Epidémiologie moléculaire et métagénomique à haut débit sur la grille

Trung-Tung Doan

N° d'ordre : D. U :
E D S P I C :

Université Blaise Pascal - Clermont II

ECOLE DOCTORALE
SCIENCES POUR L'INGENIEUR DE CLERMONT-FERRAND

T h è s e

Présentée par

DOAN TRUNG-TUNG

pour obtenir le grade de

D o c t e u r d' U n i v e r s i t é

SPECIALITE : **INFORMATIQUE**

Epidémiologie moléculaire et Métagenomique à haute débit sur la grille

Soutenue publiquement le 17 décembre 2012 devant le jury :

M. Patrick Bellot	Président
M. Alain Franc	Rapporteur et examinateur
M. Eddy Caron	Rapporteur et examinateur
Mme. Gisèle Bronner	Examinateur
M. Vincent Breton	Directeur de thèse
M. Nguyen Hong Quang	Co-directeur de thèse

REMERCIEMENTS

Je tiens à remercier le Dr. Vincent Breton, Directeur de ma thèse, pour m'avoir encouragée et stimulée dans la réalisation de mes travaux par son exigence, ses apports constructifs, sa vigilance et sa patience.

Je remercie également le Dr. Nguyen Hong Quang, Co-Directeur de ma thèse pour l'aide, les conseils et le support tout au long de cette thèse.

J'adresse tous mes remerciements à Dr. Alain Franc, ainsi qu'à Dr. Eddy Caron, de l'honneur qu'ils m'ont fait en acceptant d'être rapporteurs de cette thèse.

Je remercie très sincèrement Prof. Pattrick Bellot et Dr. Gisèle Bronner pour la coopération et la participation au jury de thèse.

Je voudrais remercier Dr. Le Thanh Hoa qui me a fourni des idées qui m'ont permis d'enrichir la réalisation de cette thèse.

Je tiens également à remercier Dr. Lydia Maigne, responsable de l'équipe PCSV et tous mes collègues / amis à l'IFI- MSI, IOIT, HPC, l'équipe PCSV, LPC, et laboratoire LGME, spécialement Jean Salzemann, Géraldine Fettahi, Ana Lucia Da Costa, Vincent Bloch, Najwa Taib, Bui The Quang, Vu Trong Hieu, Dao Quang Minh et Pham Quang Trung pour ses aides et ses supports.

Enfin, je remercie particulièrement ma famille proche pour son soutien et bien évidemment remercier mon extraordinaire femme Phuong qui m'épaule maintenant depuis 6 ans et sans qui rien n'aurait été possible.

TABLE DES MATIERES

LIST DES FIGURES

LISTE DES TABLEAUX

CHAPITRE 1 : INTRODUCTION

1.1. Contexte

Les « grilles informatiques » sont des infrastructures constituées d'un ensemble d'ordinateurs, géographiquement éloignés mais fonctionnant en réseau pour fournir une puissance globale. Elles proposent des ressources informatiques (de calcul et/ou de stockage) qui, avec l'aide de protocoles et interfaces standardisés, ouverts et génériques, offrent des qualités de service non triviales [1]. La grille informatique permet de mobiliser les ressources de plusieurs ordinateurs dans un réseau sur un seul problème en même temps - le plus souvent un problème scientifique ou technique qui nécessite un grand nombre de cycles de calcul ou l'accès à de grandes quantités de données. L'intergiciel de la grille doit permettre aux utilisateurs d'avoir un accès uniforme et transparent aux ressources dans un environnement hétérogène.

La grille informatique est devenue un élément important du panorama du calcul intensif (High Performance Computing en anglais) de la science, des affaires, et dans une moindre mesure de l'industrie [2]. Elle ouvre l'accès à des cycles de calcul abondants et moins coûteux que les supercalculateurs et constitue un environnement privilégié pour les collaborations scientifiques entre pays développés et pays en développement. Elle permet en effet à tous ses utilisateurs de disposer des mêmes services, quelle que soit leur situation géographique, sous réserve de disposer d'un accès au réseau performant. De plus, elle s'appuie sur l'infrastructure informatique existante pour optimiser les ressources et gérer les données ainsi que les charges de calcul [3].

Le travail décrit dans ce manuscrit s'est inscrit dans le démarrage des activités de grille au Vietnam. Une collaboration entre les établissements d'enseignement et de recherche Français et Vietnamiens poursuit depuis 2007 l'objectif de déployer une infrastructure informatique distribuée pour soutenir la recherche au Vietnam. Plusieurs écoles et workshops ont été organisés depuis 5 ans pour former des chercheurs, ingénieurs et étudiants vietnamiens à l'administration et à l'utilisation des ressources de la grille.

Le travail de portage et de déploiement d'une application scientifique sur une grille relève de l'e-science, qui se définit comme un nouvel environnement de recherche dans lequel des outils informatiques puissants sont mis à disposition des chercheurs au sein d'un réseau interopérable. Si la physique des hautes énergies a joué un rôle pionnier dans l'adoption et le déploiement des grilles pour le traitement de données massives, beaucoup d'applications scientifiques l'ont suivie. On peut citer les sciences du vivant (biologie moléculaire, biochimie), les sciences de la planète (climatologie, sismologie), les sciences complexes, la finance,...

Le travail décrit ici se situe dans le domaine de la bioinformatique qui, à l'interface entre biologie moléculaire et informatique, englobe des outils de calcul et les méthodes utilisées pour gérer, analyser et manipuler de grands ensembles de données biologiques. La génomique ne recouvre à ce jour qu'une petite partie de la variété des ensembles de données produites en biologie mais représente une majorité des besoins de stockage, de calcul et d'analyse. La production des données brutes (séquences), réclame des moyens importants de stockage et d'archivage et des moyens relativement modérés de calculs. Le traitement/service/support bio-informatique permettant de produire des données 'filtrées' à plus forte valeur ajoutée et exploitables par les communautés intéressées (biologie/médecine/agronomie) demande de très gros volumes de stockage sur disque mais aussi d'importantes ressources de calcul.

Essentiellement, la bioinformatique a trois composantes [4]:

- La création de bases de données, permettant de stocker et de gérer des grands ensembles de données biologiques.
- Le développement des algorithmes et des statistiques, permettant de déterminer les relations entre les membres de grands ensembles de données.
- L'utilisation de ces outils pour l'analyse et l'interprétation des différents types de données biologiques, y compris l'ADN, l'ARN et les séquences de protéines, des structures de protéines, des profils d'expression génique, et les voies biochimiques.

Les premières analyses bioinformatiques de données de biologie moléculaire ont été déployées sur la grille dès le début des années 2000 mais elles se sont heurtées aux limitations des services proposés par les intergiciels de première génération. L'évolution de la technologie a permis d'enrichir progressivement le spectre de services offerts aux utilisateurs de la grille, notamment dans le domaine de la gestion des données et de la conception de pipelines de traitement. Ces services ont permis d'envisager le portage d'applications de plus en plus complexes.

Dans le cadre de cette thèse, nous nous sommes concentrés sur l'utilisation de la grille en sciences du vivant dans deux nouveaux domaines scientifiques d'intérêt spécifique pour le Vietnam.

Le premier domaine est celui de la santé publique internationale. Les récentes pandémies ont montré que les maladies émergentes se propageaient très rapidement dans le monde entier. Le Vietnam a été l'un des premiers pays exposés au SRAS (Syndrome Respiratoire Aigu Sévère) et à la grippe H5N1. Depuis 10 ans, l'Organisation Mondiale de la Santé basée à Genève a pris conscience de l'intérêt de la grille informatique pour gérer la collecte et l'analyse des données épidémiologiques. Donner aux virologues qui isolent les souches impliquées sur les nouveaux foyers des outils très performants permettant de diagnostiquer in silico par épidémiologie moléculaire l'émergence de nouveaux virus et de caractériser leur pathogénicité pourrait sauver des milliers de vies.

Le deuxième domaine est celui de la métagénomique. La richesse de la biodiversité au Vietnam est très grande. Sa connaissance et son utilisation pour la recherche de nouveaux médicaments sont encore très incomplètes. La caractérisation de la biodiversité s'appuie de plus en plus sur la métagénomique qui exploite les technologies de séquençage de nouvelle génération. Celles-ci génèrent des volumes de données comparables à ceux de la physique des particules à l'échelle mondiale. L'utilisation de la grille pour la métagénomique constitue une voie prometteuse pour l'analyse de la biodiversité au niveau mondial. Elle rend possible des approches globales et systématiques par les ressources qu'elle peut mobiliser.

1.2. Objectifs de la thèse

Nous avons présenté brièvement le contexte dans lequel ce travail de thèse s'est inscrit. Nous avons identifié deux domaines scientifiques, l'épidémiologie moléculaire pour la surveillance des maladies émergentes et la métagénomique pour l'étude de la biodiversité, pour lesquels la grille peut constituer un environnement très performant, notamment pour la recherche au Vietnam.

L'utilisation de la grille dans ces domaines scientifiques implique de relever plusieurs défis :

- pour la surveillance des maladies émergentes, il s'agit de proposer un ensemble d'outils d'analyse d'épidémiologie moléculaire permettant une prise de décision rapide sur les nouvelles souches isolées et séquencées
- pour la métagénomique, le défi se situe dans le volume important des données produites par chaque pyroséquençage. Plusieurs centaines de milliers, voire des millions de séquences courtes doivent être analysées.

A ces deux défis s'ajoute celui de la facilité d'utilisation. En effet, l'obstacle principal à l'adoption de la grille en sciences du vivant a été et demeure la difficulté de son utilisation par des personnes peu expérimentées en informatique. Bien que ce problème ait été identifié très tôt, le développement d'environnements conviviaux a été très lent, pour plusieurs raisons :

- les services proposés par les intergiciels étaient de très bas niveau, essentiellement au niveau des lignes de commande
- le traitement des données en science du vivant requiert une chaine de traitement et donc l'utilisation d'outils de gestion de flot
- les utilisateurs (biologistes, médecins) étaient vraiment éloignés de l'informatique et avaient besoin d'environnement leur permettant de manipuler des concepts propres à leur discipline

Les objectifs de notre travail étaient donc d'explorer la pertinence de la grille pour répondre aux besoins de nouvelles communautés scientifiques, de concevoir et/ou d'offrir des services faciles et agréables d'utilisation notamment par des chercheurs actifs dans un pays en développement.

Nous nous sommes appuyés pour cela sur les services de gestion de données et

de conception de pipelines de traitement ainsi que sur les environnements de production ou de traitement de données biologiques ou biochimiques aujourd'hui disponibles sur la grille EGI.

1.3. Structure de la thèse

Après cette introduction, nous présentons dans le chapitre 2 un état de l'art des infrastructures de grille utilisées pour la production scientifique en bioinformatique ainsi que les outils les plus populaires pour le déploiement des applications (portails, plates-formes). Nous présenterons aussi les applications déployées sur la grille dans les deux domaines spécifiques auxquels nous nous sommes intéressés dans ce mémoire : l'épidémiologie et la métagénomique.

Le chapitre 3 est réservé à présenter en grands détails WISDOM (Wide In Silico Docking On Malaria), plate-forme de production sur la grille qui fut développé à partir de 2005 pour la recherche de médicaments contre les maladies négligées et émergentes. Le WPE (Wisdom Production Environnement) a été utilisé pour le portage et le déploiement de nos applications scientifiques. Le concept principal est de simplifier les soumissions des jobs sur la grille via les jobs pilotes (les jobs génériques qui peuvent contrôler et lancer les tâches réelles différentes) et le modèle PULL (les tâches sont récupérés et exécutés automatiquement). Après une description complète du WPE, nous présentons des tests de performance du WPE qui ont mis en évidence certaines limitations ainsi que les améliorations apportées. La nouvelle version du WPE offre des performances nettement améliorées par rapport à la première.

Basés sur ces études, nous présentons dans le chapitre 4 l´architecture et la conception d´une plate-forme appelée g-INFO (grid-based International Network for Flu Observation), mise à la disposition des épidémiologistes moléculaires pour étudier la grippe de type A. L'approche adoptée est de fédérer les données de séquences de la grippe et fournir aux épidémiologistes les outils de la grille pour faire les analyses sur ces données. Les outils bioinformatiques choisis dans la plate-forme sont les outils populaires qui peuvent être utilisés dans beaucoup d'analyses bioinformatiques. Nous avons intégré un moteur de flot dans la plate-forme pour accroître la flexibilité dans les traitements.

Le chapitre 5 présente le déploiement sur la grille d'une chaine de traitement déjà existante et conçue par une équipe du Laboratoire Micro-organismes Génomique et Environnement (LMGE) : la plate-forme ePANAM d'analyse de données d'écologie microbienne. L'objectif était alors d'exploiter les ressources de la grille pour accélérer le traitement des données de pyroséquençage et ainsi faire de la plate-forme ePANAM un portail ouvert aux utilisateurs potentiels.

En conclusion, nous reviendrons sur les objectifs fixés au départ et dresserons un bilan des résultats obtenus. Nous proposerons ensuite des perspectives pour poursuivre le travail accompli.

CHAPITRE 2 : ETAT DE L'ART

Ce chapitre présente un état de l'art des infrastructures de grille utilisées pour la production scientifique en bioinformatique, en France, au Vietnam et de façon plus générale, au niveau international, ainsi que les outils les plus populaires pour le déploiement des applications (portails, plates-formes).

Nous présenterons aussi les applications déployées sur la grille dans les deux domaines spécifiques auxquels nous nous sommes intéressés dans ce mémoire à savoir l'épidémiologie et la métagénomique. Nous introduirons le chapitre par un glossaire des termes utilisés de façon répétée dans le manuscrit et le terminerons par une brève conclusion.

2.1. Glossaire

Ce glossaire propose quelques termes importants qui seront répétés tout au long de cette thèse.

2.1.1. Glossaire grille

Computing Element (CE) : L'élément de calcul est le service qui représente une ressource de calcul. Sa principale fonction est la gestion des jobs (voir la définition de job ci-dessous) par exemple : la soumission des jobs, le contrôle des jobs, etc. Le CE peut être utilisé directement par un utilisateur ou à travers le Workload Management System (définition de WMS ci-dessous) qui soumet un job à un CE approprié en fonction des spécificités de ce job.

Certificat et proxy : Afin de s'authentifier auprès des ressources de la grille, un utilisateur doit avoir un certificat numérique délivré par une autorité de certification. Le certificat utilisateur, dont la clé privée est protégée par un mot de passe, est utilisé pour générer et signer un certificat temporaire, appelé proxy, qui est utilisé pour l'authentification auprès des services de la grille. Parce qu'un proxy est une preuve d'identité, le fichier qui le contient doit être lisible uniquement par l'utilisateur, et un proxy a, par défaut, une durée de vie courte (généralement 12 heures) pour des raisons de sécurité.

GUID, LFN, SURL et TURL : un fichier possède plusieurs identifiants sur la grille: son Grid Unique IDentifier (GUID), son Logical File Name (LFN), mais aussi les Storage URL (SURL) et Transport URL (TURL). Alors que les GUIDs et LFNs identifient un fichier indépendamment de son emplacement, les SURLs et TURLs contiennent des informations sur la réplique physique ou le moyen d'accès. Un fichier sur la grille peut être identifié sans ambiguïté par son GUID qui lui est attribué la première fois que le fichier est enregistré sur la grille. Afin de localiser un fichier dans la grille, un utilisateur utilise

normalement son LFN. Les LFN sont généralement plus intuitifs et lisibles car ils sont attribués par l'utilisateur.

Job : Un job est une unité de travail, un programme avec des paramètres et des entrées. Soumis par un utilisateur, il s'exécute sur les ressources de la grille pour produire des résultats de sorties. Une application de grille peut se résumer à un seul job, ou requérir l'exécution de plusieurs jobs similaires (c'est-à-dire un même programme qui est exécuté plusieurs fois avec des paramètres ou entrées différents) ou impliquer un enchaînement (flot) de jobs différents qui s'exécutent sur la grille.

Job pilote : Un job pilote est un job générique soumis par un utilisateur sur la grille dont le rôle est de réserver une ressource de la grille qui sera utilisée ultérieurement par un programme réel (dans cette thèse, on l'appelle une tâche). Un job pilote peut être utilisé pour lancer plusieurs programmes.

Job Description Language (JDL) : le langage de description de job permet de décrire un job avant de le soumettre. Il permet de préciser par exemple l'exécutable à lancer et ses paramètres, les fichiers à déplacer vers et à partir du WN (Worker Node, définition ci-dessous) sur lequel le job est exécuté, les fichiers d'entrée nécessaires sur la grille, et toutes les autres exigences du CE et du WN. Par exemple, un fichier JDL pour spécifier le programme à exécuter et les sorties se présente delà façon suivante :

```
Executable = "/bin/hostname";
StdOutput = "std.out";
StdError = "std.err";
```

Storage Element (SE): un élément de stockage fournit un accès uniforme à des ressources de stockage de données. L'élément de stockage peut contrôler des serveurs de disques simples, des piles de disques de grande taille ou d'autres moyens de stockage. Le SE peut prendre en charge différents protocoles d'accès aux données.

Tâche : une tâche est une unité de travail, un programme avec des paramètres et des entrées qui s'exécute sur la grille. Ce qui diffère une tâche à un job est que la première n'est pas soumise directement par l'utilisateur mais est lancée par un job pilote. Une tâche est donc un travail qu'un job pilote peut exécuter dans son existence sur la grille. Le concept de tâche sera présenté en détail au chapitre 3 de cette thèse.

Worker Node (WN) : c'est la machine où les jobs sont exécutés. Un Computing Element s'appuie sur une collection de WNs pour exécuter les jobs.

Workload Management System (WMS) : Le but du système de gestion de charge est d'accepter des jobs utilisateurs, de les assigner au CE le plus approprié, d'enregistrer leur statut et de récupérer leur résultat. L'utilisateur peut soumettre un job à un CE directement mais en général, il laisse le WMS choisir le CE le plus approprié à partir de la description de ce job.

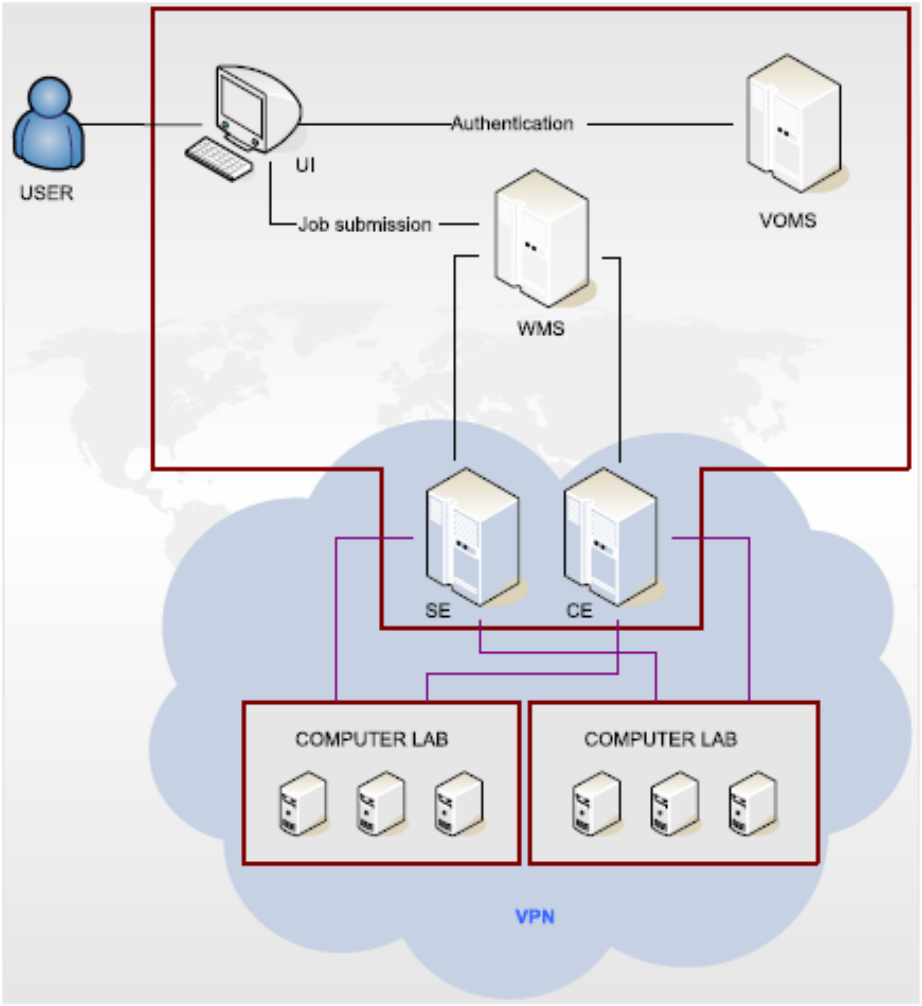


Figure 2-1 – Illustration de la soumission des jobs sur la grille (http://www.ac.usc.es/escience/gridComputing)

Virtual Organisation (VO) : une organisation virtuelle se réfère à un ensemble dynamique d'individus ou d'institutions définis autour d'un ensemble de règles et de conditions de partage de ressources. L'appartenance à une VO accorde des privilèges spécifiques à un utilisateur dans l'accès aux ressources. Pour devenir un membre d'une VO, un utilisateur doit se conformer aux règles de la VO.

Virtual Organisation Membership Service (VOMS) : VOMS est un système de gestion des données d'autorisation d'accès aux ressources de la grille. VOMS fournit une base de données de rôles et de fonctions d'utilisateur et un ensemble d'outils pour l'accès et la manipulation. VOMS utilise le contenu de cette base de données pour générer des certificats de grille pour les utilisateurs lorsque cela est nécessaire.

2.1.2. Glossaire bioinformatique

Alignement : L'alignement de séquences est un moyen d'arranger les séquences pour identifier les régions de similitude qui peuvent être une conséquence des relations fonctionnelles, structurelles ou évolutives entre les séquences. Les séquences alignées de résidus d'acides aminés ou de nucléotides sont généralement représentées par des lignes au sein d'une matrice. Des écarts sont insérés entre les résidus de sorte que les caractères identiques ou similaires sont alignés en colonnes successives.

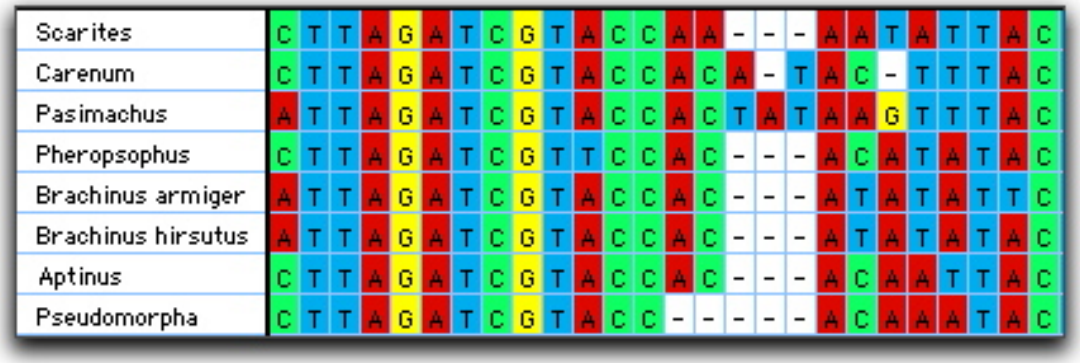


Figure 2-2 - Exemple d'un alignement de séquences de plusieurs organismes (www.sequence-alignment.com/)

Métagénomique : la métagénomique est un procédé méthodologique qui vise à étudier le contenu génétique d'un échantillon issu d'un environnement complexe (intestin, océan, sols, etc.) trouvé dans la nature (par opposition à des échantillons cultivés en laboratoire). La metagénomique étudie les organismes microbiens directement dans leur environnement sans passer par une étape de culture en laboratoire.

Paire de base (pb) : en biologie moléculaire et en génétique, une paire de base est l'appariement de deux bases azotées situées sur deux brins complémentaires d'ADN ou ARN. La taille d'un gène particulier ou du génome entier d'un organisme est souvent mesurée en paires de bases. Par conséquent, le nombre de paires de bases total est égal au nombre de nucléotides dans l'un des brins.

Phylogénie et arbre phylogénétique : la *phylogénie* est l'étude des relations de parentés entre différents êtres vivants en vue de comprendre l'évolution des organismes vivants. L'idée principale est de comparer les caractéristiques spécifiques de l'espèce en vertu de l'hypothèse que les espèces similaires sont génétiquement proches. La phylogénie a pour objectif de classifier les êtres vivants pour rendre compte des degrés de parenté entre les espèces et comprendre leur histoire évolutive. Le résultat d'une classification est représenté sous forme d'arbre. Un **arbre phylogénétique** est un arbre schématique qui montre les relations de parentés entre des groupes d'êtres vivants. Chacun des nœuds de l'arbre représente l'ancêtre commun de ses descendants.

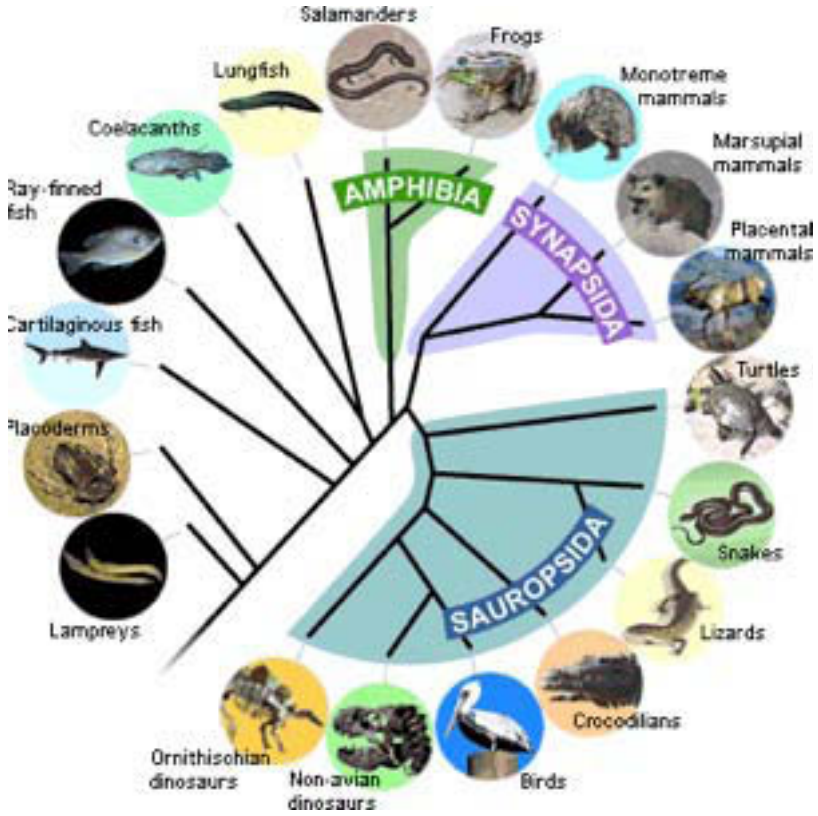


Figure 2-3 - Exemple d'un arbre phylogénétique (http://evolution.berkeley.edu/evolibrary/article/evo_08)

Séquence : Une séquence biologique est généralement une chaîne de textes codés pour représenter la structure primaire d'une macromolécule biologique. Dans le cadre de cette thèse, on parle des séquences nucléotidiques ou protéiques et des séquences ribonucléiques. Un exemple de séquence nucléotidique est le suivant :

```
ttcagttgtg aatgaatgga cgtgccaaat agacgtgccg ccgccgctcg attcgcactt
```

Séquençage de Nouvelle Génération ou Séquençage à Haut Débit (Next-Generation Sequencing / **H**igh **T**hroughput **S**equencing - HTS) : le séquence de nouvelle génération ou séquençage à haut débit est une technologie permettant de paralléliser le processus de séquençage, produisant des milliers ou des millions de séquences à la fois. Elle offre ainsi aux chercheurs et aux scientifiques un outil révolutionnaire pour obtenir des séquences rapidement et à moindre coût.

2.2. Présentation des grilles actuelles

L'écosystème des grilles est aujourd'hui d'une grande complexité. Nous allons présenter dans ce chapitre les infrastructures qui ont été directement utilisées pour le travail de la thèse, à savoir l'infrastructure internationale EGI et les infrastructures nationales en France et au Vietnam.

2.2.1. EGI

L'initiative de grille européenne EGI (European Grid Infrastructure) a pris la suite des projets européens DataGrid (http://eu-datagrid.web.cern.ch/eu-datagrid/) et EGEE (Enabling Grids for E-SciencE, http://www.eu-egee.org/). Dès septembre 2007, le projet EGI_DS (European Grid Initiative Design Study, http://web.eu-egi.eu/) a défini une feuille de route vers une e-infrastructure pour l'Europe construite sur des Initiatives de Grilles Nationales (National Grid Initiatives ou NGIs en anglais). Les NGIs sont des organisations mises en place pour coordonner les efforts au niveau national dans l'opération et l'utilisation de ressources informatiques de grille et de cloud distribué. Elles constituent des points de contact et des structures d'animation pour les communautés de recherche et les centres de ressources. L'Initiative de Grille Européenne EGI fédère les efforts des Initiatives de Grilles Nationales pour fournir une infrastructure globale ouverte à toutes les disciplines.

