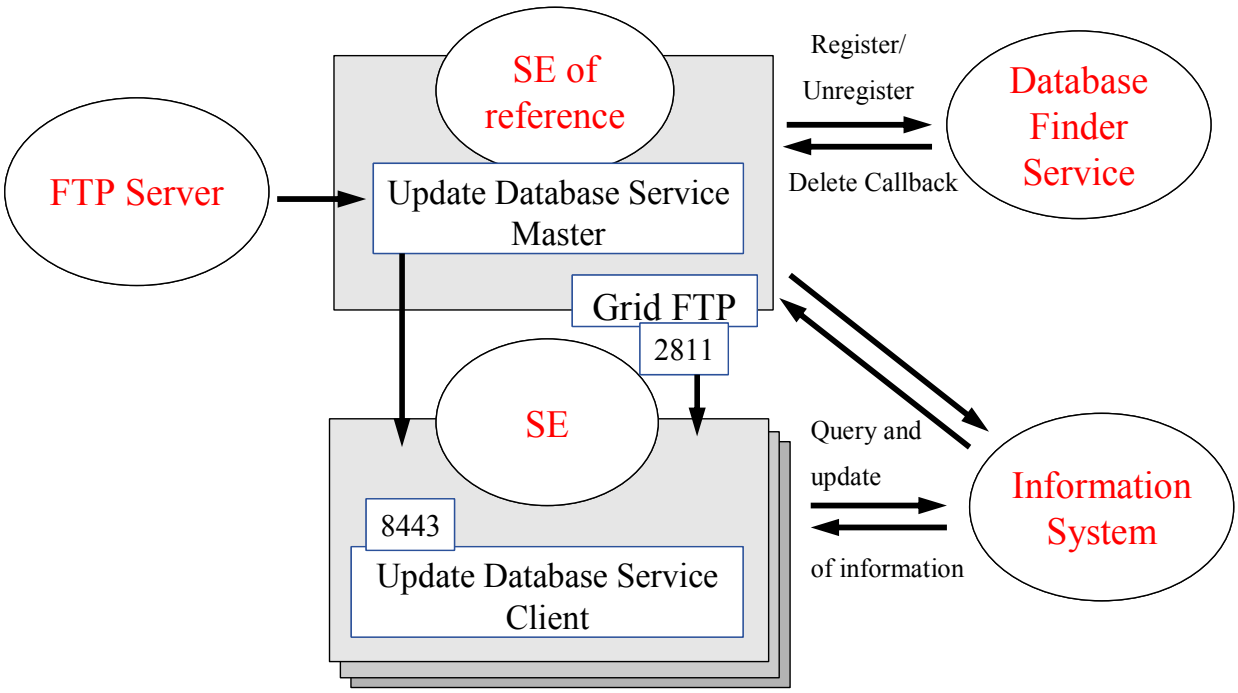


Figure 28: Architecture of the update service

As soon as the number of Storage Elements (clients) that have successfully deployed the update reaches a given threshold, the DatabaseFinder service is notified to register the new database. This implies that the working space will open and accept new jobs. If a database version is already on the Storage Element (in a working space), the same service is notified to unregister the old version. New jobs will not be allowed to run on the old version, since it is unregistered, but will use the new registered version. Running jobs, still submitted with the old version, are not stopped.

When the grid has registered the new deployed database, the XML description file of the database is updated on the Information System and the old one deleted.

As soon as no more jobs are using the old version, the server is notified by the Database Finder Service and it deletes the old version. That notification is propagated through the grid so that any Storage Element hosting this version will delete it. The whole process can be repeated through time to keep the databases updated.

For each database with a running update service, there is a database administrator. This administrator is responsible for changing the database file if the external database model changes. He can also launch himself an update when he thinks it is necessary. This transparent

service frees the user and the database administrator of technical constraints. They can concentrate on understanding and exploiting the data themselves.

Service implementation

For its integration the service must borrow some mechanisms from RUGBI and Globus Toolkit. The GridFTP protocol is used for grid data transfer. The external data transfers are done through basic ftp when the external site hosting the database is an ftp server. Signed proxy and X509 certificates are used for authentication and interoperability with several grid components. In order to allow further deployment of the service or interfacing with other grid infrastructures a basic core version was developed and the specific developments inherent to the integration with specific grid components were built on this core. Developments were made in Java for portability and modularity.

As the RUGBI grid was based first on Globus Toolkit version 3, and later on Globus Toolkit version 4, the sub services were also implemented as web services, messaging with SOAP protocol, in order to achieve better connectivity with the other services, and to avoid the handling by grid sites of exotic firewall configurations: they should just allow inbound and outbound connectivity for web services (8080 for Apache tomcat) and gridftp (2811 most of the time). The developments were done in an Apache Axis environment.

3.4.4. Result: periodical database updates and distribution on the RUGBI sites

The RUGBI grid has currently five sites in Clermont-Ferrand, Lyon and Grenoble. The client Update Service is deployed on each Storage Element of the grid (in each of these sites). The Clermont-Ferrand site hosts the Storage Element of Reference of the grid. Thus, the master service was deployed there.

The following bases are deployed and updated periodically from their repositories in Clermont-Ferrand and Grenoble: UniProtKB/SWISSPROT (700 MB), UniProtKB/TREMBL (2.4 GB), EMBL (release without annotations: 180 GB), KEGG (13 GB), PDB (2.9 GB), NCI (900 MB), representing a total of 200 GB. All database files in correct format and yet indexed are directly downloaded from the external reference sites.

These databases have an update cycle of 15 days on the grid. Once the update of a database is initiated from the portal by the database administrator, its reference site runs the update process each 15 days. The volume of the transfers required by each update varies from several kilobytes to several gigabytes depending on the databases and their activity. Performance of the complete update process depends on the network bandwidth.

3.4.5. Service limitations and perspectives

The service presents several limitations.

Limitations

A biological database records data as textual entries composed of a datum to which are associated metadata. In a protein sequence database the datum is the sequence and metadata

are items such as the identifier of the protein or the bibliographic references. The Database Update Service allows to only update the sequence flat files which will be used by software. The process is the same for a full database and could be implemented, but it requires indexing which is CPU intensive. The Database Update Service should be interconnected with indexing applications parallelized on the grid.

Only one Storage Element of Reference can host the service for a dedicated database. This is a single point of failure because the service must be permanent and can not be stopped. Other Storage Elements of Reference could be configured to host the service for the database, but it requires synchronization.

Relational databases are also used in life science, but the service is limited for updates of flat files. The CONStanza system [417] is designed for relational databases. The development of a new system merging functionalities from both services is being considered.

Perspectives

The service, packaged as a middleware component, is installed and deployed on the RUGBI grid by the site administrators as a grid plug-in. Its deployment on the French regional grid AuverGrid, which uses the EGEE middleware LCG-2, requires some interfacing with the job submission system to allow jobs to put locks on site databases to secure updates. Instead of being installed on a Storage Element, the service could be deployed on a User Interface with a specific service certificate registered in the biomedical Virtual Organization.

Used in conjunction with a database declaration server, the update service could be easily added to the DIET middleware [426], which follows the gridRPC API defined within the Global Grid Forum [427] and presents a hierarchical architecture where agents manage scheduling and some persistence mechanisms.

The new gLITE middleware [350] should be theoretically quite similar to RUGBI with its web service-friendly architecture providing APIs for new service integration. The update service could therefore be easily interfaced to gLITE.

Another perspective is to offer this service as a software application instead of a service. It could become a part of a grid workflow, and called only when a database update is necessary.

There are also future plans to optimize the deployment of databases: for example, being able to split databases and lock only parts of them. The service will mature through its deployments on grid middlewares in production environments.

3.5. Conclusion

In this third chapter, two grid-enabled services to deploy and update protein structure prediction software and database were deployed on the RUGBI grid.

The deployment service aims to ease the use of software and databases for non-grid expert users on a grid environment. It allows registration, download and installation thanks to interfaces generated by the Information System describing software and databases. The

environment is personalized for each user in the RUGBI environment. This deployment service is paramount to make grid use transparent for the scientist. Developments are still needed to improve interfaces and enrich functionalities.

The aim of the Database Update Service is to provide the grid users with the most up to date version of any biological flat file database, to do it transparently and without disturbing any running jobs. The frequent and regular update of the databases is a recurrent issue for scientists and all host or mirror bioinformatics centers. This service ensures that the user's analysis on grid will be carried out with the last available version of the database. It is installed for a precise database by its administrator but runs autonomously and periodically for small updates and for large new versions. Developments are still needed to interconnect the service with an execution service to parallelize recurrent time-consuming tasks such as indexing. There is a plan to migrate the service onto other grid environments like EGEE.

Deployment and update services for protein structure prediction software and databases contribute to personalizing the working environment for biologists and life science industrials.

The developments were carried out so that they would be easily modifiable and interoperable with new services. In the future, it will be possible to add an *in silico* experiment manager, with administration aspects, deployment of workflow, and project qualification, to valorize experimental results. This service will define and manage experiments with traceability, constituting a virtual laboratory.

Permanent bioinformatics services must be conceived for daily *in silico* drug discovery, but large scale grids allow large scale challenges in life science that could never be carried out before. The aim of the next chapters is to present the deployment of high throughput virtual screening by virtual docking against neglected and emerging infectious diseases and to report issues related to the deployment and the monitoring of the *in silico* docking experiment as well as experience with grid operations and services.

3.6. References

[404] Teo, Y.-M., et al., GLAD: a system for developing and deploying large-scale bioinformatics grid, Bioinformatics 21:794-802 (2005).
[405] http://biogrid.icm.edu.pl/
[406] http://www.apbionet.org/grid/
[407] Lavenier, D., et al., Le projet GénoGRID : une grille expérimentale pour la génomique, Journées Ouvertes Biologie, Informatique et Mathématique 2002, (2002).
[408] Gallopoulos, S., et al., Computer as Thinker/Doer: Problem-Solving Environments for Computational Science, Computational Science and Engineering IEEE, (1994).
[409] Cannataro, M., et al., Proteus, a Grid based Problem Solving Environment for Bioinformatics: Architecture and Experiments, IEEE Computational Intelligence Bulletin, (2003).
[410] Aloisio, G., et al., ProGenGrid: A Workflow Service Infrastructure for Composing and Executing Bioinformatics Grid Services, 18th IEEE Symposium on Computer-Based Medical Systems 555-560 (2005).
[411] Walker, D., et al., The Software Architecture of a Distributed Problem-Solving Environment, Concurrency: Practice and Experience 12(15) (2000).
[412] Hull, D., et al., Taverna: a tool for building and running workflows of services, Nucleic Acids Research 34:W729-W732 (2006).

[413] http://www.ebi.ac.uk/
[414] http://www.ncbi.nlm.nih.gov/
[415] Combet, C., et al., NPS@: Network Protein Sequence Analysis, Tibs 25 147-150 (2000).
[416] http://www.lionbioscience.com/
[417] Domenici, A., et al, Relaxed Data Consistency with CONStanza, 6th IEEE International Symposium on Cluster Computing and the Grid, IEEE Computer Society Press 16-19 (2006).
[418] www.uppmax.uu.se/Members/lottab/xndt-on-swegrid/
[419] Blanchet, C., Grid Deployment of Legacy Bioinformatics Applications with Transparent Data Access, IEEE Conference on Grid Computing (2006).
[420] Bairoch, A. and Apweiler, R., The SWISS–PROT protein sequence data bank and its supplement TrEMBL, Nucleic Acids Res. 27 49-54 (1999).
[421] http://jaxen.sourceforge.net
[422] Garnier, J., et al., GOR secondary structure prediction method version IV, Methods in Enzymology 266 540-553 (1996).
[423] Stoesser, G., et al., the EMBL nucleotide sequence database, Nucleic Acids Res. 27 18-24 (1999).
[424] Kanehisa, M. and Goto, S., KEGG: Kyoto Encyclopedia of Genes and Genomes. Nucleic Acids Res. 28 27-30 (2000).
[425] http://dtp.nci.nih.gov/docs/3d_database/Structural_information/structural_data.html/
[426] Caron, E. and Desprez, F., DIET: A Scalable Toolbox to Build Network Enabled Servers on the Grid, Technical report RR-5601, INRIA (2005).
[427] https://forge.gridforum.org/projects/gridrpc-wg/

Chapter 4. Grid-enabled high throughput structure-based virtual screening by docking

4.1. Introduction

Structure-based virtual screening is one of the first steps of the drug discovery process once a molecular target has been identified. It is about selecting the compounds, i.e. the molecules, which could impact the target biochemical activity. High throughput structure-based virtual screening by molecular docking is a technique which can screen millions of compounds rapidly, reliably and cost effectively. It is particularly useful in the discovery of drugs against neglected and emerging infectious diseases where reducing the duration and the cost of the process is essential.

WISDOM is an international initiative to enable a virtual screening pipeline on a grid infrastructure against neglected and emerging infectious diseases. Its first venture was to deploy large scale *in silico* docking on a public cluster grid infrastructure. Given the very large amount of data involved in the computation, such a large scale deployment is a stressful experiment for the grid infrastructure, which is called a data challenge. Several applications were already deployed at a large scale on cluster or desktop grid infrastructures. For instance, the particle physics experiments on the Large Hadron Collider currently being built at CERN are the first applications having set up such data challenges on a public cluster grid.

The preparation of the deployment included the development of an environment for job submission and output data collection. This environment had to be able to handle the submission of about 70,000 15-hour long jobs and the collection of an equivalent number of output data. One major issue was the handling of job resubmissions whenever a job failed for any reason.

The aim of this chapter is to describe the WISDOM production environment designed for a large scale deployment on the public cluster grid infrastructure EGEE.

The chapter is introduced as follows:

- After the introduction, the second section introduces the requirements for such a deployment. Then different grid-enabled initiatives to deploy large scale virtual screening or particle physics experiments are reported.
- The third section describes the bioinformatics components of a docking experiment. These first three sections are important in order to understand how to build an efficient WISDOM production system.

- Finally, the last section presents the design of the WISDOM production system: specific issues, preparatory steps on the AuverGrid infrastructure, production system design, license management and accounting data management are detailed.

4.2. Requirements

Advances in combinatorial chemistry have paved the way for synthesizing millions of different chemical compounds. Thus there are millions of chemical compounds available in the laboratories and recorded in 2D or 3D electronic databases. But screening millions of chemical compounds *in silico* is a complex process. Depending on each compound's structural capacity, screening requires from minutes to hours of computation time on a standard PC and produces several megabytes of data. Consequently, screening all the compounds in a single database would require years and hundreds of megabytes to terabytes. However, the problem is embarrassingly parallel and the computation time can be reduced very significantly by distributing data to process over a grid gathering thousands of computers. For instance, reducing the virtual screening time from 100 years on a basic PC to only 1 month requires more than 1250 dedicated processors.

Such deployment on a large scale requires:

- Multidisciplinary collaboration providing the application package adapted to the grid. The application package is composed of data and software. Bioinformatics applications are not designed for grid computing. Their code can not be modified for several reasons, such as application copyright or application invalidity in case of modifications. A common strategy is to split the application into shorter tasks, or jobs, in order to distribute them in parallel on the different Worker Nodes of a grid. The splitting factor is often dependant on the number of smaller subsets that can be produced by splitting the database.
- Robust, stable and large infrastructure access providing maintained computing and storage resources. Reliable resource estimation is required before the deployment. The application package needs to be installed on at least one resource to be available.
- System providing automated and fault-tolerant job submission, job monitoring and output retrieval. The number of jobs for a large scale deployment is very high (tens of thousands to millions). Preparation, fast submission and efficient monitoring on a distributed and heterogeneous environment such as a grid are a manually labor-intensive task. This tool must be able to manage large data or large numbers of small data. This tool must provide monitoring data for the process control by the user. Fault-tolerance involves identification, failure resolution and job resubmission process. Thus this production system is at the interface with the grid middleware managing resources.
- User interface for deployment management. Today, large scale deployment is still a task for a grid expert. Consequently, the deployment system does not require an end-

user-friendly interface. But it requires at least simple command lines management and an expert monitoring view.

- License management for commercial software. There are several types of licenses. For instance, the license price can depend of the number of processors used. Such a system is not viable in a grid infrastructure gathering thousands of processors. Editors have not yet adapted their license system to the grid.
- Efficient and responsive user support of the infrastructure. A failure in a large scale deployment can impact a large number of jobs. The deployment can be stopped to solve the problem. Thus fast feedback and quality of service are important.
- Tools providing global statistic data and figures about the large scale deployment. A large scale deployment involves many actors, such as a site administrator, an organization leader or a funding provider. The deployment must be assessed to prove its performance and to identify new issues.

Thus large scale deployment of computing and data intensive tasks requires a large infrastructure interfacing to a system providing many efficient services. Beyond these requirements, in the context of the fight against neglected diseases, the required tools and infrastructures should be freely available for the application. Additionally, in the context of the EGEE project, the deployment tools interfaced with the EGEE infrastructure should use the EGEE middleware components for preference.

The next section presents several projects and applications aiming to deploy virtual screening on grid, but also large scale application from other scientific fields.

4.3. Related works

Recently, high throughput virtual screening projects on grids have emerged with the perspective to reduce costs and time. Below are reported the main advanced projects on desktop and cluster grids. Then several deployment tools of large scale experiments on cluster grid are presented.

Several criteria, when they are available in publications or technical reports, are used to describe the grid performance of the deployment tools.

- The total CPU time corresponds to the cumulated amount of CPU used for a given application.
- The duration represents the total elapsed time between the submission of the first job and the end of the last job.
- The crunching factor represents the gain of time obtained thanks to the grid deployment. It is simply obtained by dividing the total CPU time by the execution duration.
- The approximated distribution efficiency is defined as the ratio between the crunching factor and the maximum number of jobs running in parallel on the grid. This parameter is based on the approximation that the maximum number of jobs running in parallel on the grid corresponds to the number of available processors during the

overall period of the deployment, and the crunching factor is the constant number of running jobs.

4.3.1. High Throughput Virtual screening on a cluster grid

The Virtual Laboratory project [253] was the pioneer in enabling molecular modeling for drug design on geographically distributed resources. It optimizes for time or cost the use of grid resource. The Nimrod-G [252,428] resource broker, based on Globus, is used for scheduling and on-demand processing of docking jobs on the World-Wide Grid resources. The Virtual Laboratory is used by the Australian BioGrid Portal to easily submit docking jobs [429]. The test case reported in [429] was deployed on 7 nodes with about 450 CPUs in only few hours.

The purpose of the Drug Discovery Grid [430] is to set up a desktop and cluster grid environment with aggregated computing and data resources to provide drug virtual screening services and pharmaceutical chemistry information services. BOINC is the desktop grid middleware and Nimrod-G is the cluster grid middleware. The Drug Discovery Grid composed of 5 sites with 336 CPUs was used to screen 120,000 compounds against an interesting target for the treatment of hyperlipidemia, cholelithiasis and cholestasis.

GROCK [431] is a portal that facilitates mass screening of potential molecular interactions in the Life Sciences. It aims to facilitate for users the performance of huge amounts of computational tasks using EGEE. They use LCG-submitter, a tool developed by the Experiment Integration and Support team as an interface with the EGEE middleware for job submission and monitoring. The GROCK portal is not yet available and no deployment has been reported.

The Grid-based Virtual Screening of Dengue Virus Target Proteins project, supported on the national Swiss grid SwissBioGrid, aims to find new compounds against Dengue [432]. The first massive deployment was made on 4 heterogeneous nodes (desktops and clusters) with 360 CPUs using the in-house middleware ProtoGrid. They docked 500,000 compounds and consumed about 4 CPU years. First results are promising but the technology choice for the middleware on SwissBioGrid is not yet confirmed.

These four initiatives aim to provide full virtual screening environments on large grids. But there are no reports of virtual screening deployment on thousands of processors for many weeks such as is required for high throughput virtual screening. Access to the grid infrastructures used by these applications (except Grock) is not possible for external initiatives such as WISDOM.

4.3.2. Large scale deployment on a desktop grid

There are a few efforts that use desktop grids for processing docking jobs in parallel against a specific target. One effort is the FightAIDS@Home project [433], which is based on the World Community Grid [312] and Autodock [434] docking application. World Community Grid sets 400,000 CPU units for three different projects. The Cancer Screensaver Project [296] has used 447,000 CPU years since 2001. Over 3.4 million computers have

joined the initiative. D2OL [315] is using its own platform to screen a large compound database against targets of different diseases such as malaria and avian influenza.

Novartis is a successful example of the deployment of an internal desktop grid. Their vision is to manage the knowledge and the informatics of virtual screening in their tightly protected grid environment [267]. They deployed the first automated modeling and docking pipeline on grid using the commercial United Devices platform. From 400,000 compounds, they selected a few promising molecules against the protein kinase CK2 with 1200 computers in 6 days instead of 6.4 years [435]. The crunching factor is about 387, thus the approximated distribution efficiency is only 32%.

A protein structure prediction application experience, Predictor@home, was recently reported in [149] which will aim to deploy the BOINC framework with 6786 desktop PCs for 3 months consuming 380 CPU years. Use of the BOINC framework requires specific application developments. Even if the deployment succeeds, there are still technological challenges such as server limitation and service stops due to maintenance.

These initiatives group together many computing resources, but they are based on desktop grid technologies. There are no detailed experience reports about data challenge deployment (except Predictor@home).

4.3.3. Job submission systems for large scale applications on the cluster grid EGEE

Many different applications were deployed on the large scale EGEE infrastructure. The particle physics experiments are the first applications to have taken benefit from the infrastructure. Their experience for large scale deployment is thus presented below.

The four particle physics experiments on the Large Hadron Collider

Four particle physics experiments associated with the Large Hadron Collider (LHC) are currently being built at CERN: Alice [436], ATLAS [437], CMS [438] and LHCb [439]. The volume of data generated by these experiments is expected to be in petabytes which need to be distributed to physicists around the world for analysis. Even though the detector will not take data until 2007, there is a large-scale data simulation and analysis effort underway for each experiment. Each experiment developed one or several analysis job submission systems and tested them at a real large scale on one or several grids (tens of sites, hundreds of CPU years, terabytes of managed data). Below are presented the job submission systems of each experiment used on the EGEE infrastructure.

Alien, the job submission system of the Alice experiment

Alien [440] was developed for the Alice experiment. It offers a single interface for ALICE users. It is a pull model service: a server holds a master task queue of jobs and uses the Resource Broker to submit many job agents. Once running on a Worker Node of a Computer Element, an agent calls a large job set, which is waiting in the task queue. Thus the job set is executing on all free Worker Nodes of the cluster bypassing the Resource Broker.

No Information System is included in Alien. The monitoring system of the Alice experiment uses MonaLisa [441].

This system is clearly powerful for the Alien user. But it is concurrent to the EGEE Workload Management System, composed of Resource Brokers, used by other grid users. Moreover information extracted from Resource Brokers for the global accounting of the EGEE project is consequently incomplete [442].