Les principaux services offerts par EGI à travers les NGIs sont les suivants:

- Services d´opération et de sécurité : opération d'une infrastructure de calcul et de données dans le monde entier, fiable et hautement disponible 24 heures sur 24, 7 jours sur 7
- Services de vérification et de test des intergiciels
- Services aux utilisateurs: support aux utilisateurs, formations, etc.
- Services de liaisons : connexion avec les autres infrastructures non-européennes

Selon le rapport annuel de l´EGI pour l'année 2012 (https://documents.egi.eu/document/1059), le nombre total de cœurs fournis par EGI a atteint 270.800, tandis que le nombre total de ressources de calcul des infrastructures intégrées dans EGI est de 399.300, en augmentation de 30,7% depuis l'année dernière. La capacité totale de stockage est de 139 Pétaoctets, en augmentation de 31,4% depuis l'an dernier.

La gestion des ressources de l'infrastructure de EGI est faite par plusieurs intergiciels de grille. Les intergiciels constituent le liant qui tient ensemble les éléments de la grille. Ils fournissent des services généraux pour l'ensemble de l'infrastructure : gestion de la charge, gestion de données, authentification, surveillance, etc. Les intergiciels déployés sur EGI doivent respecter le cahier des charges de la Distribution d'Intergiciel Unifiée (Unified Middleware Distribution). Quatre intergiciels sont aujourd'hui déployés sur les infrastructures d'EGI : il s'agit de gLite [5], UNICORE [6], Globus Toolkit [7] et ARC [8]. Tous les quatre reposent sur le Globus Toolkit.

En résumé, EGI fournit une plate-forme de calcul et de stockage de données pour les communautés scientifiques en Europe. EGI est un environnement où les utilisateurs peuvent se rencontrer et travailler ensemble dans le cadre de collaborations internationales. Aujourd'hui EGI est une infrastructure de ressources en expansion et elle continue à évoluer pour inclure de nouveaux

types de ressources et notamment des clouds académiques.

2.2.2. France Grilles

France Grilles (http://www.france-grilles.fr/) Initiative de Grille Nationale (National Grid Initiative) française est organisée depuis Juin 2010 sous la forme d'un Groupe d'Intérêt Scientifique de 8 organismes majeurs de recherche :

- Ministère de l'Enseignement Supérieur et de la Recherche (MESR)
- CNRS représenté par l'IdGC
- Commissariat à l'énergie atomique et aux énergies alternatives (CEA)
- Institut National de la Recherche Agronomique (INRA)
- Institut National de Recherche en Informatique et en Automatique (INRIA)
- Institut National de la Santé et de la Recherche Médicale (INSERM)
- Conférence des Présidents d'Université (CPU)
- REseau NAtional de télécommunications pour la Technologie, l'Enseignement et la Recherche (RENATER)

Les missions de France Grilles sont les suivantes :

- Établir et opérer une infrastructure nationale de grille de production, pour le traitement et le stockage de grandes masses de données scientifiques.
- Contribuer avec les autres états membre impliqués au fonctionnement de l'infrastructure européenne EGI.
- Favoriser les rapprochements et les échanges entre les équipes travaillant sur les grilles de production et les grilles de recherche.

Ces missions s'étendent aujourd'hui au domaine du cloud computing.

France Grilles intègre plus de 22.000 processeurs et 15 pétaoctets de stockage répartis dans 23 sites à travers le territoire français pour les analyses de données à haut débit. Ces ressources sont mises à la disposition des utilisateurs français – il y a aujourd'hui environ 850 titulaires de certificats de grille en France - mais aussi des chercheurs à travers le monde via des organisations virtuelles internationales. France Grilles est un des plus gros contributeurs à EGI (Figure 2-4)

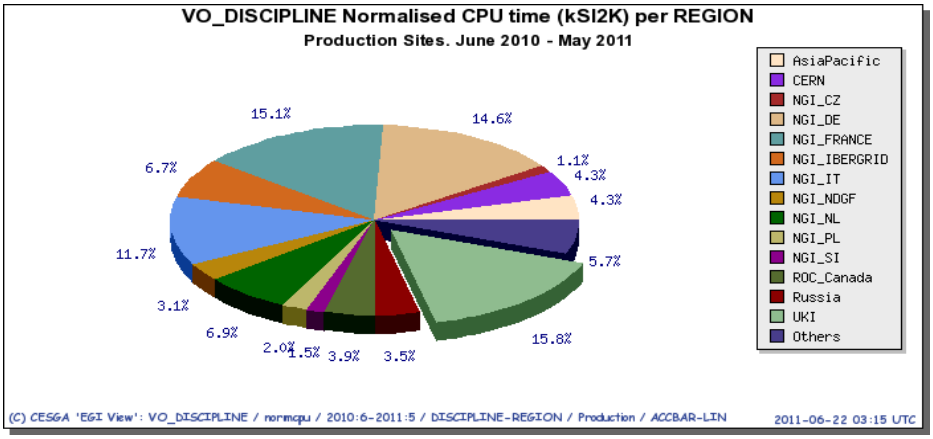


Figure 2-4 - Ressources de calcul fournis aux EGI par les différentes initiatives de grille nationales

France Grilles se concentre sur la réponse aux besoins en croissance exponentielle de nombreuses disciplines scientifiques pour les ressources informatiques. Les communautés de recherche les plus actives en France sur la grille sont les suivantes:

- L'astronomie, l'astrophysique et les astroparticules, principalement pour la simulation Monte-Carlo de nouveaux détecteurs comme CTA (Cerenkov Telescope Array), l'analyse des données issues d'instruments comme AUGER, GLAST ou LSST dans le futur, ou pour la modélisation de systèmes complexes
- La physique des particules plus gros consommateur de ressources pour l'analyse des données du LHC mais aussi des autres grandes expériences internationales, comme l'expérience D0 à FERMILAB
- Les sciences de la Planète, notamment la sismologie, les sciences de l'atmosphère, l'hydrologie, l'hydrodynamique, l'exploration géophysique et l'étude du climat
- Les sciences du Vivant aujourd'hui très actives sur la grille notamment dans les domaines suivants :
    - la bioinformatique autour du Réseau National des plateformes de Bioinformatique (RENABI), avec des infrastructures dédiées comme Décrypthon sur lesquelles nous reviendrons dans ce chapitre
    - l'imagerie médicale, au sein de la Life Science Grid Community (http://www.lsgc.org)
    - les neurosciences, à travers des projets nationaux et européens

Des communautés plus réduites sont très actives dans le domaine des systèmes complexes ou de la chimie tandis que l'observatoire de la grille (http://www.grid-observatory.org) se propose de collecter et d'analyser les données sur le comportement de la grille pour les chercheurs en informatique.

La croissance très rapide des besoins de traitement des données dans de

nombreux domaines scientifiques amène de nouvelles communautés à frapper à la porte de la grille. C'est le cas par exemple de la recherche financière mais aussi de l'étude de la biodiversité sur laquelle nous reviendrons dans ce chapitre.

2.2.3. Grilles au Vietnam

2.2.3.1. Les acteurs

Le calcul haute performance au Vietnam se concentre dans des centres de calcul dispersés à travers le pays. Ces centres de taille modeste (de quelques dizaines à quelques centaines de cœurs) sont installés dans les universités, les instituts et les organisations et fonctionnent de manière indépendante, pratiquement sans coordination, ni partage des ressources et des services.

On peut citer le Centre de Calcul de Haute Performance de l'Université Polytechnique de Hanoi (HPC-HUT), le Centre de Prévision Hydro-météorologique Nationale, le Centre de Calculs de l'Université des Sciences Naturelles - Université Nationale de Hanoi ou l'Université Nationale de Ho Chi Minh-Ville. L'Académie des Sciences et Technologies du Vietnam héberge aussi des ressources informatiques à l'Institut de Mathématiques, l'Institut de Technologie de l'Information et l'Institut Militaire de Science et Technique.

L'Institut de la Francophonie pour l'Informatique (IFI) est un institut international en informatique créé par l'Agence universitaire de la Francophonie (AUF) en 1995 pour répondre à une demande vietnamienne en formation de cadres supérieurs en informatique et pour aider à l'émergence de professeurs d'informatique vietnamiens de calibre international au niveau universitaire.

Pour le Vietnam, l'IFI représente une aide au développement et à la formation de formateurs et une passerelle vers l'acquisition, l'utilisation et le développement local des techniques avancées d'information et de communication. C'est dans ce contexte que l'IFI s'est intéressé à partir de 2007 à la technologie des grilles de calculs.

2.2.3.2. Histoire de la grille au Vietnam

La première initiative de grille au Vietnam est venue des Etats-Unis à travers le projet PRAGMA (Pacific Rim Application and Grid Middleware Assembly - http://www.pragma-grid.net/) dont l'objectif est depuis 2002 d´établir des collaborations et de promouvoir l'utilisation des technologies de grilles entre les communautés de chercheurs des grandes institutions à travers le Pacifique.

La collaboration entre la France et le Vietnam sur la grille a démarré à l'automne 2007 où une première école, baptisée ACGRID, est organisée dans les locaux de l'Institut des Technologies de l'Informatique (Institute Of Information Technology, IOIT) de l'Académie des Sciences et Technologies du Vietnam dans le cadre de la série des écoles Do-Son. Les objectifs de cette école étaient très ambitieux :

- former des administrateurs systèmes vietnamiens à la gestion des

services de grille
- démarrer 5 sites de la grille EGEE au Vietnam grâce à du matériel acheté par le CNRS pour la formation
- former des utilisateurs de ces sites pour démarrer le déploiement d'applications scientifiques

L'école n'a pas atteint tous ses objectifs mais 3 sites à Hanoï (Centre de Calcul de Haute Performance de l'Université Polytechnique de Hanoi, Institut de la Francophonie pour l'Informatique et Institut des Technologies de l'Information de la VAST) ont pu effectivement rejoindre la grille dans les mois suivants.

Mais rapidement, l'installation des sites s'est heurtée à des problèmes d'infrastructure réseau. En effet, toutes les connexions entre les sites académiques au Vietnam et en Europe utilisent le réseau TEIN2, projet international financé par la Commission Européenne avec les objectifs suivants:

- Augmenter la connectivité à Internet pour la collaboration dans les domaines de la recherche et l'éducation entre l'Europe et l'Asie
- Améliorer la connectivité intra-régionale en Asie
- Agir comme un catalyseur pour le développement des réseaux de recherche nationaux dans les pays en développement dans la région Asie-Pacifique.

Bien que TEIN2 offre une infrastructure de connexion à haute vitesse qui satisfasse aux conditions requises pour des sites de grille, en pratique, le processus de déploiement des nœuds de la grille au Vietnam s'est heurté à de multiples problèmes d'accès à TEIN2, d'instabilité de la connexion, de capacités limitées de gestion, etc.

Dans ce contexte difficile, le projet d'Initiative de Grille Nationale VNGrid a été lancé en 2008 avec pour objectif de connecter et de partager les ressources entre les organisations participantes. A cause des difficultés rencontrées, notamment au niveau de l'infrastructure réseau, ce projet n'a pas réussi à poser les fondations d'une infrastructure nationale exploitable et durable. Il a par contre produit des résultats académiques et quelques publications dans le domaine de la recherche et du développement sur cluster de calcul et sur grille.

Malgré ces difficultés, en 2009, l'Institut de la Francophonie pour l'Informatique (IFI) a eu l´opportunité de participer au projet européen EuAsiaGrid (http://www.euasiagrid.org/). Dans ce cadre, l'IFI a développé notamment le portail bioinformatique g-INFO présenté dans ce manuscrit.

L'IFI a accueilli la deuxième école ACGRID à l'automne 2009 et joué un rôle moteur dans l'animation de l'activité scientifique sur la grille et le cloud depuis lors.

D´autres projets de recherche et de développement basés sur la grille et le cloud sont en cours entre l´IFI et les partenaires de la VAST comme le projet sur le criblage virtuel des composants naturels issus de la biodiversité avec l´Institut

de Chimie des Produits Naturels (INPC) de la VAST ou celui pour la surveillance et prédiction de tsunami dans la mer de l´Est avec l´Institut Physique du Globe (IGP) et l´IOIT.

Les troisième et quatrième écoles ACGRID ont été consacrées à la grille et au cloud computing. Ce nouveau paradigme de calcul est décrit dans le paragraphe suivant.

2.2.4. Des grilles au cloud computing

Récemment, l'informatique en nuage, encore appelé communément le Cloud Computing, a généré l'intérêt intense des média en réponse aux efforts de marketing des grandes marques telles que Amazon et Google. Des recherches récentes de IDC (International Data Corporation[1]) montrent que les recettes publiques dans le monde de services du cloud computing ont dépassé 21,5 milliards de dollars en 2010 et atteindront 72,9 milliards de dollars en 2015.

Bien qu'il y ait un certain flou sur la définition précise, la plupart s'accorde pour définir le Cloud Computing comme un système où les ressources sont exposées comme des services. L'idée de fournir les technologies de l'information comme un service (IT-as-a-service) offre des avantages énormes aux clients de l'entreprise qui veulent une puissance de calcul importante sans devoir acheter des ordinateurs coûteux. La capacité à augmenter ou à diminuer la puissance de calcul à la demande en payant seulement la puissance consommée est particulièrement intéressante. Malgré tous les discours depuis des années autour du Cloud Computing et malgré les avantages qu'elle offre, la technologie n'est pas arrivée toute de suite. En effet, elle est le résultat d'un processus évolutif qui a commencé il y a 20 ans et dont la grille de calcul a constituée certainement une étape, et il y a encore du chemin à parcourir avant que les promesses du Cloud soient tenues.

Si le point commun le plus notable entre la grille et le cloud est la distribution des services, l'idée principale de la grille est le partage des services et celle du cloud est la commercialisation des services. La grille de calcul est née dans les années 1990 avec l´idée de rendre l'accès aux ordinateurs aussi facile que l'accès au réseau électrique. Cette même idée a également contribué au Cloud Computing. Le terme «cloud computing» a été d'abord utilisé par le professeur Chellappa Ramnath en 1997. En 1999, Salesforce[2] a commencé à offrir des applications aux utilisateurs en utilisant un simple site Web. Les applications ont été livrées aux entreprises sur l'Internet, et de cette façon l'idée de vendre les services informatiques via internet est devenue réalité. Bien que le service ait été un succès, il faut attendre plusieurs années avant que le Cloud Computing ne se généralise. En 2002, Amazon a commencé Amazon Web

---

[1] International Data Corporation est une entreprise d'études de marché et d'analyse spécialisée dans les technologies de l'information, des télécommunications et des consommateurs de la technologie.

[2] www.salesforce.com

Services qui fournit des services de stockage et de calcul. C'est en 2006 qu'Amazon lance l'Elastic Cloud Computer (http://aws.amazon.com/ec2/), un vrai service commercial ouvert à tous.

Une étape importante dans l'évolution du Cloud Computing est en 2009 le démarrage des services de cloud de Google.

Le Cloud Computing se situe à la convergence des grilles, de la virtualisation et de l'automatisation. Bien que la fiabilité et la sécurité des clouds pose encore question, de nombreuses entreprises ont pris les premières mesures en vue de son adoption, soit en utilisant un Software-as-a-Service (SaaS), soit en louant des cycles de calcul à des fournisseurs. Une autre approche dans les grands groupes industriels est le déploiement de clouds internes privés. À une époque où faire plus avec moins est plus important que jamais, le Cloud Computing a un bel avenir.

2.3. Présentation des infrastructures utilisées pour la bioinformatique

L'objet de cette thèse est le déploiement d'applications bioinformatiques sur grille de calculs. Nous allons donc introduire dans cette section les infrastructures de la bioinformatique sur grille en commençant par les plus virtuelles (Life Science Grid Communty, Organisation Virtuelle Biomed) pour aller vers les infrastructures physiques (Décrypthon, ReNaBi, ELIXIR), d'une envergure nationale (ReNaBi) à une envergure européenne (ELIXIR). Les travaux décrits dans ce manuscrit ont été déployés principalement sur l'Organisation Virtuelle Biomed de l'Initiative de Grille Européenne EGI.

2.3.1. VO Biomed et Life Sciences VRC

Pour accéder aux ressources de EGI, les communautés scientifiques utilisatrices utilisent des Organisations Virtuelles (VO) qui sont des ensembles dynamiques d'individus ou d'institutions définis autour d'un ensemble de règles et de conditions de partage de ressources.

La VO Biomed est la plus grande organisation virtuelle à l´échelle internationale pour les communautés en Sciences de la Vie. Tout chercheur en sciences de la vie muni d´un certificat d´authentification délivré par une autorité de certification accédée au niveau international peut rejoindre cette VO. Les 300 utilisateurs de la VO Biomed viennent aujourd'hui de plus de 20 pays différents. Leurs recherches se concentrent dans trois domaines principaux : la bioinformatique, la recherche de nouveaux médicaments et l'imagerie médicale. La VO Biomed donne accès à des ressources dans de nombreux sites différents, offrant un accès à des dizaines de milliers de cœurs de processeurs à ses utilisateurs, ce qui permet de développer des applications très exigeantes. Elle constitue aussi un vecteur de collaboration internationale dans le domaine des sciences de la vie par le partage de ressources, de moyens et de données.

Le support aux utilisateurs et la gestion du fonctionnement de la VO Biomed sont faits en groupe, sur la base du volontariat. Par exemple, le groupe de

volontaires « Biomed Shifter », que l'IFI a rejoint en 2011, a pour but de surveiller l'état des sites fournissant des ressources afin d'envoyer des alertes à l'administrateur du site pour améliorer l'utilisation des ressources et optimiser l'efficacité.

Malgré son caractère très ouvert, Biomed n'est pas la seule Organisation Virtuelle en sciences de la vie sur EGI. Plusieurs autres organisations virtuelles proposent des ressources à des communautés nationales, la VO RENABI par exemple, ou aux partenaires de projets européens come la VO WeNMR en biologie structurale ou NeuGrid en neurosciences.

Pour favoriser le développement de collaboration entre ces communautés d'utilisateurs virtuels et accroître leur visibilité et améliorer leur représentation au sein d'EGI, la Communauté de Recherche Virtuelle (Virtual Research Community ou VRC en anglais) des sciences de la vie a été créée. Les VRCs sont des communautés de recherche auto-organisés, qui confient à quelques membres au sein de leur communauté un mandat clair pour représenter les intérêts de leur domaine de recherche au sein d'EGI. Ils peuvent inclure une ou plusieurs organisations virtuelles et agissent comme le principal canal de communication entre les chercheurs qu'ils représentent et EGI.

La Communauté Virtuelle de Recherche des Sciences de la Vie (Life Science Grid Community - LSGC) rassemble les domaines scientifiques suivants: la bioinformatique, la biologie moléculaire, les bio-banques, l'imagerie médicale, la biologie structurale et les neurosciences. L'initiative de grille européenne EGI établit des partenariats avec les VRCs via une convention (Memorandum of Understanding - MoU). Après le processus d'accréditation et l'accord final, les représentants des VRCs deviennent les interlocuteurs d'EGI dans leur domaine scientifique et profitent de nombreux avantages d'un partenariat solide avec EGI. Ils représentent leur discipline pour négocier des ressources, assurer la liaison avec les Initiatives de Grille Nationales et d'autres fournisseurs de ressources à travers le monde. EGI propose des workshops et des forums pour aider et supporter les communautés sur les problèmes techniques spécifiques, et afin aussi de les impliquée dans l'évolution de l'infrastructure de production d'EGI. Les représentants des VRCs expriment à EGI le cahier des charges de leurs exigences techniques et de service qui sont prises en compte dans le développement global de l'infrastructure. Vis-à-vis des communautés qu'ils représentent, ils servent de point de contact pour les nouveaux utilisateurs et apportent de l'aide pour partager les expertises, éviter la réplication des efforts et encourager le partage des ressources, des données et des outils. Ils organisent des événements de formation et fournissent des services techniques: opérer et supporter les VOs communs, opérer des services partagés, etc. Ils contribuent enfin à diffuser les informations et les connaissances afin de faciliter la communication interne et externe avec d'autres groupes d'intérêt.

Une Communauté de Recherche Virtuelle dans le domaine de la biodiversité est en cours d'émergence autour des projets européens Creative-B et ENVRI et de l'Infrastructure de Recherche Européenne LifeWatch.

2.3.2. Décrypthon

L'infrastructure Décrypthon a commencé en 2001, puis changé de nom en Programme Décrypthon en 2004. Ce projet est une collaboration entre le CNRS, l'Association Française contre les Myopathies (AFM) et la société IBM. Son objectif est de fournir aux équipes de recherche en bioinformatique des ressources de calcul et de stockage, notamment aider la recherche sur les maladies neuromusculaires et les maladies rares pour la plupart d'origine génétique.

La grille Décrypthon fédère des ressources installées par IBM dans 6 universités en France et reliées par le réseau RENATER [9]. La grille Décrypthon version I a été réalisée à partir de la solution GridMP [10]. Dans la version II, Décrypthon utilise l'intergiciel DIET (voir section 2.4.3.1). Figure 2-5 illustre la grille Décrypthon version II [11].

La grille Décrypthon se compose des composants suivants :

- L'intergiciel DIET qui gère des soumissions des jobs ainsi que la migration des données nécessaires pour les calculs et le stockage des résultats.
- Le portail Web qui contient une application web spécifique à chaque projet bioinformatique pour aider la soumission des calculs via l'intergiciel DIET
- Le gestionnaire local de ressources (*batch scheduler*) installé sur chaque nœud de la grille Décrypthon. Il assure la répartition et le déroulement des calculs sur les machines locales.

Plusieurs projets utilisent les ressources du Décrypthon :

- MS2PH : Mutations structurales avec les conséquences sur le phénotype des pathologies humaines.
- Help Cure Muscular Dystrophy
- SpikeOmatic : tri de potentiel d'action
- GEE : Expression de l'évolution des gènes

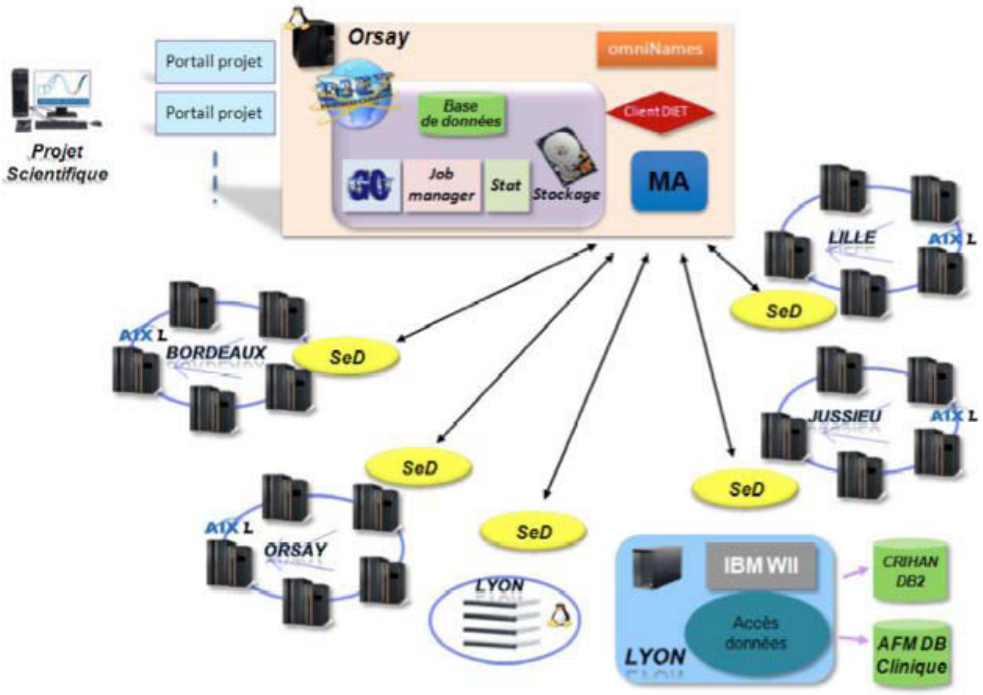


Figure 2-5 - Architecture de la grille Décrypthon version II ([11])

En conclusion, l'infrastructure Décrypthon accueille des projets scientifiques sur une infrastructure de grille dédiée dont la limitation principale est la ressource disponible et le vieillissement du matériel. La combinaison avec la grille d'internautes World Community Grid (http://www.worldcommunitygrid.org/) permet des déploiements à très grande échelle. Et l'addition d'un moteur de flot pour décrire l´enchainement de tâches par un graphe dirigé et orienté va beaucoup enrichir la flexibilité pour les utilisateurs.

2.3.3. ReNaBi / GRISBI

ReNaBi (**Ré**seau **Na**tional des Plates-formes **Bi**oinformatiques - http://www.renabi.fr/) est une infrastructure s'appuyant sur les plates-formes bioinformatiques des établissements français dans différentes villes et régions de France. L'infrastructure ReNaBi a pour objectif de favoriser la coordination de l'activité des plateformes pour mieux répondre aux besoins des équipes de recherche en biologie à l'échelle nationale. ReNaBi soutient la structuration de la bioinformatique en France à travers ses activités. Aujourd'hui, ReNaBi est structuré en 6 centres régionaux qui rassemblent 28 plates-formes dont GRISBI. Ces plates-formes offrent de nombreuses ressources pour les communautés bioinformatique et scientifiques des sciences de la vie. ReNaBi vise à promouvoir une forte coordination en bioinformatique en France et à agir comme un partenaire majeur de la bioinformatique en Europe. ReNaBi est membre du Réseau européen de biologie moléculaire, EMBnet (http://www.embnet.org/).

Les principaux objectifs de ReNaBi sont d'améliorer la coordination de toutes

les composantes de l'infrastructure nationale de bioinformatique:

- Partager les connaissances et les développements entre les plates-formes
- Promouvoir l'interopérabilité et standard de diffusion
- Optimiser et rationaliser l'utilisation des installations informatiques à l'échelle nationale
- Empêcher la duplication des efforts

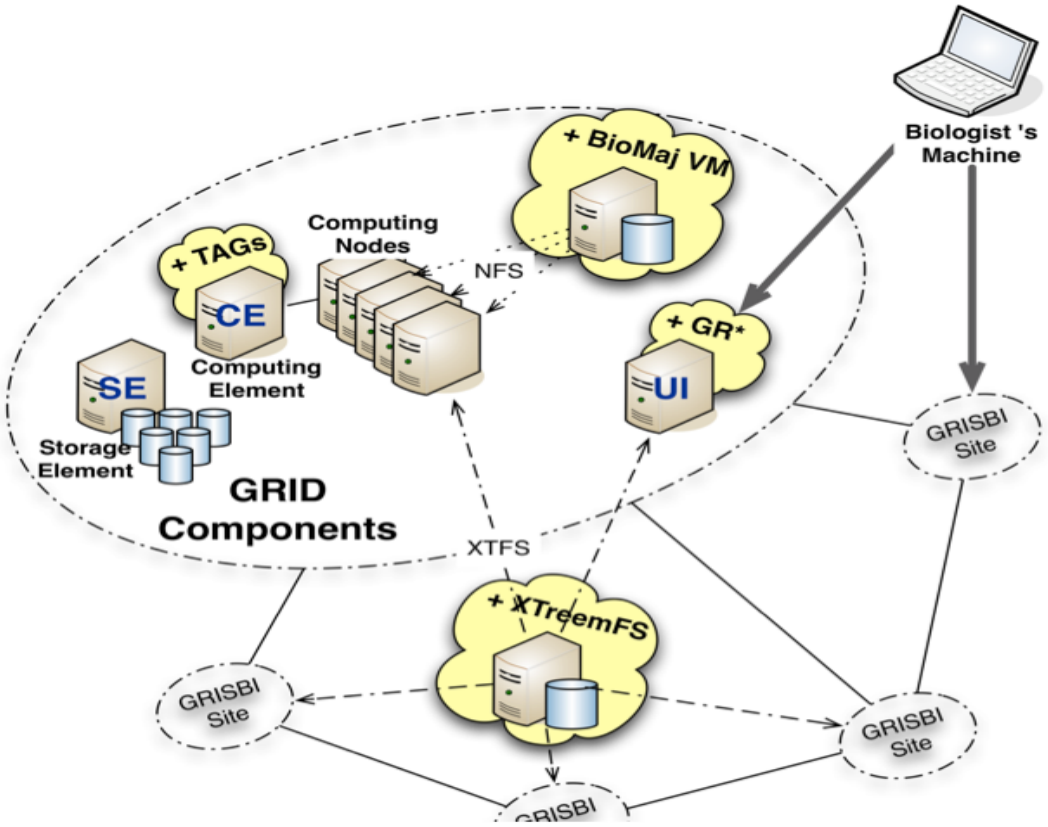


Figure 2-6 - L'infrastructure de GRISBI
(http://www.grisbio.fr/media/cms_page_media/28/grisbi_27mai_session1.pdf)

ReNaBi offre des services suivants :

- Services liés aux bases de données :
  - Distribuer des mises à jour des principales bases de données de biologie moléculaire
  - Développer et maintenir des sources originales de données
  - Fournir des accès aux bases de données et des outils bioinformatiques pour les analyser
- Services de stockage et de calcul, par exemple, sur la grille
- Services de support aux utilisateurs et services de formation

Le site www.phylogeny.fr fait par exemple partie des sources de données originales de ReNaBi. Nous nous sommes inspirés de cet exemple pour le

portail g-INFO de surveillance des virus influenza décrit dans cette thèse (chapitre 4).