The ATLAS production system, the job submission system of the ATLAS experiment

The ATLAS production system [443] uses middleware components as much as possible. A job supervisor submits, monitors and resubmits if necessary the jobs to an executor. In a first time, the executor was in-house job submitter using the Resource Brokers. Then, Lexor-G [444], using the Condor-G service [445], was integrated to avoid the Resource Brokers, increase the job submission rate to the grid and make better use of the CPU resource. The data management is based on the EGEE Data Management System. A relational database manages the job information. Ganga [446,447] is used for the User Interface. The monitoring system of the ATLAS experiment uses GridICE [448]. A few services are dedicated (like Resource Brokers) to the experiment providing better stability and management.

Several of the problems encountered during the ATLAS experiment deployments were very similar to the problems met during the WISDOM deployment (Data Management, site instability…). But they were not reported before the WISDOM production system development. Moreover, the deployment experience can not be fully validated in an open grid environment as long as the ATLAS experiment is using dedicated services.

BOSS and CRAB, the job submission system of the CMS experiment

The CMS experiment used CRAB [449] as a User Interface and BOSS [450] as a monitoring system. CRAB is a command line environment interfaced with BOSS allowing the job preparation, submission, follow-up and output retrieval using the EGEE grid technologies. BOSS (Batch Object Submission System) has been developed to provide real-time monitoring and bookkeeping of jobs submitted to a compute farm system. It parses the log files of the jobs (i.e. the standard output and standard error of the job executable). It needs to be installed on each Worker Node. The last deployment of the CMS experiment [451], involving 15,000 jobs, was done with 2 dedicated Resource Brokers during two weeks with a success rate of 90%.

CRAB is not a full automated environment because the user needs to manage a set of jobs by command line. Moreover, BOSS retrieves information only from Worker Nodes, not from Resource Brokers. The grid overhead time is thus not reported. CRAB was not validated by such a large scale experiment when the first WISDOM deployment was developed.

DIRAC, the job submission system of the LHCb experiment

The DIRAC [452,453] Workload Management System is made up of central services and distributed agents. With the same approach as Alien, agents submitted to EGEE or DIRAC sites are requesting jobs whenever the corresponding resource is available. The execution environment is checked before jobs are delivered to the Worker Nodes. Pilot agents go

through the Resource Brokers as normal jobs. The monitoring data are stored in a relational database. The Data Management System is the same than as the EGEE grid. Ganga is used for the User Interface. During a LHCb experiment deployment [454], DIRAC managed up to 5,000 simultaneous jobs. There are gains in efficiency when the number of users is reduced.

Again, the DIRAC system is well adapted for LHCb experiment users, but does not profit other grid communities.

JAM, the job submission system for an application to find functional analogous gene products

Job Application Monitoring (JAM) is the grid framework used to find functional analogous gene products [455]. It was first used by the CMS experiment. JAM manages large job submissions and monitoring thanks to the EGEE components and a relational database in the same model as the BOSS system. The output retrieval is done by scripts bypassing the Resource Brokers. The application was recently deployed on the INFN and EGEE infrastructures using 3 Resource Brokers. 2,400 CPUs were used by 95,000 jobs and 60 CPU years were consumed in 1 month on 64 Computing Elements. The crunching factor is thus 730. The success rate is 45%.

This job submission system is close to the WISDOM system, except in the use of a relational database. The low success rate is corrected by the resubmission process. The application was deployed after the first WISDOM deployment. Such a deployment also confirms issues revealed by the WISDOM experience.

GridICE and Monalisa, two monitoring services for users

To manage a large scale production, a grid monitoring and accounting tool is very helpful for tracing the progress as well as the failures of the jobs. GridICE [448] and MonaLisa [441] services collect information from agents deployed on the grid nodes and from the Information System. Such information is stored in a database which enables, via a web interface, to retrieve the real time status and historical events about jobs and services. Views focus on different aspects of job and resources monitoring.

These two distinct and user-friendly services are limited to the scope of a Virtual Organization. They are thus not adapted for an individual application inside a Virtual Organization such as the WISDOM application deployed in the biomedical Virtual Organization. Moreover, there are nodes or Resource Brokers where the GridICE and MonaLisa agents are not installed. Furthermore, the grid resources can be incorrectly configured. The partial information is thus difficult to interpret for an end-user.

4.3.4. Comparison with the WISDOM production system

Compared to these initiatives, WISDOM is the first attempt to deploy large scale *in silico* docking on a public grid infrastructure. As highlighted above, previous virtual screening deployments were either limited to grids of few clusters or to desktop grids. The job submission system (Alien, BOSS, DIRAC, the ATLAS production system, JAM) and monitoring system (GridICE, MonaLisa) review lead the WISDOM initiative to develop its

own production system. Compared to Alien and DIRAC system, the WISDOM production system aims to be developed on EGEE middleware components, without dedicated services. The BOSS system bypasses the monitoring information from the Resource Brokers. The ATLAS production system and the JAM system were not fully validated when the WISDOM production system was developed. The two studied monitoring systems GridICE and MonaLisa are not adapted for a single application in a Virtual Organization.

Before detailing the WISDOM production system, the next section describes the bioinformatics components to be deployed.

4.4. The bioinformatics components

Virtual screening deployment by docking required selection and preparation of docking software, biological targets and a library of virtual or effective compounds to test.

4.4.1. The target

The target is typically a protein which plays a pivotal role in a pathological process, e.g. the biological cycles of a given pathogen (parasite, virus, bacteria…). The goal is to identify which molecules could dock on the protein active sites in order to inhibit its action and therefore interfere with the molecular processes.

For each target to be docked, 3D structures are required. The 3D coordinates of the structures are routinely obtained from the Protein Data Bank (PDB) [118]. The active site of the protein needs also to be prepared according to the docking tool (adding charges, hydrogens, water molecules…). All available target structures can be used to generate only one composite structure, or the screening process can be conducted on each individual structure requiring more computing time. The structure should have a resolution of less than 2 Angstrom. Finally, if needed, the target structure files are converted into a format ready for use by the docking software. The size of a target structure is in the order of hundreds of kilobytes to megabytes.

Figure 29 presents a step in target preparation. 5 target structures selected from PDB are superimposed with their compounds. The different protein structures used as templates share high sequence similarity. Variations exist in the loop regions and are indicated in yellow. The co-crystallized compounds are represented as balls and sticks in red.

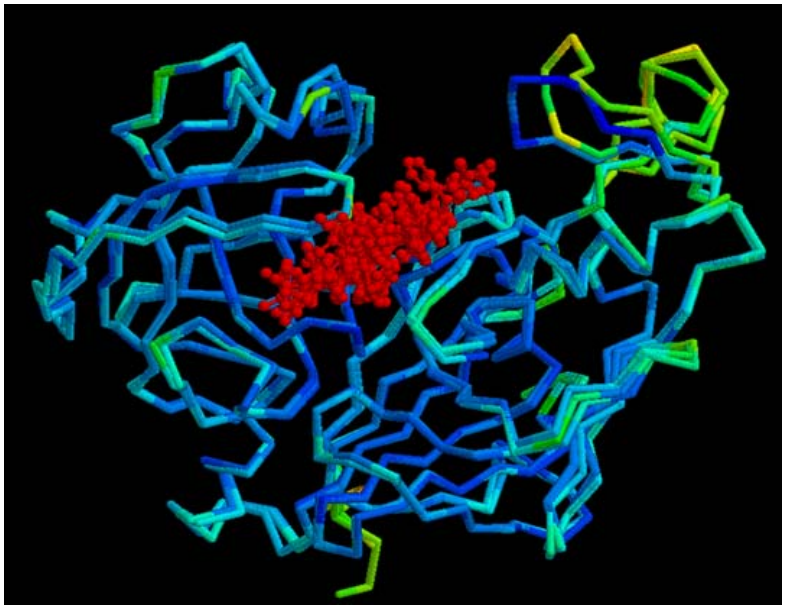


Figure 29: Five target structures with their co-crystallized compounds are superimposed. This picture is generated by using Rasmol software.

4.4.2. The chemolibrary

Chemolibraries are libraries of compound 2D or 3D structures. They are made openly available by academic laboratories or by chemistry companies which can produce them. They are updated monthly like supplier catalogs. Compounds for docking are selected before docking according to different criteria (diversity, drug likeness…) and the target (hydrophobic, docking models…). 3D structure of 2D compounds can be predicted.

Each compound database requires storage space in the order of tens of megabytes to terabytes. Information about the compound to be docked must be extracted from the database. If needed, the compound structure files are converted into a format ready for use by the docking software. Docking scores of compounds are computed after carefully adjusting the parameters of the docking software.

Figure 30 presents the two-dimensional (2D) schema of WISDOM-490500, a compound docked during the first WISDOM deployment.

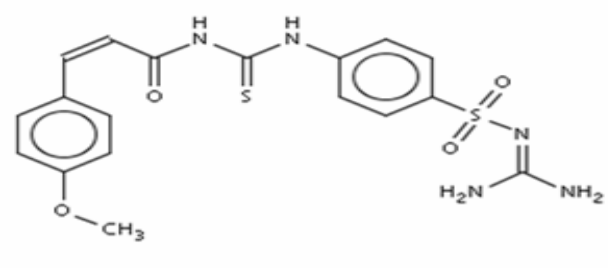


Figure 30: 2D schema of the compound WISDOM-490500

4.4.3. The software

Many docking software are available either open-source or licensed. They differ in the sampling algorithms used, the handling of compound and protein flexibility, the scoring functions they employ, and the CPU time required to dock a molecule to a given target. The docking applications share the same processing model master/worker. The tasks are independent and the data can be easily splitted. Only a small proportion of the execution time is spent on data communication.

The docking tools used in the first WISDOM initiative are FlexX [180], a commercial software made graciously available by BioSolveIT [456] for a limited time, and Autodock [434], a software which is open-source for academic laboratories and which uses a different docking method. Before a docking experiment, different parameter sets are generated according to the software and the target. The size of a software is in the order of megabytes.

The output of a docking-based screen is a set of 3D models of the predicted binding mode of each compound against the target structure, together with a ranking that is a measure of the quality of fit. The size of the output is different for each docking software. It is in the order of kilobytes for each single computation. The output size of a large scale deployment is in the order of hundreds of megabytes to terabytes.

Bioinformatics components need to be installed or sent on the grid. The next section focuses on the WISDOM production environment in charge of running the docking application.

4.5. The WISDOM docking production environment on EGEE

The grid is often perceived in life science as a tool for time-consuming applications, but no life science application had been deployed at a large scale on a cluster grid infrastructure before WISDOM. Preparation of the deployment included the development of an environment for job submission and output data collection. This environment had to be able to handle the submission of about 70,000 15-hour long jobs and the collection of the output data.

A major issue was to handle job resubmission whenever a job failed for any reason, as the grid success rate was typically of the order of 80% [457]. Large scale tests were made on the French regional grid AuverGrid to validate the environment and to identify potential issues and bottlenecks. Other issues were raised by the data challenge, like the usage of licensed software on the grid or the need for a high throughput job submission scheme.

In this section, specific issues related to WISDOM deployment are introduced. Then, the results and lessons learned from the large scale tests deployed on the French regional grid AuverGrid are presented. Finally, the WISDOM production environment which was designed to achieve production of a large amount of data in a limited time with a minimal human cost using EGEE middleware services is detailed.

4.5.1. Specific issues relating to WISDOM deployment

A docking job requires an input file (the target), a database of independent molecules (a compound file), a set of parameters provided in a file or by command line and a docking software. A number of issues need to be addressed to achieve significant acceleration from the grid deployment. Previous experience with LCG middleware indicated potential bottlenecks:

- Grid performances are impacted by the amount of data moved around at job submission. As a consequence, the files providing the 3D structure of targets and compounds should preferably be stored on grid Storage Elements in preparation for the data challenge.
- The rate at which jobs are submitted to the grid resource brokers must be carefully monitored in order to avoid their overload. The job submission scheme must take into account this present limitation of the EGEE brokering system.
- The grid submission process introduces significant delays for instance at the level of resource brokering. The jobs submitted to the grid computing nodes must be sufficiently long in order to reduce the impact of this middleware overhead.
- Use of licensed software requires designing a strategy to distribute licenses on the grid.

The WISDOM production system was designed to address the issues listed above. It had also to automatically recover from errors to avoid extremely tedious manual intervention. Indeed, the job success rate has kept increasing on the EGEE infrastructure since April 2004 but the achieved efficiency of the order of 80% required handling about 20% of failed jobs.

In the next sections, preparation of the large scale deployment is described as well as the production environment which was developed to maximize job throughput on the grid after analyzing the tests on the AuverGrid regional grid. The strategy adopted for the deployment of licensed software is also discussed.

4.5.2. WISDOM preparation

The preparation took place in several steps.

In order to limit the amount of data transferred at job submission, docking software were stored on each Computing Element. The files providing the structures of the target and the compounds were stored on the Storage Elements. The number of compounds docked per job was estimated so that the job duration was approximately 15 hours. FlexX being much faster than Autodock, different compound files were used as input to the 2 docking software. As a result, about 600 MB of data were stored on each grid Storage Element of the biomedical Virtual Organization.

To achieve this preparation step, two packages were developed. They are responsible for installing the application components on the resources and for testing these components, together with the resources and grid services. Indeed, a stress usage of the grid in a limited time requires resources to be available immediately and reliably. This requires checking the status of the biomedical Virtual Organization services and of each of the grid node

committing resources to the Virtual Organization. A very important inefficiency factor comes from sites which are wrongly configured and where jobs systematically fail.

Before WISDOM was launched at its full scale on the EGEE infrastructure in the summer of 2005, deployment went through a ramping process in order to study the performances of the production system and to identify potential bottlenecks. During Christmas break in December 2004, up to 150,000 compounds were docked on the AuverGrid infrastructure, a regional grid which brings together the main laboratories of the Auvergne region using EGEE middleware to share technologies, skills and resources. Several instances of variable size were submitted. Table 3 presents some of the parameters which were monitored for 2 of the instances submitted, one of 50 jobs (2,000 dockings) and one of 500 jobs (100,000 dockings):

- The total CPU time corresponds to the cumulated amount of CPU used for a given instance.
- The duration represents the total elapsed time between the submission of the first job and the end of the last job.
- The crunching factor represents the gain of time obtained thanks to the grid deployment. It is simply obtained by dividing the total CPU time by the execution duration.
- The grid performance is a measurement of the grid efficiency. It takes into account grid inefficiencies due to job submission failure, aborted jobs, and loss of resources due to competing jobs by other users, etc. It is computed by analyzing all error messages from the grid Information System and job log file. For instance, the second test case was launched in a period with an important number of competing jobs, which explains the relatively low grid performance.
- The CPU time for 1 job indicates the average computing time for each job on one of the grid PCs.
- The grid overhead time for 1 job indicates the extra time due to the deployment on the grid. It includes all the extra delays coming from the different grid components (scheduling, queuing...).
- The data transfer time corresponds to the time needed to transfer the input data to the working nodes at job submission time.

We can conclude from table 3 that the submission of longer jobs was definitely more efficient as it reduced the relative grid overhead. Delays due to data transfer were found to be negligible, but this was expected as the different sites used for this test on AuverGrid benefit from high bandwidth network connections.

During these tests it was also noticed that upgrading grid nodes to new versions of middleware was often generating instability and loss of efficiency. The necessity to adapt our deployment process to the grid limitations was understood, for instance the limited number of files on each Storage Element or Worker Node or the fact that each Resource Broker uses a different Berkeley Database Information Index for listing available resources. Due to the low Berkeley Database Information Index update frequency in a large scale system, a resource can still be registered in the Information System even if it is in fact out of the grid.

Metrics	2000 docking in 50 jobs	100,000 docking in 500 jobs
Total CPU time	2.5 days	188 days (6.3 months)
Duration	2.5 hours	40 hours
Crunching factor	24	113
Grid performance	50%	30%
CPU time for 1 job	1.2 hours	9 hours
Grid overhead time for 1 job	7.2 minutes	30 minutes
Data transfer time for 1 job	< 1 minute	2,5 minutes

Table 3: Relevant parameters of 2 test cases deployed on the AuverGrid infrastructure during WISDOM preparation.

4.5.3. WISDOM execution

Based on the experience acquired during the testing phase on AuverGrid, the WISDOM production system, presented in figure 31, was developed in Perl, except for the multithreaded job submission tool which was developed in Java. The entry point is a simple command line tool. Its users during the data challenge were members of the Biomedical Task Force which gathers a team of engineers with recognized expertise in application development and deployment.

This software environment was developed to allow the submission and monitoring of job sets which were called instances. The different jobs of a given instance have the same target input and docking software. They only differ in the molecules of the compound library which are docked. Tasks needed to submit an instance were automatically executed by the WISDOM execution system. The user, authenticated by a proxy certificate, had to start his or her instance execution following a precise submission schedule to avoid too much concurrency between the computation participants leading to a grid overload. Once the computation has started, the WISDOM environment takes care of monitoring jobs and registering results. The user only has to check regularly if the process is running correctly, up to the end of all the jobs belonging to the instance. The overall process progression could be monitored through an output file for follow-up messages and an error file in case of problems.

For each instance, a configuration file contained the instance information (software, target, database, parameter settings) and the grid parameters (number of jobs for the instance, Resource Brokers, Computing Elements, Storage Elements). A shell script and a Job Description Language file were created for each job and used by the submission tool.

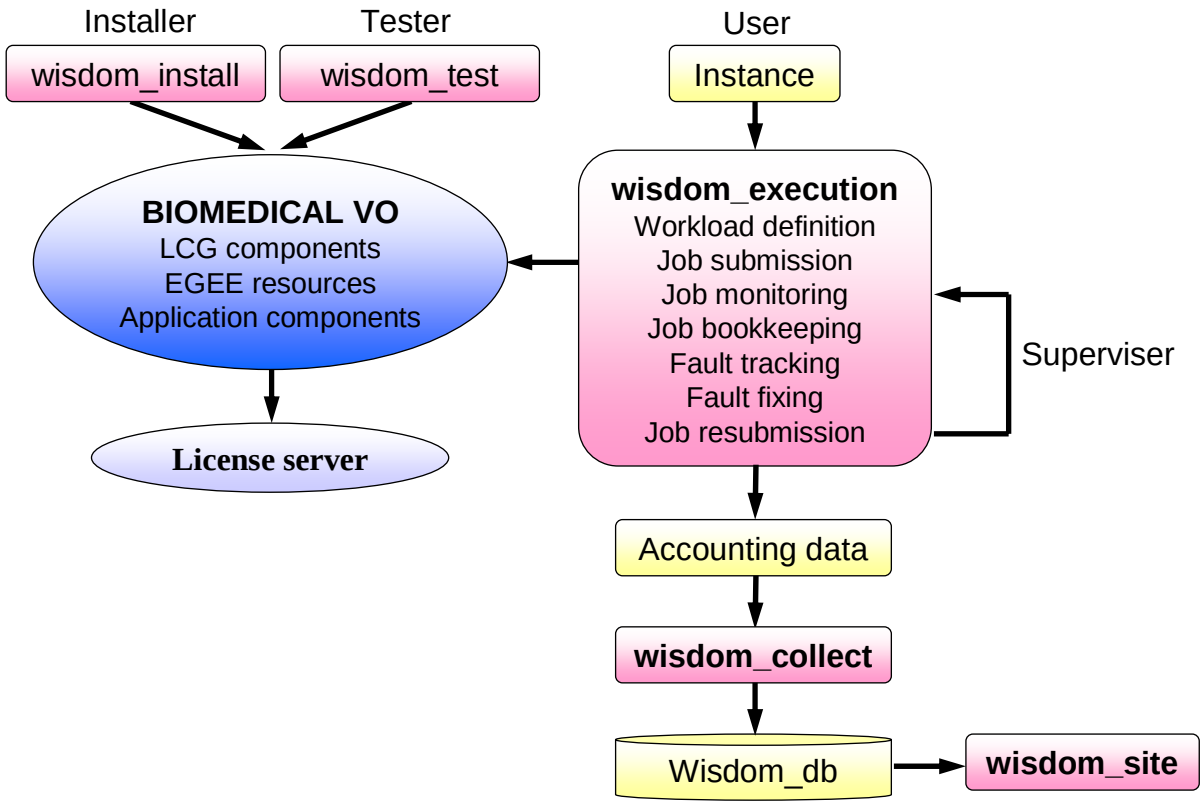


Figure 31: Design of the WISDOM production system.

On the Worker Node, after the environment was configured, the shell script downloaded the database file from a Storage Element chosen by the Information System using the LCG API. Then binaries were called with the target and parameter settings transferred with the job. The compressed result was stored on a Storage Element and registered in the grid file catalogue. A backup copy was also generated on another Storage Element. For the sake of simplification, the most relevant metadata relative to the output (software, parameter settings, compounds database, etc.) were stored in the name of the output itself. Output, errors and accounting messages were transferred on the User Interface.

A multithreaded job submission was developed specifically for a bulk and efficient submission on the Resource Brokers to address the limitation of LCG API which submits jobs one by one on one Resource Broker (with a minimal latency time of 7 seconds). The aim of the tool was to reduce the time needed for sequential submission of the jobs on Resource Brokers by parallelizing job submission on several Resource Brokers.

Once the jobs were submitted, the supervisor needed a real-time overview of the production. For a large number of jobs, the WISDOM environment includes an automatic monitoring tool which was developed to check the status of the jobs with the bulk monitoring LCG API and to prompt a reaction if needed.

Discovering and fixing failures was crucial for the process, and for the evaluation of grid performances. For each job that failed, the system was able to track the fault in the Workload Management System status message or in the job output content. After any corrective

operation, like deletion of a faulty Computing Element in the attributes of the Job Description Language file, the job was resubmitted. Preventive actions were also taken during the process. Then GridFTP was used to transfer data on the grid automatically from the Worker Node when the LCG Data Management Service failed. The output files were also carefully stored on several Storage Elements for security reasons.

4.5.4. License management

During the data challenge, commercial docking software (FlexX) with floating licenses was used on the EGEE infrastructure. 3000 floating licenses were made available by BioSolveIT for 3 weeks to be distributed on the grid.

Each job using FlexX software contacted the Flexlm server at the beginning of the job and asked for a license, namely an ASCII file with specific keys generated for this server. Then the job was able to run without connection to the license server. Accessing floating licenses on the grid behind firewalls required known IP and the opening of two specific ports for institutes hosting Worker Nodes. During the first three weeks of the data challenge, FlexX licenses were available through the license server. Afterwards, the Autodock phase started in August for about 2.5 weeks.

4.5.5. Collection and presentation of accounting data

Accounting data are required to evaluate the performance of the grid and to quantify the large-scale deployment on the grid. It is also a means to find missing docked compounds in the output file: after the data challenge, a control was done to find out if the missing compounds were due to a grid failure or to the docking process.

At the end of the instance execution, Logging and Bookkeeping services of each Resource Broker are used for bookkeeping with LCG APIs. Data are transferred on a remote machine from the User Interface to be stored in a relational database. A web site provides access to the database to dynamically build different statistics and histograms.