Dans le cadre du réseau ReNaBi, GRISBI (*Grid Support to Bioinformatics*) a été lancé pour supporter et répondre à des problèmes biologiques de grande échelle, en particulier les problèmes de Séquençage de Nouvelle Génération. GRISBI est une initiative conjointe entre six plates-formes bioinformatiques françaises qui fournit une infrastructure distribuée en bioinformatique. Cette infrastructure regroupe 6 centres de ReNaBi et 9 sites avec 7 instituts du CNRS, soit environ 1000 processeurs et 220 To de stockage.

L'infrastructure de GRISBI est illustrée dans la figure ci-dessus. GRISBI utilise 2 systèmes de stockage des fichiers :

- Le Système de Gestion de Données de gLite [5], un des intergiciels de l'Initiative de Grille Européenne EGI.
- XtreemFS [12] qui est un système de fichiers distribués et répliqués pour le cloud computing.

La plateforme GRISBI a des logiciels pré-installés sur les nœuds de la grille. Les logiciels disponibles sur GRISBI sont :

- ClustalW [13] pour faire l'alignement multiple des séquences,
- FastA [14] pour la recherche de similarité entre séquences de protéines,
- HMMER [15] pour aligner les séquences,
- PHYML [16] pour construire les arbre phylogénétiques, etc.

GRISBI gère aussi quelques banques de données biologiques, par exemple UNIPROT (http://www.uniprot.org/), UNIREF (http://www.ebi.ac.uk/uniref/) et la Brookhaven Protein Data Base [référence]. L'outil BioMAJ (http://biomaj.genouest.org/) déployé sur GRISBI permet de synchroniser et de mettre à jour les données sur chaque site. Pour simplifier l´utilisation, GRISBI fournit aux utilisateurs les commandes *gr**. Par exemple, la commande :

```
grsub jdlfile
```

permet de lancer le job `jdlfile` sur la VO renabi qui correspond aux commandes de gLite suivant :

```
voms-proxy-init –voms vo.renabi.fr
glite-wms-job-submit –a –o jobid jdlfile
```

Pour les services centraux de la grille, GRISBI collabore avec France Grilles en utilisant ses services.

L'article [17] présent les résultats du traitement de données NGS (New Generation Sequencing) en utilisant la plateforme GRISBI pour l'assemblage initial, dit de novo, d'un génome complet à partir des données issues de séquenceurs à haut débits. «*Le résultat démontre donc la faisabilité et l'apport considérable de l'utilisation d'une grille de calcul pour ces traitements. En effet, d'une part, les temps de calculs sont réduits par rapport à des serveurs*

locaux de capacité moyenne, mais il est également possible, de lancer, de n'importe où, plusieurs analyses simultanées dont un paramètre varie, comme pour le cas d'assemblages de novo ».

2.3.4. ELIXIR

ELIXIR (http://www.elixir-europe.org/) est une Infrastructure de Recherche Européenne (ESFRI) pour la gestion des informations de biologie moléculaire en Europe. Portée par un consortium de 32 membres incluant des grandes agences de financement et les instituts de recherche en bioinformatique, sa mission est de construire et d'exploiter une infrastructure durable pour les données de biologie moléculaire en Europe pour supporter la recherche en sciences de la vie mais aussi en médecine et dans les sciences de l'environnement, les bio-industries et la société. « ELIXIR réunit les principales organisations en sciences de la vie en Europe pour la gestion et la sauvegarde des quantités massives de données générées chaque jour par la recherche publique» dit Prof Janet Thornton, directeur de l'EMBL-European Bioinformatics Institute.

Les objectifs d'ELIXIR sont les suivants:

- Construire une infrastructure fiable distribuée pour fournir un accès à l'information biologique à travers toute l'Europe
- Fournir un financement durable pour la collection de données biologiques (génomes, séquences, structures, etc.) au niveau européen ou mondial
- Poursuivre un processus pour :
    - Le développement de nouvelles bases de données
    - Le développement de l'interopérabilité des outils bioinformatiques
    - L'élaboration de normes bioinformatiques
- Améliorer l'utilisation de l'information biologique dans la recherche universitaire, l'industrie pharmaceutique, la biotechnologie, l'agriculture et pour la protection de l'environnement

ELIXIR est construit sur les ressources de données et des services existants. Il suit un modèle « hub-and-nodes », avec un centre de gravité, hub unique situé à l'EMBL-EBI à Hinxton (Royaume-Uni), et un nombre croissant de nœuds (nodes) situés dans les centres d'excellence dans toute l'Europe. Dans ce modèle, un nœud devra être (ou être représenté par) une entité légale habilitée à gérer des moyens et capable de s'engager à offrir et à maintenir dans la durée un ou plusieurs éléments de l'infrastructure. Ces éléments pourraient être une ou des collections de données, des moyens de déployer ou permettre l'accès / l'exploitation de ces collections y compris par la définition de standard et l'offre de formation. Une telle structure distribuée permet de s'assurer que les données sont conservées en toute sécurité et facilement accessibles.

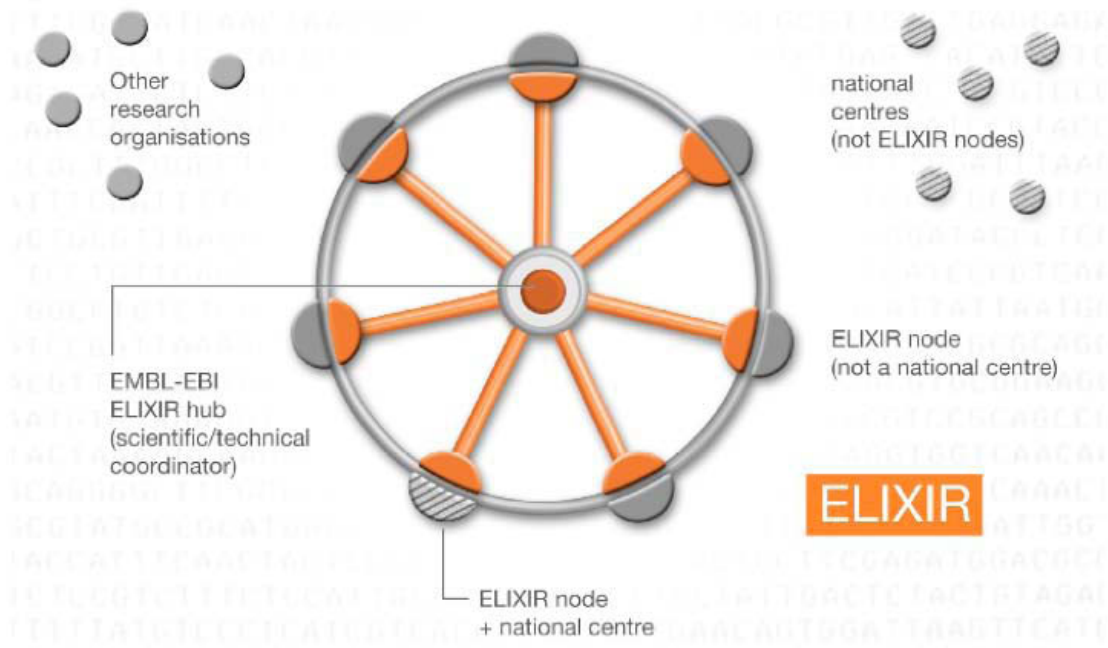


Figure 2-7 - Modèle hub-and-nodes de ELIXIR (http://www.elixir-europe.org/about/how-does-elixir-work)

Les composantes importantes dans l'infrastructure ELIXIR sont :

- Les collections de données biomoléculaires et les méta-données associées
- Les ressources de calcul (clusters ou grille)
- Les normes, standards et ontologies
- L'infrastructure de formation
- Les outils et les services d'intégration

Si l´on considère ReNaBi comme l'infrastructure support de la coopération en bioinformatique au niveau national, ELIXIR est l'infrastructure support de la coopération au niveau international, notamment en Europe. En résumé, l'infrastructure ELIXIR contribue à la science européenne par les actions suivantes:

- Optimiser l'accès et l'exploitation des données des sciences de la vie.
- Assurer la pérennité des données et la protection des investissements déjà réalisés dans la recherche.
- Augmenter la compétence et la taille de la communauté des utilisateurs des données de biologie moléculaire en renforçant les efforts nationaux en matière de formation.
- Améliorer l'influence de l'Europe dans la recherche en science de la vie au niveau mondial.

ELIXIR montre que la grille permet de construire une infrastructure

continentale pour un domaine autre que la physique des particules.

2.4. Portails bioinformatiques et outils de déploiement sur la grille

Après avoir présenté les différentes infrastructures disponibles ou en cours de structuration fournissant des ressources informatiques pour les sciences de la vie, nous allons présenter quelques portails bioinformatiques déployés sur la grille ainsi que les principaux outils utilisés par ces portails pour déployer les calculs sur la grille.

2.4.1. Portails et plate-formes bioinformatiques

Depuis une dizaine d'années, de nombreuses équipes à travers le monde ont étudié comment mettre les ressources de la grille à la disposition des biologistes et des bioinformaticiens. L'analyse des données de biologie moléculaire présente des spécificités :

- Ces données sont d'une très grande complexité. Il en existe de multiples représentations qui requièrent des traitements et donc des algorithmes spécifiques
- Les nouvelles données doivent être constamment comparées aux données déjà existantes.
- Les nouveaux résultats doivent être continuellement intégrés aux bases de données publiques existantes de biologie moléculaire
- De nouvelles bases de données voient continûment le jour à mesure que la compréhension avance et que de nouvelles représentations des données apparaissent.

Face à ces besoins spécifiques, portails et plates-formes bioinformatiques ont fleuri sur les infrastructures de grille dans le monde.

Dans cette section, nous nous limiterons à des exemples représentatifs et qui exploitent des ressources de grille en Europe, aux Etats-Unis et en Asie.

2.4.1.1. Bio-Grid

Le projet Bio-GRID [18] est l'un des premiers en Europe à avoir exploré l'utilisation de la grille pour la biologie moléculaire sur l'infrastructure fournie par le projet EUROGRID. Financé par la Commission européenne du 1er novembre 2000 au 31 janvier 2004, les objectifs de Bio-Grid étaient déjà de fournir un accès facile et uniforme aux systèmes de calcul à distance avec les fonctionnalités suivantes :

- Un portail pour les ressources de modélisation bio-moléculaires.
- Les interfaces permettant aux chimistes et biologistes de soumettre leurs jobs aux ressources de calcul.
- Les outils de visualisation de champ électrostatique généré par une molécule.

L'intergiciel utilisé par Bio-GRID était Unicore [6]. L'utilisateur accède à l'interface utilisateur graphique et à un client Unicore qui offre les fonctions de

préparation, de contrôle des jobs, de mise en place et de maintenance d´un environnement de sécurité de l'utilisateur. L'interface utilisateur graphique offre des fonctions pour maintenir la sécurité, afin de préparer, de soumettre et de surveiller les jobs. Le job peut être constitué de plusieurs parties exécutées de façon asynchrone ou dépendante sur différents systèmes à des sites différents. Bio-Grid utilise des Abstract Job Objects (AJO) de Unicore pour des logiciels de chimie computationelle très populaire comme Gaussian98, Gromos96 [19], Amber [20] et Charmm [21]. Ces plugins (Gaussian98, Gromos96, etc.) sont écrits en Java et sont chargés au client Unicore lors du démarrage, ou à la demande. Une fois qu'ils sont disponibles dans le client, l'utilisateur obtient l'accès au menu qui permet de préparer un job de l'application spécifique. L'utilisateur peut spécifier le système à exécuter et les ressources nécessaires pour les jobs en utilisant le client Unicore. Par exemple, l'utilisateur peut vérifier l'état d'un job, surveiller son exécution et récupérer la sortie sur une machine locale. Toutes ces fonctions peuvent être exécutées à partir du client Unicore.

La plate-forme Bio-GRID a été mise à la disposition des utilisateurs des institutions partenaires de EUROGRID.

2.4.1.2. LIBI

Le projet *International Laboratory of Bioinformatics* (Laboratorio Internazionale di BIoinformatica, LIBI, http://www.libi.it) a été lancé par le CNR italien dans le but de développer un environnement pour résoudre des problèmes bioinformatiques sur la grille. Construit sur les infrastructures de grille, il devait permettre la soumission et le suivi des jobs spécifiques aux expériences complexes en bioinformatique. LIBI a été pensé comme un laboratoire de bioinformatique "sans murs" avec les objectifs suivants :

- encourager les interactions entre les chercheurs dans les institutions publiques dans le domaine biomédical et biotechnologique.
- Promouvoir la recherche en bioinformatique au niveau (inter) national : maintenir et créer de nouvelles bases de données spécialisées, développer de nouveaux algorithmes, développer et mettre en œuvre des processus nouveaux et plus efficaces et des mécanismes pour l'analyse bioinformatique automatisée.
- S'appuyer sur une infrastructure de calcul haute performance distribuée pour l'accès à de vastes ensembles de données et une grande puissance de calcul.
- Promouvoir le transfert technologique, la formation et la provision de services bioinformatiques.

La plate-forme LIBI utilise un environnement de résolution de problèmes (Problem Solving Environment) [23] pour répondre aux exigences en bioinformatique. Elle fournit les fonctionnalités suivantes :

- L'accès à des outils partagés (par exemple, des outils d'affichage et

d'analyse de données) sous la forme de services simples ou composés.

- La distribution des données et leur accès transparent (la fédération des bases de données).
- Le développement d'un environnement collaboratif permettant aux utilisateurs de résoudre des problèmes biologiques complexes.
- La possibilité de personnaliser l'environnement de l'utilisateur et de gérer ou partager / échanger des données personnelles en respectant des politiques d'autorisation bien définies.

Figure 2-8 - L'architecture de la plate-forme LIBI
(http://www.cs.unibo.it/~margara/page14/assets/LIBIchapter.pdf)

L'architecture de la plate-forme LIBI est illustrée dans la Figure 2-8. Dans la couche supérieure, se trouvent des algorithmes qui peuvent être exécutés séparément ou sous forme d'une chaine de traitement (flot). Parmi les algorithmes disponibles se trouvent BLAST et PSI-BLAST [22], PatSearch [24], DNAFan [25], FT-COMAR [26]. Le service de *Resource Management* contient la logique d'exécution des applications et un *Meta Scheduler* qui met en œuvre les fonctionnalités comme la soumission et la surveillance des jobs, le transfert de fichiers, etc. Le *Meta Scheduler* est construit au-dessus des services de base offerts par des intergiciels de grille différents, tels que gLite [5], Unicore [6] et Globus Toolkit [7]. Les autres fonctionnalités de *Resource Management* sont notamment la planification et le suivi des demandes des utilisateurs ainsi que l'allocation optimale des ressources dans un environnement distribué. Le service *Data Management* intègre la logique de grille pour la gestion avancée des données, permettant un accès transparent et dynamique à des données hétérogènes et géographiquement distribués. Le composant *Security* s'occupe de la gestion et l'accès aux services et aux ressources au sein de l'environnement LIBI distribué. Il fournit des mécanismes pour l'authentification, l'autorisation et la protection des données.

Malgré la fin du projet LIBI en 2009, il continue de garantir un espace de travail virtuel pour les partenaires académiques et industriels. Même si ce projet ne comporte que des partenaires italiens, son utilisation est ouverte à

d'autres groupes de recherche internationaux afin de tester et donc d'améliorer les services offerts.

gHnHI HVH□□□d □

GPS@ (*Grid Protein Sequence Analysis*) [27] est un portail bioinformatique développé depuis une dizaine d'années par l'Institut de Biologie et Chimie des Protéines sur le portail NPS@ d'analyse protéomique [28]. GPS@ fournit aux biologistes une interface conviviale pour utiliser les ressources de calcul et de stockage de la grille et à terme du cloud. L'objectif de GPS@ est de gridifier les données et les algorithmes bioinformatiques pour l'analyse des séquences de protéines. L'architecture de GPS@ a été conçue sur la base des services fournis par l'infrastructure du projet européen EGEE (2004-2005) comme illustré dans la Figure 2-9.

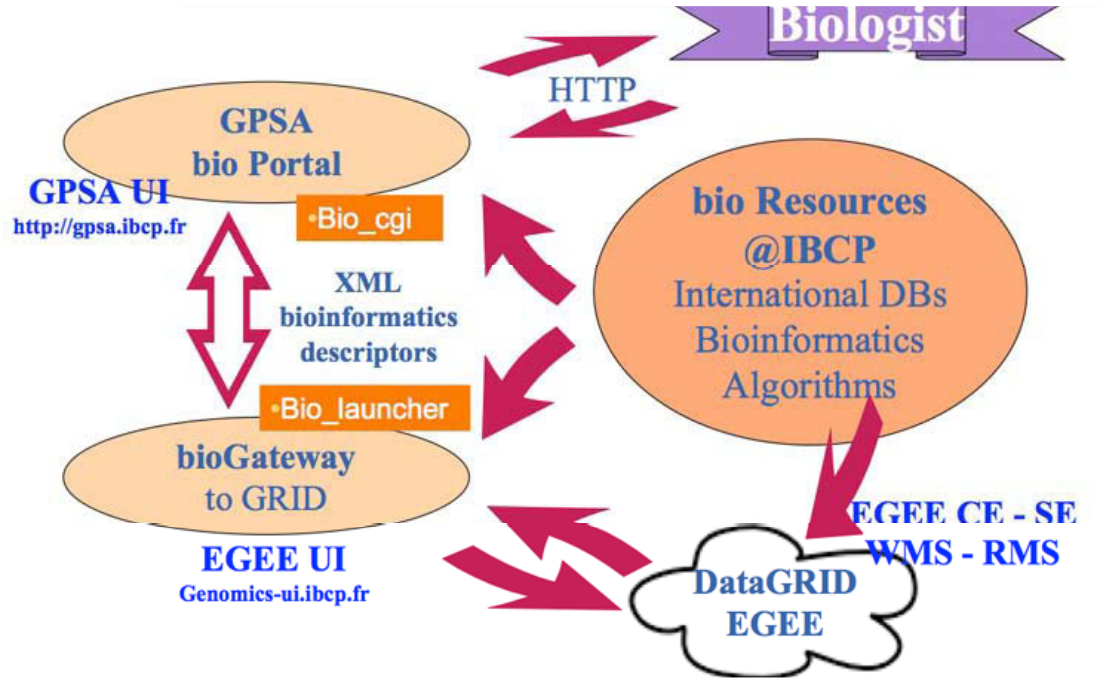


Figure 2-9 - L'architecture du GPS@ [http://egee.pnpi.nw.ru/presentation/06.03.01.egeeUF.GPSA.pdf]

GPS@ simplifie et automatise la soumission de jobs sur la grille EGEE par l´utilisation des mécanismes de gestion des données et du langage XML pour décrire les algorithmes bioinformatiques et les banques de données. GPS@ propose une grande variété d'algorithmes pour l'analyse de protéine :

- FastA [14], SSearch [29], BLAST [22] pour la recherche de similarité entre séquences de protéines
- ClustalW [13] pour faire l'alignement multiple des séquences
- PattInProt [28] pour chercher des motifs dans une séquence

GPS@ déploie quelques bases de données biologiques :

- SwissProt [30]: banque de séquences de protéines

- Prosite [31]: banque de domaines protéiques, de motifs et de profils

GPS@ propose 2 interfaces pour les utilisateurs : le portail et les services web. Le portail fournit aux biologistes une interface web où ils peuvent insérer les données d'entrée de protéines dans les champs correspondants de la page Web de soumission. Puis, l'exécution des jobs sur les ressources EGEE peut être lancé en cliquant simplement sur le bouton «Submit». Toute la gestion des jobs de EGEE est encapsulée dans le GPS@ backoffice: par exemple, la planification et le statut des jobs soumis. Enfin, les résultats des jobs soumis sur la grille sont affichés dans une nouvelle page. L'utilisateur peut aussi construire des pipelines d'analyse en utilisant les services web. Les moteurs de flot Triana [32] et Taverna [33] sont disponibles avec les services web de GPS@.

En résumé, GPS@ cache les mécanismes de l'exécution des analyses bioinformatiques sur l'infrastructure de grille. Pour permettre cette exécution, les algorithmes bioinformatiques et les banques de données sont préalablement distribués et enregistrés sur la grille. De cette façon, le portail de GPS@ permet aux biologistes d'utiliser l'infrastructure de grille de façon transparente pour analyser les grandes séries de données biologiques. Notre portail g-INFO est basé sur une approche similaire pour l'analyse des séquences de la grippe aviaire.

2.4.1.4. NC Bioportal

NC Bioportail (North Carolina University Bioinformatic portal) ou Bioportal [57] intègre des outils de domaines spécifiques et des outils communautaires standard pour fournir un environnement intégré et collaboratif. Le BioPortail a d'abord été déployé pour un usage communautaire en 2005. Depuis lors, il a été prolongé et amélioré avec de nouvelles applications, des pipelines de traitement intégrés et l'accès aux ressources de l'infrastructure TeraGrid (https://www.xsede.org). TeraGrid est une infrastructure nord-américaine qui a fourni de 2004 à 2011 jusqu'à un PetaFlops de capacité de calculs et 30 PBytes de stockage à plus de 4000 utilisateurs issus de plus de 200 universités. Le projet XSEDE (Extreme Science and Engineering Discovery Environment) a pris le relais du projet TeraGrid avec un financement de 121 Millions de dollars sur 5 ans. Les calculs en biologie moléculaire ont représenté environ 20% des ressources utilisées sur TeraGrid.

Le public ciblé par le Bioportal ne se limite pas aux chercheurs et aux enseignants à North Carolina University. L'objectif du portail est de répondre aux besoins de tous les utilisateurs de la bioinformatique à travers les États-Unis. Il a donc été conçu sur les objectifs suivants:

- Être facile d'utilisation à la fois par des enseignants, des étudiants et des chercheurs.
- Passer à l'échelle pour répondre aux besoins de stockage et d'analyse d'un volume exponentiellement croissant de données expérimentales

- Être extensible: le Bioportal doit permettre aux utilisateurs d'intégrer facilement de nouveaux algorithmes et de nouvelles applications
- Partager: le Bioportal doit permettre de partager des ressources et des données, permettant la collaboration entre de nombreuses communautés.

Dans ce contexte, le Bioportal a été construit avec plusieurs outils :

- l'OGCE (*Open Grid Computing Environment*) [34] développe, commissionne et diffuse des éléments logiciels pour construire des portails de grille (grid portals), des portails scientifiques (scientific gateways) et des outils d'accès aux ressources de calcul grille et cloud
- La boîte à outils Globus (Globus Toolkit) [7] est une bibliothèque logicielle Open Source utilisée depuis 2000 pour construire des grilles. Développée principalement par l'alliance Globus autur de l'université de Chicago, elle est l'intergiciel de référence avec lequel des grilles ont été construites dans le monde entier.
- Le système de gestion de flot Taverna [33] propose une suite d'outils utilisés pour concevoir et exécuter des pipelines d'analyse et aider pour l'expérimentation in silico

La Figure 2-10 - L'architecture du Bioportal montre les quatre composantes principales du Bioportal et leur interaction: User Workspace (espace de travail utilisateur), Application Framework (cadre d'application), Grid Framework (cadre de grille), et Security and Account Management (gestion de compte et de la sécurité). Comme son nom l'indique, le User Workspace est le point d'interaction pour les actions de l'utilisateur avec les services à l'arrière. Le cadre d'application permet de décrire la logique d'interface et de traitement nécessaire pour interpréter les entrées de l'utilisateur. Les services Grid Framework et Security and Account Management coordonnent l'authentification, la soumission des jobs et le transfert de fichiers.

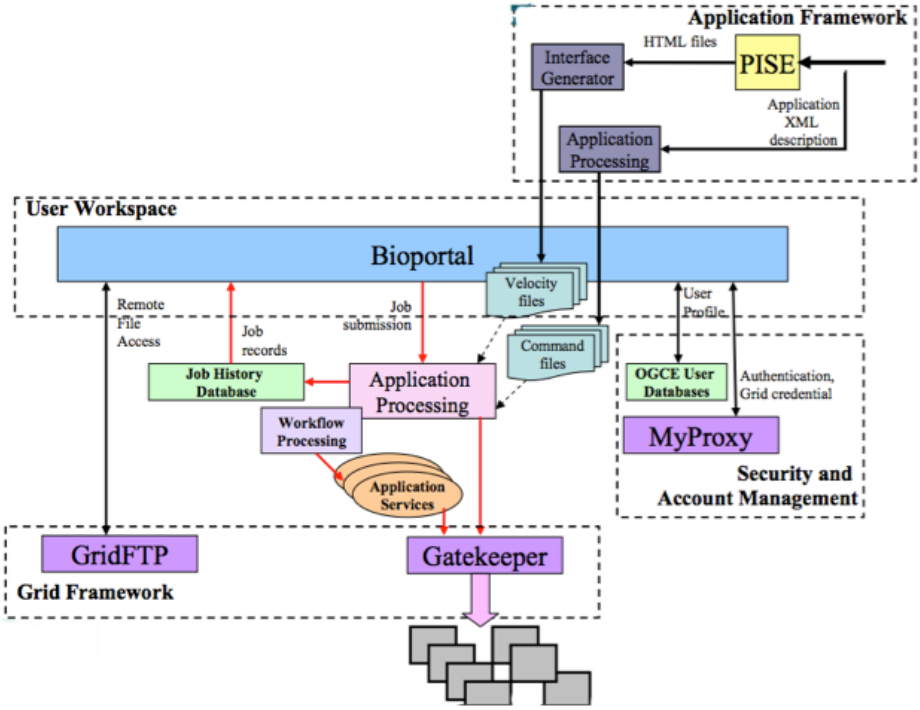


Figure 2-10 - L'architecture du Bioportal

L'espace de travail utilisateurs utilise des « portlets » qui sont des composants de portail qui contrôlent des éléments configurables par l'utilisateur dans le navigateur web de l'utilisateur. Le Bioportail utilise des éléments logiciels de l'Open Grid Computing Environment mais aussi CHEF (CompreHensive collaborativE Framework) [35] pour supporter le renforcement des communautés et le partage des informations. Les services Jetspeed (http://portals.apache.org/jetspeed-2/) sont utilisés pour la personnalisation de l'interface utilisateur, la mise en cache, la persistance et l'authentification des utilisateurs. Velocity (http://velocity.apache.org/) est utilisé pour la spécification d'application.

L'*Application Framework* permet aux utilisateurs de sélectionner, configurer et lancer des applications à travers le portail. Les interfaces bioinformatiques sont basées sur PISE, un outil qui génère des interfaces web pour des applications biomédicales en utilisant des descriptions XML des entrées de l'application. En arrière-plan, la chaine de traitement est implémentée en utilisant Taverna. Développé dans le cadre du projet myGrid (http://www.mygrid.org.uk/), Taverna permet aux utilisateurs de composer et d'exécuter des analyses complexes en utilisant des composants disponibles localement et sur des sites distants par les interfaces standard de services Web. Bioportal utilise Taverna pour déployer les applications de manière dynamique à travers une interface de service Web.

Le *Grid Framework* utilise les technologies de réseau et les protocoles existants pour gérer les jobs et les fichiers : GSI (Grid Security Infrastrucutre) [36], GRAM (Grid Resource Allocation and Management) [37] et GridFTP (Grid File Transfer Protocol) [38].

Le *Security and Account Management* offre une sécurité à plusieurs niveaux: Secure Socket Layer (SSL), GSI, et un audit MyProxy [39]. Pour accéder aux services du portail et la grille, chaque utilisateur a besoin d'un compte dans le portail, un certificat d'utilisateur et des comptes sur les ressources de calcul.

En bref, Bioportal est un excellent exemple de portail d'accès à des données et des ressources distribuées par une infrastructure standard et extensible pour la conception de pipelines de traitement bioinformatiques et leur exécution.

2.4.1.5. GVSS

Développé à l'Academia Sinica Computing Center de Taïwan, acteur majeur de la grille en Asie, GVSS (Grid-enabled Virtual Screening Services, http://gap.grid.sinica.edu.tw/) est un portail dédié à la biologie structurale et plus spécifiquement au « docking » pour la recherche de nouveaux médicaments. Le principe du *docking* est de calculer l'énergie de liaison d'un composé chimique, le ligand, avec le site actif d'une cible biologique. Le calcul de l'énergie de liaison entre le ligand et le site actif utilise des logiciels de chimie computationnelle dont le plus connu est Autodock [56] et requiert typiquement quelques minutes. Ce logiciel sera utilisé dans des tests qui seront décrits dans le chapitre 3 de ce manuscrit.

L'intérêt du docking est de permettre de tester *in silico* si des molécules sont actives sur une cible biologique, réduisant ainsi le coût des tests *in vitro*. Il existe aujourd'hui des millions de composés chimiques qui sont des médicaments potentiels. La structure tridimensionnelle de ces molécules est stockée dans des bases de données appelées criblothèques. Certaines sont privées et jalousement gardées par leurs propriétaires. D'autres sont publiques et mises à disposition par les entreprises qui produisent ces composés. La plus grande de ces bases de données publiques, ZINC (http://zinc.docking.org/), regroupe ainsi environ 10 Millions de ligands.

La structure tridimensionnelle des sites actifs des cibles biologiques est fournie par la base de données Brookhaven Protein Data Base (PDB, http://www.rcsb.org/).

La plate-forme GVSS offre à l'utilisateur la possibilité de personnaliser la préparation de la base de données ligand et donne accès à plus de 700 protéines de la PDB pour soumettre la simulation de docking. Après la simulation, l'utilisateur peut visualiser chaque meilleure conformation et des résultats tels que l'histogramme d'énergie et l'analyse en composante principale 2D/3D.