4.6. Conclusion

In this fourth chapter, a grid-enabled high throughput structure-based virtual screening by docking application was defined.

A large scale deployment of computing and data intensive tasks, called a data challenge, requires a large infrastructure interfacing to a job submission system providing many efficient services. In the context of the fight against neglected diseases, the required tools and infrastructures should be freely available for the application. Additionally, in the context of the EGEE project, the tools interfaced with the EGEE infrastructure to deploy the application should use the EGEE middleware components for preference.

Many initiatives aim to provide a full virtual screening environment on large grids. But there are no reports of virtual screening deployment on thousands of processors for several weeks such as is required for high throughput virtual screening. Other initiatives are based on

desktop grid technologies, but this system is limited for a full virtual screening deployment. The particle physics experiments, associated with the Large Hadron Collider currently being built at CERN, are the first applications having deployed such a data challenge. But their job submission systems bypass the EGEE middleware components, use dedicated services or were not validated at the time of the WISDOM development.

WISDOM is the first attempt to deploy large scale *in silico* docking on a public grid infrastructure. A docking application is composed of target structures, a compound database and software associated with parameter settings. Such an application can be deployed on the EGEE grid thanks to the WISDOM production system. Experience acquired during the testing phase on AuverGrid infrastructure brings to light some issues, such as grid nodes instability or low update frequency of the Information System. The WISDOM production system is designed to achieve production of a large amount of data in a limited time with a minimal human cost using EGEE middleware services. It is able to interact with a license server. Accounting data are collected.

The aim of the next chapter is to present the first achieved deployment of the WISDOM production system against malaria. Issues related to the deployment and the monitoring of the *in silico* docking experiment as well as experience with grid operation and services will be reported.

4.7. References

[428] Buyya, R., et al., Nimrod/G: An architecture for a resource management and scheduling system in a global computational grid, Proceedings of the 4th International Conference on High Performance Computing in Asia–Pacific Region (2000).

[429] Gibbins, H., et al., The Australian BioGrid Portal: Empowering the Molecular Docking Research Community, Proceedings of the 3rd APAC Conference and Exhibition on Advanced Computing, Grid Applications and eResearch (2005).

[430] Zhang, W., et al., Drug Discovery Grid, Proceedings of the UK e-Science All Hands Meeting (2005).

[431] Garcia Aristegui, D.J., et al., GROCK: High-Throughput Docking Using LCG Grid Tools, The 6th IEEE/ACM International Workshop on Grid Computing 85-90 (2005).

[432] Podvinec, M., et al., The SwissBioGrid Project: Objectives, Preliminary Results and Lessons Learned, To be published in 2nd IEEE International Conference on e-Science and Grid Technologies and Applications - Workshop on Production Grids, IEEE Computer Society Press (2006).

[433] http://fightaidsathome.scripps.edu/

[434] Morris, G.M., et al., Automated Docking Using a Lamarckian Genetic Algorithm and Empirical Binding Free Energy Function, J. Computational Chemistry, 19 1639-1662 (1998).

[435] Ziegler, R., Pharma GRIDs: Key to Pharmaceutical Innovation ?, Proceedings of the HealthGrid conference 2004, (2004).

[436] http://aliceinfo.cern.ch

[437] http://atlas.web.cern.ch/Atlas/

[438] http://cms.cern.ch/

[439] http://lhcb.web.cern.ch/lhcb/

[440] Saiz, P., et al., AliEn - ALICE environment on the GRID, Nucl. Instrum. Meth., A502 437-440 (2003).

[441] Legrand, I.C., et al., Monalisa: a distribute monitoring service architecture, Proceedings of Computing in High Enery and Nuclear Physics 2003 (2003).

[442] Aggarwal, M., A Statistical Analysis of Job Performance within LCG Grid, Proceedings of Computing in High Enery and Nuclear Physics 2006 (2006).
[443] Poulard, G., ATLAS Experience on Large Scale Productions on the Grid, Proceedings of Computing in High Enery and Nuclear Physics 2006 (2006).
[444] De Salvo, A., et al., LEXOR, the LCG-2 Executor for the ATLAS DC2 Production System, Proceedings of Computing in High Energy and Nuclear Physics 2004, (2004).
[445] http://www.cs.wisc.edu/condor/condorg/
[446] Harrison, K., et al., Ganga: a Grid user interface for distributed analysis, Fifth UK e-Science All-Hands Meeting, (2006).
[447] Egede, U., et al., GANGA - A GRID User Interface, Proceedings of Computing in High Enery and Nuclear Physics 2006 (2006).
[448] Andreozzi, S., et al., GridICE: a monitoring service for the Grid, 3rd Cracow Grid Workshop, (2003).
[449] http://cmsdoc.cern.ch/cms/ccs/wm/www/Crab/
[450] http://boss.bo.infn.it/
[451] Andreeva, J., et al., Distributed Computing Grid Experiences in CMS, Nuclear Science, IEEE Transactions on 52(4):884-890 (2005).
[452] Paterson, S., and Tsaregorodtsev, A., DIRAC Infrastructure for Distributed Analysis, Proceedings of Computing in High Enery and Nuclear Physics 2006 (2006).
[453] Tsaregorodtsev, A., et al., DIRAC - the LHCb Data Production and Distributed Analysis system, Proceedings of Computing in High Energy and Nuclear Physics, (2006).
[454] Egede, U., et al., Experience with distributed analysis in LHCb, Proceedings of Computing in High Energy and Nuclear Physics, (2006).
[455] Donvito, G., et al, A GRID approach for finding functional analogous gene products, Proceedings of the NETTAB 2006 workshop, (2006).
[456] http://www.biosolveit.de/
[457] http://egee-jra2.web.cern.ch/EGEE-JRA2/QoS/JobsMetrics/JobMetrics.htm

Chapter 5. First large scale deployment against malaria

5.1. Introduction

The first large scale deployment within the framework of WISDOM (World-wide In Silico Docking On Malaria) is motivated by three goals:

- The grid goal is the deployment of a CPU consuming application generating large data flows to test the grid operation and services. Given the very large amount of data involved in the computation, such a large scale deployment is a stressing experiment for the grid infrastructure called a data challenge.
- The bioinformatics goal is the computation of a large scale virtual docking experiment, involving millions of compounds.
- The biological goal is to propose new inhibitors for a family of proteins produced by *Plasmodium falciparum*.

The first large scale docking experiment ran on the EGEE grid production service from 11 July 2005 to 19 August 2005 against targets relevant to research on malaria and saw over 41 million compounds docked for the equivalent of 80 years of CPU time. Up to 1,700 computers were simultaneously used in 15 countries around the world. Resources were made freely available from computing centers and laboratories of institutions in the framework of the EGEE project.

The WISDOM production system described in the previous chapter was used to achieve this production of a large amount of data in a limited time with a minimal human cost using EGEE middleware services. The aim of this chapter is to report issues related to the deployment and the monitoring of the *in silico* docking experiment as well as experience with grid operation and services. Docking experiment preparation and output analysis were leaded by Faunhofer Institute SCAI.

The chapter content is as follows:

- After the introduction, the second section presents in detail the three objectives of the data challenge.
- Then an analysis of the large scale deployment is proposed, followed by a description of achievements in terms of scale.
- Perspectives about the biological results from the deployment, the grid-enabled virtual screening against malaria and a new data challenge against neglected diseases will be presented in the last section.

5.2. Objectives

5.2.1. Grid objective

As discussed in the previous chapter, a large number of applications are already running on grid infrastructures. Even if many have passed the proof of concept level [475], only a few are ready for large-scale production with experimental data. Large Hadron Collider experiments at CERN, like the ATLAS collaboration [394], have been the first to test a large data production system on grid infrastructures [458]. In a similar way, WISDOM aimed at deploying a scalable, CPU consuming application generating large data flows to test the grid infrastructure, operation and services in very stressing conditions.

Docking is, along with BLAST [459] homology searches and some folding algorithms, one of the most prominent applications that has successfully been demonstrated on grid testbeds [460]. It is typically an embarrassingly parallel application, with repetitive and independent calculations. Large resources are needed in order to test a family of targets, a significant amount of possible drug candidates and different virtual screening tools with different parameter and scoring settings. This is both a computational and data challenge problem to distribute millions of docking comparisons with millions of small compound files.

Moreover, docking is the only application for distributed computing that has prompted the uptake of grid technology in the pharmaceutical industry [435]. The WISDOM scientific results are also a means of making a demonstration of the EGEE grid computing infrastructure for the end users community, of illustrating the usefulness of a scientifically targeted Virtual Organization, and of fostering an uptake of grid technologies in this scientific area.

5.2.2. Bioinformatics objective

The bioinformatics objective is to deploy high throughput virtual screening by docking on a public cluster grid. Docking is a first step for *in silico* virtual screening. Basically, protein-compound docking is about computing the binding energy of a protein target to a library of potential drugs using a scoring algorithm. The goal is to identify which molecules could dock on the protein active sites in order to inhibit its action and therefore interfere with the molecular processes essential for the pathogen. Libraries of compound 3D structures are made openly available by chemistry companies which can produce them.

Many docking software are available either open-source or licensed. The docking tools used in this project are FlexX [180], a commercial software made graciously available by BioSolveIT [456] for a limited time, and Autodock [434], a software which is open-source for academic laboratories and which uses a different docking method.

FlexX predicts the geometry of the protein-ligand complex using an incremental construction algorithm and estimates the binding affinity in less than 15 seconds. It offers a fast method for docking conformationally flexible ligands, full specification of the active site,

including oxidation states, metals ions, and side chain protonation states, and automatic ligand positioning.

Autodock carries out quick conformation search of small compounds in the binding sites, fast calculation of binding energies of possible binding poses, prompt selection for the probable binding modes, and precise ranking and filtering for good binders.

Altogether 4 different parameter variations, or assay conditions, were generated for FlexX (location of water molecules, overlap volume) and 2 for Autodock (2 genetic algorithms).

Chemical compounds were obtained in sybyl mol2 format from the ZINC database [461]. ZINC is a free database of commercially-available compounds for virtual screening. It contains over 4.6 million compounds. They are filtered with the Lipinski drug like rules and prepared in ready-to-dock and 3D formats. Docking scores of compounds taken from a subset of the ZINC database, the ChemBridge [462] database (~500,000 compounds) were computed on the above scenarios using FlexX and Autodock. Another drug like subset (~500,000 compounds) of ZINC database was docked on the above scenarios using only FlexX because the computing resources made available to the data challenge were not sufficient to do it also with Autodock. The ChemBridge database molecules were converted from mol2 format to pdbq format in preparation of the Autodock run.

5.2.3. Biological objective

As discussed in chapter 1, malaria is a dreadful disease affecting 300 million people and killing 1.5 million people every year [17]. Malaria is caused by a protozoan parasite, plasmodium. There are several antimalarial drugs presently available. But the constant emergence of resistance and the costs of the present drugs are worsening the disease condition [17]; therefore it is important to keep exploring new strategies to fight malaria. The one investigated within WISDOM aims at the haemoglobin metabolism, which is one of the key metabolic processes for the survival of the parasite.

There are several proteases involved in human haemoglobin degradation inside the food vacuole of the parasite inside the erythrocytes (see figure 2). Plasmepsin, the aspartic protease of Plasmodium, is responsible for the initial cleavage of human haemoglobin and is later followed by other proteases [463], as showed in figure 32 [464].

There are ten different plasmepsins coded by ten different genes in *Plasmodium falciparum* (Plm I, II, IV, V, VI, VII, VIII, IX, X and HAP) [465]. High levels of sequence homology are observed between different plasmepsins (65-70%). Simultaneously they share only 35% sequence homology with their nearest human aspartic protease, Cathepsin D4 [466]. This and the presence of accurate X crystallographic data make plasmepsin an ideal target for rational drug design against malaria.

Though several peptidic and non-peptidic inhibitors have been described as inhibitors for plasmepsins, none of them were effective in killing the parasite in cell culture. This is due to the fact that large size compounds cannot easily penetrate the food vacuole where hemoglobin degradation occurs.

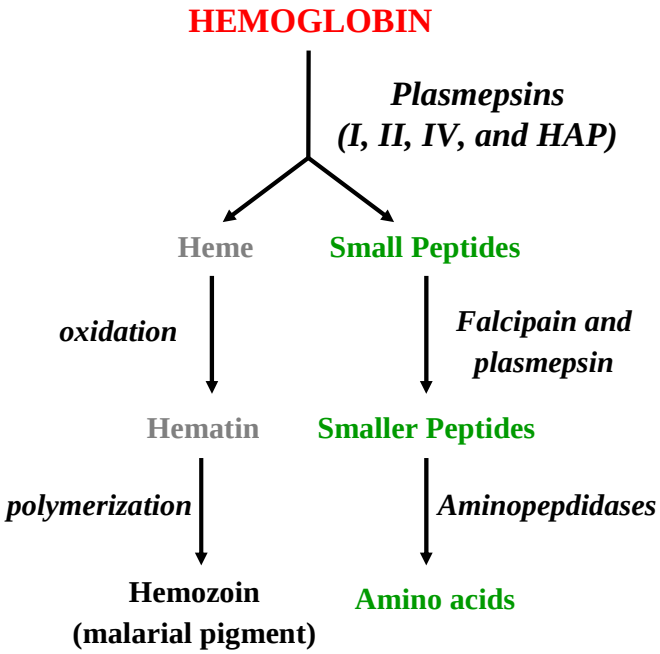


Figure 32: The hemoglobin degradation inside the food vacuole during the erythrocytic phase of the life cycle.

The 3D coordinates of three plasmepsin II structures (1lee, 1lf2, 1lf3) and one plasmepsin IV structure (1ls5) were obtained from the PDB. The protein targets are prepared in a format ready to use for Autodock (pdbqs) and FlexX (mol2). 8 target scenarios for FlexX and 10 target scenarios for Autodock were prepared based on the inclusion of different water molecules in the proteins during the docking process. Table 4 summarizes the experimental setup for the WISDOM deployment.

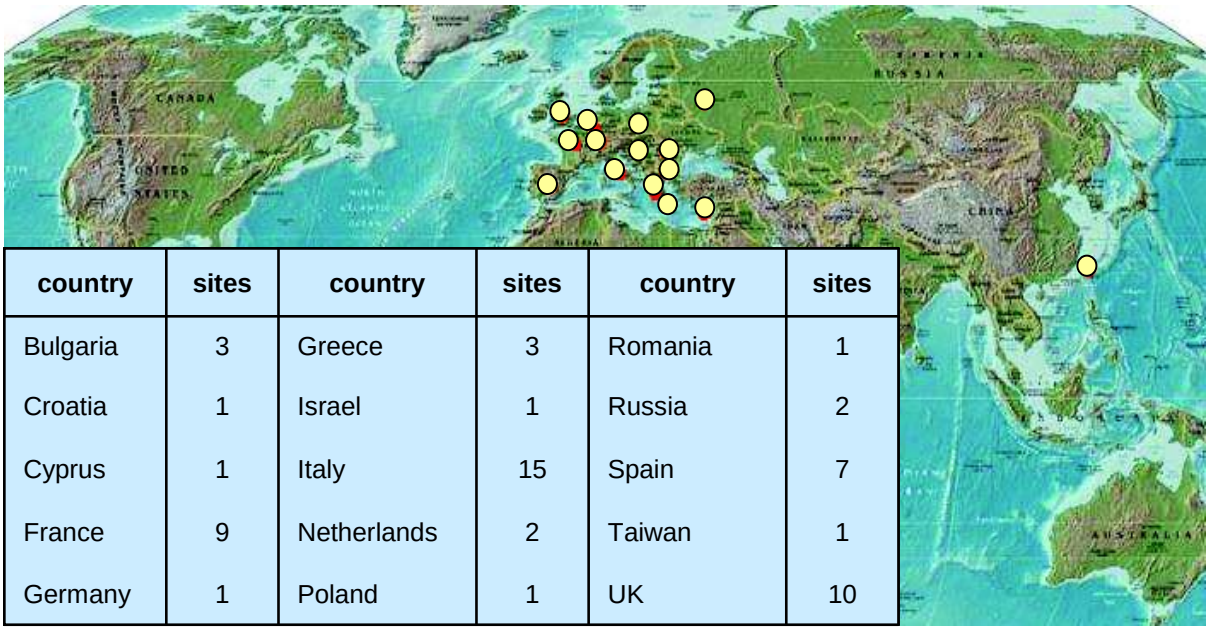
Parameter	Value for FlexX assay	Value for Autodock assay
Number of structures	8	10
Number of assay conditions	4	2
Number of docked compounds	999,810	499,918

Table 4: Experimental setup for the WISDOM data challenge

5.3. The first WISDOM deployment

The deployment took place in July and August 2005. During this period, 10 users launched jobs from 5 User Interfaces, monitored the process with the help of the WISDOM environment and interacted with the user support of the EGEE project and the nodes administrators. 72,751 jobs were launched for a total of 80 CPU years, producing 1 TB of data (500 GB, doubled for the back-up). Figure 33 gives an overview of the 15 countries with 58

grid sites available for the first WISDOM deployment in the biomedical Virtual Organization, which scaled up to about 3,000 CPUs and 21 TB disk space in the summer of 2005.



country	sites	country	sites	country	sites
Bulgaria	3	Greece	3	Romania	1
Croatia	1	Israel	1	Russia	2
Cyprus	1	Italy	15	Spain	7
France	9	Netherlands	2	Taiwan	1
Germany	1	Poland	1	UK	10

Figure 33: Countries with grid sites contributing to the first WISDOM deployment during the summer 2005

In the next section, the deployment is further documented. Firstly, achievements in terms of scale are described, and then the grid performances measured during the deployment are discussed. Finally, the performances of the different grid services are also discussed.

5.3.1. Achieved deployment for the first data challenge

For the sake of simplicity, the WISDOM deployment has been split into 6 phases:

- Phase n°1 corresponds to the high throughput submission of FlexX jobs against the ChemBridge database of compounds. Many problems due to the grid (sites, Resource Broker…) and to the WISDOM production system (load balancing, process management…) were discovered during this early period.
- Phase n°2 corresponds to the resubmission of failed jobs after phase 1
- Phase n°3 corresponds to a second high throughput submission of FlexX jobs with the second drug like compound base. This phase ended on August the 1st, when the number of available FlexX licenses was reduced to 100.
- Phase n°4 corresponds to the resubmission of failed jobs after phase 3
- Phase n°5 corresponds to a first high throughput submission of Autodock jobs with the ChemBridge database of compounds
- Phase n°6 corresponds to the resubmission of failed jobs after phase 5.

Figure 34 shows the number of docked compounds over time during the months of July and August 2005. Figure 35 shows the number of running and waiting jobs vs. time. Figure 36 shows the amount of transferred output vs. time.

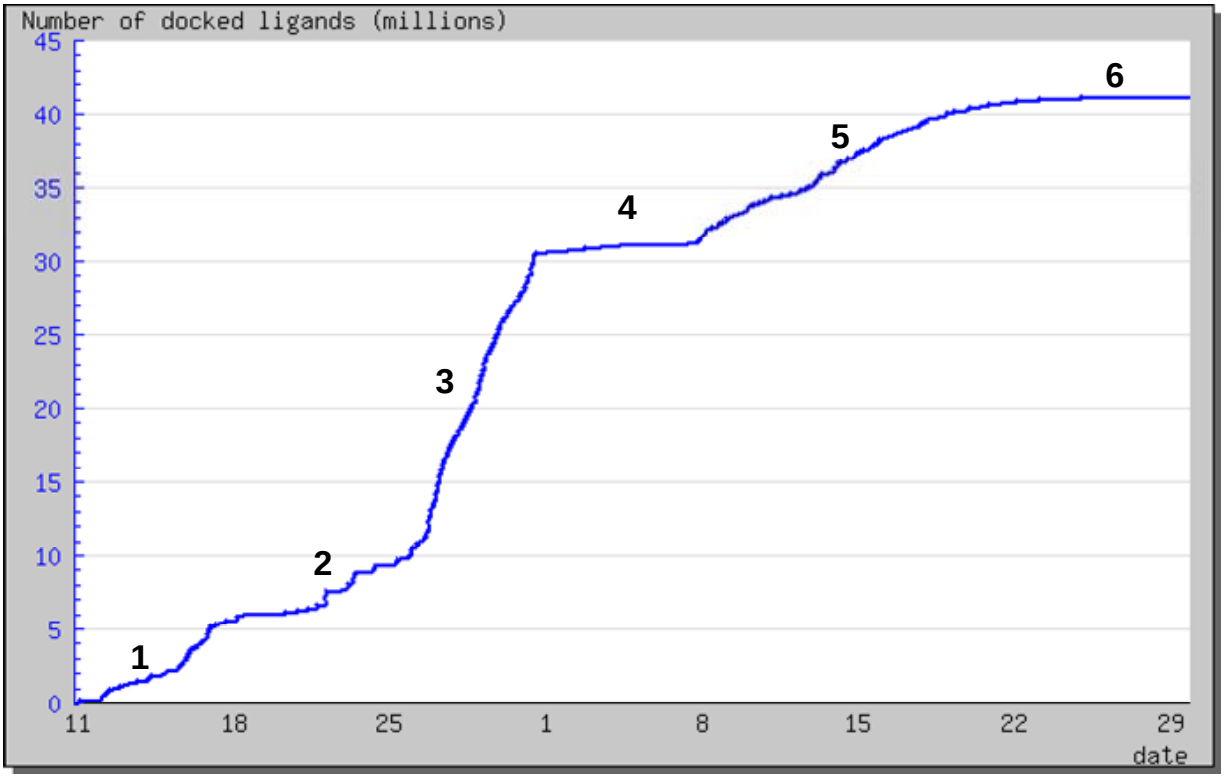


Figure 34: Number of docked compounds vs time.

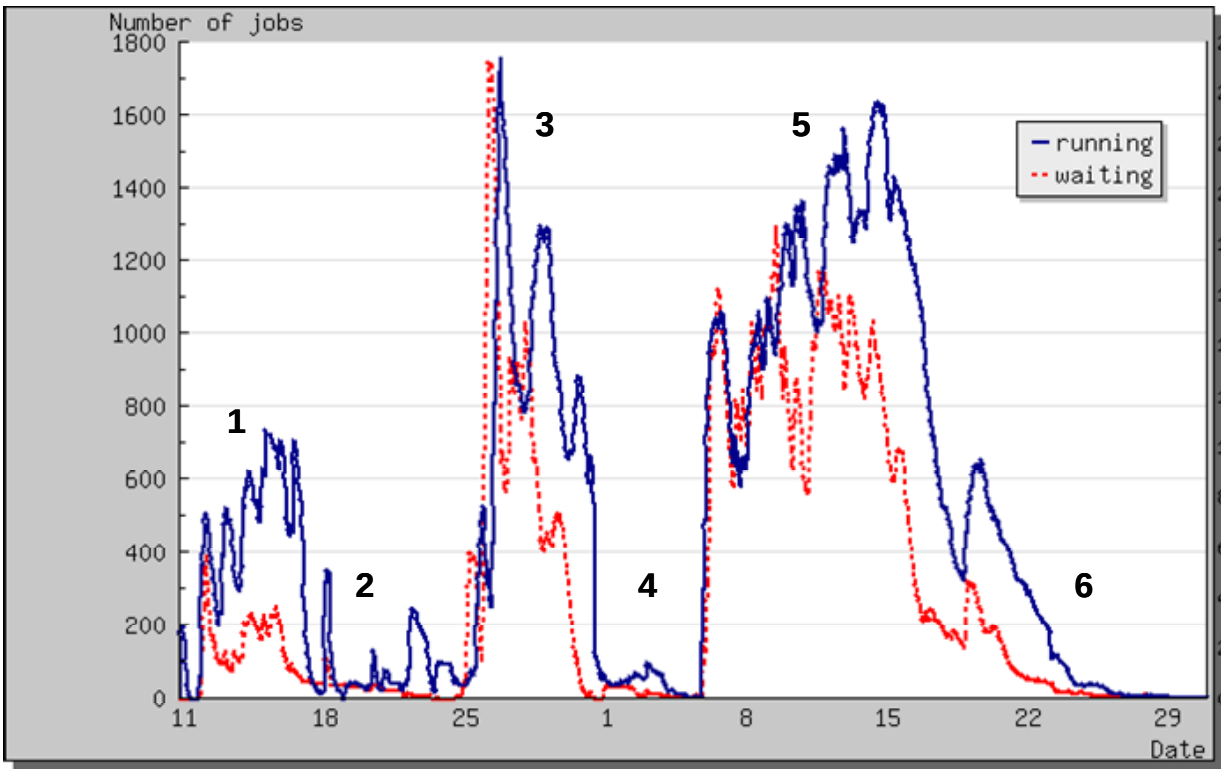


Figure 35: Number of running and waiting jobs vs. time.

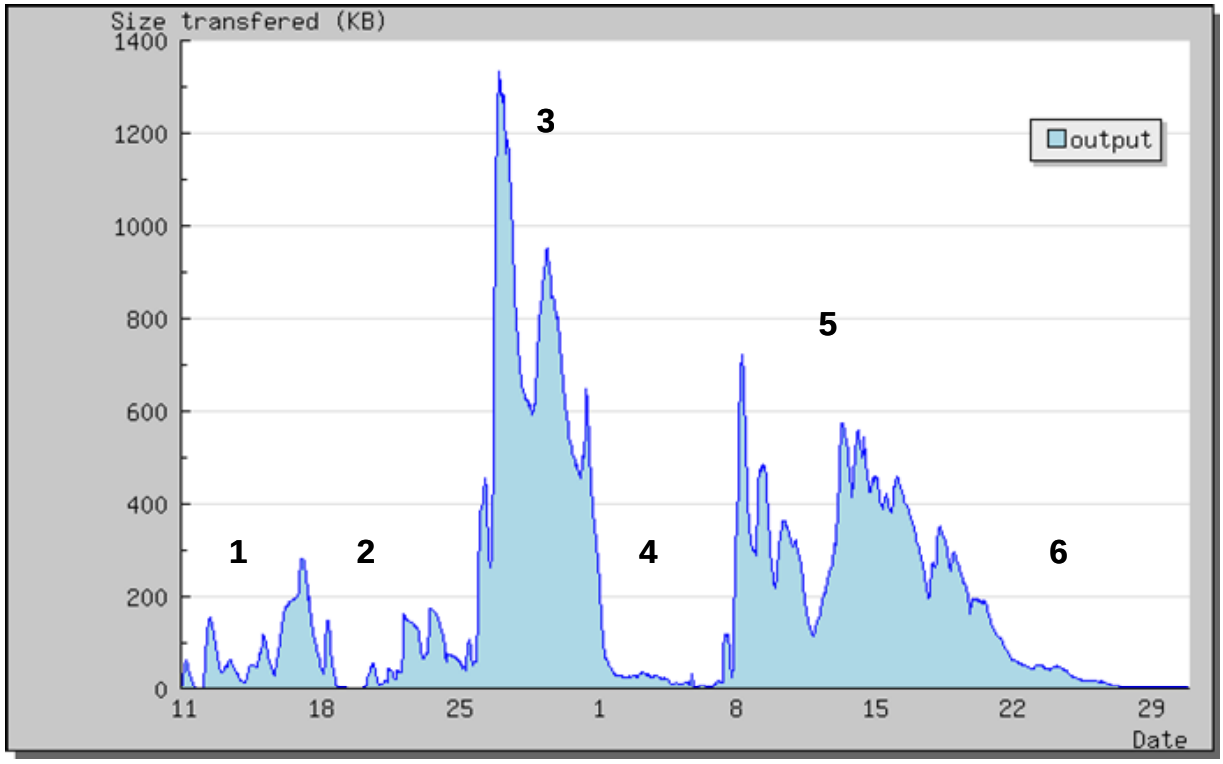


Figure 36: Amount of transferred output vs. time.

The different phases show very different patterns:

- Phases 2, 4 and 6 are resubmission phases where the number of jobs submitted and the amount of data produced are significantly lower than in the high throughput docking phases 1, 3 and 5
- Phase 1 corresponds to a ramp-up phase where many bugs were identified and dealt with
- Data output as well as docking throughput were highest in phase 3 of production with FlexX. Indeed, Autodock software is about 3 times slower than FlexX software.

The shift between the curves of waiting and running time on figure 35 illustrates the latency introduced by the grid. This latency is further documented in the "grid node performances" section but it is worth noticing already that such latency, of the order of a few hours, is acceptable only if the submitted jobs are themselves hour-long jobs.

In what follows, the phases 1 to 4 will also be called the FlexX phase while the last 2 phases will also be called the Autodock phase.

Table 5 presents several parameters relevant in evaluating WISDOM deployment scale for the Autodock and FlexX phases. The following points are worth mentioning:

- The number of docked compounds is a critical parameter for computational chemists. In only 37 days of effective computing, 41.27 million of number compounds were docked. As stated above, FlexX is significantly faster than Autodock and the FlexX phase allowed the computing of three times more compounds than the Autodock one.

- The number of docked compounds per hour is not a factor 3 larger for the FlexX phase compared to the Autodock phase because the number of FlexX licenses was limited to 1,000 while the number of CPU used during the Autodock phase reached more than 1,500 nodes.
- The average crunching factor is 662, and reached 1,031 during the Autodock phase.

Measures	Total	FlexX phase	Autodock phase
Cumulated number of docked compounds (in millions)	41.27	31.41	9.87
Effective duration	37 days	22 days	15 days
Number of docked compounds / hour	46,475	59,488	27,417
Crunching factor	662	411	1,031
Number of jobs submitted	72,751	41,520	31,231
Number of grid Computing Elements used	58	56	57
Number of Resource Brokers used	12	12	11
Maximum number of jobs running in parallel on the grid	1,643	1,008	1,643
Volume of output data	946 GB	506 GB	440 GB
Total CPU time	80 years	29.5 years	50.5 years
Effective CPU time used by successful jobs	67.2 years	24.8 years	42.4 years
Overhead time	77.1 years	25.9 years	51.2 years

Table 5: Performance measures of the WISDOM deployment.

The percentage of CPU time provided by each EGEE computing resource is presented in figure 37. Half of the CPU time was consumed by only 7 of 58 sites. The WISDOM application did not use each site equally (ponderated by the computing capacity of the site), but distributed the tasks to the most important and robust sites.

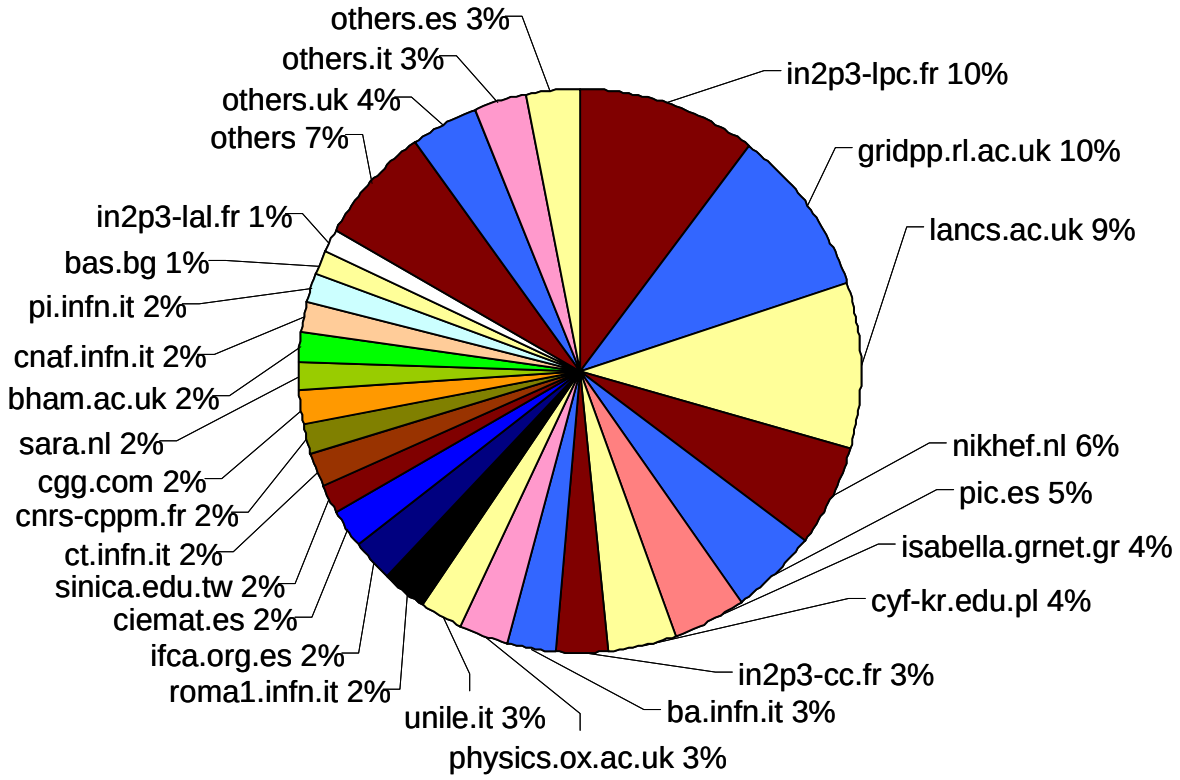


Figure 37: Relative CPU time provided by EGEE computing resources. Some sites have been grouped to improve readability (groups "others")

In the next section the inefficiencies that generated the overhead time are detailed.

5.3.2. Grid node performances

Figure 38 illustrates the different delays introduced by the grid deployment on three among the 58 sites of the biomedical Virtual Organization used for WISDOM. As already explained in chapter 2.5.1, jobs undergo different states when submitted to a grid Computing Element:

- "Submitted" corresponds to jobs submitted by the user through the User Interface and not yet handled by the Resource Broker. It corresponds also to jobs failed and automatically resubmitted by the Resource Broker.
- "Waiting" corresponds to jobs accepted by the Resource Broker but which are not yet allocated to a Computing Element
- "Ready" corresponds to jobs for which the matching resources are found and which are submitted to a Computing Element
- "Scheduled" corresponds to jobs accepted by a Computing Element and which are queuing for execution
- "Running" corresponds to jobs executed on a Worker Node.

Grid node performance depends on several parameters like scheduling policies, Worker Node configuration and system failures. Figure 38 shows the average time spent in the different states by the jobs submitted to the three sites.

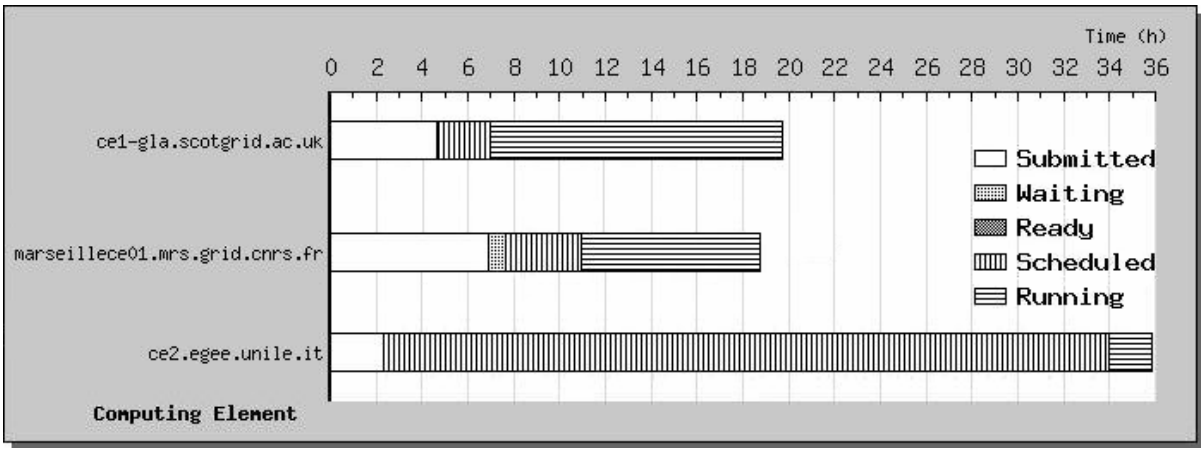


Figure 38: Average time for the different status of a job on three Computing Elements.

Delays introduced by the grid have different explanations for the three sites presented on these figures:

- Scotgrid.ac.uk is a node with about 200 Worker Nodes that did not experience breakdown during the data challenge. The short time spent as "scheduled" shows that this node was not busy with concurrent jobs during the data challenge. The time spent at submission, corresponding to the time needed to access the Resource Broker, is more important.
- The second grid node located in Marseille hosts 30 Worker Nodes. Additional time spent as "submitted" can be attributed to the fact that the node limited the number of scheduled jobs to 30. This limitation was ignored by the Resource Broker and therefore, the extra jobs submitted to this site were discarded immediately. This example shows how a node can have a special configuration not taken into account by the middleware.
- The last node shown on the figure is unile.it with only a few Worker Nodes. Such a small node should not receive a large number of jobs. But a failure of the grid Information System directed almost 100 jobs to the node which increased very significantly the scheduled time.

5.3.3. Analysis of the job success rate

The success rate definition used in this section is also used by EGEE Activity JRA2 in charge of Quality of Service. The formula is as follows:

success rate = successful jobs / (submitted jobs – cancelled jobs)

where successful jobs are jobs which have been executed successfully, submitted jobs are the jobs launched by the user from the User Interface, cancelled jobs are jobs cancelled by the user.

The proposed definition of the success rate is not completely relevant from a user point of view as a successful job as seen from the grid can be unsuccessful from a user perspective if it did not produce the expected output data. Finally, some of the jobs have to be cancelled for reasons external to the grid, for instance failures of the WISDOM execution environment or break downs of the FlexX license server.

Table 6 shows the success rates during the data challenge and more specifically in its two specific phases. An analysis of the origin of failures is summarized in table 7 with their corresponding rates.

Measures	Total	FlexX phase	Autodock phase
EGEE success rate	77 %	80.4 %	71.9 %
Success rate after checking output data	46.2 %	33.8 %	64.4 %
Success rate after checking output data and subtracting WISDOM and server license failures	63 %	61.6 %	65 %

Table 6: Efficiency measures of the WISDOM deployment.

During the FlexX phase, the dominant source of failures was the license server which was distributing tokens to all the jobs running on the grid. This bottleneck can be overcome by having several license servers available.

The second major source of failure was workload management and site failures, including overload, disk failure, node mis-configuration, disk space problems, air-conditioning and power cuts. To improve the job submission process, an automatic resubmission of jobs was included in the WISDOM execution environment. However, the consequence of automatic resubmission was the creation of several “sink-hole” effects where all the jobs are attracted to a single node. These sink-hole effects were observed when the status of a Computing Element was not correctly described in the Information System. If a Computing Element already loaded is still viewed as completely free by the Information System, it keeps receiving jobs from the Resource Broker. If the Computing Element goes down, all jobs are aborted. If the Computing Element can support the excessive number of jobs, the processing time is going to be very long.

Most of the Data Management System failures were circumvented by the back-up system. Finally, unclassified failures account for 4% inefficiency. This illustrates the work which is still needed to improve grid monitoring.

Other large scale deployments on EGEE report similar success rates [458] although the reported causes of job failures are partially different.

	Rate	Reasons
Success rate after checking output data	46 %	
Workload Management failure	10 %	Overload, disk failure
		Mis-configuration, disk space problem
		Air-conditioning, power cut
Data Management failure	4 %	Network / connection
		Power cut
		Other unknown causes
Sites failure	9 %	Mis-configuration, tar command, disk space
		Information system update
		Job number limitation in the waiting queue
		Air-conditioning, power cut
Unclassified	4 %	Lost jobs
		Other unknown causes
Server license failure	23 %	Server failure
		Power cut
		Server stop
WISDOM failure	4 %	Job distribution
		Human error
		Script failure

Table 7: Origin of failures during the WISDOM deployment with their corresponding rates.

5.3.4. Analysis of grid services

The WISDOM data challenge was the opportunity to test the different grid components within the framework of a large scale deployment. Each service has an important role in the process. In this section, the main issues encountered with the different grid services are described.

Information System

To process a job, a Resource Broker chooses the best Computing Element using the information provided by the Information System. The following issues in relation to EGEE Information System were identified:

- a Computing Element can have a policy unknown to the Information System like a limitation on the number of authorized biomedical jobs.
- When the Grid Information Index Service of a Computing Element is down or very slow, it provides obsolete information on its status. This is misleading for the Resource Broker which may keep allocating jobs to this Computing Element although it is saturated.
- The Information System is updated every 2 minutes. This has to be compared to the normal submission time to a Resource Broker which is 7 seconds. As a consequence, when jobs are simultaneously submitted to several Resource Brokers, these services have exactly the same image of the system and therefore distribute jobs in a very similar manner. This generates overload of the best ranked sites.

In summary, our experience shows that detailed information on Computing Element configuration must be known to the Information System in order to manage large scale submission of jobs. Because of the limitations identified, a strategy was adopted not to submit jobs in bursts but rather to have a constant submission flow to limit the impact of missing information and sink-hole effects.

Workload Management System

The Resource Broker allocates jobs to a Computing Element depending on the job attributes described using the Job Description Language. Ranking of the Computing Elements depends on several criteria including the number of free Worker Nodes or the number of scheduled or running jobs.

Resource Brokers turn out to be significant bottlenecks in the system. Several failures of the most used Resource Brokers were observed including disk crashes or overloaded services. As a consequence, hundreds of jobs were lost or retrieved with difficulty. In practice, the measured time for job submission was closer to 30 seconds than 7 seconds. WISDOM users had to change regularly Resource Brokers to limit their workload and instability. They were often left with the task of manually allocating jobs to Computing Elements.

Our experience shows that there is a real need for synchronization between Resource Brokers to be able to send a job through a given Resource Broker and check its status or retrieve results via another one. This means that job databases should not be stored on the Resource Brokers but somewhere on the grid where they can be reached by any Resource Broker and possibly replicated. The idea would be to have a job management similar to the data management, with the notion of a Logical File Name for a job.

On such large scale experiments, bulk submission, grouped monitoring, and partial error mechanisms are critical. On the EGEE infrastructure, bulk submission and grouped monitoring are implemented to some extent, although the application needs to limit the

stressing level on the workload management services. Partial error recovery is still handled at the level of the WISDOM environment.

The Data Management System

The Data Management System certifies data copy, registration and replication. This service is also a potential bottleneck, because there is only one file catalogue per Virtual Organization. As it was the main cause of failure for the previous ATLAS Rome Production Experiment [394], the EGEE Data Management System was significantly improved for the WISDOM data challenge. In case of breakdown of the Data Management System, the back-up solution for data transfer was to use GridFTP retry system for the input copy from a Storage Element to a Computing Element and for the output copy, registration and replication from a Computing Element on two Storage Elements. But failures like a power cut of the Computing Center of Lyon, IN2P3, hosting the unique file catalogue, proved the limitations of the system.

The Data Management System could be improved by replicating the service and file information in several places. An automatic retry after checking the size and the integrity of the copied data could be developed similar to the retry system for job failures.

Resource centers

The resource centers, i.e. the Computing Elements, the Worker Nodes and the Storage Elements, were essential to the success of the data challenge. The 3 Storage Elements storing the output data on disk or tape worked very well despite the concurrent accesses. Difficulties encountered in using the Computing Elements have been discussed previously.

The need for regularly testing all the resources of the biomedical Virtual Organization emerged during the preparation of the large scale deployment. Today, automatic testing is only performed on the EGEE infrastructure for a special test Virtual Organization. These tests allow the automatic extraction failing resources from the production grid.

User Interface

The User Interface is the entry door where the user is authenticated and authorized to submit jobs and manage data. Its disk capacity must be sufficient to receive all the information during and after execution. Preliminary tests are needed on each User Interface to evaluate its capacity to control a large number of jobs with EGEE APIs. Java and Perl are also required.

License server

The license server aims at authorizing the use of commercial software for a large number of jobs. Only a few institutes were not able to open the necessary ports for security reasons. They did not participate in the FlexX part of the data challenge. The server failed a few times because of a limited number of file descriptors on the system which prevented new socket connections. It was necessary to increase this number to allow new jobs to get a license (the number of possible socket connections must be at least equal to the number of available

licenses). In the future, an interesting improvement would be to integrate the server into the grid, so that it can check the certificates of the users requiring a license.

Error messages

When a job is aborted, a status message is sent to help to understand and solve the problem. There is a list of possible reasons for job abortion, but the message is often not clear and not sufficient to know exactly what has happened and what the source of the error is. The help from user support and site administrators is therefore crucial.

5.4. Perspectives

The experiment demonstrated how grid infrastructures have a tremendous capacity to mobilize very large CPU resources for well targeted goals during a significant period of time. But beyond successful large scale deployment, the produced outputs need to be analyzed, post-processed and shared in a collaborative environment. This section summarizes the biological outputs and presents the next steps of the grid-enabled virtual screening against malaria.

5.4.1. Biological results

Post-processing of the huge amount of data generated was a very demanding task as millions of docking scores had to be compared. At the end of the large scale docking deployment, the best 1000 compounds based on scoring were selected thanks to post-processing ranking jobs deployed on the grid. They were inspected individually. Several strategies were employed to reduce the number of false positives. A further 100 compounds were selected for post processing. These compounds had been selected based on the docking score, the binding mode of the compound inside the binding pocket and the interactions of the compounds to key residues of the protein.

There are several scaffolds in the 100 compounds selected for post processing. The scaffolds diphenyl urea, thiourea, and guanidino analogues are most repeatedly identified in the top 1000 compounds. Some of the compounds identified were similar to already known plasmepsin inhibitors, like the diphenyl urea analogues which were already established as micro molar inhibitors for plasmepsins (Walter Reed compounds) [467]. This indicates that the overall approach is sensible and large scale docking on computational grids has potential to identify new inhibitors. Figure 39 represents a diphenyl urea analogue with a good score, well inside the binding pocket of plasmepsin, and interacting with key protein residues. The guanidino analogues can become a novel class of plasmepsin inhibitors.