GVSS est basé sur la plate-forme GAP (Grid Application Platform) [40] et utilise l'outil de déploiement DIANE (voir section 2.4.2.4) qui déploie des jobs pilotes sur la grille. Les données biochimiques sont stockées en utilisant le catalogue de métadonnées AMGA (voir section 3.2.4.1) L'architecture du système de GVSS est illustrée dans la Figure 2-11.

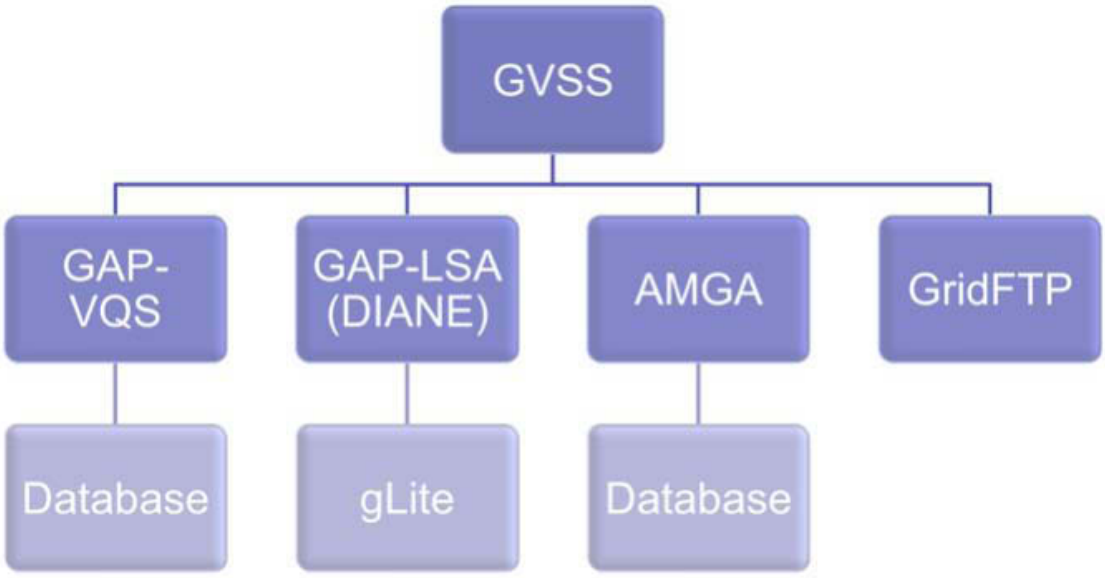


Figure 2-11 – Architecture du système de GVSS

Grâce à DIANE et à GAP, GVSS peut diviser et exécuter les jobs soumis en plusieurs sous-tâches indépendantes. L'exécution de ces sous-tâches indépendantes tire profit de la disponibilité de la ressource de la grille.

GVSS est utilisé pour la recherche de médicaments *in silico* sur les maladies négligées et émergentes [41].

2.4.2. Outils de déploiement

Après quelques exemples d'environnements ou de portails de bioinformatique, nous allons maintenant faire un état de l'art des principaux outils de déploiement utilisés sur les grilles en Europe, en commençant par DIET qui fournit à la fois un intergiciel et un outil de déploiement.

gHnHgH H□ □□□□

Né au sein de l'équipe GRAAL à l'ENS Lyon, le projet *Distributed Interactive Engineering Toolbox* (DIET) [42], [43] s'est concentré sur le développement d'un intergiciel de grille de type NES (*Network Enabled Server*) respectant le standard GridRPC. Ce standard est l'implantation dans un environnement grille du mécanisme d'appel de procédure distante (*Remote Procedure Call*). Le GridRPC [93] est défini par un travail commun au sein de l'Open Grid Forum (OGF). DIET est constitué de plusieurs composants logiciels qui permettent de construire une application répartie sur une grille. L'intergiciel base sa recherche sur la description de la requête du client (nom du service, paramètres et taille des données), les capacités de traitement des serveurs et la localisation des données utilisées pour la résolution du service. Un mécanisme de gestion de données permet de stocker de manière permanente ou temporaire les données dans l'intergiciel en vue d'une utilisation future pour une autre requête. L'originalité de DIET repose entre autre sur une architecture

hiérarchique d'agents.

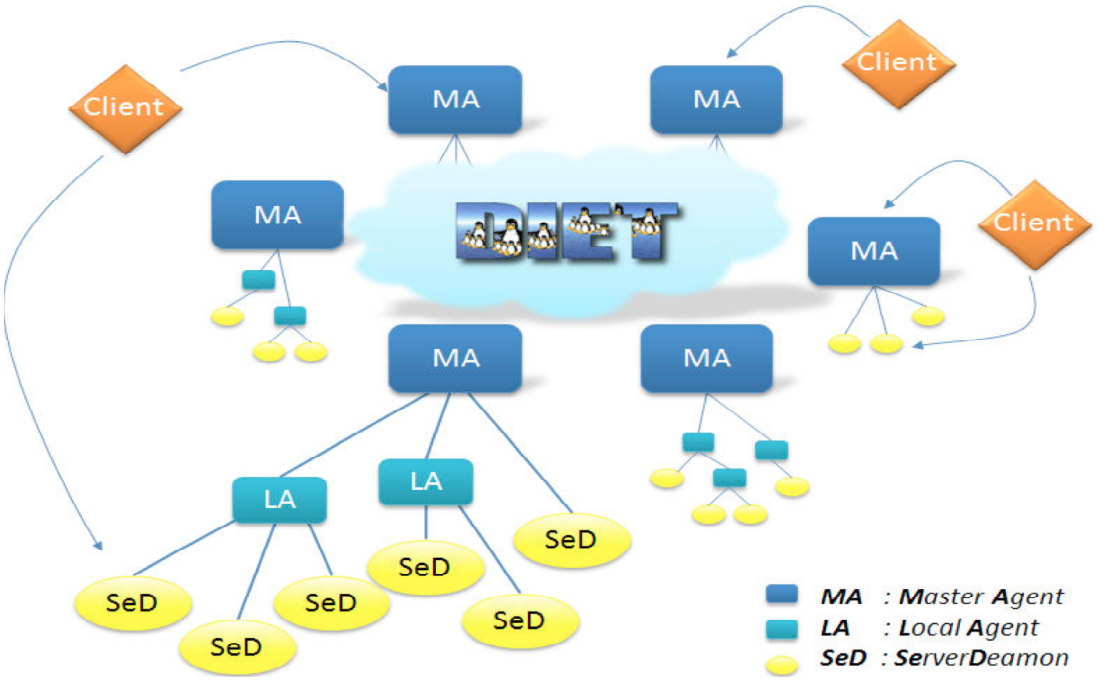


Figure 2-12 – Architecture de DIET

DIET est basé sur un modèle Client-Agent-Serveur. Il se différencie notamment par la distribution d'agents en une hiérarchie d'agents. On distingue parmi les agents, le MasterAgent (MA) et les LocalAgent (LA). Cet arbre d'agents est enraciné sur le MA qui est le point d'entrée pour un client. Chaque agent (MA) peut être relié à des Server-Deamons (SeDs) qui sont les points d'entrée aux ressources de calcul et de stockage. Plusieurs arbres d'agents peuvent être interconnectés dynamiquement ou statiquement par l'intermédiaire de chaque point d'entrée (i.e. les MA) de chaque hiérarchie. La Figure 2-12 montre une architecture complète formée de plusieurs hiérarchies d'agents. Le client contacte son point d'entrée grâce au service de nommage standard CORBA, dans lequel chaque MA s'enregistre avec son nom.

Le fonctionnement de DIET est décrit en détail en [9]. Le portage ou « gridification » d'une application dans DIET utilise un ensemble d'interfaces permettant aux développeurs de publier leurs applications sur la grille, et aux clients (les utilisateurs) de faire appel à celles-ci. DIET fournit une abstraction de l'application afin de la rendre disponible sur un ensemble de ressources réparties sur une grille.

La soumission d'une requête par un client DIET à un ensemble de ressources gérées par DIET est une succession d'étapes masquées dans l'appel d'une unique fonction *diet call*.

DIET dispose de son propre les services de gestion de données [44] *Data Tree Manager* (DTM), JuxMEM et DAGDA. Ce système propose une gestion des données basée sur deux éléments clés : des identifiants pour les données et une gestion en arbre qui colle à la hiérarchie DIET. Dans le but d'éviter des transmissions inutiles de la même donnée d'un client vers les serveurs de

calcul, DTM ajoute la possibilité de laisser les données à l'intérieur de la plate-forme après la fin d'une requête. Un identifiant est alors attribué à la donnée et retourné au client. Il pourra s'en servir pour faire référence à sa donnée lors d'une autre requête. Un client peut choisir, pour chacune de ses données, si elle doit être persistante ou non à l'intérieur de la plate-forme. Plusieurs modes de persistance sont proposés.

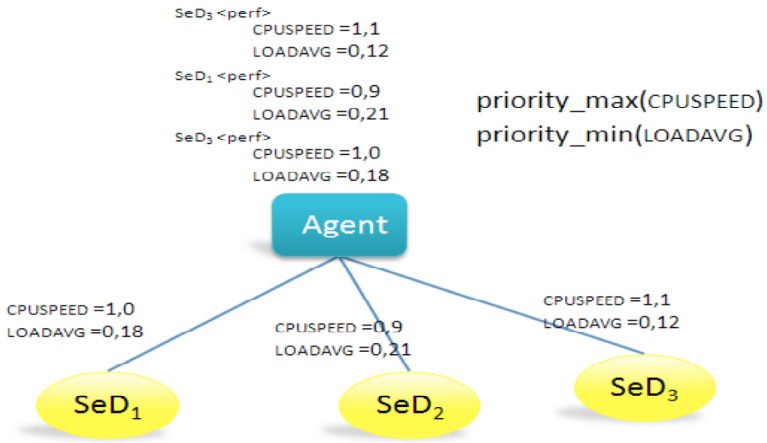


Figure 2-13 – Fonctionnements des ordonnanceurs spécifiques

Une autre particularité originale de DIET est son mécanisme d´ordonnancement appelé *plug-in scheduler* [46]. En fait, en dehors de l´ordonnanceur par défaut qui utilise une stratégie de quasi *round-robin* sur toutes les ressources disponibles, DIET offre la possibilité aux développeurs de la couche applicative serveur, d'affiner l'ordonnancement fait par la hiérarchie DIET en fonction des spécificités de son application. Pour se faire, le codeur définit son propre vecteur d'estimation et spécifie le comportement (i.e l'ordonnancement) que la hiérarchie doit avoir pour trier les réponses fournies par les SeDs (*ServerDemons*). En addition de ce mécanisme de *plug-in scheduler,* DIET offre un collecteur modulaire d'informations sur les ressources gérées par le SeD appelé CoRI. Cette fonctionnalité est complémentaire à la possibilité de définir un *plug-in scheduler*.

DIET a été testé sur la grille Grid´5000 [45] et utilisé dans plusieurs applications réelles. Par exemple, en géologie, DIET a été utilisé pour distribuer la charge de calcul de l´application DEM (Digital Elevation Model) sur un cluster de 125 processeurs avec une interface web facile à utiliser. Dans une autre application en aérospatial de l´analyse des éléments finis avec le programme massivement parallèle Ipsap, DIET a permis de déployer les calculs sur un large cluster coréen de 500 CPU connecté avec quelques machines en France.

Pour la bioinformatique, DIET a été utilisé pour implanté et supporté la grille Décrypton version II [46]. L'objectif qui a guidé la mise en place de la grille Décrypthon sous l'intergiciel DIET a été la continuité de service. En effet, pendant toute la phase de migration, la grille était active pour les projets scientifiques et aucune interruption n'a été à déplorer.

2.4.2.2. DIRAC

Développé initialement au CPPM à Marseille puis dans la collaboration LHCb au CERN, DIRAC (Distributed Infrastructure with Remote Agent Control) [47] est à la fois un système de gestion de charge (Workload Management System - WMS) et un système de gestion de données (Data Management System - DMS), qui est développé pour supporter les activités de production ainsi que l'analyse des données. DIRAC forme une couche intermédiaire entre une communauté particulière et des ressources de calcul diverses pour permettre une utilisation optimisée, transparente et fiable. DIRAC a été développé dans le cadre de la collaboration LHCb au CERN (http://lhc.web.cern.ch/lhc/) pour satisfaire les besoins liés à la génération d'un grand volume de données de simulation Monte-Carlo. DIRAC a été le premier système de soumission de jobs pilotes dès le projet DataGrid au début des années 2000 (http://eu-datagrid.web.cern.ch/eu-datagrid/).

Le système de production distribuée DIRAC a les fonctionnalités suivantes:

- Définition des tâches de production
- Installation de logiciels sur des sites de production
- Planification et surveillance des jobs
- Gestion et réplication de données

Pour un système de production, le composant en charge de la planification des jobs (*job scheduling*) est très important. Deux paradigmes différents peuvent être choisis :

- Dans le paradigme *push*, le planificateur (*scheduler*) utilise les informations sur la disponibilité et l'état des ressources de calcul afin de trouver le meilleur match aux exigences particuliers d'un job.
- Dans le paradigme *pull*, en revanche, c´est la ressource de calcul qui cherche des tâches à exécuter. Les jobs sont d'abord accumulées par un service de production, validés et mis dans une file d'attente. Lorsque une ressource de calcul est disponible, elle envoie une requête pour un job à faire au service de production. Le service de production choisit un job, selon les capacités des ressources et l'envoie à la ressource de calcul pour qu'il puisse s'y exécuter.

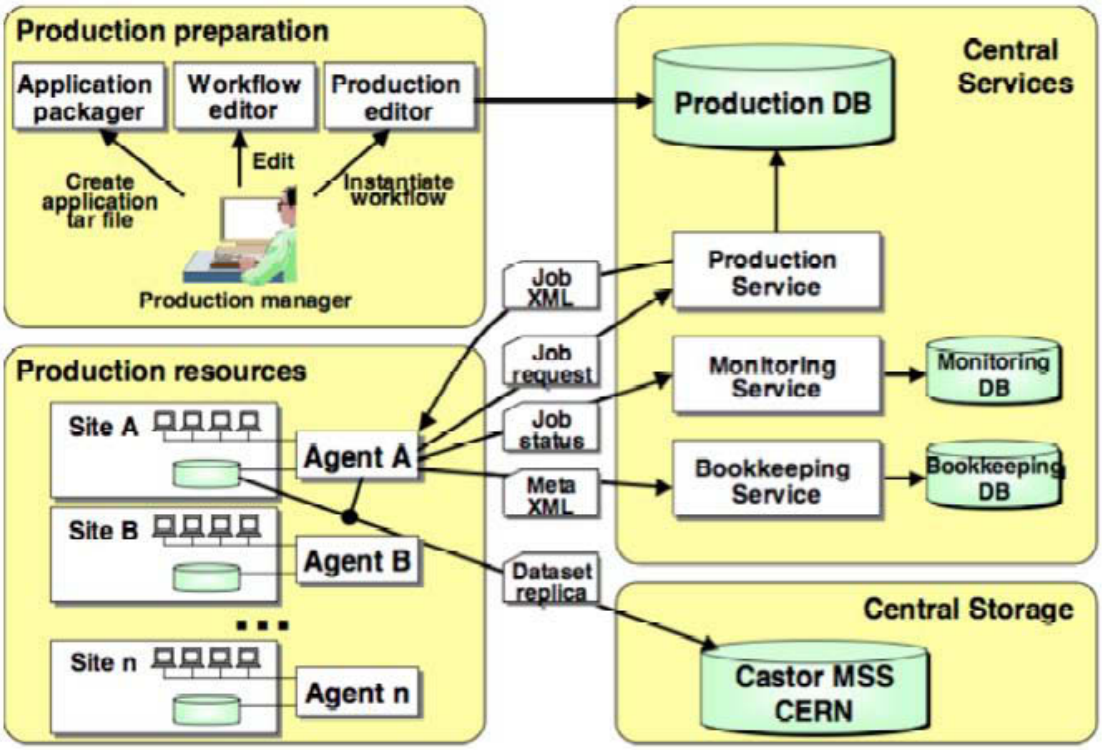


Figure 2-14 - L'architecture de DIRAC (http://diracgrid.org/)

Il y a des avantages et des inconvénients dans les deux approches. Dans le paradigme *push*, l'information sur l'état dynamique de toutes les ressources est généralement collectée dans un seul endroit. Pour chaque job, le meilleur choix possible peut être fait. Un problème est que le nombre d´opérations à effectuer pour faire ce choix est proportionnel au produit du nombre de ressources et de jobs. Cela conduit à des problèmes d´escalade, parce que ces deux nombres sont en constante augmentation. Dans le paradigme *pull*, une ressource ralentie se manifeste immédiatement. Pareillement, la ressource la plus puissante sera la plus sollicitée. Par conséquent, l'équilibrage de la charge est atteint plus aisément. En plus, il est plus facile d'intégrer de nouveaux sites de production parce que leurs informations ne sont pas nécessaires au service de production central.

DIRAC, comme des autres systèmes de production de la grille de calcul (DIANE – section 2.4.3.4 ou WPE – chapitre 3) est développé avec le paradigme *pull*. Ce paradigme sera présenté de façon plus détaillée dans la section 3.3).

L'architecture de DIRAC est présentée dans la Figure 2-14. Elle se compose de services centraux: *Production*, *Monitoring* et *Bookkeeping*. Les *Agents* de production s´exécutent en permanence sur chaque site de production. Les services centraux ont pour rôle de préparer les productions, surveiller l´exécution des jobs et la comptabilité des paramètres des jobs. Les *Agents* examinent l'état des queues de production locales. Si les ressources locales sont capables d'accepter la charge, l'Agent envoie une demande de job au service de

production centrale et assure l'exécution et la surveillance de ce job. Après l´exécution du job, l'Agent transfère les données de sortie et met à jour le catalogue de métadonnées.

DIRAC est maintenant largement utilisé sur de nombreuses infrastructures dans le monde entier. Son concept permet d'agréger dans un seul système informatique des ressources de nature différente, tels que des grilles de calcul, des *clouds* ou des *clusters*, de façon transparente aux utilisateurs.

Parmi les limitations de DIRAC, le fait d'utiliser des jobs pilotes à la demande faite que, chaque fois on veut lancer un job réel, un job pilote correspondant est soumis. Bien que la soumission des jobs soit bien optimisée, il arrive encore qu'un job pilote ne réussisse pas à se lancer [48]. Heureusement, on peut resoumettre les jobs facilement avec un interface web simple fournie par DIRAC.

2.4.2.3. OpenMole

Il existe plusieurs outils tels que G-Eclipse [50], Ganga [51] ou JSAGA [52] qui simplifient l'accès aux environnements d'exécution distribués. Cependant, ils ne cachent pas le matériel et l'hétérogénéité du logiciel de ressources de calcul. OpenMOLE [49] a été conçu pour que des algorithmes scientifiques puissent être déployés indépendamment de l'architecture de l´environnement d'exécution. Il fournit un environnement d'exécution virtualisée en prenant les avantages de l'interface générique de JSAGA. La délégation transparente des algorithmes scientifiques à l'environnement à distance devient possible avec OpenMOLE : l'utilisateur peut être assuré qu´une expérience exécutée avec succès sur son ordinateur local sera exécutée avec succès sur un autre environnement d'exécution, sans tests préliminaires ou installation de logiciels.

Il y a quelques contraintes pour l'environnement d'exécution d'OpenMOLE :

- L'utilisateur doit pouvoir définir et intégrer ses composants logiciels dans OpenMOLE, qui doit pouvoir les exécuter à distance.
- OpenMOLE ne doit pas dépendre du logiciel installé initialement sur l'environnement
- OpenMOLE doit être autonome pour accéder aux différents environnements d'exécution à distance

- L’ordinateur de l’utilisateur peut être connecté à l’Internet sans être nécessairement le propriétaire d'une adresse IP publique.

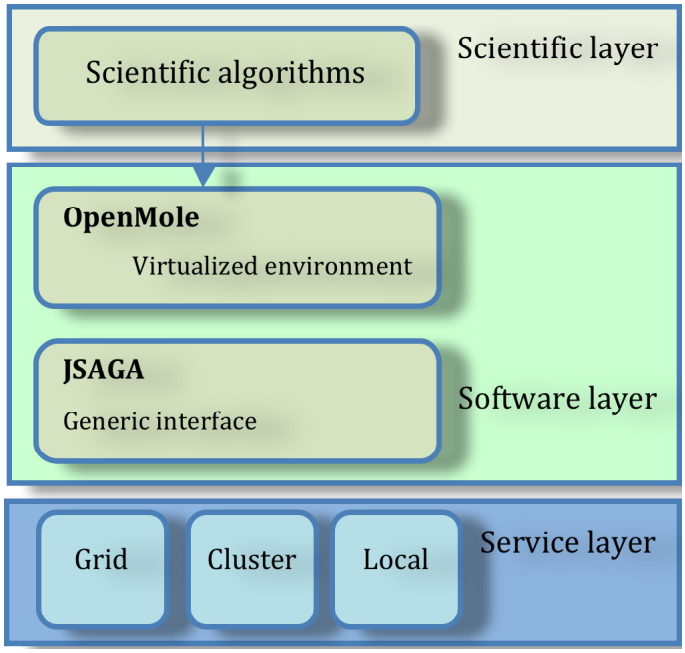


Figure 2-2-15 – L’architecture de couche de OpenMOLE (http://www.openmole.org/)

Les deux premières contraintes ont été mis en place afin que l'utilisateur puisse travailler sur ses propres algorithmes, sans savoir comment les déployer ou exécuter sur des environnements d'exécution différents. La troisième contrainte assure l´escalade et la convivialité de l'application. La dernière contrainte assure que l’ordinateur de l’utilisateur ne fonctionne pas comme un serveur. Ces contraintes posent deux problèmes majeurs :

- Comment pouvons-nous aborder la question de la portabilité des composants logiciels des utilisateurs sur les environnements d'exécution hétérogènes ?
- Comment pouvons-nous accéder à des environnements d'exécution différents de façon générique sans compter sur un serveur tiers ?

Afin de résoudre le premier problème, OpenMOLE fournit des services génériques mises en œuvre dans la classe *Task*. Une *Tâche* est exécutée dans un *Contexte* spécifique et utilise des *Ressources* indiquées. Cette conception a pour but de migrer et d'exécuter des tâches à distance. La classe Task est polymorphe et chaque spécialisation de Task est fournie comme un service nouveau. Les ressources d’une tâche peuvent être les fichiers, les bibliothèques ou les logiciels, qui doivent être disponibles lors de son exécution. Le contexte contient les variables qui sont utilisées par la tâche lors de l'exécution. Chaque contexte correspond aux conditions d'exécution particulières. Comme les tâches peuvent être déplacées, elles doivent être portables d'un environnement à un autre. Sur OpenMOLE, l'hétérogénéité des environnements d'exécution

différents est gérée à l'aide du système de virtualisation JVM (Java Virtual Machine).

Pour la délégation des tâches, OpenMOLE utilise une méthode développée pour fonctionner sur n'importe quel environnement contenant au moins à la fois un système de stockage de fichiers et un système de soumission des jobs. La première étape de cette méthode est de stocker les fichiers requis pour l'exécution à distance :

- une archive (s'appelant *runtime*) qui contient la machine virtuelle Java et le framework OpenMOLE,
- un objet de la classe Task ainsi que son contexte d'exécution sérialisé dans un fichier
- les fichiers nécessaires pour le déploiement des ressources sur l'environnement cible

Une fois cette étape complétée, un job est transmis au système de soumission. Au début de son exécution, le job prépare l'environnement sur le nœud de calcul: il télécharge le *runtime*, désérialise la tâche et son contexte d'exécution, télécharge et déploie les ressources. Puis il exécute la tâche dans ce contexte. Enfin, il stocke les sorties sur le système de stockage dans un lieu prédéfini. Au cours de l'exécution, le *framework* OpenMOLE suit son état. Une fois terminé, les résultats sont téléchargés sur l'ordinateur local de l'utilisateur et ils peuvent être utilisés pour une autre tâche. OpenMOLE profite des couches d'abstraction de JSAGA pour la soumission de job et le transfert de données. En haut de JSAGA, OpenMOLE fournit à la fois la portabilité transparente de la tâche et l'accès transparent à des environnements d'exécution différents. Par conséquent, OpenMOLE peut mettre en œuvre la fonctionnalité de transférer l'exécution de la tâche d'un environnement à l'autre de manière déclarative.

2.4.2.4. DIANE

DIANE (Distributed Analysis Environment) [53] est un projet qui se concentre sur l'analyse et la simulation semi-interactive et parallèle de données à distance. DIANE fournit l'infrastructure pour les applications scientifiques dans le modèle *master-worker*. Autrement dit, DIANE est un gestionnaire de flot générique pour des applications distribuées, construit au-dessus de l'intergiciel de la grille pour fournir des facilités de haut niveau pour le développement et déploiement d'applications. Les principales caractéristiques de DIANE sont les suivantes:

- L'exécution de la tâche est entièrement contrôlée par le Maître. Cela permet de faire l'ordonnancement dynamique des jobs sans interférer avec la logique d'application
- Le modèle de retrait des tâches (Task Pull) supporte l'équilibrage de charge automatique (automatic load balancing).
- La vérification périodique (Heartbeat check) des *workers* et la définition

de la reprise après incident au niveau du *job* améliorent la stabilité de l'exécution des jobs sur la grille.

- La fusion des ressources de calcul entre des ressources locales et des ressources de la grille.
- L'interface claire et facile à programmer. A titre d'exemple, la distribution des jobs AutoDock sur la grille requiert moins de 500 lignes de code python.

L'architecture de DIANE est illustrée sur la Figure 2-14. Dans le modèle *master-worker*, le client envoie les paramètres du job au *Planner* qui partitionne le job en petites tâches exécutées par les *Workers*. L'*Integrator* est responsable de la fusion des résultats de l'exécution des tâches et de l'envoi du résultat final au client.

Les composants d'application (*Integrator, Planner* et *Worker App)* sont déployés sur les *Masters / Workers* qui fournissent le contexte d'exécution et l'environnement de mise en place. La couche de communication physique fournit des mécanismes de l'interaction des Masters/Workers et la transmission de messages de données produits par les composants d'application. Avec l'architecture de DIANE, les développeurs d'applications ne doivent pas implémenter les mécanismes de communication explicitement. Au lieu de cela, ils n'ont qu'à mettre en œuvre des composants d'application (en décrivant le contenu des messages d'entrée et de sortie) et implémenter des interfaces. La plate-forme prend en charge la gestion de la chaine de traitement et le passage de messages. L'avantage de cette approche est que la topologie de déploiement des éléments de l'intergiciel peut être modifiée sans affecter l'application au niveau du code source. Le code des applications n'est donc pas affecté par des changements dans la mise en œuvre de mécanismes de communication interne.

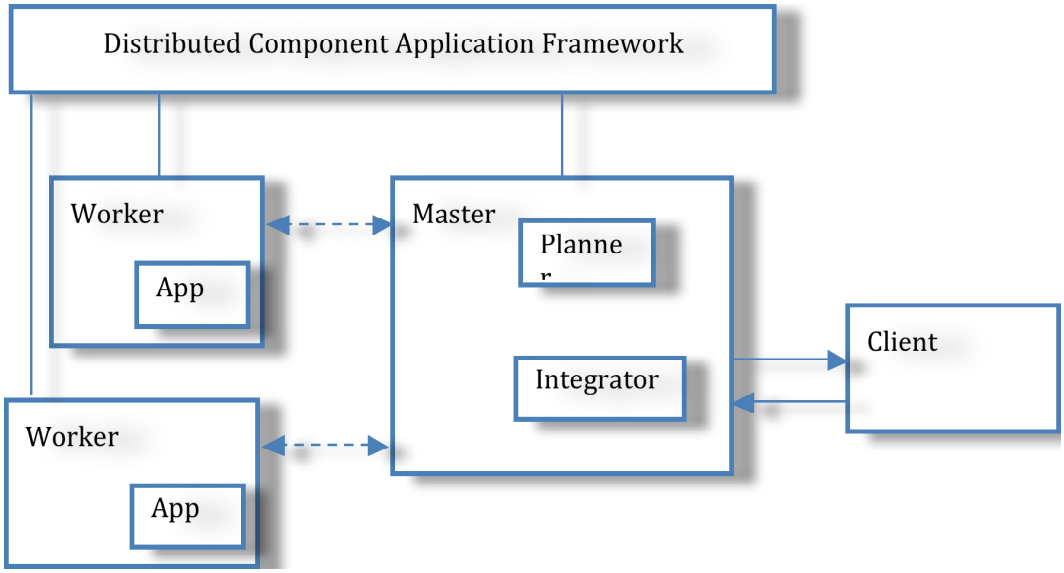


Figure 2-2-16 - Architecture de DIANE

Des tests de performance [54] ont montré que DIANE pouvait atteindre environ 75% de l'efficacité théorique qui correspond à une accélération

directement proportionnelle au nombre de processeurs utilisés. DIANE est utilisé dans quelques applications comme : ATHENA (ATLAS Data Analysis) [55], AutoDock [56] et les simulations avec la bibliothèque Geant4 [57].

2.4.2.5. WISDOM

Démarré en 2006, le projet WISDOM (***W**ide **I**n **S**ilico **D**ocking **O**n **M**alaria*) [75] avait pour objectif de déployer à très grande échelle des calculs de docking pour la recherche de nouveaux médicaments contre le paludisme. Il s'agissait de calculer l'énergie de liaison de plusieurs centaines de milliers de composés chimiques avec les sites actifs de cibles biologiques d'intérêt pour le traitement de cette maladie.

En 2006, les outils disponibles pour gérer un déploiement à très grande échelle sur la grille EGEE n'étaient pas légion. Aussi, l'approche adoptée par la collaboration WISDOM fut de développer un système permettant de centraliser la gestion de plusieurs milliers de tâches soumises simultanément sur la grille.

Ce système était basé au départ sur une stratégie « PUSH » de pousser les jobs sur la grille. Lorsque les jobs finissaient avec succès, les données étaient collectées sur des éléments de stockage de la grille et lorsque les jobs échouaient pour une raison connue ou inconnue, ils étaient resoumis. Cette approche pragmatique a permis de faire le premier déploiement à très grande échelle de calculs et ainsi le premier déploiement d'un « data challenge » dans un autre domaine que la physique des hautes énergies sur la grille EGEE.

La description détaillée de l'environnement WISDOM fait l'objet du chapitre 3.

2.5. Etat de l'art de l'utilisation des grilles en épidémiologie et en métagénomique

Nous avons présenté dans les paragraphes précédents un bref état de l'art des infrastructures, portails, plates-formes et outils développés et déployés pour répondre aux besoins des biologistes et des bioinformaticiens sur la grille. Nombre de ces environnements sont déjà conçus pour s'adapter au cloud computing (DIRAC, OpenMole) dès que des ressources importantes seront disponibles sur des clouds académiques.

Nous allons maintenant nous intéresser à la science qui peut être faite sur la grille. Avant de nous concentrer sur deux disciplines, l'épidémiologie et la métagénomique, nous allons apporter des éléments généraux de contexte.

2.5.1. Introduction

Dès le premier projet de grille européen DataGrid (2001-2003), les sciences de la vie ont été identifiées parmi les disciplines qui, aux côtés de la physique des particules, pouvaient le mieux tirer parti des infrastructures de grille. Très vite, l'imagerie médicale autour du laboratoire CREATIS et la bioinformatique autour de l'Institut de Biologie et de Chimie des Protéines ont apporté des applications requérant des services qui n'existaient pas alors. Ces disciplines ont donc été des moteurs puissants de l'évolution des services de la grille,

alimentant les développeurs d'intergiciels avec leurs cahiers des charges et leurs usages spécifiques.

Des algorithmes de génomique comme BLAST [22] ou CLUSTAL [13] ont ainsi été déployées dès le début des années 2000 sur les infrastructures de grille. Bien que la génomique ne recouvre à ce jour qu'une petite partie de la variété des données produites en biologie, elle représente une majorité des besoins de stockage, de calcul et d'analyse. Avec l'arrivée des nouvelles technologies de séquençage à haut débit, les besoins en ressources informatiques de la génomique sont devenus considérables. La métagénomique est un procédé méthodologique qui vise à étudier le contenu génétique d'un échantillon issu d'un environnement complexe (intestin, océan, sols, etc.) trouvé dans la nature (par opposition à des échantillons cultivés en laboratoire). Elle étudie les organismes microbiens directement dans leur environnement sans passer par une étape de culture en laboratoire. Par ses objets d'études et ses procédés, elle est confrontée aux défis de l'analyse des données de séquençage à haut débit liés aux volumes énormes de données et aux temps d´exécution des algorithmes. Par conséquent, les solutions de calcul distribué comme la grille doivent être considérées.

Parallèlement à la génomique, l'épidémiologie a été identifiée en 2005 parmi les champs d'application les plus prometteurs des grilles informatiques dans le domaine de la santé [58]. On peut définir l'épidémiologie moléculaire comme l'utilisation des techniques de la biologie moléculaire à la résolution de problèmes épidémiologiques.

Après ces quelques lignes d'introduction, nous allons maintenant faire un bref état de l'art de l'utilisation des grilles en épidémiologie et en métagénomique.

2.5.2. Epidémiologie

L'épidémiologie conventionnelle requiert des collections très complètes de données concernant les populations, les paramètres de santé, les maladies ainsi que des facteurs environnementaux comme le régime alimentaire, le climat et le contexte social.

Une étude peut se focaliser sur une région particulière ou un foyer épidémique. Elle peut aussi s'intéresser à une condition sur un domaine très étendu. Les données requises dépendent des études entreprises mais certains éléments se retrouvent de façon récurrente : un niveau de confiance est nécessaire pour que la provenance des données puisse être garantie et un niveau de standard dans la qualité des pratiques cliniques doit être garanti.

Quand des données sont collectées dans des environnements cliniques différents, il est nécessaire de pouvoir établir une équivalence sémantique, pour s'assurer que la comparaison ou l'agrégation des données est légitime. Une dimension éthique doit aussi être prise en compte si des données collectées initialement à des fins thérapeutiques sont aussi utilisées pour la recherche.

L'analyse de données agrégées requiert la construction de modèles complexes et l'utilisation d'outils statistiques sophistiqués. Cela a nécessité la

collaboration de médecins et de biostatisticiens et l'émergence de l'épidémiologie comme une discipline à part entière. L'impact des analyses génomiques a enrichi le spectre des données collectées.

La technologie pour fédérer les bases de données dans les hôpitaux se développe depuis de nombreuses années. Ces bases de données doivent pouvoir être consultées en préservant l'anonymat des patients. De telles requêtes peuvent être gérées et supervisées par les hôpitaux responsables des données, s'assurant du bon respect du cadre légal et juridique.

Les grilles sont des outils privilégiés d'intégration des bases de données. Elles permettent l'interopérabilité des outils et des services d'analyse et permettent de définir des standards communs et une sémantique commune pour le contenu des bases de données et les outils d'entrée/sortie. La fédération de bases de données médicales requiert donc une sémantique claire. Si les efforts actuels autour de la OBO Foundry (Open Biological and Biomedical Ontologies, http://www.obofoundry.org/) sont prometteurs pour la biologie moléculaire, il n'existe pas aujourd'hui de sémantique médicale faisant l'objet d'un consensus aussi large.

L'utilisation des grilles en épidémiologie recouvre les champs d'application suivants :

- épidémiologie génétique pour identifier les gênes impliqués dans des maladies complexes
- études statistiques pour un travail sur les populations de patients. Un exemple est le projet européen ENS@T-CANCER [59] de suivi des patients atteints de cancer des glandes surrénales
- évaluation de l'efficacité d'un médicament par l'analyse des populations
- suivi d'une pathologie par des études longitudinales
- développement humanitaire : les grilles ouvrent de nouvelles perspectives pour la préparation et le suivi de missions médicales dans les pays en développement ainsi que pour le suivi et le support de centres de soin isolés en termes de téléconsultation, télédiagnostic, suivi des patients et e-learning [60]
- grilles de surveillance épidémiologiques pour répondre à des alertes éventuelles

Une équipe de l'Universidad Polechtnica Valencia autour de Ignacio Blanquer s'intéresse particulièrement aux problématiques de partages d'images médicales dans un contexte d'épidémiologie [61].

De son côté, l'équipe de Richard Sinnott au Royaume-Uni (National e-Science center à Glasgow) puis à l'Université de Melbourne s'est particulièrement intéressée aux problématiques liées à la gestion d'essais cliniques et de cohortes de patients sur la grille [62].

Nous allons présenter maintenant plus en détails le projet GINSENG de fédération de bases données médicales avant de présenter l'initiative Global Public Health Grid dont le service g-INFO décrit dans le chapitre 4 de ce

manuscrit a été l'un des prototypes.

gHmHgH H

Le projet GINSENG (Global Initiative for Sentinel E-health Network on Grid) a pour but de construire une infrastructure de grille pour l'e-santé et l'épidémiologie en France. Financé par l'Agence Nationale de la Recherche, GINSENG fédère des bases de données médicales distribuées de plusieurs sites hospitaliers et les laboratoires de pathologie. Ces données peuvent ensuite être interrogées pour obtenir des informations statistiques d'épidémiologie. Le projet GINSENG propose une solution alternative à la gestion centralisée des données sur un seul serveur. Le portail à l'adresse http://www.e-ginseng.org permet aux utilisateurs de s'authentifier et d'avoir une vue d'ensemble de l'activité de santé pour le dépistage et le diagnostic des cancers et la périnatalité.

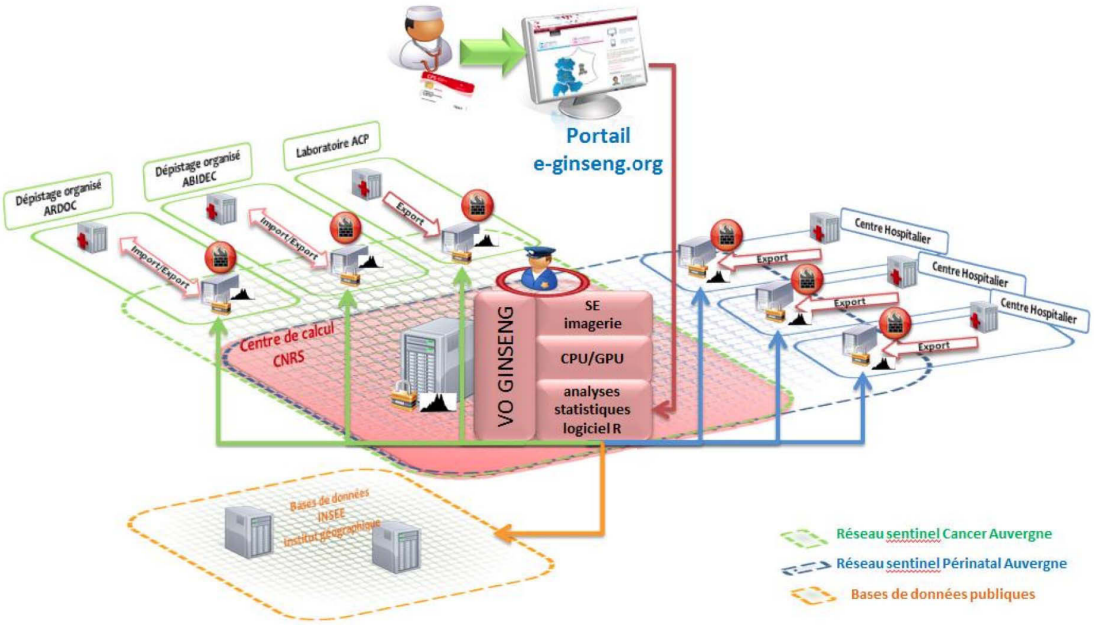


Figure 2-2-17 – Architecture du GINSENG

L'architecture du GINSENG est illustrée dans la Figure 2-17. Les composants principaux dans cette architecture sont :

- Un portail qui est le point d'accès unique au GINSENG. Il a pour but de fournir l'information sur le projet, d'identifier les utilisateurs et gérer des requêtes épidémiologiques. Ce portail s'adapte son interface à chaque type d'utilisateur selon ses besoins.

- La sécurité dans l'infrastructure est basée sur CAS (Central Authentification Service) et CPS (Carte de Professionnel de Santé). GINSENG dispose de sa propre Organisation Virtuelle et son propre service de gestion VOMS qui gère l'autorisation via certificat et proxy de la grille.

- Le logiciel d'analyses statistiques R (http://www.r-project.org/) qui est

un projet open source très familier pour les statisticiens.

- Un serveur de métadonnées pour agréger les informations qui serviront de base aux requêtes R des épidémiologistes.

L'infrastructure GINSENG est en cours de déploiement pour la gestion des données de soin périnatal en Auvergne ainsi que pour fédérer les bases de données des cabinets d'anatomo-pathologie pour la surveillance de certains cancers.

En bref, l'infrastructure GINSENG propose une solution pour les données médicales distribuées entre différents hôpitaux et les outils pour les interroger dans un but statistique. La grille permet de réaliser les améliorations majeures de GINSENG par rapport aux solutions existantes comme le temps de transfert, l'automatisation ou la sécurité et ouvre la perspective de traiter aussi les images et les données génomiques ainsi que d'étendre le réseau hors de l'Auvergne.

2.5.2.2. Global Public Health Grid

Le projet g-INFO décrit dans le chapitre 4 de ce manuscrit s'est développé dans le cadre très large d'une réflexion internationale autour de l'utilisation des grilles pour la santé publique globale. Cette réflexion a démarré en 2002 lors des premiers contacts entre le projet européen EGEE et l'Organisation Mondiale de la Santé (OMS).

Ces contacts se sont ensuite élargis à d'autres acteurs de la grille et de la santé publique et le concept de la Grille Globale de Santé Publique (Global Public Health Grid - GPHG) [63] est né d'une réflexion commune entre l'Organisation Mondiale de la Santé (WHO), le Centre for Disease Control (CDC) d'Atlanta et les porteurs de l'initiative HealthGrid en Europe, notamment le CNRS et l'Université de Bristol.

La Grille Globale de Santé Publique a pour objectif de permettre l'échange global de données et le développement collaboratif de systèmes, outils et services interopérables. Elle vise à améliorer la santé publique globale en fournissant une infrastructure informatique basée sur des standards et des services pour le développement d'une large palette d'applications de santé publique dans un cadre collaboratif très ouvert.

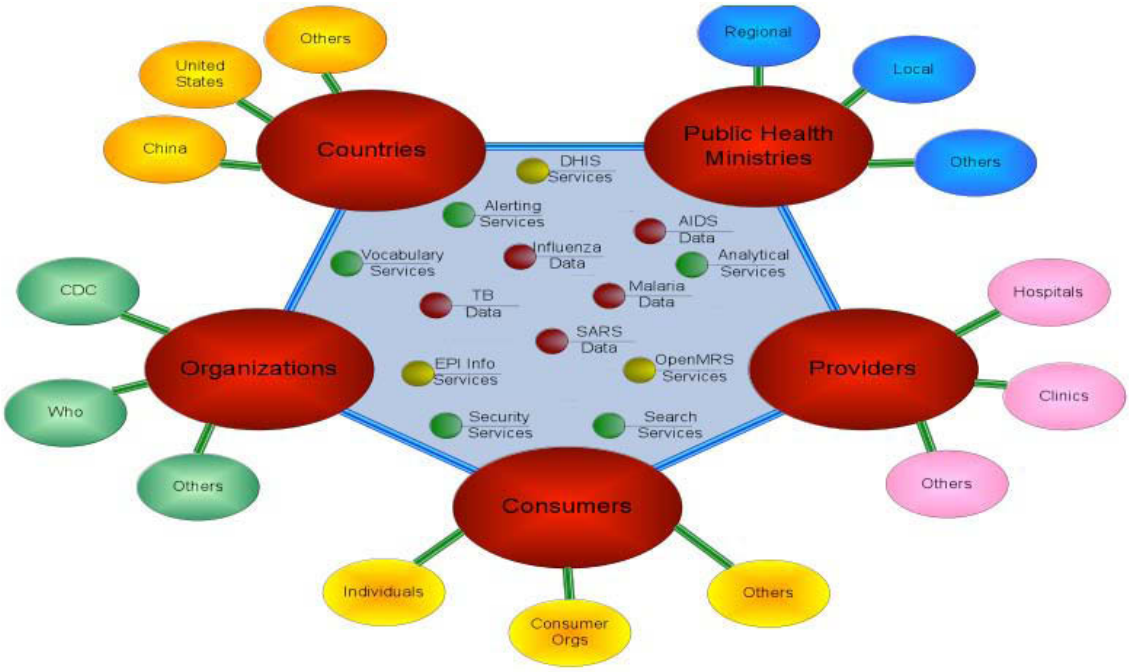


Figure 2-18 – Global Public Health Grid

La GPHG doit venir en soutien des efforts de l'Organisation Mondiale de la Santé pour créer un Observatoire Global de Santé (Global Health Observatory - GHO) dont le but est d'accélérer les échanges et l'accès aux données, encourager la transparence et promouvoir l'utilisation de standards. La figure ci-dessous présente les acteurs d'une telle initiative en termes de fournisseurs et consommateurs de ressources et de services.

L'installation de nœuds de la GPHG dans les Centres Régionaux de l'OMS et dans les Ministères de la Santé constitue une voie très prometteuse dans cette optique.

Le concept de Grille Globale pour la Santé Publique a suscité l'intérêt à un très haut niveau de l'Organisation Mondiale de la Santé, du Center for Disease Control d'Atlanta et de la Commission Européenne en 2009. Pour en démontrer la faisabilité et l'impact potentiel, l'idée de développer des démonstrateurs a grandi et c'est dans ce contexte que le projet g-INFO a vu le jour. Son objectif était de démontrer qu'une grille pouvait significativement améliorer la surveillance des maladies émergentes dans le monde en permettant aux acteurs en première ligne de s'appuyer sur la grille pour accéder à l'ensemble des données mondiales et disposer de la puissance de milliers d'ordinateurs pour une caractérisation rapide des souches responsables de nouveaux foyers épidémiques. L'intérêt spécifique dans le cadre de cette thèse est que le Vietnam a été au cours des dernières années en première ligne de plusieurs maladies émergentes, notamment du *Syndrome Respiratoire Aigu Sévère (SRAS)* et de la grippe aviaire H5N1. Nous présenterons dans le chapitre 4 le contexte spécifique de la lutte contre les maladies émergentes au Vietnam.

2.5.3. Métagénomique

Comme nous l'avons vu précédemment, la métagénomique est confrontée par

ses objets d'études et ses procédés aux défis de l'analyse des données de séquençage à haut débit liés aux volumes énormes de données et aux temps d´exécution des algorithmes.

Sur la base de l'analyse d'un groupe de travail sur les e-Infrastructures pour la biologie et la génomique à grande échelle [64], on peut distinguer deux besoins principaux:

- la production des données brutes (séquences), qui réclame des moyens importants de stockage et d'archivage et des moyens relativement modérés de calculs. Ces infrastructures doivent être proches des lieux de production, en raison, en particulier, des transferts fréquents de données très volumineuses qui excluent, pour l'instant, l'utilisation du réseau internet standard.
- le traitement/service/support bio-informatique permettant de produire des données 'filtrées' à plus forte valeur ajoutée et exploitables par les communautés intéressées (biologie/médecine/agronomie). Les volumes de stockage sur disque y sont d'égale importance mais les ressources de calcul nécessaires y sont beaucoup plus conséquentes. Par ailleurs, le bon fonctionnement de ce type de centre de calcul repose sur l'existence d'une expertise bio-informatique, en lien avec la recherche, actuellement distribuée sur le territoire national.

L'utilisation de la grille a été étudiée pour ces deux besoins.

2.5.3.1. Utilisation de la grille GRISBI pour le traitement de données NGS

Les approches de séquençage de nouvelle génération (NGS) ont permis et permettent désormais d'obtenir très rapidement des séquences de génome entier. Cependant, l'obtention de séquences « de haute qualité », totalement assemblées, sans régions manquantes ni ambiguïté reste problématique. En effet, dans le cas de séquençage de novo, l'étape de l'assemblage est très coûteuse en temps de calcul et en mémoire.

Afin d'évaluer la faisabilité et l'apport d'une grille de calcul telle que RENABI GRISBI pour l'analyse de données de type NGS, les auteurs de [65] ont défini plusieurs cas d'utilisation et testé plusieurs solutions logicielles sur des jeux de lectures artificiels simulant des séquençages de type Illumina et de type Roche 454. Après comparaison avec d'autres infrastructures de calculs, les premiers résultats ont démontré la faisabilité et l'apport considérable de l'utilisation d'une grille de calcul pour ces traitements. En effet, d'une part, les temps de calculs sont réduits par rapport à des serveurs locaux de capacité moyenne, mais il est également possible, de lancer, de n'importe où, plusieurs analyses simultanées dont un paramètre varie, comme pour le cas d'assemblages de novo. Une telle infrastructure distribuée est un atout lorsque les ressources de calcul, de mémoire, ou de stockage ne sont pas accessibles localement.

2.5.3.2. PUMA2

Le système PUMA2 [66] est un environnement interactif intégré du NCBI basé

sur la grille pour l'analyse de séquences génétiques à haut débit et la reconstruction métabolique à partir des données de séquence. PUMA2 fournit un framework d'analyse des données génomiques comparative et évolutive et des réseaux métaboliques dans le contexte de la taxonomie et de l'information phénotypique. L'infrastructure de grille est utilisée pour effectuer des tâches de calcul intensif.

PUMA2 prend en charge l'annotation de génomes automatisée et interactive, en utilisant une variété d'outils bioinformatiques publics. Il contient également une suite d'outils (BLAST, Blocks, Pfam, Chisel, etc.) pour l'attribution automatique de la fonction des gènes, l'analyse évolutionaire des familles de protéines et l'analyse comparative des voies métaboliques. PUMA2 permet aux utilisateurs de soumettre des données de séquence pour l'analyse fonctionnelle automatisée et la construction de modèles métaboliques. Les résultats de ces analyses sont mis à la disposition des utilisateurs dans l'environnement PUMA2 pour l'analyse et l'annotation des séquences.

PUMA2 diffère de autres solutions (KEGG [67], MetaCyc [68], WIT2 [69]) sur les points suivants:

- Il soutient le développement interactive des modèles d'utilisation pour les génomes publics et soumis par l'utilisateur
- Il utilise la technologie de grille pour les tâches de calcul intensif;

Il fournit de nouveaux outils pour l'analyse comparative des génomes et des réseaux métaboliques dans le cadre de la taxonomie et de l'information phénotypique.

2.5.3.3. GNARE

GNARE (GeNome Analysis Research Environment) [70] est un serveur bioinformatique du NCBI qui prend en charge l'analyse automatisé et interactif des génomes et métagénomes soumis par les utilisateurs. Ces analyses incluent la prédiction de la fonction des gènes et le développement de la reconstructions métaboliques des organismes spécifiques à partir des données de séquence. GNARE fournit un *framework* d'analyse comparative et évolutive ainsi que l'annotation des génomes et des réseaux métaboliques dans le contexte de l'information phénotypique et taxonomique. Les résultats des analyses et des modèles métaboliques sont visualisés et abondamment annotés avec des informations à partir de bases de données publiques. GNARE utilise des pipelines de traitement automatisés déployés sur la grille de calcul pour effectuer l´analyse à haut débit des génomes. GNARE permet à des groupes scientifiques et utilisateurs individuels d'analyser les données génomiques en utilisant un environnement bioinformatique intégré et automatisé sur la grille.

GNARE utilise des fonctionnalités du système PUMA2 pour la représentation des résultats d'analyse. Divers modèles sont utilisés dans PUMA2 : page protéique représentant les résultats du BLAST ou Blocs, reconstruction métabolique et comparaison des voies métaboliques. GNARE utilise GADU

pour accéder à des milliers de processeurs de diverses ressources de la grille à grande échelle tels que l'Open Science Grid (OSG) (http://www.opensciencegrid.org) et TeraGrid (http://www.teragrid.org).

2.5.3.4. metaSHARK

Le projet metaSHARK (metabolic SearcH And Reconstruction Kit) [71] offre aux utilisateurs une interface intuitive, de manière totalement interactive pour explorer le réseau métabolique KEGG [67] via un navigateur Web. Les informations de reconstruction métabolique pour des organismes spécifiques, produit par l'outil automatisé SHARKhunt [72] ou d'autres programmes, peuvent être téléchargées sur le site et superposées sur le réseau générique. Les données additionnelles peuvent également être incorporées, permettant la visualisation de l'expression du gène dans le contexte du réseau métabolique prédit. metaSHARK est différent des autres solutions comme KEGG ou PUMA2 par la visualisation du réseau de données en utilisant SHARKView. Les visualisations de voies métaboliques sont entièrement personnalisables, permettant aux biologistes d'explorer le réseau de voisinage des enzymes et de formuler des itinéraires hypothétiques pour la synthèse ou le catabolisme des composés particuliers.

Le projet metaShark est originalement une solution non-grille mais il peut utiliser SHARKhunt en parallel avec une grille basée sur Condor [73].

2.5.3.5. MetaVIR

MetaVIR [74] est un serveur web dédié à l'analyse de métagénomes viraux. En plus des approches classiques pour analyser les métagénomes, MetaVIR permet d'explorer la diversité virale en construisant automatiquement des phylogénies pour des gènes marqueurs sélectionnés, d'estimer la richesse en gènes en générant des courbes de raréfaction et de faire des comparaisons avec d'autres viromes en utilisant des similarités de séquences. Les calculs de MetaVIR sont parallélisés sur un cluster au Centre Régional de Ressources Informatiques du PRES Clermont Universités qui joue le rôle de mésocentre régional.

2.6. Conclusion

Dans ce chapitre, nous avons essayé de présenter une vue globale de l'utilisation des grilles en bioinformatique en présentant les infrastructures et les outils utilisés ainsi que les applications actuelles dans les domaines d'intérêt spécifique pour ce mémoire, à savoir l'épidémiologie et la métagénomique.

Depuis plus de 10 ans, la biologie moléculaire a constitué, après la physique des particules, le domaine privilégié de déploiement d'applications scientifiques sur la grille. L'arrivée des nouvelles générations de séquenceurs à haut débit ne fait que poursuivre la croissance exponentielle du besoin en traitement et en stockage de données des biologistes.

L'épidémiologie ne génère pas des besoins de traitement comparables à la génomique. Par contre, la nature des données à traiter est distribuée,

notamment dans les problématiques liées à la surveillance des maladies émergentes. La valeur ajoutée de la grille va être sa capacité à fédérer des bases de données hétérogènes et à automatiser des traitements complexes.

Mais un des freins majeurs à l'adoption des grilles a été la difficulté d'utilisation de ces infrastructures informatiques et leur rigidité. Nous allons dans le prochain chapitre présenter la plate-forme WISDOM, conçue dès sa genèse pour le déploiement d'applications scientifiques dans le domaine des sciences du vivant sur des ressources hétérogènes.

CHAPITRE 3 : WISDOM

En général, les utilisateurs de la grille ne travaillent pas directement sur la grille. En fait, ils utilisent des intergiciels ou au niveau plus haut, des outils ou plate-formes, des portails, etc. Dans ce chapitre, nous étudions en détail un tel outil, une plate-forme parmi celles présentées brièvement dans le chapitre précédent : L'Environnement de Production WISDOM (**W**orld-wide **I**n **S**ilico **D**ocking **O**n **M**alaria Production Environment - WPE).

La première version de WISDOM a été développée en 2005 au Laboratoire de Physique Corpusculaire (LPC) à Clermont-Ferrand (France), pour répondre au besoin du déploiement de centaines de milliers de calculs de criblage virtuel pour la recherche de nouveaux médicaments *in silico*. L'outil a ensuite évolué pour devenir un environnement (plate-forme) capable de gérer les déploiements de tâches sur la grille et sur des clusters. Il facilite l'accès à la Grille pour les non-experts grâce à un ensemble de script Shell permettant d'automatiser le déploiement et la surveillance des tâches. La plate-forme a été utilisée efficacement pour plusieurs projets scientifiques.

Dans ce chapitre, nous allons d´abord présenter brièvement l'histoire de WISDOM. En suite, l'architecture informatique du WISDOM Production Environment (WPE) et ses composants, tels qu'ils ont été conçus et développés au LPC de 2005 jusqu´à très récemment seront présentés en grands détails. Comme WISDOM utilise principalement les ressources gérées par l'intergiciel gLite, nous rappellerons le fonctionnement de gLite et le rôle de ses principaux services. Malgré un processus continu d'amélioration, les générations successives du WPE présentaient encore des limitations en termes de disponibilité, de stabilité et de connexion entre les services au démarrage de cette thèse. Nous avons effectué une série des tests de performances pour détecter et identifier des limitations de la plate-forme. Ensuite, nous avons proposé des modifications afin de remédier à ces limitations. Ces tests, les modifications qui ont suivi et leur impact sur la performance de la plate-forme seront décrits dans la suite du chapitre.

3.1. Présentation historique de WISDOM

Comme son nom l'indique, WISDOM (World-wide In Silico Docking On Malaria) est né en 2005 d´une collaboration scientifique entre le LPC et l'Institut Fraunhofer en Allemange dont l'objectif était de déployer des calculs à grande échelle sur la grille pour les criblages virtuels (dockings) pour la recherche de nouveaux médicaments contre le paludisme.

L´auteur de WISDOM est Jean Salzemann, ingénieur de recherches au LPC du 2004 à 2010. Il est le développeur principal de cette plate-forme, en étroite

collaboration avec Vincent Bloch, aussi ingénieur de recherches au LPC pendant la même période.

Des déploiements à grande échelle se sont succédés de 2005 à 2011 en ciblant des maladies différentes.

En 2005, le premier déploiement à grande échelle sur l'infrastructure de la grille européenne EGEE, baptisé data challenge WISDOM-I, a utilisé avec succès plus de 80 années de temps CPU en 6 semaines pour une première expérience « *in silico* » [75], [76], [77]. Il s'agissait de tester environ un million de composés chimiques médicaments potentiels avec deux logiciels de docking sur 5 protéines de la famille des plasmepsines, cible biologique potentielle pour combattre le paludisme. Cette première étape a permis de sélectionner environ 100.000 composés qui ont fait l'objet d'une deuxième étape de tri [78] par une analyse de dynamique moléculaire. Celle-ci a permis de réduire à 100 le nombre de composés qui ont fait l'objet d'une expérimentation biologique in vitro et finalement in vivo. En parallèle avec cette chaine complète d'analyse des résultats du premier data challenge WISDOM-I, deux autres « data challenges » ont été déployés en 2006 et en 2007:

- En 2006, au moment où les premiers cas de grippe aviaire étaient confirmés en France, une collaboration internationale [80] étudiait l'impact de certaines mutations de la neuraminidase N1 sur l'efficacité des traitements de la grippe H5N1 sur plusieurs milliers de processeurs appartenant à trois infrastructures de grille différentes (Auvergrid (France), EGEE (Europe) et TWGrid (Taiwan)). Ces études montrèrent que certaines mutations impactaient très significativement l'efficacité du tamiflu. Elles permirent aussi de tester 300.000 composés chimiques et d'en sélectionner environ 2000 dont les plus prometteurs ont fait l'objet de tests in vitro en Corée du Sud. Ces travaux [80] ont inspiré la vision de l'utilisation de la grille pour la surveillance de la grippe aviaire qui a conduit à la plate-forme g-INFO, présentée dans le chapitre suivant de cette thèse.
- En 2007, le data challenge WISDOM-II [79] mobilisait environ 4 siècles de temps CPU pour le criblage virtuel de quatre autres cibles biologiques d'intérêt pour la lutte contre le paludisme, élargissant la recherche à des cibles biologiques du parasite Plasmodium vivax, vecteur d'une forme atténuée de paludisme.