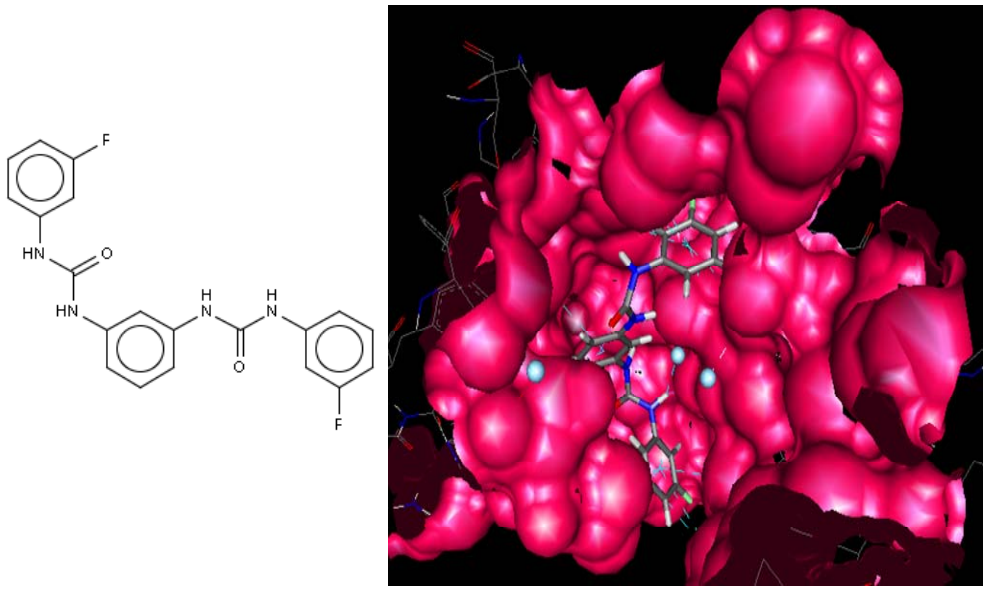


Figure 39: Presentation of a Urea analogue inside the binding pocket of plasmepsin

Before *in vitro* testing, the selected compounds need to be checked using molecular dynamics.

5.4.2. Next step: virtual screening by molecular dynamics

While docking methods have been significantly improved in the last years by including more thorough ligand orientation searches, additional energy contributions and/or refined parameters in the force field, general opinion suggests that docking results need to be post-processed with more accurate modeling tools before biological tests are undertaken. Molecular dynamics has great potential at this stage, as explained in chapter 1. One of the aims is to re-rank molecules based on more accurate scoring functions.

For the same number of compounds, molecular dynamics analysis requires much heavier computing than docking. As a consequence, molecular dynamics can only be applied to a restricted number of compounds, usually the best hits coming out of the docking stage. Molecular Dynamics and subsequent free-energy analysis most often changes significantly the scoring of the best compounds and it is therefore very important to apply it to as many compounds as possible. As a consequence, grids appear very promising in the improvement of the virtual screening process by increasing the number of compounds that can be processed using molecular dynamics.

For instance, running molecular dynamics analysis with Amber software version 8 [468] on one protein target and 10,000 compounds requires about 50 CPU years for a model with a few thousand atoms and a few tens of thousands of steps. This process produces only tens of gigabytes of data. One run with only one compound takes 44 hours (the computing time heavily depends on the choice of conditions, like explicit water simulations, or generalized born simulations).

Both grids of clusters such as EGEE and grids of supercomputers such as DEISA [245] are relevant for molecular dynamics computing. Molecular dynamics computations of large

molecules are significantly faster on a supercomputer. Within the framework of the BioinfoGRID European project [384], focus will be put on the reranking of the best scoring ligands coming out of WISDOM. The goal will be to deploy on at least one of the European grid infrastructures a molecular dynamics software to re-rank the best compounds before *in vitro* testing.

A molecular dynamics software deployment on the EGEE infrastructure to analyze the docking hits against plasmepsin is in progress. At the same time, a second data challenge against neglected diseases is being prepared.

5.4.3. Next data challenge against neglected diseases: WISDOM-II

The impact of the first WISDOM computing challenge has significantly raised the interest of the research community on neglected diseases so that several laboratories all around the world have expressed an interest in proposing targets for a second computing challenge. One of the interests for such deployment is the access to high throughput *in silico* virtual screening thanks to large scale computing resources that majority of life science laboratories can not use, particularly in the research for neglected diseases or in least developed countries. The cost for high throughput experimental screening of millions compounds is estimated to millions euros.

This next WISDOM data challenge called WISDOM-II will take place in autumn 2006. Contacts have been established with research groups from Italy, the United Kingdom, Venezuela and South Africa and a list of targets to be docked against malaria and leishmania has been established.

Also, several grid projects have expressed interest in contributing to the WISDOM initiative by providing computing resources: AuverGrid, South East Asia Grid [388], South America grid EELA [382], SwissBiogrid [432]) or by contributing to the development of the virtual screening pipeline (Embrace [469], BioinfoGRID). A rough estimation of the needed resources is about 500 CPU years and about 4 terabytes storage

5.5. Conclusion

This fifth chapter has described how this application allowed the testing of the grid operation and services for a CPU consuming application generating large data flows.

The WISDOM data challenge is the first large deployment of a biomedical application on a grid infrastructure. From 11 July 2005 until 19 August 2005, up to 1,700 computers were simultaneously used in 15 countries around the world to dock over 41 million compounds. On the biological side, the aim of the application was to identify new inhibitors for a family of proteins produced by *Plasmodium falciparum* through *in silico* virtual docking on a grid infrastructure.

The data challenge has been a very useful experience in identifying the limitations and bottlenecks of the EGEE infrastructure. The WISDOM production system developed to

submit the jobs on the grid accounted for a small fraction of the failures, along with the grid management system. On the other hand, the resource brokers significantly limited the rate at which the jobs could be submitted. Another significant source of inefficiency came from the difficulty for the grid Information System to provide all the relevant information to the resource brokers for the distribution of the jobs on the grid. As a consequence, job scheduling was a time-consuming task for the WISDOM users during the data challenge due to the encountered limitations of the Information System, the Computing Elements and the resource brokers. Finally, the necessity to deploy licensed software during one of the deployment phases has generated a single point of failure ignored by the Information System. The development of a grid component to manage license software is under way in the EGEE project to address this limitation.

The experiment demonstrated how grid infrastructures have a tremendous capacity to mobilize very large CPU resources for well targeted goals during a significant period of time. As a consequence of this challenge, middleware developers of the EGEE project are incorporating features required by the biomedical community.

On the bioinformatics side, the WISDOM data challenge has demonstrated that collaborative production grids can be used for steps in the drug discovery process. Grid-enabled high throughput structure-based virtual screening by docking is the first step to enable the virtual screening pipeline on a grid environment. The next step is the deployment of molecular dynamics simulations in grid environments. A second data challenge against neglected diseases, called WISDOM-II, is being prepared. A rough estimation of the required resources is about 500 CPU years and about 4 terabytes storage.

On the biological side, the data challenge produced a large amount of output for analysis. Results extracted 10% of the compounds with key interactions and good scoring. Top scoring compounds possess basic chemical groups like thiourea, guanidino or diphenyl urea core structure. These compounds are currently under further analysis thanks to grid-enabled molecular dynamics. This large docking analysis is the initial proof of the concept: biologists need to come up with challenges that exploit the resources now available. The cost of the discovery of new inhibitors against neglected diseases would thus be lower.

The final development of drug candidates could be awarded to a laboratory based on competitive bids. The drug itself would go into the public domain, for generic manufacturers to produce. This would achieve the goal of getting new medicines to those who need them at the lowest possible price. This model is currently supported by various United Nations programs and involves several commercial organizations, including large pharmaceutical companies.

The first large scale docking experiment focused on virtual screening for neglected diseases but new perspectives appear also for using grids to address emerging infectious diseases. The next chapter will present a new data challenge dedicated to *in silico* docking against avian influenza deployed on the EGEE, AuverGrid and TWGrid infrastructures during the spring of 2006.

5.6. References

[458] Bird, I., et al. Operating the LCG and EGEE production Grids for HEP, Proceedings of Computing in High Enery and Nuclear Physics 2004, (2004).
[459] Altschul, S.F., et al., Basic local alignment search tool, J. Mol. Biol., 215 403-410 (1990).
[460] Chien, A., et al., Grid technologies empowering drug discovery, Drug Discovery Today 7(20) 176-180 (2002).
[461] Irwin, J.J. and Shoichet, B.K., ZINC-a free database of commercially available compounds for virtual screening, J. Chem. Inf. Model. 45(1):177-82 (2005).
[462] http://chembridge.com/
[463] Francis, S. E., et al., Hemoglobin metabolism in the malaria parasite plasmodium falciparum. Annu.Rev. Microbiol. 51, 97-123 (1997)
[464] Gutierrez-de-Teran, H., et al., Inhibitor Binding to the Plasmepsin IV Aspartic Protease from Plasmodium falciparum, Biochemistry 45(35):10529-10541 (2006).
[465] Coombs, G.H., et al., Aspartic proteases of plasmodium falciparum and other protozoa as drug targets, Trends parasitol. 17 532-537 (2001).
[466] Silva, A.M., et al., Structure and inhibition of plasmepsin II, A haemoglobin degrading enzyme from Plasmodium falciparum, Proc. Natl. Acad. Sci. USA 93, 10034-10039 (1996).
[467] Jiang, S., et al., New Class of Small Nonpeptidyl Compounds Blocks Plasmodium falciparum Development In Vitro by Inhibiting Plasmepsins, Antimicrobial agents and Chemotherapy 45 2577-2584 (2001).
[468] Case, D.A., et al., The Amber biomolecular simulation programs, J. Computat. Chem. 26 1668-1688 (2005).
[469] http://www.embracegrid.info/

Chapter 6. First large scale deployment against avian influenza

6.1. Introduction

Large scale grids for *in silico* drug discovery open opportunities of particular interest to neglected and emerging infectious diseases. The first large scale docking experiment described in chapter 5 focused on virtual screening for neglected diseases, but new perspectives appear also for using grids to address emerging infectious diseases. While grid added value for neglected diseases is related to their cost effectiveness as compared to *in vitro* testing, grids are also extremely relevant when time becomes a critical factor.

In April and May 2006, the second biomedical data challenge of the EGEE project was kicked off to tackle the computational challenge of screening about 300,000 compounds against the avian influenza virus H5N1. The goal was to find potential compounds that can inhibit the activities of an enzyme on the surface of the influenza virus, the so-called neuraminidase, subtype N1. Using the grid to identify the most promising leads for biological tests could speed up the development process for drugs against the influenza virus.

This new deployment required a much shorter preparation, about a month, and took advantage of the experience acquired with WISDOM. In order to compress the overhead so that biomedical chemists could have the best responses to instant threads while the mutation of the virus happened, more than 1600 CPUs in the EGEE, AuverGrid and TWGrid grid infrastructures were mobilized to perform large scale distributed virtual screening during 6 weeks. About 750 Gigabytes of output data were produced and archived on the grid with one additional backup. In addition to re-exercising the same framework used for the first WISDOM deployment, a light-weight framework, DIANE, also took part in this second large scale deployment. The purpose was to investigate the performances of DIANE's special scheduling and failure recovery mechanisms under the pressure of handling jobs on a large scale.

The aim of this chapter is to report the effects of the deployment and the monitoring of improvements of the WISDOM production system concerning deployment stability and to compare it with the DIANE framework, another production system framework. Docking experiment preparation and output analysis were leaded by Genomics Research Center from Academia Sinica of Taiwan and Institute for Biomedical Technologies from CNR, Italy.

The chapter content is as follows:

- After the introduction, the second section presents in detail the three objectives of the data challenge.

- The second section describes deployment strategy improvements for the WISDOM production system and presents the DIANE framework.
- Then an analysis of the large scale deployment is proposed, followed by a description of achievements in term of scale, focused on the comparisons of the WISDOM and DIANE production systems.
- Perspectives about the biological results from the deployment and a grid-enabled virtual screening service will be presented in the last section.

6.2. Objectives of the deployments

6.2.1. Grid objective

There were two grid technology objectives for this activity: one was to improve the performance of the *in silico* high throughput screening environment based on what was learnt in the previous challenge against malaria; the other was to test another environment which enables users to have efficient and interactive control of the massive molecular dockings on the grid.

Therefore, two grid tools were used in parallel in the second data challenge. An enhanced version of WISDOM high throughput workflow was designed to achieve the first goal and a light-weight framework called DIANE [470,471] was introduced to carry a significant portion of the deployment for implementing and testing the new scenario. Grid performance measures will be compared between the first and the second data challenge.

Compared to the first WISDOM deployment which used only the EGEE infrastructure, the avian influenza computing challenge used three infrastructures which are share the same middleware (LCG2) and also common services: AuverGrid, EGEE and TWGrid.

6.2.2. Bioinformatics objective

The first WISDOM data challenge demonstrated the large scale deployment of virtual screening by docking. The bioinformatics objectives of this second deployment were to validate such a successful acceleration of the discovery of novel potent inhibitors and to improve the efficiency of high throughput structure-based screening.

The docking tool used in this project was Autodock [434]. Only one parameter set was defined, but it had a strong impact on the computing time. Using AutoDock to evaluate one compound structure for 50 poses within the target enzyme would take 300 kilobytes storage and 30 minutes on an average PC.

200,000 compounds from the ZINC database [461] and 100,000 compounds from a chemical combinatorial library were selected. Filters used were the presence of a benzyl, a six or five member-ring as the compound core, the presence of at least one acid group, and drug-like consideration. These criteria were defined according to the natural inhibitor and known drugs of the target. For instance, in figure 40 presented below, the zanamivir and oseltamivir core is composed of a benzyl.

6.2.3. Biological objective

The potential for reemergence of influenza pandemics has been a great threat since the report that avian influenza A virus (H5N1) could acquire the ability to be transmitted to humans. Indeed, an increase of transmission incidents suggests the risk of human-to-human transmission. Moreover, the report of development of drug resistance variants is another potential concern.

Two of the present drugs, oseltamivir and zanamivir, were discovered through structure-based drug design targeting influenza neuraminidase (NA). Viral neuraminidase is an enzyme that helps the release of new virions by cleaving human host receptors (terminal sialic acid residues from glycoconjugates), whose action is essential for virus proliferation and infectivity (see figure 5). Therefore, blocking its activity generates antivirus effects. Figure 40 presents the neuraminidase binding pocket and the structures of oseltamivir and zanamivir.

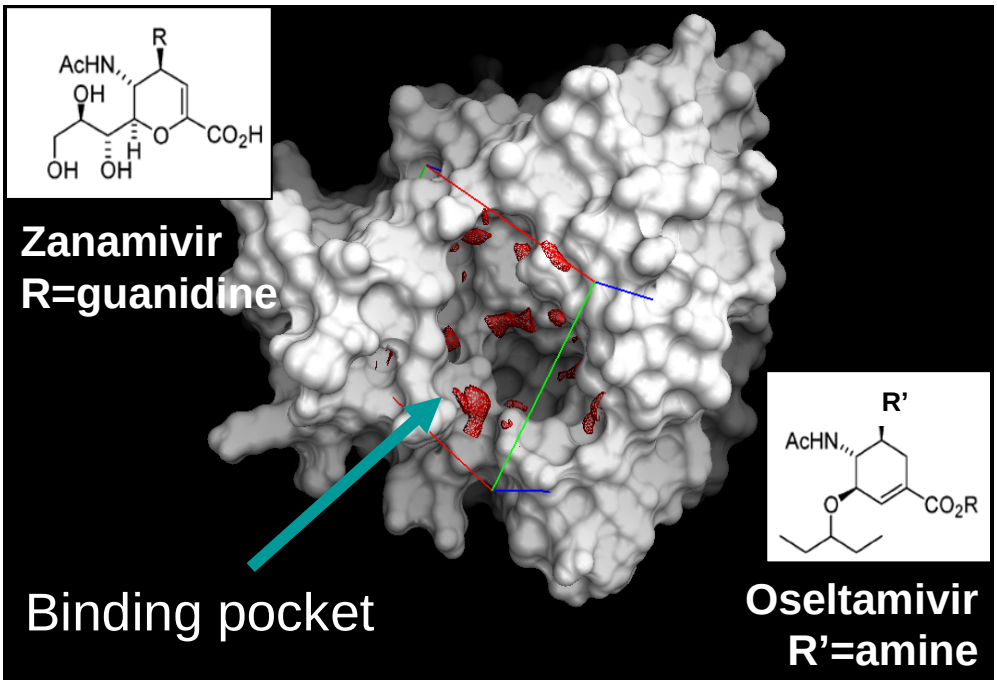


Figure 40: Neuraminidase binding pocket and structures of oseltamivir and zanamivir

The development of drug resistance variants is one of the potential concerns of influenza pandemics. Moreover, to date, there is no NA subtype one (the N1 of H5N1 virus) available for structural study. To minimize non-productive trial-and-error approaches and to accelerate the discovery of novel potent inhibitors, medical chemists take advantage of modeled NA variant structures and structure-based design.

The biological goal of the grid deployment was to evaluate the impact of mutations on the efficiency of the existing drug and to find potential compounds that can inhibit the activities of influenza A neuraminidase N1 subtype variants. The idea was to compile the results from *in silico* screening to know the kinds of compounds and chemical groups (fragments) equipped for blocking the active neuraminidases if mutations are to occur at some specific

sites. 4 amino acids are expected from literature and the analysis within the range of binding pocket to possibly mutate. 8 structures from the same target protein (from PDB) were prepared by considering each with one amino acid mutated only. For validation, these preparations were compared with crystal structures of other subtypes, except N1, available in PDB. Figure 41 shows reported and predicted mutation sites in the binding pocket of N1. One example is E119, a glutamic acid at the position 119 in the amino-acid sequence. It is reported in [60,61], that the substitution of the residue 119 can affect the efficiency of oseltamivir.

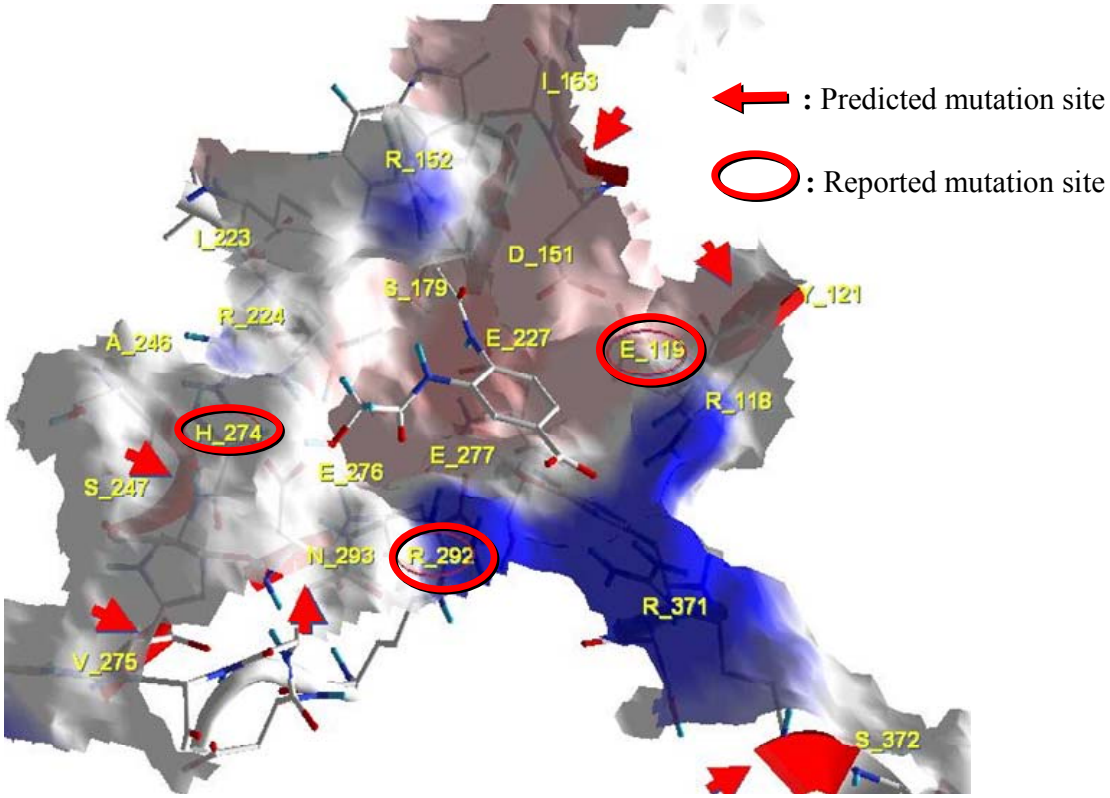


Figure 41: Reported and predicted mutation sites in the binding pocket of the neuraminidase N1 subtype

The grid helps to reduce the time and the cost of the initial investment on structure-based drug design. Table 8 summarizes the experimental setup for the avian influenza data challenge.

Parameter	Value for Autodock assay
Number of structures	8
Number of assay conditions	1
Number of docked compounds	308,585

Table 8: Experimental setup for the avian influenza data challenge

The next section details the two production systems taking advantage of the EGEE, AuverGrid and TWGrid infrastructures.

6.3. Docking production environments on EGEE, AuverGrid and TWGrid

Taking advantage of the experience acquired in the previous WISDOM data challenge on malaria, the grid-enabled *in silico* process was implemented in less than a month on three different grid infrastructures: AuverGrid, EGEE, and TWGrid, paving the way for a virtual drug screening service at a large scale. The majority of computing was conducted on the WISDOM platform; in addition, a light-weight application framework called DIANE was also adopted in this challenge and used to perform a sizeable portion of the total activity to enable efficient computing resource integration and usage.

6.3.1. WISDOM execution improvement

The environment was improved to address limitations and bottlenecks identified during the first data challenge against malaria deployed in the summer of 2005 on the EGEE infrastructure. The main requirement was the improvement of the stability to increase the success rate, even if the grid performance could be lower.

- The number of resource broker machines and the rate at which the jobs were submitted on them was extended to avoid or limit their overloading. Consequently, the time to saturate the grid resources would probably be longer, but the constant submission flow would limit the impact of missing information and sink-hole effects.
- The resubmission process after a job failure was redesigned to avoid a “sink-hole” effect on a failing grid computing node. Automatic resubmission was replaced by the manual intervention of the WISDOM production user. A perl script module was developed to automate the resubmission.
- During the first data challenge, docking process information produced during the job, such as process time, output size or failure messages, retrieved from the User Interface thank to the job Output Sandbox could be lost due to Resource Brokers crashes. Such an output file was now directly stored and registered on two Storage Elements and downloaded on the User Interface at the end of the job.
- An instance is a job set submitted and monitored on the grid by the production system. The different jobs of a given instance have the same target input and docking software. They only differ by the molecules of the compound library which are docked. The number of instances was reduced for the avian influenza data challenge in order to decrease the needed manpower time to manage them and consequently the human failure rate.