En parallèle de l'analyse des résultats obtenus au cours de ces différentes campagnes de calcul, l'activité dans le cadre de la collaboration WISDOM s'est développée en Corée du Sud où plusieurs centaines de milliers de composés chimiques ont été testés sur des cibles biologiques du SARS [81] et du diabète en 2010 et 2011 [82].

Un certain nombre de molécules actives ont fait l'objet d'un dépôt de brevets. Si l'environnement de production WISDOM a été utilisé de façon continue depuis 2005, la stratégie d'utilisation de la grille a elle évolué très

significativement depuis 2005.

Au départ, la grille était utilisée par WISDOM de façon très simple en soumettant les tâches de façon massive. L'environnement de production WISDOM parvenait ainsi à un déploiement à grande échelle au prix d'une re-soumission manuelle d'un grand nombre de tâches. Mais le taux d'échec restait élevé malgré des améliorations permanentes de l'environnement [83]. Les principales limitations restaient liées aux performances des machines courtier de ressources (*resource brockers*) et à la stabilité des nœuds de calcul de la grille.

Pour contourner ces difficultés, l'environnement de production a été complètement repensé pour adopter l'utilisation des jobs pilotes. Un job pilote est un job générique soumis par un utilisateur sur la grille dont le rôle est de réserver une ressource de la grille qui sera utilisée ultérieurement par un programme réel. Un job pilote peut être utilisé pour lancer plusieurs programmes. Ce changement de stratégie a permis d'améliorer significativement les performances et de s'affranchir de certaines limitations.

La nouvelle stratégie a été testée pour recalculer toutes les structures de la Protein Data Bank [84], la base de données mondiales rassemblant toutes les structures tridimensionnelles connues de macromolécules biologiques. Environ 17000 entrées de la PDB ont ainsi été recalculées en quelques semaines pour l'équivalent de 17 années CPU [85].

Nous allons maintenant présenter en détails l'architecture du WISDOM Production Environment (WPE).

3.2. Architecture informatique du WPE

Le WPE peut être considéré comme un intergiciel installé sur des ressources de calcul pour gérer des données et des jobs et pour partager la charge (*workload*) sur l'ensemble des ressources intégrées, même si elles utilisent des technologies différentes. Basé sur cet intergiciel, il est possible de construire des services web qui interagissent avec le système. L'intergiciel est considéré comme un ensemble de services génériques fonctionnant comme un niveau d'abstraction supplémentaire pour que les services d'application puissent utiliser l'un des systèmes sous-jacents de manière transparente.

Le WPE est composé de 3 éléments principaux (Figure 3-1):

- Le **Gestionnaire de Tâches** (TM-*Task Manager*) interagit avec le client et stocke les tâches à accomplir ;
- Le **Gestionnaire de Jobs** (JM - *Job Manager*) soumet les travaux (*jobs*) aux éléments de calcul (Compute Element - CE) sur la grille, où les tâches gérées par le TM seront exécutées;
- Le **Service d'information de WISDOM** (WIS - *WISDOM Information Service*) utilise AMGA (voir section 3.2.4.1) pour stocker les méta-donnés nécessaires pour le JM et les configurations du WPE.

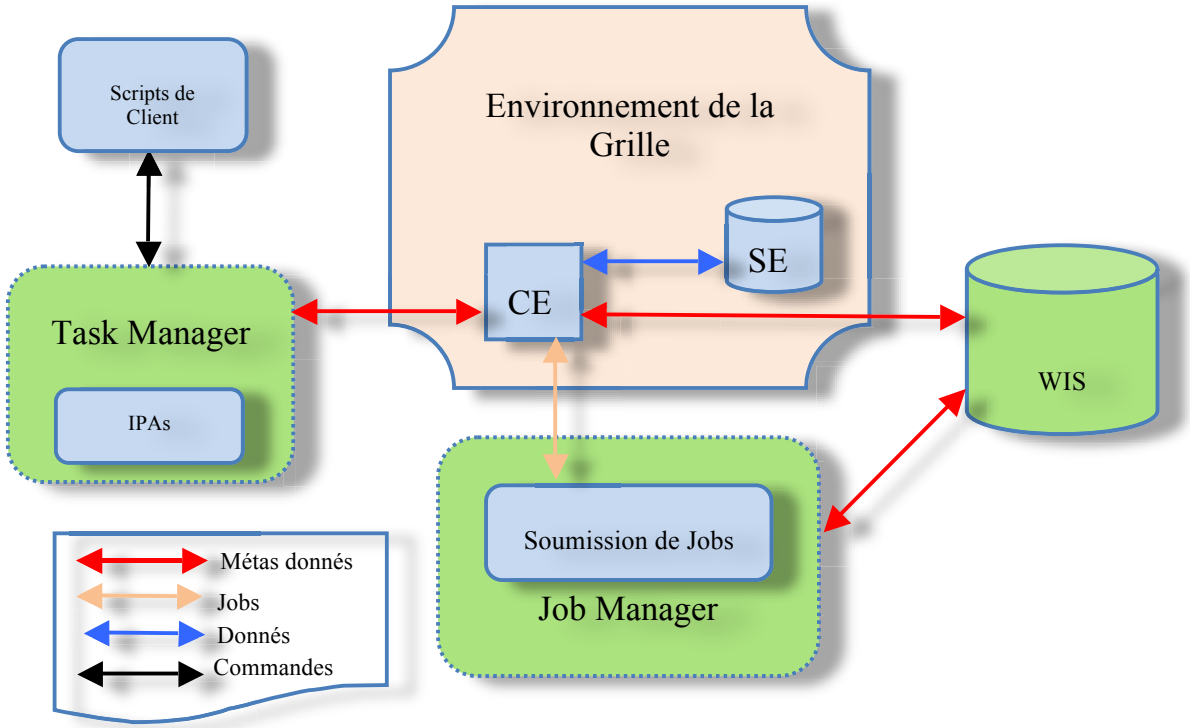


Figure 3-1 - Architecture du WPE

Tous ces 3 éléments sont déployés sur l'infrastructure de la grille basée sur l'intergiciel gLite. Avant de décrire ces trois éléments, nous allons présenter brièvement l'intergiciel gLite et ses principaux services.

3.2.1. Introduction de gLite

La distribution gLite [5] est un ensemble intégré de composants conçus pour permettre le partage des ressources. En d'autres termes, c´est un intergiciel pour la construction d'une grille.

L'intergiciel gLite a été réalisé par le projet EGEE [86] et il est actuellement mis au point par le projet EMI[1]. En plus du code développé au sein du projet, la distribution gLite rassemble les contributions de nombreux autres projets, y compris LCG [87] et VDT (http://vdt.cs.wisc.edu/). Le modèle de distribution est de construire différents services («node-types») de ces composants, puis assurer une installation et une configuration faciles sur les plates-formes choisies (actuellement Scientific Linux 4 et 5, et également Debian 4 pour les Nœuds de Calcul).

Le rôle de gLite est de faciliter l'utilisation de ressources (calcul et stockage) distribuées géographiquement. La soumission d'un job sur une grille utilisant gLite nécessite l'utilisation de logiciels spécifiques, fournis par l'Interface Utilisateur (User Interface - UI).

La Figure 3-2 présente une vue schématique de ces services ainsi que leurs

[1] http://www.eu-emi.eu/

interactions. Les acronymes sont expliqués dans le Tableau 3-1. L'utilisateur gère les jobs à partir d´une Interface Utilisateur (UI). Avant de soumettre le job au Système de Gestion de Charge (Workload Management System - WMS), il peut copier des fichiers nécessaires pour le job sur un Elément de Stockage (Storage Element - SE). Ces données seront accessibles depuis les nœuds de calcul. Une fois le job transféré sur le WMS, ce service choisit un Elément de Calcul (Compute Element – CE) devant effectuer les calculs. Pour réaliser ce choix, il se base sur la disponibilité de l'Elément de Calcul et le respect des prérequis du job définis par l'utilisateur. Une fois qu'un CE est sélectionné, le WMS soumet le job à ce CE. Le CE distribue les calculs sur ses nœuds de calcul (Worker Node - WN) et collecte les résultats qui sont ensuite transférés au WMS. Pendant ce temps, l'utilisateur peut interroger le WMS pour connaître l'état de son job. Enfin, les résultats sont disponibles et peuvent être récupérés à partir du WMS et/ou du SE.

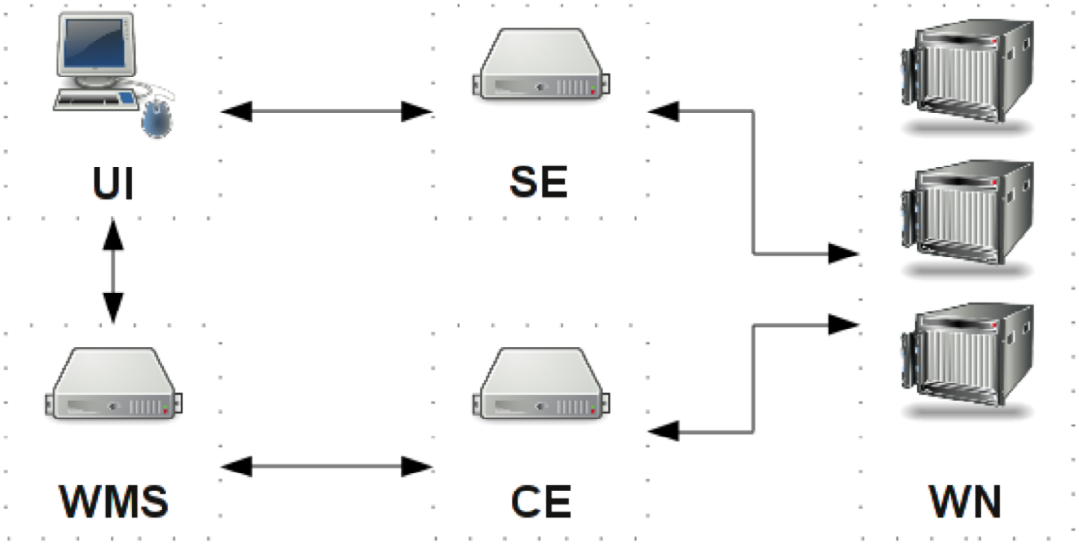


Figure 3-2 – Shémas des services de gLite

Élément	Rôle
UI	L'UI (User Interface) est l'interface utilisateur gLite. C'est une machine à partir de la quelle l'utilisateur va soumettre des jobs sur la grille. A travers cette interface, l'utilisateur interagit avec le WMS pour la soumission des jobs et avec le SE pour le transfert des fichiers. Cette interface permet de : • gérer un job (soumission, suivi de l'état, annulation) ; • récupérer les résultats d'un job ; • copier, répliquer ou supprimer des fichiers sur la grille.
SE	Le SE (Storage Element) est l'élément d'une grille pour gérer le

Élément	Rôle
	stockage. Il est accessible via différents protocoles.
WMS	Le WMS (Workload Manager) est le serveur qui aide l'utilisateur à soumettre des jobs, les suivre et récupérer des résultats. Le WMS choisit le nœud de grille (CE) correspondant au pré-requis du job et interagit avec le CE pour la soumission des jobs, le transfert des sandbox et le suivi de l'état d'un job.
CE	Le CE (Computing Element) est le serveur interagissant directement avec le gestionnaire de queues.
WN	Le WN (Worker Node) est le serveur effectuant les calculs d'un job. Il peut se connecter au SE pour récupérer les données nécessaires au calcul. Il peut également copier les résultats du calcul sur le SE.

Tableau 3-1 - Rôles des éléments de gLite

3.2.2. Task Manager (TM)

L'objectif du *Task Manager* est de fournir des tâches aux jobs tournant sur n'importe quelle infrastructure de calcul selon les capacités des jobs. Dans le WPE, les calculs scientifiques à accomplir sont représentés par des tâches. Les jobs sont génériques et ils peuvent exécuter n'importe quelle tâche. Dans le cas de la grille de calcul, le TM permet de diminuer le temps d'attente de la soumission des jobs. En effet, supposons que nous ayons 1000 calculs similaires à exécuter sur la grille. La solution normale est de soumettre 1000 jobs, chaque job correspond avec un calcul. Supposons maintenant que l'on ne puisse pas soumettre 1000 jobs à la fois à cause de la surcharge sur les WMS. On ne peut soumettre, par exemple, que 250 jobs à la fois. Il faut alors soumettre ces 1000 jobs en 4 fois, chaque fois 250. Si certains jobs échouent (cela arrive assez souvent sur la grille), ils doivent être resoumis. Cette méthode manuelle fait perdre beaucoup de temps pour attendre les jobs qui se terminent avec succès ou échec. Le TM propose une solution élégante pour surmonter ce problème.

L'idée principale du TM est d'utiliser les jobs génériques pour lancer des tâches. Un job générique est un job normal sur la grille mais il peut exécuter n'importe quelle tâche du TM. Au lieu d'écrire et de soumettre des jobs différents pour résoudre un problème, nous écrivons et créons des tâches et utilisons les jobs génériques pour lancer ces tâches. Parce que les jobs sont indépendants des tâches, un job, après avoir exécuté une tâche, peut être réutilisé pour lancer une autre tâche. Par conséquent, les tâches peuvent être exécutées immédiatement à condition que des jobs génériques soient toujours disponibles sur la grille.

Le TM fonctionne via les services web. Il faut avoir donc un serveur web et des scripts clients. Le serveur web du TM contient les services web qui peuvent être appelés à partir de n'importe quelle machine sur la grille. Les clients du TM sont les scripts écrits en langage C. Afin de manipuler des tâches, l'utilisateur utilise ces scripts pour appeler les services web. Les services web

du TM sont les suivants :

- `createTask` : créer une nouvelle tâche et la stocker dans le TM
- `getTask` : récupérer une tâche dans la liste d'attente du TM.
- `setTaskWaiting` : remettre une tâche dans la liste d'attente
- `deleteTask` : enlever une tâche de la liste d'attente
- `getStatus` : récupérer les états d'une tâche
- `getInfo` : récupérer les états des tâches d'un service

Afin de gérer l'accès concurrent, le TM est basé sur le système de fichier Unix. Chaque tâche est un fichier ayant un nom unique assigné lorsque la tâche est créée. Les jobs sur la grille utilisent les scripts clients pour interagir avec le TM. Un job peut demander s'il y a une tâche à exécuter puis la récupérer et l'exécuter. La Figure 3-3 illustre le fonctionnement du TM.

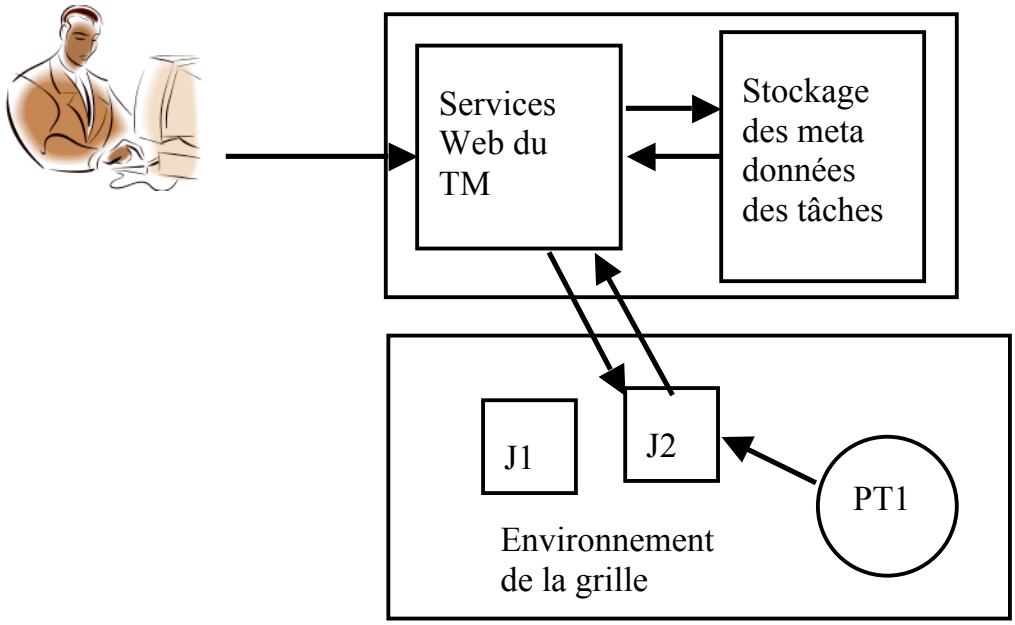


Figure 3-3 - *Fonctionnement du Task Manager*

Tout d'abord, l'utilisateur doit emballer un paquet de tâche (PT1 par exemple) et le sauvegarder sur la grille. Un paquet de tâche contient un fichier exécutable (le programme principal) et tous les fichiers nécessaires pour exécuter ce programme. Par exemple, un paquet de tâches de *docking* peut contenir le programme Autodock et tous les fichiers nécessaires pour exécuter Autodock. A chaque fois que l'utilisateur lance une tâche sur la grille, il ne fait que créer la tâche dans le TM en utilisant un script client. Toutes les informations concernant cette tâche (nom, paramètres, ...) sont stockées dans la zone de stockage des métadonnées (Figure 3-3). Pour lancer une tâche, il faut avoir un

job générique disponible sur la grille. Un job générique est soumis par un utilisateur sur la grille mais il n'est pas un programme réel.

En fait, il ne sert qu'à lancer un autre programme réel que nous appelons tâche. Un job générique peut lancer plusieurs programmes (tâches) l´un après l´autre. Si un job générique (J2 sur la Figure 3-2) sur la grille est disponible, il contacte le TM et demande s'il reste une tâche à exécuter. Le TM lui donne les informations d'une tâche. Le job peut maintenant télécharger sur la grille le paquet correspondant à la tâche (PT1 – dans ce cas) et lancer le programme principal dans le paquet avec ses paramètres. Après avoir fini la tâche, le job est à nouveau disponible et il peut lancer une autre tâche. Cette approche s'appelle le « pull model » et elle sera présentée en détails dans la section 3.2.3.2.

Dans le cas de l'exemple cité au premier paragraphe de cette section, l'utilisateur doit d'abord soumettre un nombre de jobs génériques sur la grille (dans la limite de ressources disponibles, soit 250). Si toutes les ressources sont disponibles, il peut créer 1000 tâches dans le TM. Les 250 jobs maintenant s'occupent de lancer ces 1000 tâches donc chaque job peut lancer approximativement 4 tâches. Dans cet exemple, l'utilisateur a besoin de 250 jobs. Il peut soumettre ces jobs manuellement mais il peut aussi utiliser les jobs soumis par le Job Manager expliqué dans la section suivante.

3.2.3. Job Manager (JM)

L'objectif du JM est de soumettre des jobs sur la grille à la place de l'utilisateur. Celui-ci a juste besoin de définir le nombre de jobs génériques qu'il souhaiterait avoir ainsi que l'organisation virtuelle (VO) qu'il veut utiliser et le JM se chargera de sélectionner les ressources disponibles et d'y envoyer les jobs. Le JM se chargera ensuite de suivre l'évolution des jobs et, entre autre, de resoumettre les jobs qui n'ont pas réussi à tourner ou qui se sont arrêtés. L'objectif est donc de réaliser de manière automatique et à grande échelle les actions que ferait un utilisateur, à savoir :

- Évaluation des ressources disponibles
- Soumission des jobs sur les ressources les plus appropriées
- Suivi des jobs et prise de décision en cas d'erreurs

Le JM est basé sur l'intergiciel gLite dont il utilise la plupart des services. Le JM n'a pas pour but de contourner gLite mais bien d'essayer d'exploiter aux mieux ses services. Les jobs sont donc bien soumis selon l'approche habituelle pour gLite, en utilisant le Système de Gestion de Charge (*Workload Management System* - *WMS*). Les jobs soumis par le JM sont des agents ou jobs pilotes qui ont pour objectif de récupérer des tâches auprès du TM et de les exécuter.

3.2.3.1. Fonctionnement du Job Manager

A son lancement, le JM lit un fichier de configuration. Ensuite, il fait un état des lieux des ressources disponibles sur la grille. Il commence par faire la liste

des WMS disponibles pour l'Organisation Virtuelle (*Virtual Organisation - VO*) choisie et faire la liste des Éléments de Calcul (*Computing Element - CE*) connus par ces WMS. En effet, comme tous les WMS n'utilisent pas les mêmes systèmes d´information de la grille (*Berkeley Database Information Index - BDII*), il est nécessaire de vérifier quelles sont les ressources connues par un WMS pour éviter que le JM ne demande à un WMS de soumettre un job sur une ressource qui lui est inconnue. Le JM pour chaque CE essaye de récupérer différentes informations afin d'obtenir un état des lieux le plus proche possible de la réalité. Le JM va donc établir une liste des sites en incluant les informations suivantes: le nombre de CPUs total, le nombre de CPUs disponibles et le nombre de jobs en attente. Le JM exclut les sites que l'utilisateur aurait préalablement choisi de ne pas utiliser. Dans la mesure où l´approche « pull model » est utilisée pour minimiser les temps de soumission, Le JM utilise toutes ces informations pour classer les différents sites en affectant des scores plus ou moins élevés en fonction des ressources disponibles. La stratégie d´utilisation de files d´attente courtes où un job sera arrêté au bout de 10 à 30 minutes d´inactivité (sans tâche) peut réduire fortement l'intérêt de cette approche. Plus un site aura des ressources disponibles, plus son rang sera élevé. Le JM choisit toujours le site qui a le meilleur rang, sachant que ce rang évolue au cours du temps.

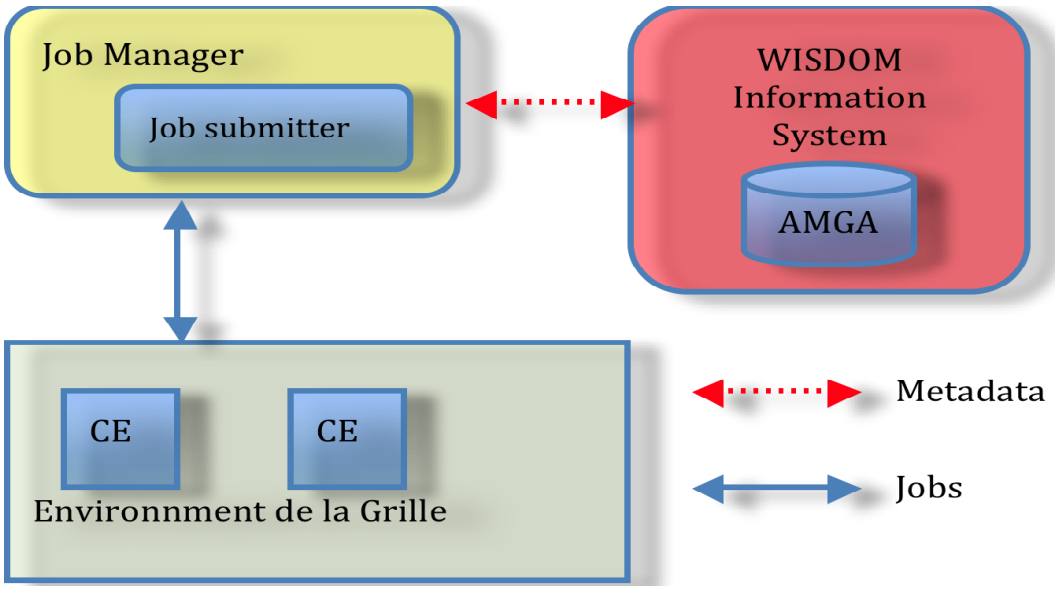


Figure 3-4 – Schéma du Job Manager

L'étape suivante est la soumission réelle des jobs. La soumission se fait selon le modèle « producteur – consommateur » avec d'un côté une liste représentant les jobs qui doivent être soumis et de l'autre, la liste des WMS. Si la liste des jobs à soumettre n'est pas vide et qu'un WMS est libre, il va prendre un job dans la liste et tenter de le soumettre. Chaque fois qu'un job est envoyé vers un CE, un compteur lié à ce CE est incrémenté. Ce compteur est utilisé dans le calcul du rang du CE et permet de diminuer le classement du site en attendant que ces informations soient publiées dans le BDII. Sans ce compteur, il y a de fortes chances que l'on soumette trop de jobs vers le site offrant le plus de ressources au départ, avec le risque que la majorité des jobs finisse leur vie

dans la file d'attente du site au lieu de tourner sur d'autres sites.

En parallèle à la soumission, le JM effectue le suivi des job soumis selon le même modèle « producteur – consommateur » avec d'un côté la liste des jobs soumis et de l'autre, la liste des WMS. Les WMS prennent un job dans la liste, vérifient son statut et effectuent l'action la plus appropriée selon son statut. Les actions possibles sont :

- Ne rien faire si le statut est « *waiting* », « *ready* », « *scheduled* » ou « *running* »
- Resoumettre le job si le statut est « *aborted* »
- Récupérer les résultats et resoumettre le job si le statut est « *done* »

Lorsqu'un job est à resoumettre, la resoumission est faite par le WMS qui a vérifié son statut. Si un job a échoué sur un site (statut « *aborted* »), le rang du site est diminué afin de favoriser les sites qui font tourner les jobs correctement. Il existe cependant un cas particulier : celui des jobs qui échouent à cause de l'expiration du proxy associé. En effet dans le modèle « pull», les jobs sont paramétrés pour tourner aussi longtemps que possible, ce qui signifie le plus souvent jusqu'à ce que le proxy expire. Si un job s'arrête à l'expiration du proxy, on ne le considère pas comme un dysfonctionnement du site mais comme une situation normale et le site n'est pas pénalisé.

3.2.3.2. Pull Model

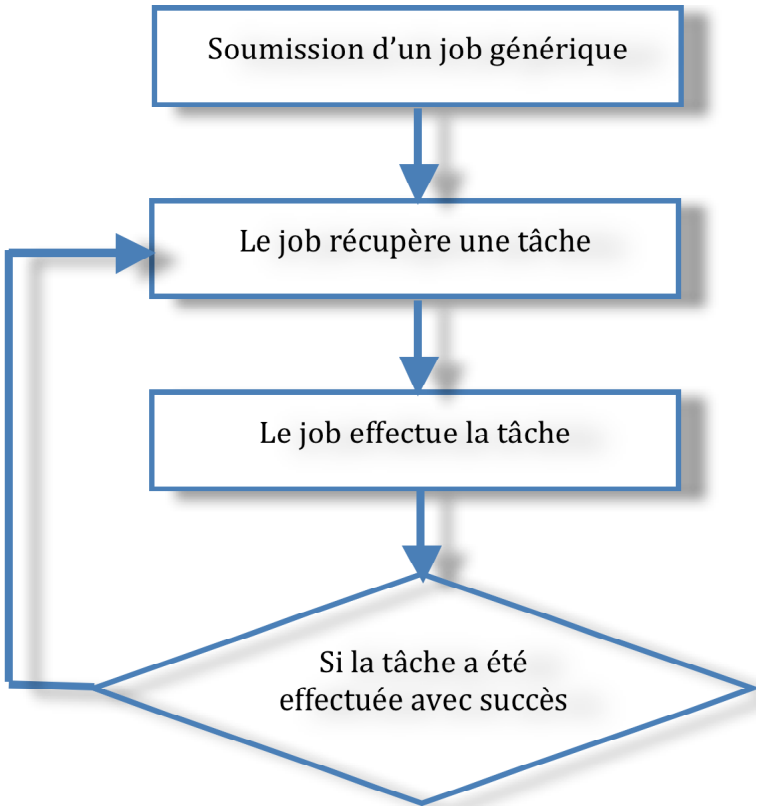


Figure 3-5 – Schéma du modèle Pull du Job Manager

Le modèle « pull » permet de se concentrer plus sur les tâches à effectuer que sur les jobs en eux-mêmes afin de limiter au maximum les temps d'attente des tâches et d'optimiser l'utilisation des jobs. Lors d'une soumission « normale », l'utilisateur doit définir les données à récupérer. Cela implique en particulier que pour que toutes les tâches soient effectuées, il est nécessaire que tous les jobs tournent avec succès. De plus, c'est à l'utilisateur d'effectuer l'évaluation de la charge de chaque job.

La solution pour ce problème est d'utiliser une charge dynamique. L'objectif est d'exploiter au maximum des jobs qui sont en train de tourner en leur faisant effectuer des tâches tant qu'ils tournent mais de ne pas lier les tâches aux jobs. Le modèle « pull » est illustré dans le Figure 3-5.