As Autodock is an open-source software, there is no license server.

6.3.2. DIANE

DIANE [470,471] is a lightweight distributed framework for parallel scientific applications in a master-worker model. It assumes that a job may be split into a number of independent tasks which is a typical case in many scientific applications. It has been applied in a number of applications ranging from image rendering to data analysis in high-energy physics.

As opposed to standard message passing libraries such as MPI [472], the DIANE framework takes care of all synchronization, communication and workflow management details on behalf of the application. The execution of a job is fully controlled by the framework which decides when and where the tasks are executed. Thus the application is very simple to program and contains only the essential code directly related to the application itself without the need for networking details.

Aiming to efficiently link underlying distributed computing environments and application centric User Interface as illustrated in figure 42, DIANE itself is a thin software layer which can easily work on top of more fundamental middleware such as LSF, PBS or the grid Resource Brokers. It may also work in a standalone mode and does not require any complex underlying software. DIANE hides the scheduling details of application distribution.

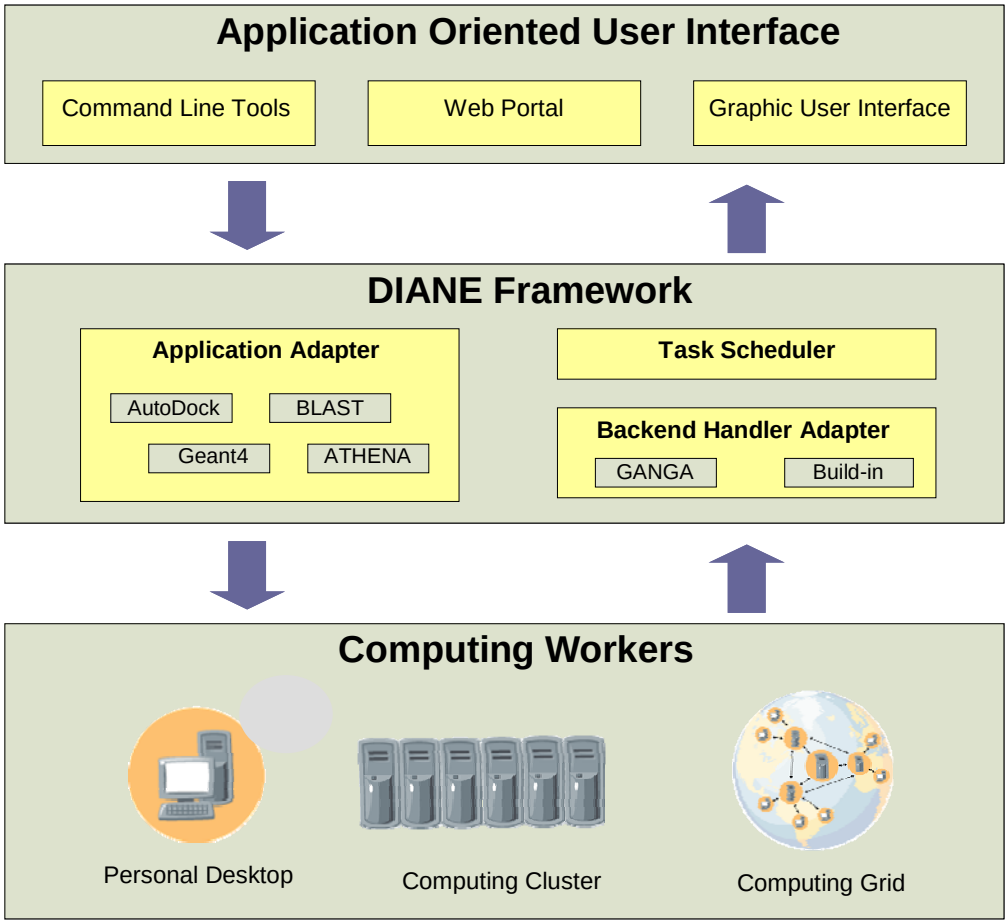


Figure 42: The DIANE framework

As a framework, DIANE provides an adapter for applications. Figure 43 shows the template of DIANE application plug-ins. A complete DIANE application plug-in should implement three major Python objects: the Planner and the Integrator objects implement the job splitting and result merging, respectively; while the logic of the Worker object concentrates on the execution of the individual task. When a DIANE job is started by a user, both the Planner and the Integrator objects are invoked by a master agent usually executed on the user's desktop, and typically the worker agents are submitted to run on distributed CPUs such as the grid worker nodes.

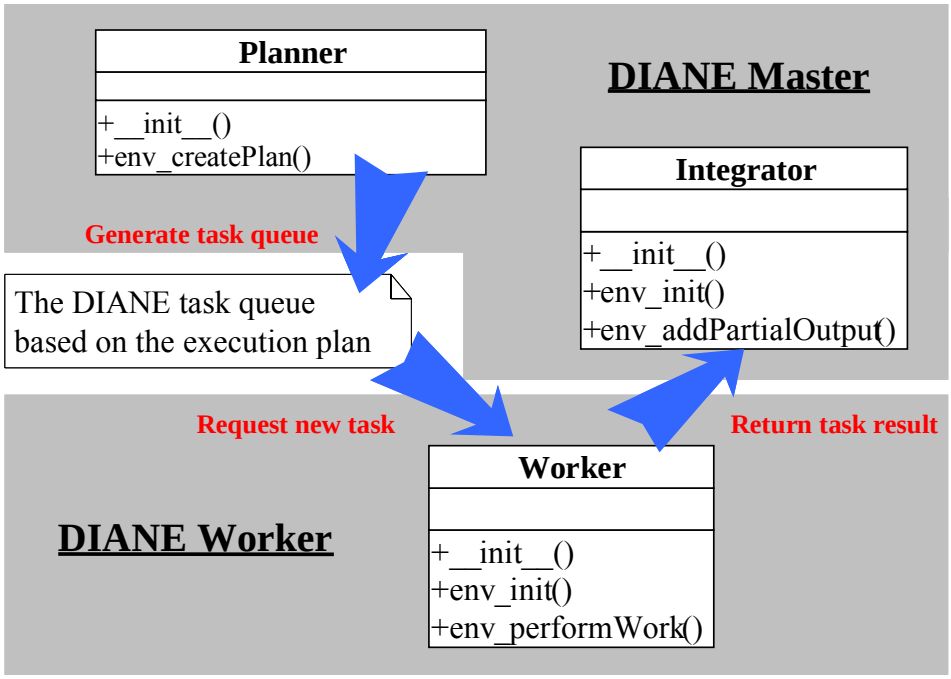


Figure 43: The template of DIANE application plug-ins as well as the cooperation model between the three major objects: Planner, Worker and Integrator.

The deployment process described in this paragraph is based on the EGEE middleware. Once the worker agent is launched from a User Interface on a Worker Node of a Computing Element thanks to a Resource Broker, first it registers itself with the master agent. In the second step, a channel is established for extracting the tasks from the queue held by the master agent. When the individual task is completed by the worker agent, the result is returned and merged on the master. The extracting-executing- returning cycle will continue until all the tasks are completed. A DIANE job is thus longer than a WISDOM job, in which the execution time is calculated to be inferior to 20 hours. The same channel is also used to profile the worker agent's health and to support user interaction with the task. The whole DIANE framework is written in Python and the communication between the master agent and the worker agents is based on the CORBA protocol. The Data Management System of EGEE is thus used only to store the concatenated output files from the User Interface.

Since the DIANE framework takes care of the control of the communication and the workflow on behalf of the application, implementing an AutoDock adaptor for DIANE costs approximately 3 days and the effort is less than 500 lines of Python codes.

6.3.3. Preparation of the deployment for the DIANE and WISDOM production systems

The WISDOM production system is a workflow of grid job handling: automated job submission, status check and report, error recovery. It is a push model job scheduling and a batch mode job handling. The DIANE framework has a master-worker model. It is a pull mode job scheduling and an interactive mode job handling with flexible failure recovery features.

Based on what was learnt in the previous data challenge, the deployment work including the prediction of the N1 variants took about 1 month. Before data challenge kick-off, the compounds were pre-staged on three Storage Elements, and the Autodock executable was widely deployed on most of the available Computing Elements. The centralized LCG File Catalog system was used to index all the files distributed on the grid. DIANE's workflow and WISDOM production system's workflow were synchronized for the policy of job taking and result archiving, the job execution statistics and the job submission agenda.

Input for the avian influenza data challenge consists of 8 protein targets predicted from the N1 to simulate the possible mutations of the H5N1 virus and 308,585 chemical compounds selected from ZINC and a chemical combinatorial library. By dividing the 308,585 chemical compounds into 2 subsets, the whole data challenge activity was broken down into 16 instances; each instance corresponded to the dockings of an N1 variant against the compounds in one of the 2 subsets. To avoid the concurrent executions of all the instances overloading the grid system and reducing the grid efficiency, the initialization time of each instance was well scheduled.

The majority of the data challenge instances were executed using the WISDOM production environment since its scalability had been demonstrated already in the first data challenge. Taking into account the approximation that the computing time of each single docking is about 30 minutes (The measurement was done on a PC with one Xeon 2.8 GHz CPU and 2 Gigabytes physical memory), each WISDOM job was prepared to run on 40 dockings. Thus each instance represented 7715 grid jobs of about 20 hours long. Two instances were deployed using the AuverGrid infrastructure with the AuverGrid Virtual Organization.

In parallel with the WISDOM activity, DIANE was used to run as many dockings as it could handle during the data challenge activity. As the estimated elapsed time of each docking is significantly longer than the startup overhead of the task, each DIANE task was defined to correspond to the docking of one compound. As a master-worker model, DIANE submitted worker agents instead of docking tasks to the grid. Due to the fact that the CPU wall time of most of the grid Computing Elements is restricted to 24 hours, more worker agents need to be submitted once the limitation is reached. During the data challenge, a DIANE master was

maintained on the User Interface to hold a queue of the waiting docking jobs and a separate process for submitting DIANE worker agents was manually triggered. This strategy allowed the use of more CPU power to ramp up the docking throughput without interfering with the running master. The result of each docking was interactively returned back to the User Interface once the task was successfully completed. All the results were also concatenated and archived into the grid. DIANE jobs were submitted on the EGEE and TWGrid infrastructures at the same time.

6.4. Second large scale deployment

The deployment took place in April and May 2006. For 6 weeks, 9 users launched jobs from 7 User Interfaces, monitoring the process with the help of the WISDOM and DIANE environments and interacting with the user support of the EGEE project and the nodes administrators. 74,850 jobs were launched for a total of 105 CPU years, producing 752 GB of data (376 GB, doubled for the back-up). 69 grid sites from 22 countries were available for this second large scale deployment in the biomedical Virtual Organization. Figure 44 shows the distribution of jobs on the AuverGrid (7%), TWGrid (4%) and EGEE (89%) infrastructures.

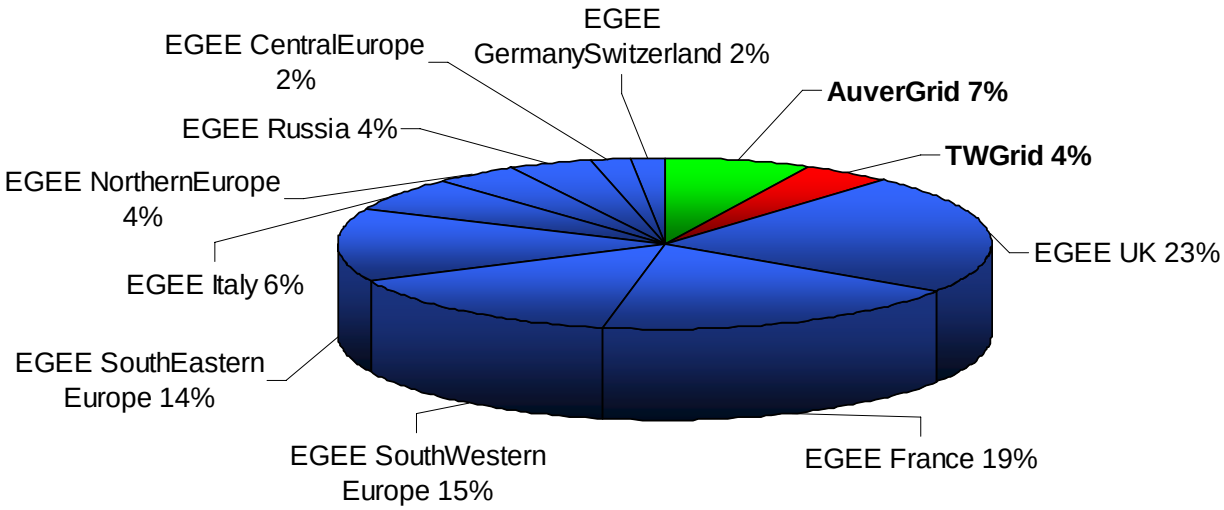


Figure 44: Job distribution rate for AuverGrid, TWGrid and EGEE infrastructures

In the next section, the deployment is further documented. Firstly, achievements in terms of scale are described, and then the grid performances measured during the deployment are discussed. Finally, the performances of the different grid services are also discussed.

6.4.1. Achieved deployment for the second data challenge

For a sake of simplicity, the WISDOM deployment has been split into 4 phases:

- Phase n°1' corresponds to a first submission of Autodock jobs. The first problems due to the grid (sites) and to WISDOM users (an user certificate change, User Interface overload) were discovered during this early period. Moreover, the job submission

frequency was slower to control the grid stability (2 minutes between 2 submissions, then 1 minute).

- Phase n°2' corresponds to an intensive job submission phase. The job submission frequency was increased (30 seconds between 2 submissions). No failures were reported
- Phase n°3' corresponds to a limited job submission phase due to grid problems (sites, Resource Broker).
- Phase n°4' corresponds to the last resubmissions of failed jobs.

Figure 45 shows the number of docked compounds over time during the months of April and May 2006. Figure 46 shows the number of running and waiting jobs vs. time.

The different phases show different patterns:

- Phase 4' is a resubmission phase where the number of jobs submitted is significantly lower than in the high throughput docking phases 1', 2' and 3'.
- Phase 1' corresponds to a ramp-up phase where many bugs were identified and resolved
- Docking throughput was highest in phase 2' of production.

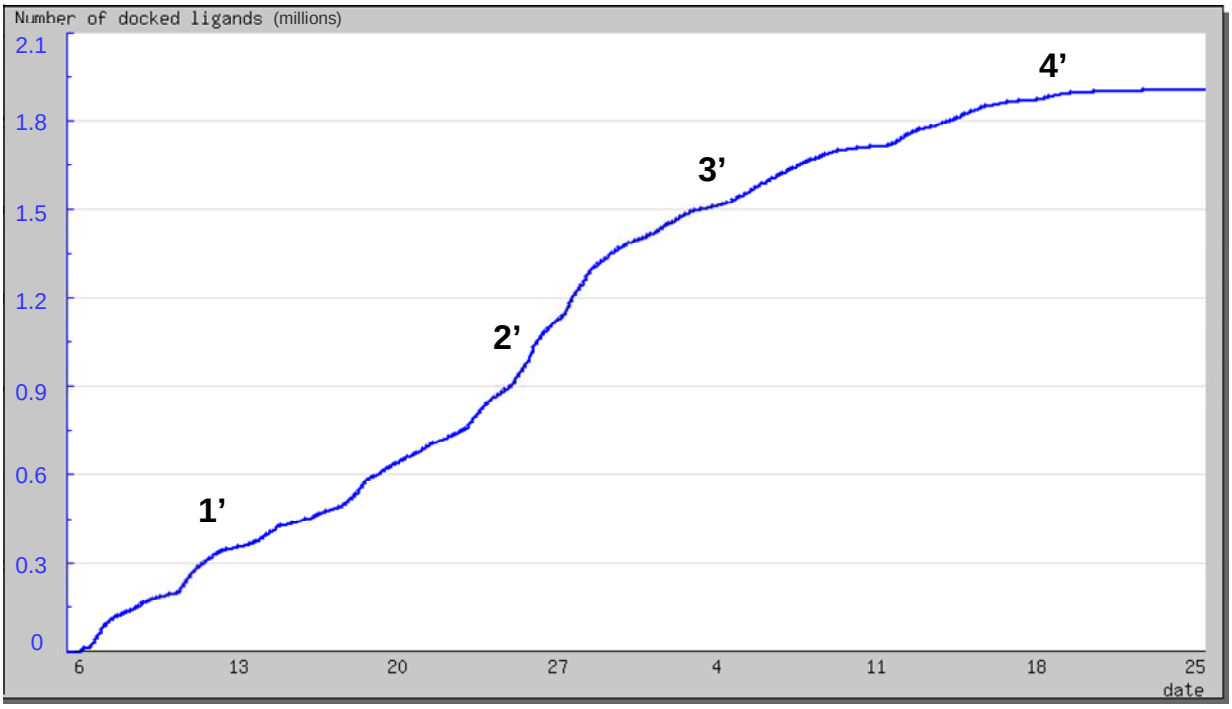


Figure 45: Number of docked compounds vs time.

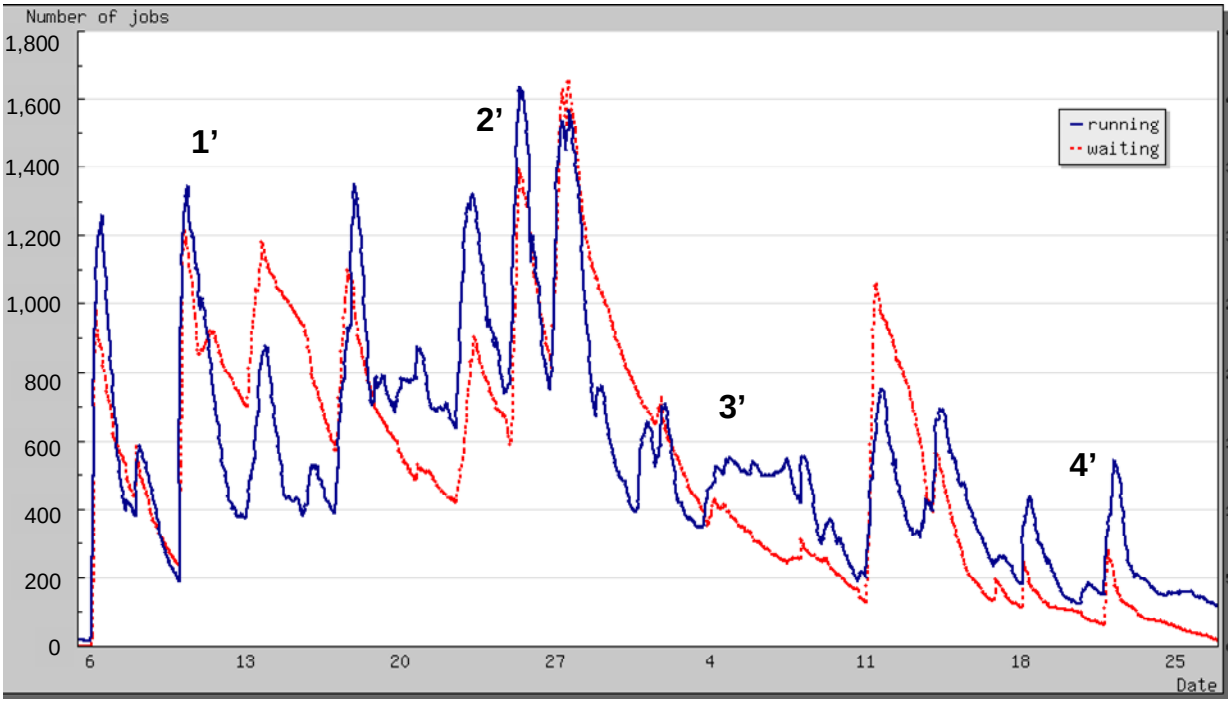


Figure 46: Number of running and waiting jobs vs. time.

A comparison of the achieved deployment phases between the first and the second large scale deployment shows differences:

- The number of docked compounds increases more regularly than during the first WISDOM deployment. This more regular production demonstrates the lower number of failures.
- The instances were submitted continually without resubmission interruption. The single resubmission phase is thus more important than the three resubmission phases of the first WISDOM deployment.
- The gradient degree of phases 1' and 3' are similar to the gradient degree of phase 5 of the first deployment, the Autodock production phase, whereas phase 2' is the highest. The production efficiency was thus better during the second deployment.
- The gradient degree of phase 3 of the first deployment, the best FlexX production phase, is significantly higher than phase 2'. FlexX is faster than Autodock; its docked compound production phase is thus more efficient.

Table 9 presents several parameters relevant to evaluate WISDOM and DIANE deployment scale. The following points are worth mentioning:

- The number of docked compounds of the second deployment, 2.47 million, is lower than the number of docked compounds during the first deployment. As stated above, FlexX is significantly faster than Autodock. Moreover, Autodock parameters for the second deployment increased by a factor of 10 the execution time of Autodock jobs by comparison to the execution time with the Autodock parameters of the first deployment.

- The average crunching factor for WISDOM deployment is 767, whereas the average crunching factor for DIANE is 203. This can be explained by the lower maximum number of jobs running in parallel on the grid for DIANE: 240.
- The overall crunching factor of the second deployment is higher than that of the first deployment (912 > 662). But this can be explained by the low crunching factor of the FlexX phases during the first deployment due to the low success rate. The crunching factor of the Autodock phase (1031) is slightly better than the crunching factor of the second deployment, but it was measured only during 2 weeks (the Autodock phase). The two crunching factors can be considered as equal.

	Total	**WISDOM**	**DIANE**
Cumulated number of docked compounds (in millions)	2.47	2.16	0.31
Effective duration	42 days	42 days	30 days
Number of docked compounds / hour	2,450	2,143	431
Crunching factor	912	767	203
Number of jobs submitted	74,850	72,000	2,850
Number of grid Computing Elements used	69	69	36
Number of Resource Brokers used	20	19	1
Maximum number of jobs running in parallel on the grid	1,839	1,639	240
Volume of output data	752	700	52
Total CPU time	105 years	88.3 years	16.7 years

Table 9: Statistical summary of the WISDOM and DIANE deployment

In the next section, the performance of the two deployment tools is compared.

6.4.2. Performance comparison of WISDOM and DIANE production systems

The WISDOM and DIANE production system performance can be compared thanks to two parameters: the approximated distribution efficiency and the docking throughput.

Approximated distribution efficiency

The approximated distribution efficiency is defined as the ratio between the crunching factor and the maximum number of jobs running in parallel on the grid. This parameter is based on the approximation that the maximum number of jobs running in parallel on the grid

corresponds to the number of available processors during the overall period of the deployment, and the crunching factor is the constant number of running jobs.