Ce modèle a des avantages :

- La charge des jobs est dynamique. Au cours de sa durée de vie, un job sur un processeur puissant pourra fournir plus de résultats que celui sur un processeur moins puissant. La charge des jobs n'est pas nivelée par le bas et l'efficacité des jobs est augmentée.
- Les gestions des jobs et des tâches sont séparées. Il n'est pas nécessaire d'avoir 100% des jobs qui tournent pour effectuer 100% des tâches.
- Si un job est disponible lorsqu'un utilisateur crée une nouvelle tâche, cette dernière peut se mettre à tourner immédiatement. Les temps de soumissions et d'attente sont considérablement réduits. Avec le modèle « pull », il est possible d'utiliser la grille pour des algorithmes requérant des temps de calculs réduits. En effet, sans ce modèle « pull », utiliser la grille pour effectuer une tâche de 2 minutes est totalement inefficace.
- La création à la volée de nouvelles tâches permet la création de chaînes de traitement (flots) simples.
- Les agents génériques peuvent être facilement adaptables à d'autres grilles ou environnements de calculs, les clusters par exemple.

Il y a cependant quelques inconvénients dans l'utilisation du modèle « pull » :

- Les logiciels doivent être adaptés pour pouvoir être appelés par les jobs génériques. La solution simple est de créer un script intermédiaire qui sera appelé par le job et qui se chargera de lancer le logiciel avec les paramètres nécessaires.
- Les résultats de l'exécution d'une tâche doivent être stockés sur la grille ou dans une base de données.
- Les ressources de calcul sont occupées par les jobs génériques. Si un job générique n'a pas de tâches à exécuter, il se met à fonctionner en mode ralenti en quelques secondes (rôle prédéfini dans le JM). Puis il redemande au TM une tâche à exécuter. Ce processus se répète jusqu'à ce qu´il reçoive une tâche ou que le nombre des boucles atteigne une limite prédéfinie. La réservation de ressources est nécessaire si l'on veut

avoir des bons temps de réponse sur de nouvelles tâches, mais elle peut être mal vue par les administrateurs des sites. Il faut donc ne pas soumettre trop des jobs inutiles.

Ce mode contourne les ordonnanceurs mis en place et du coup agit de façon "égoïste". Il offre de bons résultats d'un point de vue individuel mais sur l'ensemble de la plateforme les conséquences peuvent être désastreuses en termes de gaspillage des ressources potentiel.

3.2.4. Système d'Information de WISDOM

Le Système d'information de WISDOM (WIS) a pour but de stocker les métadonnées nécessaires au bon fonctionnement du Job Manager (JM). Il stocke aussi les informations de configuration pour le WPE. Le WIS est déployé sur un serveur AMGA.

3.2.4.1. AMGA

En général, les métadonnées sur la grille sont les informations sur des fichiers. Elles sont utilisées pour décrire les fichiers sur la grille et localiser ces fichiers sur la base de leur contenu. AMGA (ARDA Metadata Grid Application [88]) a été introduit par ARDA (**A R**ealisation of **D**istributed **A**nalysis pour le Large Hadron Collider) comme une interface pour accéder aux métadonnées sur la grille et a été intégré dans la liste des services de l'intergiciel gLite.

AMGA n'est pas seulement l'interface de métadonnées. AMGA fournit aussi une base de données simplifiée sur la grille. Il y a beaucoup d'applications sur la grille qui n'ont besoin que de schémas simples ou de données structurées. AMGA fournit donc un moyen utilisable pour ces applications. Les avantages de AMGA sont :

- Bien que la grille ne supporte pas un format de base de données spécifique (MySQL, DB2, etc.), AMGA fait partie de l'intergiciel gLite.
- AMGA peut utiliser la sécurité technique de la grille pour contrôler les accès aux métadonnées.

L´utilisation d´AMGA est relativement simple et permet de cacher l'hétérogénéité de la base de données sur la grille

3.2.4.2. Structure du WIS

La structure du WIS comprend des répertoires suivants :

- `agent` : les informations des jobs pilotes sont stockées dans ce répertoire. Les attributs importants sont *job_id* et *status*. L'attribut *job_id* est une référence à un job pilote sur la grille et l'attribut *status* est l'état de ce job. Les statuts possibles d'un job pilote sont :
    - *unsubmitted* : le job est créé dans le WIS et prêt pour la soumission
    - *queued* : le job est dans la queue de soumission

- *waiting* : le job est soumis mais il attend d'être exécuté sur un CE.
- *running* : le job est en cours d'exécution et entrain de lancer une tâche
- *idle* : le job est en cours d'exécution mais il ne s'occupe pas de tâches ou il n'y a plus de tâches dans le TM. Dans ce cas, le job dort un certain temps et essaye à nouveau de récupérer une nouvelle tâche quand il se réveille.
- *killed* : le job est tué pendant l'exécution d'une tâche.
- *tobekilled* : si le statut d'un job est toujours *waiting,* il peut être signalé *tobekilled* et le JM pourra le détruire.
- *terminated* : le job se tue lui-même si le nombre maximum de tentatives de récupération d'une tâche sans succès a été atteint. Cela évite qu'un job occupe trop longtemps des ressources pour rien. Un job dont le statut est *terminated* est ignoré par le JM.

- `grid` : ce répertoire stocke les informations de configuration pour des *grilles*. En fait, il peut s'agir de n'importe quelle infrastructure de calcul. Il contient aussi les informations sur les jobs qui tournent sur chaque *grille*. Une *grille* configurée dans le WIS peut être une grille publique ou une grille privée ou même un cluster. Les configurations de chaque *grille* contiennent les informations des CE et des SE. Pour chaque *grille*, une limitation sur le nombre de jobs est définie ici.
- `parameters` : ce répertoire contient les valeurs des paramètres comme le nombre maximum des agents que le JM peut soumettre.

3.2.5. Fontionnement du WPE

Le fonctionnement de l´Environnement de Production de WISDOM (WPE) peut être décrit par le fonctionnement du JM. Lorsque le JM fonctionne, il interagit avec le reste du WPE. Le JM peut fonctionner en mode Maître ou en mode Esclave. Pour chaque session, une seule instance de Maître et plusieurs instances d'Esclave peuvent être lancées. Le Maître s'occupe de la charge des jobs. Il surveille et contrôle les jobs via les métadonnées dans le Système d'Information de WISDOM (WIS). Le Maître crée des jobs selon :

- La configuration des grilles du WPE : le nombre maximum de jobs autorisé pour chaque grille et sa priorité.
- Le statut courant des jobs du WPE : le nombre de jobs dont les statuts sont « *unsubmitted* », « *queued* », « *terminated* », …

Les Esclaves vont soumettre les jobs créés par le Maître. Chaque Esclave correspond à une grille et il ne soumet les jobs que sur cette grille.

Dans sa configuration actuelle, 3 grilles sont configurées dans le WPE :

- EGEE : les ressources de la VO biomed.

- Auvergrid : les ressources de la VO auvergrid, grille régionale en Auvergne
- EuAsia : les ressources de la VO EuAsia créée pour les partenaires du projet EuAsiaGrid.

Le nombre maximum des agents est fixé à 2.000. Le nombre maximum des jobs de la grille EGEE est de 10.000 avec la priorité 1. Le nombre maximum des jobs de la grille AuverGrid est de 500 avec la priorité 2. Le nombre maximum des jobs de la grille Euasia est de 200 avec la priorité 3.

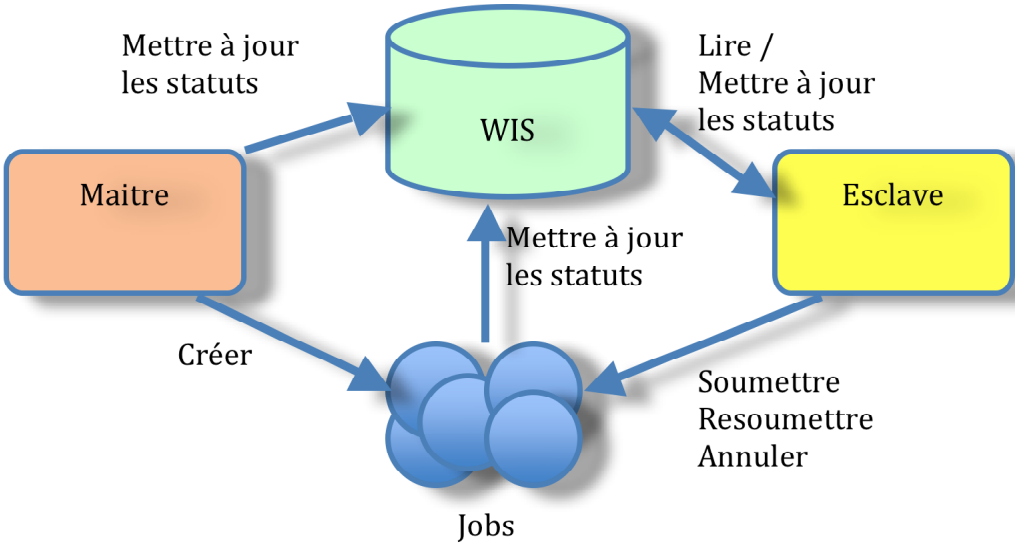


Figure 3-6 - Schéma des Maitre et Esclaves du JM

Tout d'abord, le Maître est lancé et il va créer 2.000 jobs :

- 200 jobs sur la grille Euasia parce qu'elle a la priorité la plus haute,
- puis 500 jobs sur la grille Auvergrid,
- finalement, 1.300 jobs sur la grille EGEE.

Bien que le nombre maximum des jobs de la grille EGEE soit fixé à 10.000, le Maître ne pourra soumettre que 1.300 jobs parce que le nombre maximum des agents est fixé à 2.000. Pour soumettre ces 2.000 jobs, il faut lancer 3 instances d'Esclave sur 3 Interfaces d´Utilisateur (UI) différentes avec les proxy qui correspondent aux 3 VO (biomed, auvergrid et euasia).

Les statuts des jobs sont stockés dans le WIS et ils sont mis à jour par le Maître, les Esclave et les jobs eux-mêmes.

Le Maître met à jour les statuts suivants :

- *unsubmitted* : lorsque un job est créé par le Maître, son statut est *unsubmitted*
- *killed* : si un job est tué pendant l'exécution d'une tâche, le Maître signale son statut *killed* pour que l'Esclave puisse le resoumettre
- *tobekilled* : si un job attend toujours, le Maître signale son statut

tobekilled pour que l'Esclave puisse l'annuler et le resoumettre

- *waiting* : le Maître signale ce statut pour un job s'il est en attente d'être exécuté sur un CE

L'Esclave met à jour le statut *queued* d'un job lorsqu'il met ce job dans la queue de soumission.

Un job peut lui-même mettre à jour son propre statut :

- *running* : job en exécution entrain de lancer une tâche
- *idle* : job en exécution sans tâches
- *terminated* : le job se tue lui-même si le nombre maximum de tentatives de récupération de tâche sans succès a été atteint.

Lorsqu´un job tourne sur une grille, il utilise les scripts clients fournis par le Task Manager (TM) pour demander ou pour récupérer une tâche. S´il y a une tâche disponible, il la récupère. Le statut de ce job est mis à jour dans le WIS à *running* par lui-même. Sinon, le job met à jour son statut à *idle*. Après avoir fini une tâche, le job continue la boucle *demander-récupérer-exécuter* jusqu'à l´expiration de son proxy ou qu´il soit inoccupé pendant longtemps.

L'ajout d'une nouvelle grille au WPE est assez facile. Il suffit de configurer la nouvelle grille dans le WIS et de lancer une instance de l'Esclave sur une Interface Utilisateur avec un proxy correspondant. Comme le TM est totalement indépendant des grilles, la nouvelle grille peut donc être intégrée facilement.

3.3. Performances du WPE

Nous allons présenter dans cette partie une étude de performance que nous avons menée sur la version actuelle du WPE, les problèmes identifiés par cette étude et enfin, les améliorations que nous avons apportées au WPE afin d´améliorer sa performance.

La performance de calcul du WPE dépend de nombreux facteurs : les ressources de la grille où le WPE est déployé, la configuration du WIS et la nature des tâches déployées. Le WPE a été pensé pour le déploiement à très grande échelle de calculs indépendants de *docking* pour le criblage virtuel [75], [76], [77]. Comme nous l'avons vu précédemment, le WPE est déployé actuellement sur plusieurs VO. Sa performance dépend en conséquence des ressources de ces VO :

- Les Éléments de Calcul (CE): Les jobs du WPE sont lancés sur les CE. La performance de calcul du WPE dépend de façon évidente des performance de ces CE.
- La charge de la VO: Le JM soumet les jobs pilotes via les Workload Management Systems (WMS). Plus les ressources de calcul de la VO sont surchargées, moins les jobs du WPE sont lancés. Cela affecte la performance des applications scientifiques qui demandent l'exécution en parallèle de beaucoup de jobs comme par exemple le *docking*.

La configuration du WIS affecte aussi sa performance :

- Le nombre d´agents indique le nombre des jobs soumis par le JM en même temps. Dans le cas où le nombre des tâches est supérieur à celui des agents, plus le nombre d'agents disponibles est grand, plus vite les tâches seront exécutées.
- Le choix des CE pour soumettre les jobs : les CE sont choisis en priorité pour leur performance. Mais le nombre des CE de haute performance peut être limité ou réduit.

A ces facteurs externes s'ajoutent des problèmes liés à l'implémentation du WPE, à savoir la disponibilité, la connexion entre le Task Manager, le Job Manager et le WIS ainsi que la stabilité. Ci-dessous, nous allons présenter brièvement les résultats de premiers tests de performance puis nous proposerons une analyse des limitations observées au cours de tests ciblés et discuterons les améliorations apportées ainsi que leur impact sur les performances.

3.3.1. Premiers tests de performances de WISDOM

3.3.1.1. Autodock

Dans la première section de ce chapitre, nous avons décrit la genèse du développement de la plate-forme WISDOM. Elle a été conçue pour des déploiements à très grande échelle du criblage virtuel. Nous avons donc testé ses performances avec Autodock [56], le logiciel open source de référence, conçu pour le calcul de l'énergie de liaison entre une petite molécule, par exemple un médicament potentiel, et le récepteur d'une protéine, cible biologique dont la structure tridimensionnelle est connue. Le test contient plusieurs petites tâches indépendantes dites de *docking*. Chaque tâche de *docking* prend généralement de 3 à 5 minutes de temps de CPU pour calculer la l'énergie de liaison d'un ligand et d'une protéine sur un CE.

Rappelons que l'objectif de la thèse est le traitement massif de données d'épidémiologie moléculaire et de métagénomique. Ce traitement va impliquer une chaîne de tâches parallélisées à une échelle réduite par rapport au criblage virtuel pour lequel la plate-forme WISDOM a été conçue initialement. C'est le cas par exemple de l'application ePANAM qui sera étudié en détail au chapitre 5 de ce mémoire : au lieu de requérir le déploiement de plusieurs milliers ou dizaines de milliers de tâches indépendantes, l'analyse d'un *run* de pyroséquençage va requérir au maximum quelques dizaines de tâches en parallèle. Par contre, les résultats de toutes ces tâches doivent être collectés pour la dernière étape du traitement des données. Ce scénario d'utilisation de la grille est significativement différent de celui du criblage virtuel. Nous avons donc effectué une deuxième série de tests avec la plate-forme ePANAM et une chaine de traitements de phylogénie.

Tous les tests ont été effectués avec l'hypothèse que les ressources de la VO biomed ne sont pas surchargées et que le rang pour choisir les CE est « `Rank`

= other.GlueHostBenchmarkSF00; ». C'est un rang de base qui se concentre sur la performance de calcul selon le *benchmark* **SpecFloat2000** (Cette suite de test contient 14 programmes *benchmark*, afin d'évaluer la performance des opérations en virgule flottante d'un système donné - http://www.spec.org/). Le CE disponible avec le rang plus haut de *benchmark* SpecFloat2000 sera choisi.

Le premier test de performance a été effectué avec l'application Autodock. Le test a été répété avec 100, 500, 1000, 5000 et 10000 tâches configurées dans le WPE. Le nombre maximum des agents (ou job pilot) est configuré aussi respectivement de 100 à 1000. Pour chaque paire {nombre de tâches, nombre maximum des agents}, le test a été répété 5 fois et on calcule la moyenne du temps total pour finir tous les tâches afin de s'affranchir de l'impact de la charge de la grille.

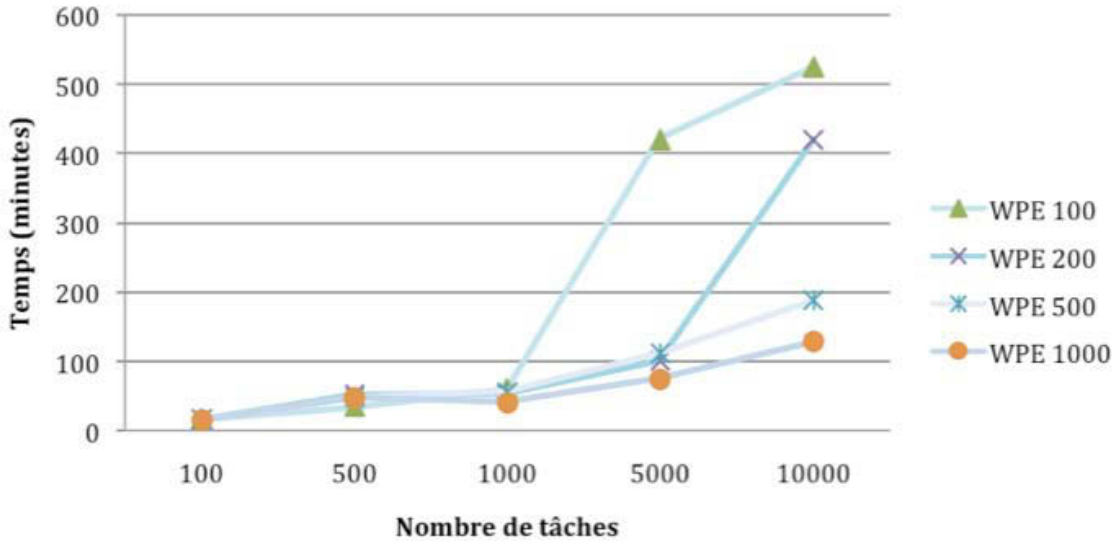


Figure 3-7 - Test de performance avec Autodock

Les temps moyens d'exécution sont montrés dans la Figure 3-7 en fonction du nombre de tâches. Nous constatons que si le nombre de tâches à effectuer est inférieur à 1000, la différence entre les temps d'exécution des tests correspondant aux différentes configurations WPE100/200/500/1000 n'est pas significative. Au-delà de 1000 tâches, le test montre l'importance du nombre de jobs disponibles dans le WPE. Plus ce nombre est grand, plus le temps d'exécution est réduit. Ce test a donc confirmé l'efficacité du WPE pour le criblage virtuel à grande échelle où des millions de calculs de *docking* , donc des millions de tâches, doivent être effectués.

VHVHI HgH

Le deuxième test a été réalisé avec ePANAM (voir chapitre 5), plate-forme d'annotation phylogénétique automatisée adaptée aux données de Séquençage de Nouvelle Génération (Next Génération Sequencing). Cette plate-forme permet de faire un contrôle sur la qualité des séquences expérimentales, mais également de leur attribuer une taxonomie et de comparer les écosystèmes sur

la base d'une analyse phylogénétique. Les détails de son déploiement sur la grille seront décrits au chapitre 5. A la différence d'Autodock, l'analyse phylogénétique effectuée par ePANAM ne requiert que quelques dizaines de tâches en parallèle mais de plus grande durée. De plus, l'exécution de l'analyse d'un ensemble de données de séquençage sur ePANAM contient les étapes suivantes :

- Une tâche de départ (`g-panam-split`)
- Plusieurs tâches indépendantes (`g-panam-alnphy`). Le nombre de ces tâches dépend du nombre de séquences en entrée.
- Une tâche de fin (`g-panam-clade`) qui attend les résultats des tâches `g-panam-alnphy` pour les combiner.

La dernière étape de l'analyse phylogénétique requiert la collecte des résultats de toutes les tâches précédentes. A des fins de tests, 3 jeux de données de séquençage haut débit ont été utilisés : 250.000 séquences, 500.000 séquences et 1.000.000 séquences.

Pour chaque ensemble de données, nous avons répété 20 fois l'exécution de la chaîne de traitement.

Nombre de séquences	Temps de calcul avec le WPE (secondes)	Temps de calcul sur un PC (secondes)
250000	6140	42552
500000	10725	139500
1000000	20929	571428

Tableau 3-2 - Temps moyen de ePANAM sur la grille et sur une machine locale

Le Tableau 3-2 nous donne les temps moyens d'exécution sur la grille et sur une machine locale des différents tests ePANAM avec les 3 ensembles de données. La Figure 3-8 montre la comparaison graphique entre ces résultats.

Grâce à la parallélisation des tâches indépendantes de ePANAM sur le WPE, les temps de calcul de ePANAM sur la grille sont nettement meilleurs que ceux sur une seule machine. Le WPE s´est montré efficace même avec les problèmes bioinformatiques à haut débit pour lequel il n´était pas conçu au départ.

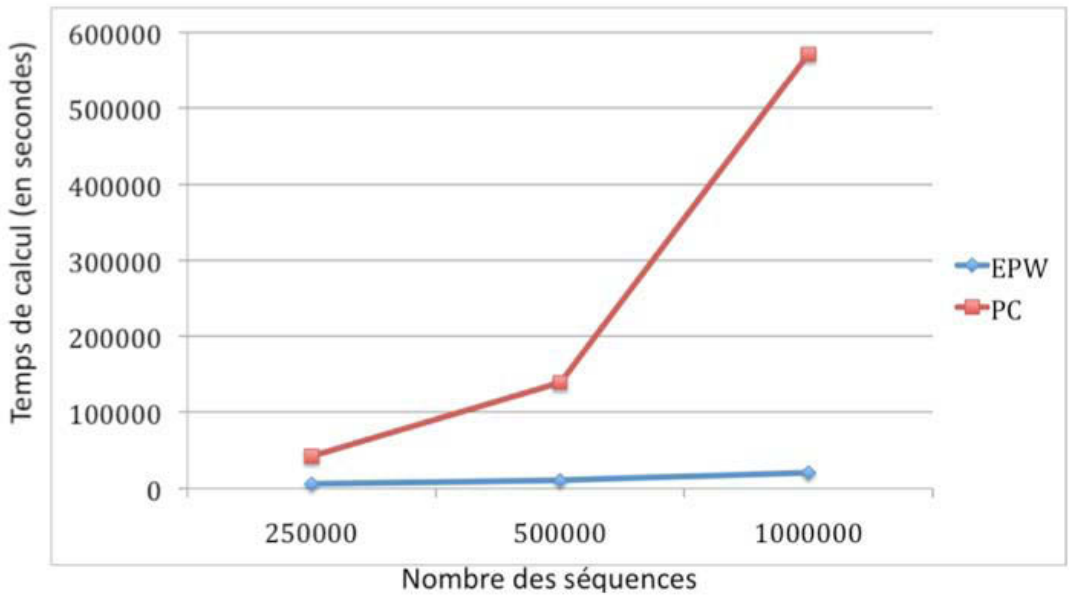


Figure 3-8- Comparaison la performance de ePANAM sur la grille et sur une machine local

Il existe pourtant des limitations déjà identifiées par les développeurs du WPE [48]. Dans les sous-sections suivantes, nous allons analyser en détails ces limitations.

3.3.2. Analyse des limitations actuelles

VHVigH H□□eL1□□□□□

Le Job Manager du WPE soumet des jobs pilotes sur la grille avec un proxy. Par conséquent, la durée de vie des jobs pilotes est basée sur la durée de vie du proxy. Au moment d'expiration du proxy, tous les jobs sont tués : les tâches ne peuvent plus être exécutées et restent dans la queue du Task Manager parce qu'il n'y a plus de jobs disponibles. Dans ce cas, l'administrateur du WPE doit renouveler le proxy et relancer le JM. Malheureusement, ce travail doit être effectué manuellement parce qu'il n'y a pas encore des scripts automatiques pour le faire. La vie d'un proxy sur la grille dure normalement de 12 heures à 24 heures. L'administrateur du WPE doit alors relancer le JM au moins une fois par jour. Cependant, la relance du JM ne résout que le problème avec les tâches qui sont créées après l'expiration du proxy. Pour les tâches qui sont entrain de tourner, les jobs correspondants sont tués à cause de l'expiration du proxy et les tâches sont perdues. Nous avons proposé une solution pour résoudre ce problème en utilisant MyProxy (voir section 3.3.3).

Un autre facteur peut influencer la disponibilité du WPE. Comme nous l'avons abordé dans la section 3.2.3, pour éviter l'occupation de ressources de la grille, les jobs pilotes se terminent eux-mêmes après une période de temps s'ils sont libres ou s'il n'y a pas de tâches dans le Task Manager. Dans ce cas, le Job Manager ne les resoumet plus parce qu'ils occuperaient les ressources inutilement. Le nombre des jobs disponibles diminue ainsi avec le temps si les tâches ne sont pas créées à un rythme régulier. De nouvelles tâches ajoutées au TM doivent attendre plus longtemps dans la queue du TM à cause de cette

diminution. Dans le pire des cas, s'il n'y a plus de jobs disponibles, le WPE ne fonctionne plus. Pour rafraîchir le WPE dans cette situation, il faut remettre les statuts des jobs dans le WIS et relancer le JM. Pour mettre en évidence ce problème, nous avons surveillé la diminution de nombres des jobs disponibles. Pour le faire, le TM a été vidé de toutes ses tâches pendant une période courte (10 minutes) et une période longue (1 heure) respectivement. A chaque minute, nous comptons le nombre de jobs pilotes sur la grille et le nombre de tâches dans le TM.

Le WPE démarre avec 100 jobs. Ensuite, 100 tâches sont poussées au TM. Après avoir terminé ces tâches, nous avons attendu pendant 10 minutes et pousser 100 nouvelles tâches dans le TM. Après avoir terminé ces nouvelles tâches, nous laissons le TM rester vide pendant une heure puis finalement poussons à nouveau 100 nouvelles tâches dans le TM.

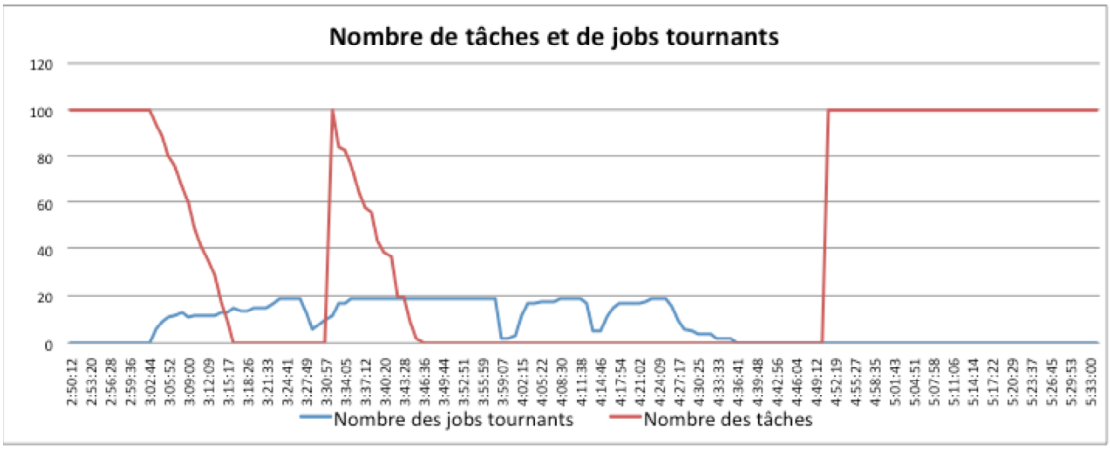


Figure 3-9 - La diminution de nombre des jobs disponibles

La Figure 3-9 montre la relation entre le nombre des jobs tournants et le nombre des tâches. Au début, 100 tâches sont créées et quelques jobs les récupèrent pour exécuter. La deuxième fois, 100 nouvelles tâches sont créées et cette fois, il y a toujours des jobs disponibles pour les exécuter. Finalement, après une longue période, 100 tâches sont créées mais cette fois elles restent toujours dans la queue du TM parce qu'il n'y a plus de jobs disponibles. Tous les jobs sont tués (ou ils se tuent eux-mêmes) après une longue période d'inactivité.

En pratique, cela n'est pas un problème remarquable parce que en mode de production, le WPE reçoit toujours des nouvelles tâches. Les jobs du JM sont occupés donc tous le temps. Dans le cas où un job est tué pour d'autre raison, le JM le resoumet.

3.3.2.2. Communication

Le système d'information WISDOM (WIS) utilise AMGA pour stocker les metadonnées nécessaires au fonctionnement du JM et les données de configuration du WPE. Tous les jobs doivent donc communiquer avec le WIS pendant leur temps de vie sur la grille. Dans la pratique, on a constaté un

ralentissement sensible de la communication entre les jobs pilotes et le WIS après une longue exécution du WPE. Le test suivant nous a permis de mettre en évidence ce problème.

Nous avons lancé le WPE dans un environnement neuf et enregistré chaque jour le temps moyen de communication entre des jobs et le WIS. Cette surveillance a duré pendant un mois et demi.