The distribution efficiency was estimated at 47% for the WISDOM deployment. The low ratio for the WISDOM deployment is due to the overhead time in the current EGEE production system. The task pull model adopted by DIANE allows the isolation of the scheduling overhead of the grid jobs and is therefore expected to achieve a better distribution efficiency. During the data challenge, DIANE was able to push the efficiency to higher than 80% within the scope of handling the intermediate scale of distributed docking. Figure 47 presents the resource utilization by the DIANE framework for a period of 15 days.

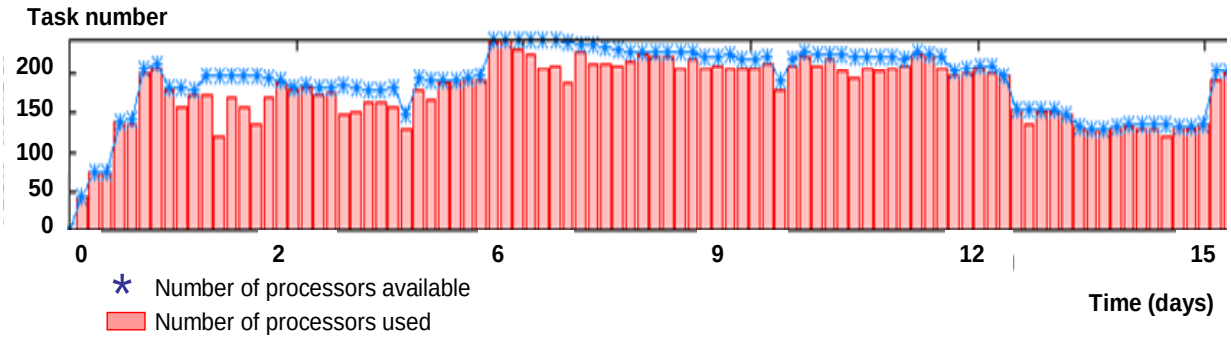


Figure 47: The resource utilization of a DIANE job

But a fair comparison with WISDOM could only be made if the improvement of the DIANE framework is tested in a large scale as with the WISDOM exercise.

Docking throughput

In this section, the docking throughput is the number of docked compounds by second. It depends on the execution time of the software and for this reason the docking throughput of the first and second deployments cannot be compared.

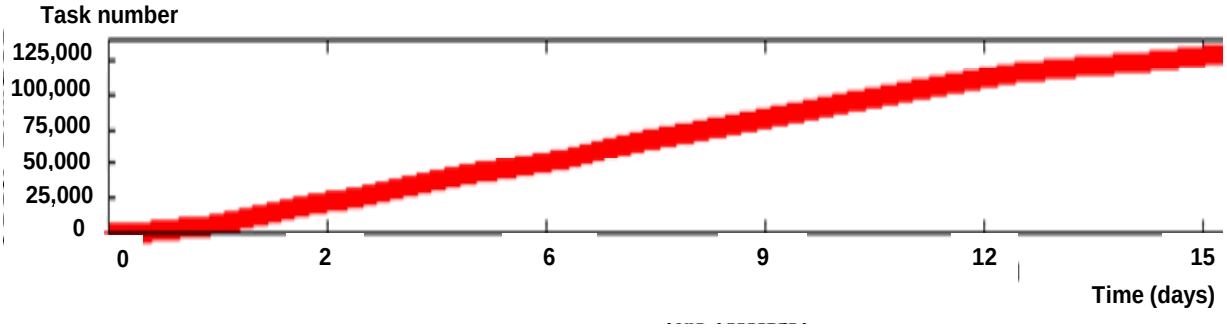


Figure 48: Cumulative plot of the completed dockings by the DIANE framework

Because of the highly scalable nature of the WISDOM framework, high throughput docking could be achieved at a rate of 1.7 seconds per docking. As DIANE was handling not more than a few hundred concurrent jobs, its throughput was limited to about one docking every 8.4 seconds. The cumulative plot of the completed dockings by the DIANE framework

in figure 48 also demonstrates that a constant throughput can be effortlessly maintained for a few weeks using the task pull model.

In the next section, the job success rate of each deployment is reported.

6.4.3. Analysis of the job success rate

Table 10 shows the success rates during the data challenge and more specifically for the WISDOM and DIANE activities.

Measures	Total	WISDOM	DIANE
EGEE success rate	83.1 %	83 %	85 %
Success rate after checking output data	70.1 %	70 %	71 %
Success rate after checking output data and subtracting WISDOM and/or DIANE failures	80.3 %	80 %	83 %

Table 10: Efficiency measures of the WISDOM and DIANE deployment.

In the WISDOM activity, about 83% of the jobs were reported as successfully finished according to the status logged in the grid Logging and Bookkeeping system. The observed failures were mainly due to errors of Resource Brokers (overload, disk crashes, change of a Resource Broker name), of sites (mis-configuration of Computing Elements) and of WISDOM users. The WISDOM user failures were the change of a user certificate during the data challenge leading to the loss of submitted jobs for two instances. Another WISDOM user source of failure was the overload for an unidentified reason of a User Interface due to the Java job submission tool.

The success rate went down to 70% after checking the content of the data output file. The main causes for these failures were frequent errors in the transfer of results to the Storage Elements and failures of the Computing Elements to access Autodock stored and registered in the Computing Element software repository (NFS and configuration failures). The success rate after checking output data and subtracting WISDOM failures is about 80%.

Compared to the previous data challenge, improvement is significant as the observed success rate was 63% for the success rate after checking output data and subtracting WISDOM and server license failures. Moreover, the WISDOM job success rate has reached the level of typical EGEE grid success rate. Several parameters explain this improvement:

- In order to balance the load on the Workload Management System, WISDOM submitted the jobs to 19 resource brokers in a round-robin order. An average of 10 Resource Brokers was used simultaneously.
- Because of the limitations identified during the first deployment, a strategy was adopted not to submit jobs in bursts but rather to have a constant submission flow to limit the impact of missing information and sink-hole effects.

- Resubmission by hand allows the process to be checked precisely and limits any sink-hole effect.
- The measured time for job submission was between 30 seconds and 2 minutes to limit Resource Broker overload.
- Several failures of the most used Resource Brokers were observed including disk crashes or overloaded services. As a consequence, hundreds of jobs were lost or retrieved with difficulty. But the problem was limited thanks to the save on grid of the job process information and the non automatic resubmission.

In the DIANE activity, a similar job failure rate was also observed; nevertheless, the failure recovery mechanism in DIANE automated the re-submission and guaranteed a finally fully complete job. The feature of interactively returning part of the computing efforts during the runtime (e.g. the output of each docking) introduces a more economical way of using the grid resources.

The scalability issue of the DIANE framework is due to the fact that the DIANE master needs to keep the connections with the distributed DIANE workers for task dispatching and worker health checking. Performance evaluation during the data challenge showed that the current implementation of the DIANE master is restricted to handling a few hundred DIANE workers at the same time. The main reason for this restriction is still under investigation. An alternative solution is to adopt stateless protocols (e.g. web services or stateless CORBA); however, this will introduce an overhead for establishing every connection and a tradeoff should be made between the performance and scalability.

For instance, to give users more flexible control over their DIANE jobs, the master of DIANE is usually executed on the User Interface. This feature will turn into a performance issue when the payload of result integration is high. The heavily loaded integration process will affect the performance of the User Interface. A possible approach to address this issue is to run the DIANE master as a Grid job on a Worker Node; however, one should make sure that the master is always started before the workers and the network connectivity between two Worker Nodes becomes yet another problem.

6.4.4. Issues related to the Grid middleware

The different grid components within the framework of a large scale deployment were tested again during this large scale deployment. Improvements were noticed, as was reported in the previous section. But there was no significant change in the problems surrounding the services of LCG-2 middleware since the last deployment because the new gLite middleware was still under development. The only new middleware service was the LCG File Catalogue which replaced the Replica Location Service of the EGEE biomedical Virtual Organization during the spring of 2006 for the second deployment. The LCG File Catalogue has the same functionalities as the Replica Location Service. It certifies data copy, registration and replication. This service was again a potential bottleneck, because there is only one file catalogue per Virtual Organization.

The output data transfer process was described in chapter 5. A first output copy registration is done and two replications are executed. The output storage is then controlled by a LCG File Catalogue command line. More than 100,000 files were stored on the grid, meaning that the commands were executed more than 400 000 times.

Failures have been observed regularly with the LCG File Catalogue for the preliminary installation of the compound database subsets and during the data challenge itself. The failure ratio was measured at about 1/20 during the database installation step and the data challenge deployment. The error can occur on any occasion, for no particular reason, and seems quite random. Most of the time, the command line must be rerun once more to be successful. Failure was reported to the EGEE grid user support.

However, thanks to the back-up solution for data transfers by the GridFTP retry system, outputs were saved despite the instability of the LCG File Catalogue system.

Among the new gLite services soon to be deployed on the EGEE infrastructure, the gLite Workload Management System is supposed to overcome several of the middleware limitations highlighted by the two WISDOM data challenges.

It will be possible to submit several jobs in bulk mode i.e. in a single interaction. This will avoid the overhead of the authentication step per single interaction and will speed up the submission process. In addition, the Logging and Bookkeeping information will be collected by dedicated services (the gLite Information Service) to avoid overloaded systems. Preliminary tests of the gLite Workload Management System have shown an improvement in the job submission speed of a factor of 5 to 10 with respect to the LCG Resource Broker.

The gLite middleware is currently deployed on the EGEE infrastructure but not sufficiently for the next data challenge WISDOM-II scheduled for autumn 2006.

6.5. Perspectives

The experiment demonstrated how grid infrastructures have a tremendous capacity to mobilize very large CPU resources for well targeted goals during a significant period of time. But beyond successful large scale deployment, the produced outputs need to be analyzed, post-processed and shared in a collaborative environment. This section summarizes the biological outputs and presents a grid-enabled high throughput virtual screening.

6.5.1. Biological results

About 120,000 files in total were archived in Taiwan and in France. The most interesting docked compounds were identified and selected by ranking binding free energies of resulted docked models obtained from this data challenge. The example presented below shows how the mutation of only one amino acid can affect the energy of docked complexes.

Figure 49 presents the energy distribution of docked complexes with the wild-type neuraminidase, one of the studied structures during the data challenge. The compound GNA, the core compound of the zanamivir drug, is docked with an energy located in the top 5% database compounds. The compound 4AM, the core compound of the oseltamivir drug, is

docked with an energy located in the 15%. These results are an argument to validate the overall docking process.

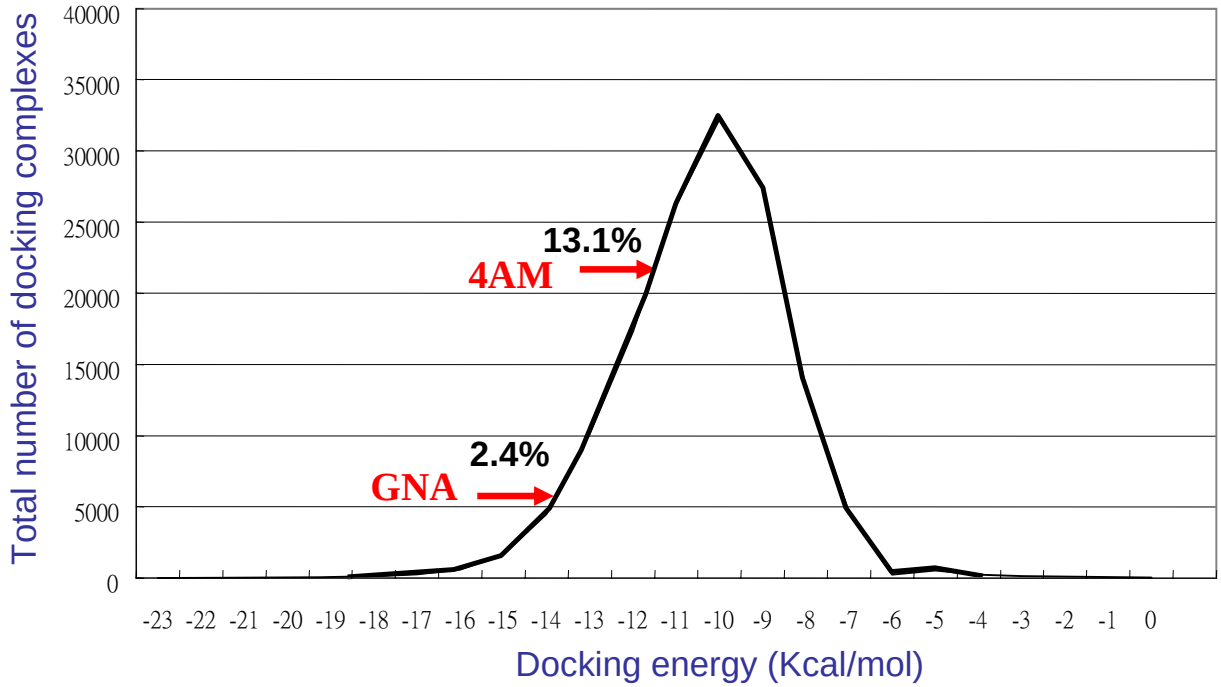


Figure 49: Energy distribution of docked complexes with the wild-type neuraminidase

Figure 50 presents the energy distribution of docked complexes with a substitution E119A in the active site of neuraminidase. E119A signifies that the glutamic acid amino-acid in position 119 is substituted by an alanine. The mutation affects the energy of the docking complexes: GNA is no longer in the top 5% and 4AM is now a similar very bad docking energy.

This example is an argument to validate the docking experiment: the docking conditions allow selection of a drug core. They also allow the retrieval of the mutation effect on the docking complex.

Two sets of re-ranked data for each target were made (QM-MM method and selection by the study of the interactions between the compound and the target active site). The top 15% is about 45,000 compounds for each target. This set will be publicly available for the scientific community working on the avian influenza neuraminidase. The top 5% is about 2,250 compounds for each target. This set will be refined by different methods (molecular modeling, molecular dynamics…). The analysis will indicate which residue mutation is critical, which chemical fragments are preferred in the mutation sub sites and other information for lead optimization to chemists. Finally, at least 25 compounds will be assayed experimentally at the Genomic Research Center from Academia Sinica.

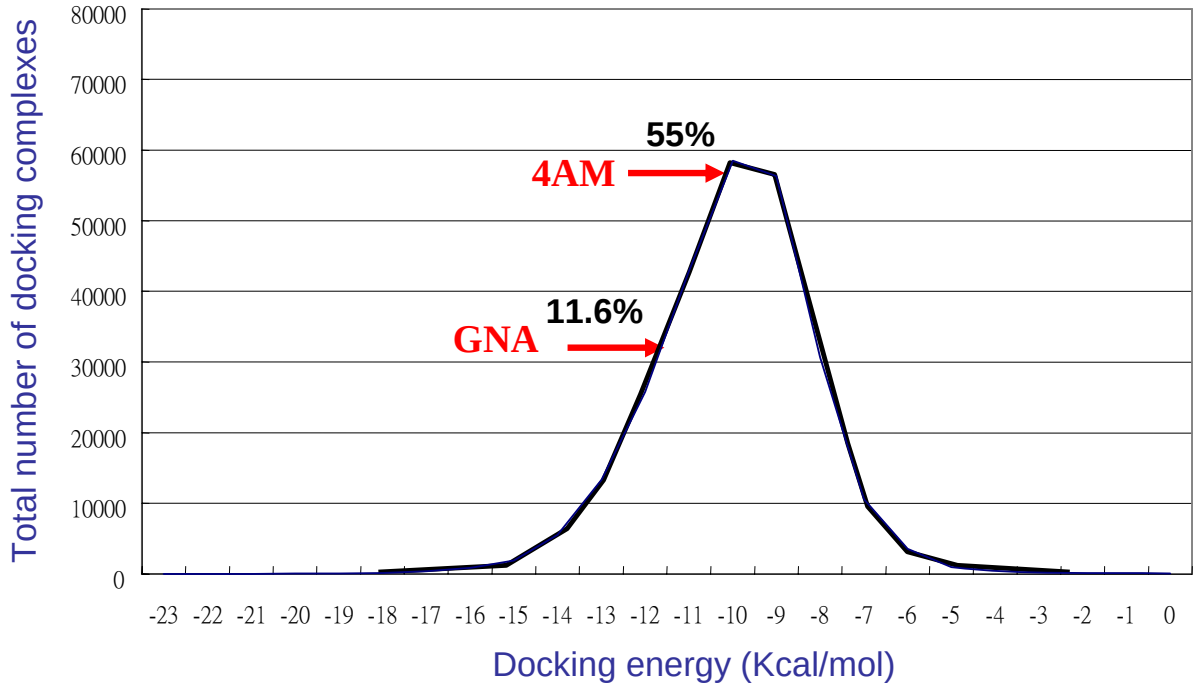


Figure 50: Energy distribution of docked complexes with the E119A mutated neuraminidase

6.5.2. High throughput virtual screening service

Improvements in the WISDOM environment during the second deployment demonstrate that a high throughput virtual screening service could be deployed for academic laboratories, and in the long-term for companies. Through a user-friendly interface, this service could register the customer data, applications and parameters, automate the grid deployment (installation, submission, monitoring, output retrieval), and provide the ranked outputs stored on the grid.

The automatic resubmission applied during the first data challenge was a cause of failure and consequently a time-consuming correction task for the job supervisor. The removal of this process during the second deployment increased the success rate, but it is a limit to the building of an automatic pipeline of grid-enabled virtual screening. An issue for the future will be to improve the WISDOM production environment to manage efficient automatic resubmission with only the relevant resource brokers and grid computing nodes. Other issues for a successful service deployment are cost and time reduction for scientists, security and data protection, fault tolerant and robust services and infrastructure, resource access policy and transparent and easy use of the interfaces.

Such a service could be progressively enriched by other services integrated in a grid infrastructure. Figure 51 illustrates a grid-enabled drug discovery pipeline. Grid service providers would deploy their drug discovery services on a grid infrastructure such as docking or molecular dynamics services. Grid service customers would access these drug discovery services through grid-enabled high throughput services such as that described in the previous paragraph.

Such service requires check points with easy data access. This is the main next issue for the WISDOM production system. The aim is to improve output data collections and post-docking analysis in order to reduce as much as possible the time needed for experts to analyze the results. Real-time storage of job output in a relational database is under way for the WISDOM-II data challenge.

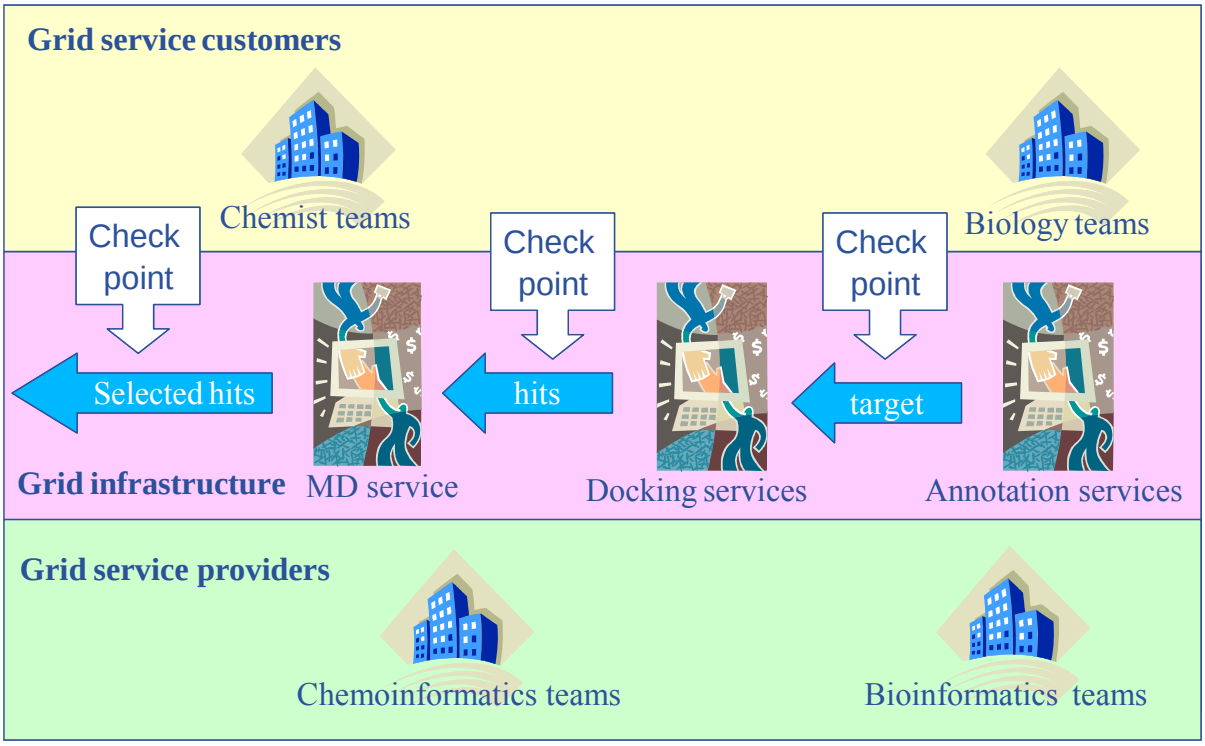


Figure 51: Grid-enabled *in silico* drug discovery pipeline

This model would be relevant for competence and resource sharing between developed and least developed countries. Services would be available routinely and in an emergency, in order to compress the overhead so that biomedical chemists can have the best response to instant threads, for instance while the mutation of a virus happens.

6.6. Conclusion

This sixth chapter has demonstrated that reliability of the improved WISDOM environment which reached the average EGEE grid success rate despite the deployment of tens of thousands jobs and consummation of almost 90 CPU years in the same time period.

This second data challenge is the first large deployment for an emerging infectious disease on three grid infrastructures. In April and May 2006, almost 1700 computers were simultaneously used in 22 countries around the world to dock over 2 million compounds. This new deployment required a much shorter preparation, about a month, and took advantage of the experience acquired with the first WISDOM deployment against malaria. On the biological side, the goal was to find potential compounds that can inhibit the activities of an

enzyme on the surface of the influenza virus, the so-called neuraminidase, subtype N1 through *in silico* virtual docking on a grid infrastructure.

Compared to the first WISDOM deployment, the deployment scale and the grid performance were similar. But the job success rate was greatly improved even if middleware issues reported in chapter 5 did not change between the two deployments. The rate is now at the level of the average EGEE grid success rate for lower scale grid applications. The main reasons for this improvement are a constant and slower job submission flow and a manual control of re-submission process. Compared to the DIANE deployment, the WISDOM production system allows a higher throughput for a similar job success rate. The DIANE light-weight framework offers better distribution efficiency and saves progressively the outputs during the job instead of at the end of the job as does the WISDOM system. The development of a new system merging functionalities from both WISDOM and DIANE frameworks is being considered. The next development steps of the WISDOM production system will be a migration to gLite middleware framework and the improvement of output data collection and post-docking analysis.

The results are now under analysis and the outcome will help biomedical chemists to reduce the cost of the first investment in the process of structure-based drug design. A permanent high throughput virtual screening service could offer a shared space for collaboration between developed and least developed countries. It could also be integrated into the *in silico* drug discovery pipeline. Moreover, the much shorter preparation time for this large scale deployment opens interesting perspective in the use of the large scale EGEE grid for urgent needs for disaster management like emerging infectious disease pandemic.

These achievements demonstrated the relevance of grids for the drug discovery process against neglected and emerging infectious diseases and in enabling world-wide and multidisciplinary collaboration.

6.7. References

[470] Moscicki, J.T., DIANE - Distributed Analysis Environment for GRID-enabled Simulation and Analysis of Physics Data, NSS IEEE 2004, (2004)

[471] Moscicki, J.T., et al., Biomedical Applications on the GRID: Efficient Management of Parallel Jobs, NSS IEEE 2003, (2003).

[472] Gropp, W. and Lusk, E., Dynamic process management in an MPI setting, Proceedings of the 7th IEEE Symposium on Parallel and Distributed Processing, (1995).

General conclusion

Neglected and emerging infectious diseases are major public health concerns in the beginning of the 21st century. Neglected diseases are one of the main reasons of deaths in least developed countries. They affect hundreds of millions of people world-wide, but they continue to suffer from a lack of research and development. Emerging infectious diseases have the potential to cause a large-scale pandemic and the question is how long would be required to develop a vaccine or a drug should such a pandemic occur.

Grids offer great perspectives to foster international collaboration in order to reduce costs and time for *in silico* drug discovery for neglected and emerging infectious diseases. This document describes innovative approaches to protein structure prediction and virtual screening taking advantage of grid technologies:

- a grid-enabled service to deploy protein structure prediction software and database on the RUGBI grid.
- a grid-enabled service to update protein structure prediction database on the RUGBI grid
- a grid-enabled deployment of high throughput structure-based virtual screening by docking against malaria
- a second grid-enabled deployment of high throughput structure-based virtual screening by docking against avian influenza

The deployment service aims to ease the use of software and databases for non-grid expert users on a grid environment. It allows registration, download and installation thanks to interfaces generated by the Information System describing software and databases. Developments are still needed to improve interfaces and enrich functionalities.

The Database Update Service aims to provide the grid users with the most up to date version of any biological flat file database, to do it transparently and without disturbing any running jobs. This service ensures that the user's analysis on grid will be carried out with the last available version of the database. It is installed for a precise database by its administrator but runs autonomously and periodically for small updates and for large new versions. Developments are still needed to interconnect the service with an execution service to parallelize recurrent time-consuming tasks such as indexing. There is a plan to migrate the service onto other grid environments like EGEE.

These two services are deployed on the RUGBI grid, which is available for commercial users, such as companies of the Biopôle Clermont-Limagne, and for academic users. Services are also proposed by Communication & Systems in the framework of grid solution for small and medium enterprises. The Database Update Service will be improved in the framework of

the Embrace network of excellence in collaboration with developers of the CONStanza system.

The deployments of computing and data intensive tasks against malaria and avian influenza are the first attempt to deploy large scale *in silico* docking on a public grid infrastructure. They allowed the testing of the grid operation and services for a CPU consuming application generating large data flows. Issues related to the deployment and the monitoring of the *in silico* docking experiment were reported.

The data challenges have been a very useful experience in identifying the limitations and bottlenecks of the EGEE infrastructure. The WISDOM production system developed to submit the jobs on the grid accounted for a small fraction of the failures, along with the grid management system. The current success rate of the WISDOM production system is now at the level of the average EGEE grid success rate for lower scale grid applications.

Compared to the DIANE deployment, the WISDOM production system allows a higher throughput for a similar job success rate. The DIANE light-weight framework offers better distribution efficiency and saves progressively the outputs during the job instead of at the end of the job as does the WISDOM system. The development of a new system merging functionalities from both WISDOM and DIANE frameworks is being considered. The next development steps of the WISDOM production system will be a migration to gLite middleware framework and the improvement of output data collection and post-docking analysis. The goal is to reduce as much as possible the time needed for experts to analyze the results.

On the bioinformatics side, the WISDOM data challenge has demonstrated that collaborative production grids can be used for steps in the drug discovery process. Grid-enabled high throughput structure-based virtual screening by docking is the first step to enable the virtual screening pipeline on a grid environment. The next step is the deployment of molecular dynamics simulations. The deployment against avian influenza required a short preparation period, about a month, and took advantage of the experience acquired with the first WISDOM deployment against malaria. Less than three months were required between the first contacts and the achievement for all the required virtual screening. A second data challenge against neglected diseases, called WISDOM-II, is currently running.

On the biological side, the data challenges produced a large amount of output for analysis. Results extracted compounds with key interactions and good scoring. These compounds are currently under further analysis thanks to grid-enabled molecular dynamics in the framework of BioinfoGRID project. The outcome will help biomedical chemists to reduce the cost of the first investment in the process of structure-based drug design. This large docking analysis is the initial proof of the concept: biologists need to come up with challenges that exploit the resources now available.

The experiments demonstrated how grid infrastructures have a tremendous capacity to mobilize very large CPU resources for well targeted goals during a significant period of time and in enabling world-wide and multidisciplinary collaboration.

The results described in this document demonstrate the relevance of grids in addressing neglected and emerging infectious diseases. It is however a first step and much work still needs to be done before *in silico* drug discovery on grids is adopted by the research community. Nonetheless, our results open important perspectives.

A permanent high throughput virtual screening service could offer a shared space for collaboration between developed and least developed countries. It could also be integrated with the grid-enabled service for deployment and update. The cost of the discovery of new inhibitors against neglected diseases would thus be lower. The final development of drug candidates could be awarded to a laboratory based on competitive bids. The drug itself would go into the public domain, for generic manufacturers to produce. This would achieve the goal of getting new medicines to those who need them at the lowest possible price. This model is currently supported by various United Nations programs and involves several commercial organizations, including large pharmaceutical companies.

Emerging infectious diseases can evolve into a crisis situation. In case of a pandemic, public authorities, like governments, have to take urgent decisions. Various resources such as health-care centers and infrastructures can be requisitioned. Industries can be solicited to produce on a large scale drugs or vaccines.

A grid infrastructure providing easily and quickly thousands of processors could be used with the same approach for urgent needs. Such a grid infrastructure could contribute to an international drug or vaccine discovery effort by large computing power access. Mission critical applications (e.g. disease diagnosis, drug discovery, etc.) running on a grid require a high level of Quality of Service. Taking the example of the data challenge, the throughput is one of the key Quality of Service parameters as time may become a critical factor to address emerging infectious diseases. The avian influenza virus might spread at an unexpected speed once the variant with the ability of human-to-human transmission comes out.

According to the definition of the Quality of Service taxonomy [249], the Quality of Service in the current EGEE middleware is implemented in a software way based only on the ranking and match-making mechanism provided by the Workload Management System. The Workload Management System relying highly on the Information System has no way to guarantee that its resource selection will meet the user's Quality of Service requirement. Thus how to ensure the Quality of Service within the current grid middleware is still an open question.

Beyond virtual screening, grid technology provides the collaborative Information Technology environment to enable the combining of molecular biology research and goal-oriented field work. The long term vision of the WISDOM initiative is to build a grid for

malaria. The aim would be to provide services to research laboratories working on malaria. One of these services could be to collect and analyze epidemiological data from least developed countries.

It proposes a new paradigm for the collection and analysis of distributed information where data are no longer centralized in one single repository. On a grid, data can be stored anywhere and still be transparently accessed by any authorized user. The computing resources of a grid are also shared and can be mobilized on demand so as to enable very large scale genomics comparative analysis and virtual screening.

A web site would allow chemists and biologists to volunteer their expertise on certain areas of the disease. They would examine and annotate shared databases and perform experiments. The results would be fully transparent and discussed in chat rooms. The research would be initially mainly computational, based on resources provided by grid infrastructures, and not carried out in “wet” laboratories. Scientists would collaborate on the data and the software.

An environment supported by the grid-enabled facilities proposed in this document can contribute to the fostering of international collaborations in the fight against neglected and emerging infectious diseases. The very real hope is that this can accelerate the discovery of new drugs and therefore save lives.

List of figures

List of tables

List of publications and other work

International journal papers

[473] Kasam, V., Zimmermann, M., Maaß, A., Schwichtenberg, H., Wolf, A., **Jacq, N.**, Breton, V., Hofmann-Apitius, M., Design of New Plasmepsin Inhibitors: A Virtual High Throughput Screening Approach On The EGEE Grid, accepted for publication in JCIM, (2007)

[474] Salzemann, J., **Jacq, N.**, Breton, V., Replication and Update of Molecular Biology Databases, IEEE Transactions on Nanobioscience, 6 131-135 (2007) doi: 10.1109/TNB.2007.897468.

[475] Breton, V., **Jacq, N.**, Kasam, V., Hofmann-Apitius, M., Grid Added Value to Address Malaria, IEEE Transactions on Information Technology for Biomedicine, 11 (2007) doi: 10.1109/TITB.2007.895930.

[476] **Jacq, N.,** Breton, V., Chen, H.-Y., Ho, L.-Y., Hofmann, M., Lee, H.-C., Legré, Y., Lin, S.C., Maaß, A., Medernach, E., Merelli, I., Milanesi, L., Rastelli, G., Reichstadt, M., Salzemann, J., Schwichtenberg, H., Sridhar, M., Kasam, V., Wu, Y.-T., Zimmermann, M., Virtual Screening on Large Scale Grids, Parallel Computing, 33 289-301 (2007) doi:10.1016/j.parco.2007.02.010

[477] **Jacq, N.,** Salzemann, J., Jacq, F., Legré, Y., Medernach, E., Montagnat, J., Maaß, A., Reichstadt, M., Schwichtenberg, H., Sridhar, M., Kasam, V., Zimmermann, M., Hofmann, M., Breton, V., Grid-enabled Virtual Screening against malaria, to be published in Journal of Grid Computing, (2007).

[478] Birkholtz, L.-M., Bastien, O., Wells, G., Grando, D., Joubert, F., Kasam, V., Zimmermann, M., Ortet, P., **Jacq, N.**, Saidani, N., Hofmann-Apitius, S., Hofmann-Apitius, M., Breton, V., Louw, A.I., Marechal, E., Integration and mining of malaria molecular, functional and pharmacological data: how far are we from a chemogenomic knowledge space?, Malaria Journal, 5 110 (2006) doi:10.1186/1475-2875-5-110.

[479] Lee, H.-C., Salzemann, J., **Jacq, N.,** Chen, H.-Y., Ho, L.-Y., Merelli, I., Milanesi, L., Breton, V., Lin, S.C., Wu Y.-T., Grid-enabled High-throughput *in silico* Screening against influenza A Neuraminidase, IEEE Transactions on Nanobioscience, 5 288-295 (2006).

[480] **Jacq, N.,** Blanchet, C., Combet, C., Cornillot, E., Duret, L., Kurata, K., Nakamura, H., Silvestre, T., Breton, V., Grid as a bioinformatics tool, Parallel Computing 30 1093-1107 (2004).

Proceedings editor

[481] **Jacq, N.**, Müller, H., Blanquer, I., Legré, Y., Breton, V., Hausser, D., Hernandez, V., Solomonides, T., Hofmann-Apitius, M. (Eds), From Genes to Personalized Healthcare: Grid Solutions for the Life Sciences, proceedings of the HealthGrid 2007, Studies in Health Technology and Informatics, Geneva, Switzerland, (2007)

Proceedings in international conferences

[482] Breton, V., Blanquer, I., Hernandez, V., **Jacq, N.**, Legré, Y., Olive, M., Solomonides, T., Roadmap for a European Healthgrid, proceedings of the HealthGrid 2007 conference, Studies in Health Technology and Informatics, 154-163 (2007)

[483] Salzemann, J., Kasam, V., **Jacq, N.**, Maass, A., Schwichtenberg, H., Breton, V., Grid enabled High Throughput Virtual Screening against Four Different Targets Implicated in Malaria, proceedings of the HealthGrid 2007 conference, Studies in Health Technology and Informatics, 47-54 (2007)

[484] **Jacq, N.,** Breton, V., Chen, H.-Y., Ho, L.-Y., Hofmann, M., Lee, H.-C., Legré, Y., Lin, S.C., Maaß, A., Medernach, E., Merelli, I., Milanesi, L., Rastelli, G., Reichstadt, M., Salzemann, J., Schwichtenberg, H., Sridhar, M., Kasam, V., Wu, Y.-T., Zimmermann, M., Grid-enabled High Throughput Virtual Screening, proceedings of the International Workshop Distributed, High-Performance and Grid Computing in Computational Biology (GCCB) in ECCB 2006

conference, Lecture Notes in Computer Science, Eilat, Israel, 4360 45-59 (2007), doi: 10.1007/978-3-540-69968-2_5.

[485] **Jacq, N.,** Breton, V., Chen, H.-Y., Ho, L.-Y., Hofmann, M., Lee, H.-C., Legré, Y., Lin, S.C., Maaß, A., Medernach, E., Merelli, I., Milanesi, L., Rastelli, G., Reichstadt, M., Salzemann, J., Schwichtenberg, H., Sridhar, M., Kasam, V., Wu, Y.-T., Zimmermann, M., Large Scale In Silico Screening on Grid Infrastructures, Proceedings of the Third International Life Science Grid Workshop (LSGRID) 2006, Yokohama, Japan, (2006).

[486] Lee, H.-C., Salzemann, J., **Jacq, N.,** Chen, H.-Y., Ho, L.-Y., Merelli, I., Milanesi, L., Breton, V., Lin, S.C., Wu Y.-T., Grid-enabled High-throughput *in silico* Screening against influenza A Neuraminidase, Proceedings of the Distributed Applications, Web Services, Tools and GRID Infrastructures for Bioinformatics (NETTAB) 2006 workshop, Italy, (2006).

[487] Salzemann, J., **Jacq, N.,** Le Mahec, G., Breton, V., Replication and Update of Molecular Biology Databases in a Grid Environment, Proceedings of the Distributed Applications, Web Services, Tools and GRID Infrastructures for Bioinformatics (NETTAB) 2006 workshop, Italy (2006).

[488] **Jacq, N.,** Salzemann, J., Legré, Y., Reichstadt, M., Jacq, F., Zimmermann, M., Maaß, A., Sridhar, M., Kasam, V. Schwichtenberg, H., Hofmann, M., Breton, V., Demonstration of In Silico Docking at a Large Scale on Grid Infrastructure, proceedings of the HealthGrid 2006 conference, Studies in Health Technology and Informatics, Valencia, Spain, 120 155-157 (2006), PMID: 16823133.

[489] Breton, V., **Jacq, N.,** Hofmann, M., Grid added value to address malaria, Proceedings of the 6-th IEEE International Symposium on Cluster Computing and the Grid (BioGrid), Singapor, 40 (2006).

[490] Jacq, F., Bacin, F., Meda, N., Donnarieix, D., Salzemann, J., Vayssière, V., **Jacq, N.**, Renaud, M., Traore, F., Meda, G., Nikiema, R., Breton, V., Towards grid-enabled telemedicine in Africa, Proceedings of Information Society Technology-Africa 2006 Conference, Pretoria, South-Africa, Paul Cunningham and Miriam Cunningham (Eds). IIMC International Information Management Corporation, (2006) ISBN: 1-905824-01-7

Proceedings in national conferences

[491] **Jacq, N.,** Reichstadt, M., Jacq, F., Salzemann, J., Zimmermann, M., Maas, A., Sridhar, M., Kasam, V., Schwichtenberg, H., Hofmann, M., Breton, V., Les grilles pour le développement médical, Proceedings of "Le Médicament, de la Recherche au Terrain" conference, Lyon, France, (2005).

[492] Hernandez, F., Nicoud, S., **Jacq, N.,** Datagrid, projet européen de grille de calcul, Actes de la conférence JRES 2001, Lyon, France, (2001)

Invited seminar and contributions for international and national conferences

[493] **Jacq, N.**, World-wide in silico docking against malaria on grid infrastructures, First Belief Conference, Dehli, India, (2006).

[494] **Jacq, N.**, World-wide in silico drug discovery against neglected and emerging diseases on grid infrastructures, invited seminar in the University of Cyprus, Nicosia, Cyprus, (2006).

[495] **Jacq, N.,** Lee, H.C., Salzemann, J., Chen, H.Y., Milanesi, L., Wu, Y.T., Breton, V., In Silico Docking on Grid Infrastructure to Accelerate Structure-based Design against Influenza A Neuraminidases, invited contribution in the proceedings of Anti-Avian Influenza 2006 conference, Paris, France, (2006).

[496] **Jacq, N.** and Breton, V., In Silico Docking on Grid Infrastructures, 2nd Nordic Grid Neighbourhood Conference, Stockholm, Sweden, (2006).

[497] **Jacq, N.** and Breton, V., Grid Infrastructure for VHTS, Journée De la molécule au médicament, Toulouse, France, (2006).

Contribution for international conferences and workshops without proceedings

[498] Jacq, F., Bacin, F. Meda, N., Donnarieix, D., Salzemann, J., Vayssiere, V., **Jacq, N.**, Renaud, M., Traore, F., Meda, G., Nikiema, R., Breton, V., La grille au service du développement

médical en Afrique, 12èmes Journées Francophones Informatique Médicale, Bamako, Mali, (2007)

[499] **Jacq, N.,** Salzemann, J., Legré, Y., Reichstadt, M., Jacq, F., Zimmermann, M., Maaß, A., Sridhar, M., Kasam, V. Schwichtenberg, H., Hofmann, M., Breton, V., In Silico Docking on EGEE infrastructure: the case of WISDOM, EGEE User Forum, Geneva, Switzerland, (2006).

[500] Thiam, C.O., **Jacq, N.,** Donnarieix, D., El Bitar, Z., Maigne, L., Breton, V., Biomedical Research on a Computing Grid Environment, International Symposium on Web Services for Computational Biology and Bioinformatics, ?, USA (2005).

[501] Breton, V. and **Jacq, N.,** Data challenge on Drug Discovery, Bioinformatics mini-symposia in the Parallel Computing 2005 conference, Malaga, Spain, (2005).

Contribution for national workshops without proceedings

[502] **Jacq, N.,** Lee, H.C., Salzemann, J., Chen, H.Y., Milanesi, L., Wu, Y.T., Breton, V., In Silico Docking on Grid Infrastructure to Accelerate Drug Design against Influenza A Neuraminidases, Workshop Ontologie, Grille et Intégration Sémantique pour la Biologie in the Journées Ouvertes Biologie Informatique Mathématiques, Bordeaux, France, (2006).

[503] **Jacq, N.** and Breton, V., *In silico* docking on grid infrastructure, the case of WISDOM, Journée CINES De l'expérimentation à la simulation, Montpellier, France, (2006).

[504] **Jacq, N.** and Breton, V., Recherche de nouveaux médicaments sur grille informatique pour des maladies négligées, Journées CINES Le calcul et la simulation pour le médicament, Montpellier, France, (2005).

[505] **Jacq, N.** and Breton, V., Recherche de nouveaux médicaments à l'aide d'une grille pour des maladies négligées, Réunion conjointe sur les GrIlles et les Ontologies/Métadonnées pour la GéNomique 2004, Lyon, France (2004).

[506] **Jacq, N.,** Projet RUGBI: réalisation et utilisation d'une grille pour la bioinformatique, GrIlles pour la GéNomique 2004, Lyon, France, (2004).

[507] **Jacq, N.** and Cornillot, E., Bioinformatique distribuée : application dans le domaine de la parasitologie, GrIlles pour la GéNomique 2003, Lyon, France, (2003).

Invited lectures

[508] **Jacq, N.,** Biomedical activities, EGEE France training event, Clermont-Ferrand, France, (2006).

[509] **Jacq, N.,** WISDOM design and deployment, EGEE training event in the University of Cyprus, Nicosia, Cyprus, (2006).

[510] **Jacq, N.,** In silico docking on grid infrastructure, the case of WISDOM, BioinfoGRID initial training course, ?, Italy, (2006).

[511] **Jacq, N.,** Health grids: status, perspectives and user point of view, First Latinamerica Grid Workshop, Merida, Venezuela, (2004).

Book chapter in preparation

[512] Breton, V., **Jacq, N.**, Kasam, V., Salzemann, J., Chapter 2: Deployment of grid life sciences applications, Book chapter in preparation, Talbi, E.-G., Zomaya, A. (eds.) Grids for Bioinformatics and Computational Biology (2007).

Demonstrations, posters and web pages

[513] http://wisdom.healthgrid.org

[514] http://wisdom.eu-egee.fr/

[515] http://wisdom-demo.healthgrid.org/

[516] **Jacq, N.,** Lecluse, A., Nougarède, R. Reynaud, S., Roebuck, J.P., Démésy, N., Salzemann, J., Freret, L., De Vuyst, F., Y Legré, Y., Blanchet, C., Hernandez, F., Déléage, G., Prévoteau, H., Breton, V., Langlois, S., Déploiement de services bioinformatiques dans un environnement de grille de calcul, poster, Journées Ouvertes Biologie Informatique Mathématiques, Bordeaux, France, (2006).

[517] **Jacq, N.,** Salzemann, J., Legré, Y., Reichstadt, M., Jacq, F., Zimmermann, M., Maaß, A., Sridhar, M., Kasam, V. Schwichtenberg, H., Hofmann, M., Breton, V., Demonstration of In Silico Docking at a Large Scale on Grid Infrastructure, HealthGrid 2006 conference, demonstration, Valencia, Spain, (2006)

[518] Kasam, V., Zimmermann, M., Hofmann, M., Maass, A., Mahendrakar, S., **Jacq, N.,** Breton V., Large In Silico Drug Discovery On Grids For Neglected Diseases, eCheminfo Colyaer poster session, poster, (2005) and http://echeminfo.colayer.net/COMTY_mekasamv.

[519] Blanchet, C., Breton, V., Deléage, G., Démésy, N., Hernandez, F., **Jacq, N.,** Langlois, S., Lecluse, A., Legré, Y., Linglin, D., Musso, J.F., Nougarede, R., Prévoteau, H., Reynaud, S., Roebuck, J.P., Demonstration of the First Prototype of RUGBI, Design and Deployment of a Grid for Bioinformatics, demonstration, HealthGrid 2005 conference, Oxford, United Kingdom (2005).

[520] Blanchet, C., Breton, V., Deléage, G., Démésy, N., Hernandez, F., **Jacq, N.,** Langlois, S., Lecluse, A., Legré, Y., Linglin, D., Musso, J.F., Nougarede, R., Prévoteau, H., Reynaud, S., Roebuck, J.P., RUGBI, Design and Deployment of a Grid for Bioinformatics, HealthGrid 2004 conference, poster, Clermont-Ferrand, France, (2004).

[521] **Jacq, N.,** Bouet, M., Cornillot, E., A Meta-Model for the Distribution of Bioinformatic Data in the Field of Genomics and Post-genomics on a Grid, HealthGrid 2003 conference, Lyon, France, (2003).