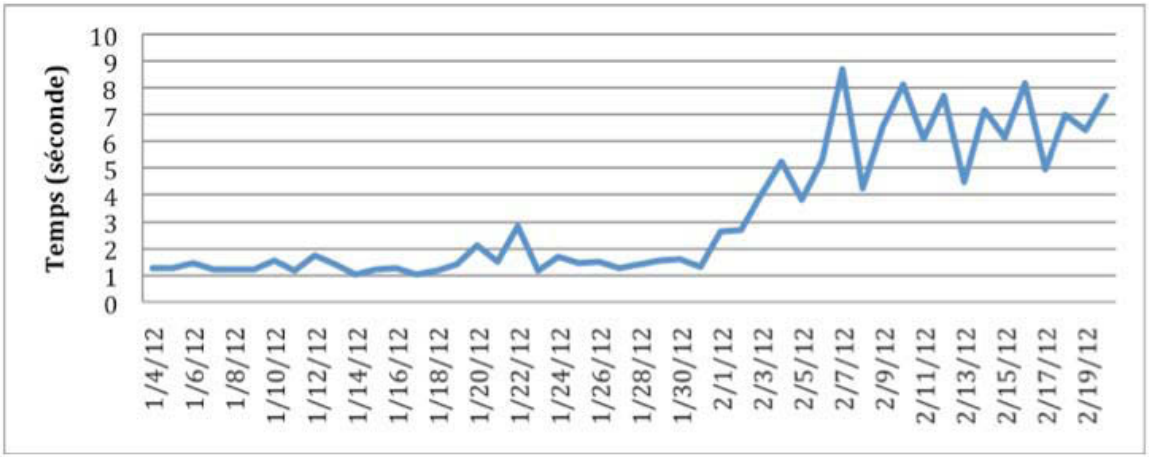


Figure 3-10 - Surveillance de temps de communication entre le WIS et les jobs

Ces temps enregistrés sont montrés par la Figure 3-10. Nous constatons que le temps moyen de communication commence à augmenter après un mois (du 4 janvier 2012 au 5 février 2012). Ensuite, cette augmentation continue significativement et le WIS est très fortement ralenti.

Ce problème illustre la nécessité de rafraîchir le WIS mensuellement pour que le WPE puisse fonctionner normalement. Pour ce test, le WPE gérait un nombre limité de tâches et de jobs chaque jour. S'il fonctionne en mode production pour un data challenge, nous avons constaté que le ralentissement du WIS pouvait se produire beaucoup plus tôt. A l'origine de ce ralentissement est l'utilisation dans le WPE d'une ancienne version de AMGA qui ne gère pas très bien les grands volumes de métadonnées qui sont stockées et accumulées dans le WIS chaque jour.

VHVHgHVH

La stabilité du WPE est mesurée par le taux de succès des tâches. Il y a de nombreux facteurs qui peuvent affecter ce taux :

- La stabilité de la grille où le WPE est déployé.
- L'expiration du proxy. Une tâche entrain de tourner échouera à l'expiration du proxy du JM.
- La communication entre les jobs et le WIS.
- La communication entre les jobs et le TM.

Nous avons fait plusieurs tests avec Autodock et ePANAM sur le WPE pour analyser sa stabilité. Les tests de Autodock ne contiennent que des tâches indépendantes qui ne durent que quelques minutes. La Figure 3-11 montre le

taux moyen de succès de 1000 tâches de Autodock. Environ 25% des tâches finissent par échouer. Ce taux est comparable au taux d'échec des jobs soumis sur l'infrastructure EGI.

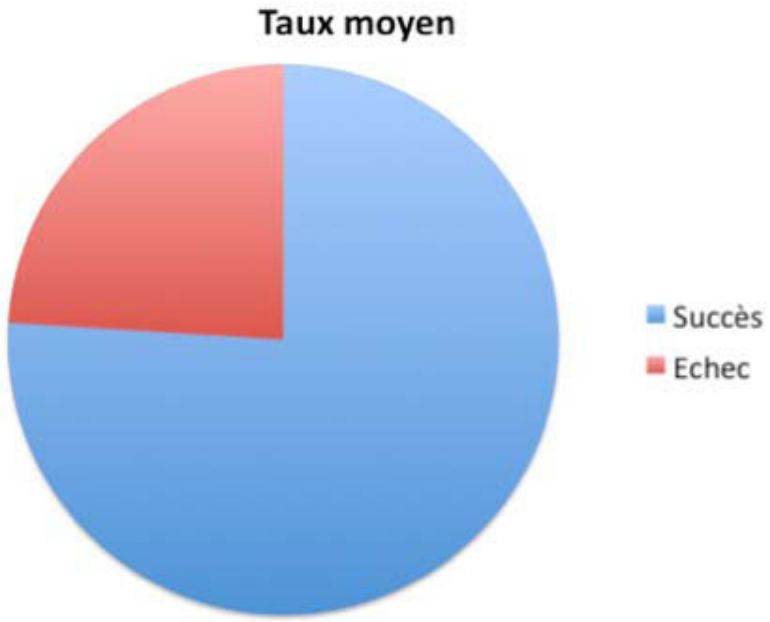


Figure 3-11 - Taux moyen de succès des tâches Autodock

Dans le cas d'ePANAM, les tâches ne sont pas indépendantes. L'analyse d'un ensemble de séquences n'est donc réussie que si toutes les tâches sont réussies. Pour chaque ensemble de données (200000, 500000 et 1000000 séquences), nous avons répété 20 tests et calculé le taux moyen de succès pour ces tests. Par rapport aux tests avec Autodock, la durée des tâches à exécuter pour ePANAM accroît la probabilité d'échec. La Figure 3-12 montre les taux de succès en exécutant ePANAM avec 3 ensembles de données : 250000, 500000 et 1000000 séquences.

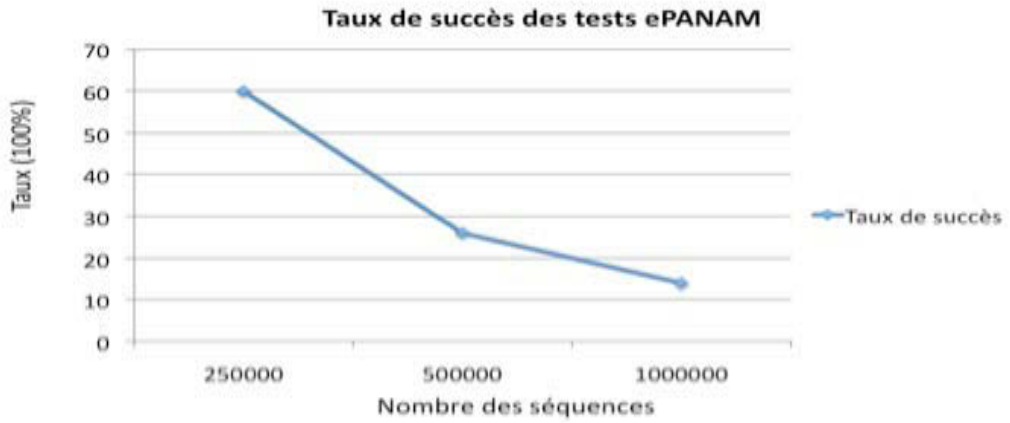


Figure 3-12 - Taux de succès des tests ePANAM avec 3 ensembles de données

Le taux de succès pour ePANAM-250000 est de 60%. Il est significativement plus faible que le taux de succès pour les tâches Autodock. Si on regarde le taux de succès des tâches de façon indépendante, il est de 72%, comparable au taux de succès de pour les tâches Autodock (75%), comme le montre la Figure 3-13.

Le taux de succès de ePANAM-500000 diminue significativement à 26%. Pour les tests de 1000000 séquences, ce taux n'est que 14% ! Si nous assumons que toutes les tâches soient indépendantes, les taux de succès de ces 2 ensembles de données sont 62% (500000 séquences) et 58% (1000000 séquences). Il est clair que si le temps de calcul augmente, le taux de succès diminue.

Ces faibles taux de succès ne sont pas acceptables si nous voulons mettre ePANAM comme un service déployé sur la grille par le WPE. Nous avons donc introduit des modifications qui ont significativement amélioré les performances du WPE.

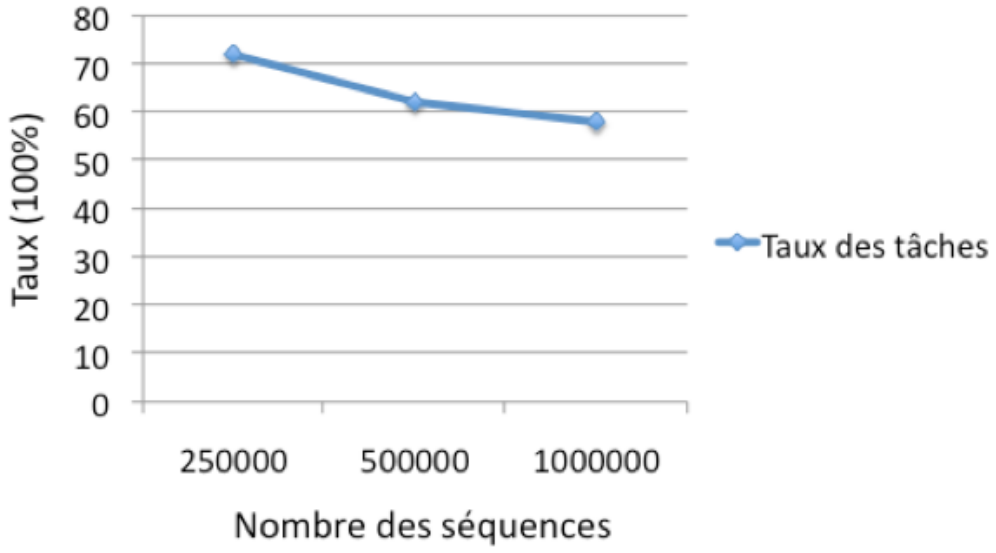


Figure 3-13 – Taux de succès des tâches

3.3.3. Amélioration

Après avoir présenté les 3 problèmes principaux du WPE (disponibilité,

connexion avec le système d'information et stabilité), nous allons discuter des modifications et améliorations apportées pour résoudre ces problèmes.

VHVHVH H 3 .Ls dL 1 eL 1

Les causes du problème de la disponibilité sont :

- La limitation de la durée du proxy qui est utilisé pour lancer des jobs.
- Le mécanisme pour tuer les jobs non sollicités par le Task Manager pour éviter d'occuper inutilement les ressources de la grille.

La durée du proxy peut être prolongée par MyProxy [39]. Normalement pour soumettre les jobs sur la grille, il faut avoir un proxy valide sur une Interface Utilisateur. Lorsqu'un job est soumis à un Élément de Calcul, il copie ce proxy avec lui. Pour prolonger la durée des jobs il faut stocker un proxy à long terme sur un serveur MyProxy et indiquer ce serveur dans les fichiers de description des jobs. Les jobs soumis avec une copie du proxy normal sont liés au proxy à long terme. Lorsque le proxy copié d'un job est expiré mais le job est toujours entrain d'exécuter, le proxy copié peut être rafraîchi après avoir contacté le proxy à long terme stocké sur le serveur MyProxy. Grâce à ce mécanisme, les jobs ne sont plus terminés à cause de l'expiration d'un proxy. Un proxy à long terme peut durer par défaut une semaine sur la grille. Par conséquent, au lieu de redémarrer le WPE quotidiennement, cette amélioration nous permet de ne le redémarrer qu'une fois par semaine. Cette nouvelle version du WPE est indispensable pour la plate-forme ePANAM dont l'exécution de certains algorithmes requiert plus de 24 heures sur la grille.

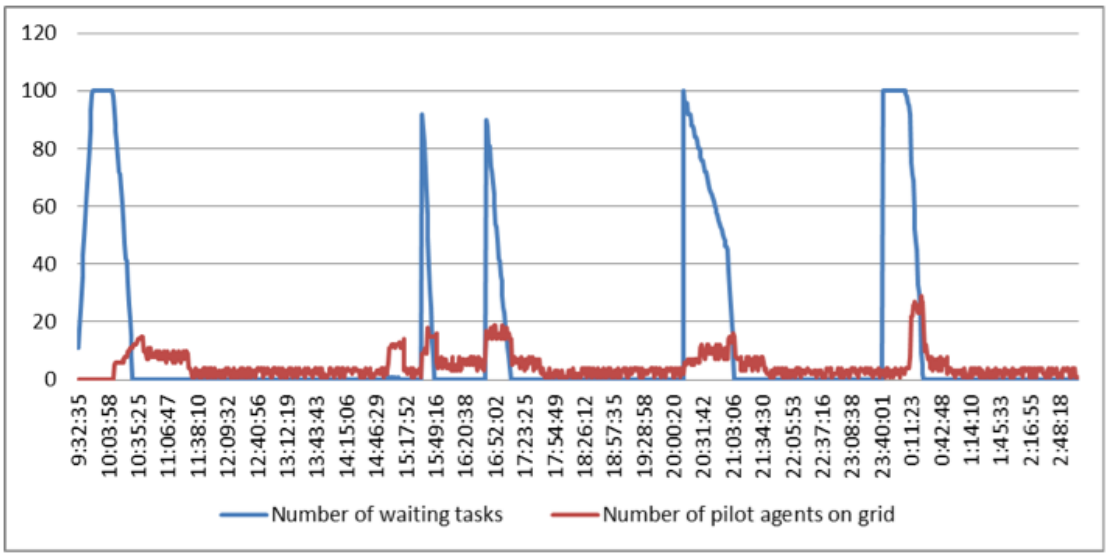


Figure 3-14 – Surveillance de la disponibilité des agents

Pour gérer le problème des jobs qui se suicident pour libérer les ressources de la grille, nous proposons d'ajouter un module dans le JM pour surveiller le nombre des tâches et le nombre des jobs. Si ce module constate que le nombre des jobs disponibles n'est pas suffisant et que de nombreuses tâches sont encore à traiter dans le JM, il va chercher les jobs pilotes dont le statut est *killed* et rafraîchit leur statut à *unsubmitted* pour que le JM puisse les

resoumettre plus tard. Nous avons refait le test décrit dans la sous section 3.3.2 pour analyser la disponibilité du WPE avec ce module. Le résultat de ce test est montré dans la Figure 3-14. Malgré le fait que le TM soit resté vide sur deux périodes de 3 heures (17h00 au 20h00 et du 21h03 au 23h40), de nouveaux jobs pilotes ont été soumis lorsque des nouvelles tâches ont été créées.

VHVHVHgH 3 .L s dL 1 L 1t L 1

L'origine des problèmes de connexion entre les jobs et le WIS est que l'ancienne version de AMGA ne gère pas bien des données volumineuses. Malheureusement nous ne pouvons pas facilement remplacer l'ancienne version de AMGA par une nouvelle version à cause de conflits de version entre les outils utilisés pour développer le WPE. En revanche, nous pouvons nettoyer les métadonnées qui ne sont plus nécessaires, particulièrement celles concernant des indices de jobs et tâches. Ces indices ont pour but d'accélérer les requêtes. Cependant, lorsque le nombre de ces indices augmente et dépasse un certain niveau, il commence à ralentir le WIS, il faut donc les nettoyer de temps en temps pour accélérer le WIS.

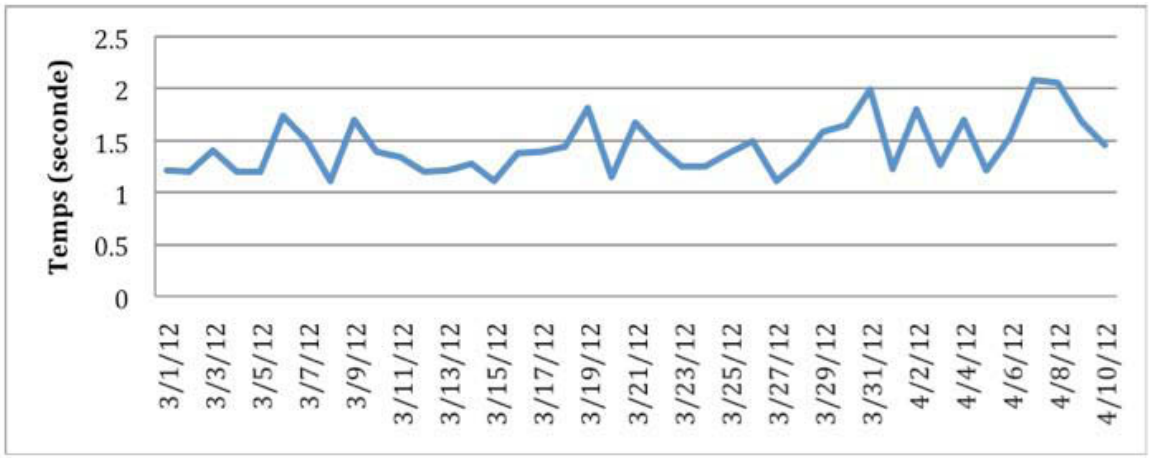


Figure 3-15 - Surveillance de temps de communication entre le WIS et les jobs

Pour ce fait, un module de nettoyage des anciens indices de jobs et de tâches a été ajouté dans le WPE et démarré chaque fois que le WPE était relancé. Normalement, le WPE doit être relancé chaque semaine. La Figure 3-15 montre la surveillance de temps de communication entre le WIS et les jobs après ajout du module de nettoyage dans le WIS. Les temps moyens sont ainsi stables dans la limite de 1 seconde à 2 secondes.

VHVHVHVH 3 .L s dL 1 d at

Finalement, nous considérons des solutions pour le problème de la stabilité du WPE. Il y a de nombreux de raisons à l'échec d'un job sur la grille. Les causes les plus fréquentes d'échec des jobs pilotes soumis par le WPE sont les suivantes :

- Le job pilote est interrompu par un Élément de Calcul à cause d'une erreur inconnue. C'est l'erreur la plus commune sur la grille.
- Le job pilote sur certain CE ne peut pas télécharger les fichiers stockés

sur des Éléments de Stockage (SE) malgré le fait que l'existence des fichiers a bien été vérifiée. Dans ce cas, il faut tester la capacité de téléchargement des jobs pilotes avant de les laisser récupérer et exécuter les tâches. Si un job ne peut pas télécharger des fichiers de tests, il se suicide en mettant à jour son statut à *unsubmitted* pour que le JM puisse resoumettre un autre job.

De toute façon, si un job est interrompu pendant l'exécution d'une tâche, il faut resoumettre le job et recréer la tâche. Dans la version originale du WPE, la recréation d'une tâche était effectuée par le job qui l'exécutait. Si la tâche produisait des erreurs, le job arrêtait cette tâche et la remettait dans la file d'attente du TM. Par contre, dans les cas des erreurs inconnues des jobs, les jobs ne pouvaient pas recréer les tâches parce qu'ils avaient été tués. Le JM a été modifié pour que lui incombe le rôle de surveiller les jobs pilotes et de recréer les tâches dans tous les cas où les jobs pilotes sont interrompus. Nous avons donc introduit un nouveau paramètre du WPE, le nombre maximum de fois où une tâche peut être recréée dans le JM.

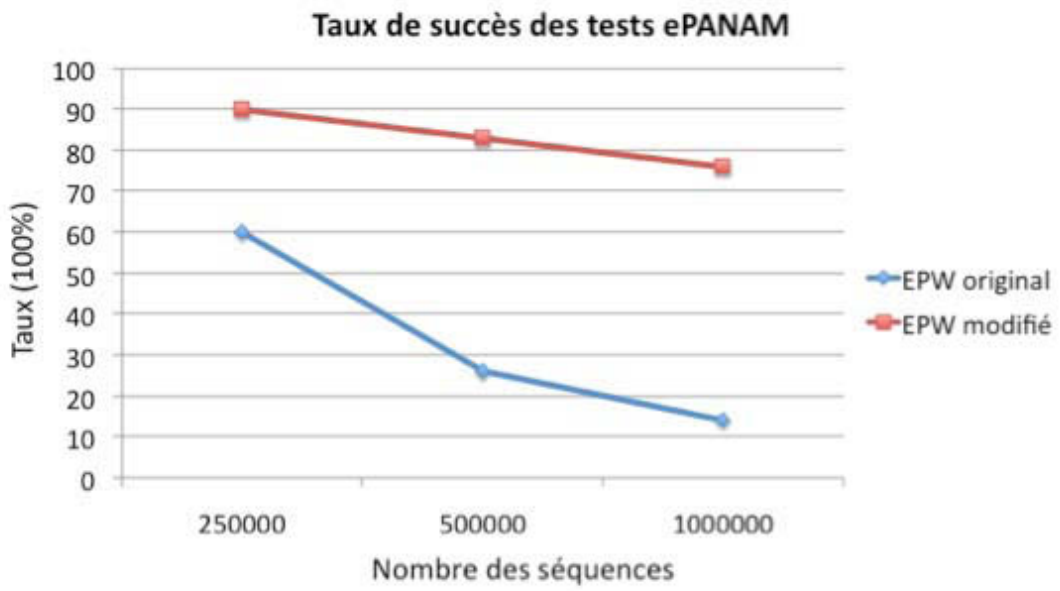


Figure 3-16 - Comparaison les taux de succès entre deux versions du WPE

Nous avons refait les tests de ePANAM dans la sous-section 3.3.2.3 pour mesurer la stabilité du WPE après ces modifications. Dans ces tests, le nombre maximum de recréation d'une tâche est configuré à 5.

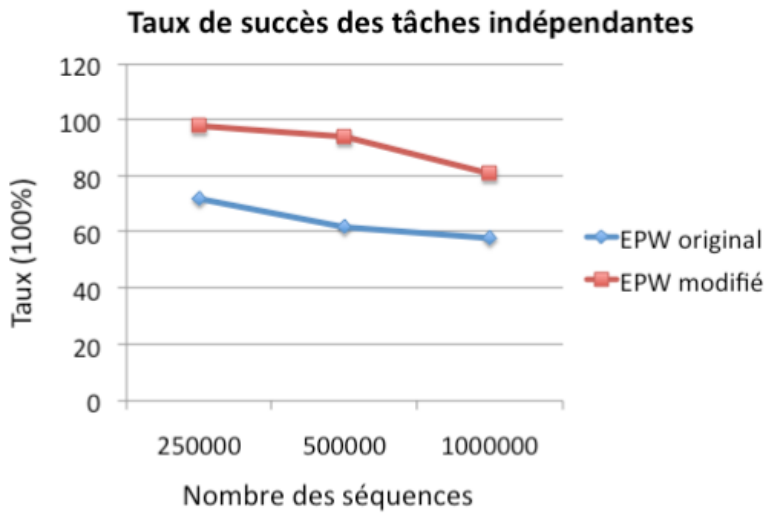


Figure 3-17 - Comparaison les taux de succès entre 2 versions du WPE en considérant que les tâches soient indépendantes

La Figure 3-16 nous montre que le taux de succès a été amélioré significativement (90% au lieu de 60% pour un ensemble de 250000 séquences, 83% au lieu de 26% pour 500000 séquences et 76% au lieu de 14% pour un million de séquences). En ce qui concerne les tâches elles-mêmes, des améliorations significatives du taux de succès sont aussi enregistrées comme illustrées sur la Figure 3-17 (98% - 72%, 94% - 62% et 81% - 58%).

Avec cette modification, la stabilité du WPE est améliorée significativement. Le WPE est donc approprié au mode de production requis pour ePANAM.

3.3.4. Comparaison des performances de WISDOM et de DIRAC

Nous avons présenté dans le chapitre 2 la plate-forme DIRAC. Un des enjeux du présent travail était d'évaluer les performances de WISDOM pour la metagénomique. Ces performances ont été considérablement améliorées grâce aux modifications décrites plus haut. Mais ces performances devaient être comparées à celles des autres environnements de production et notamment à DIRAC.

VHVHnH H L3 e s L1 eLcs i ee. d L1 cd L L2

Comme précédemment, le test de comparaison entre DIRAC et WISDOM a été réalisé pour différents nombres des tâches configurées dans le WPE : 100, 500, 1000, 5000 et 10000. Le nombre maximum des agents est varié aussi de 100 à 1000. Pour chaque paire {nombre de tâches, nombre maximum des agents}, le test est répété 5 fois et la moyenne du temps total pour finir tous les tâches est calculée afin de s'abstraire des effets de fluctuation de la charge de la grille.

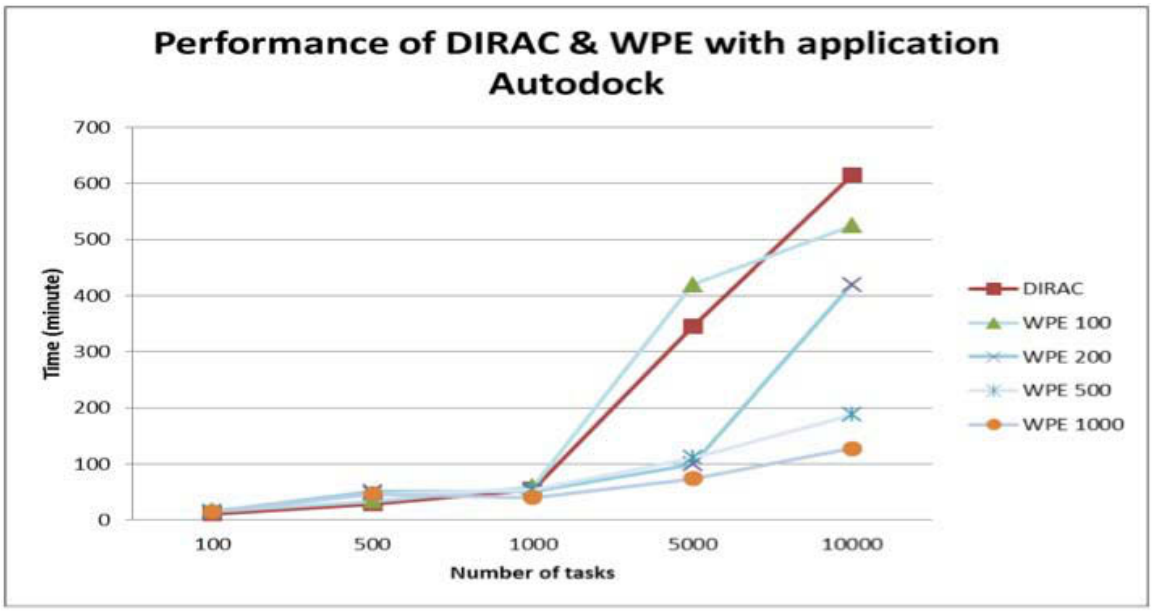


Figure 3-18Test de performance avec Autodock

La Figure 3-18 montre que le WPE (version originale) a des performances très supérieures à DIRAC lorsque le nombre de tâches est supérieur à 1000, à condition que le nombre maximum de jobs pilotes soit très supérieur à 100. Le WPE, conçu pour le criblage virtuel, est donc supérieur à DIRAC pour ce type de déploiement.

VHVhHgH L3 e s L1 eLcs i ee. dL1

La Figure 3-19 montre les performances de WPE en comparaison avec la performance de DIRAC pour le déploiement des calculs de la plate-forme ePANAM. Le déploiement sur la grille, que ce soit avec DIRAC ou avec ePANAM, réduit considérablement le temps de calcul, mais la plate-forme DIRAC a des performances significativement supérieures à WISDOM.

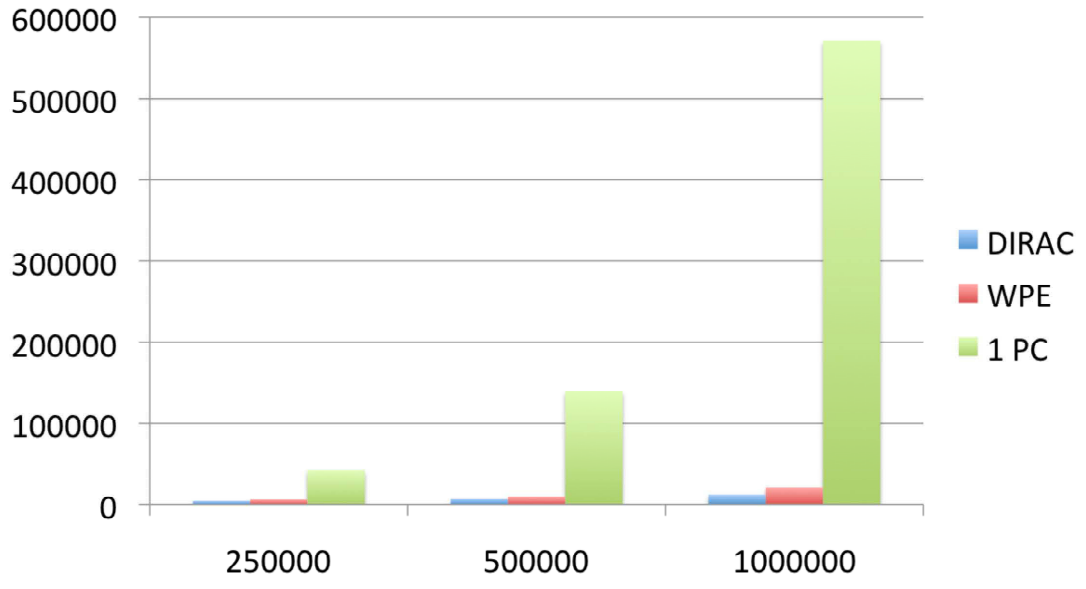


Figure 3-19 – Test de performance avec ePANAM

Le Tableau 3-3 et le Tableau 3-4 présentent une comparaison détaillée des performances pour deux algorithmes (méthodes LCA et NN) de la version originale du WPE (WPE v0), la version modifiée (WPE v2) et DIRAC.

Tests	WPE v0	DIRAC	WPE v2	WPE v0	DIRAC	WPE v2	WPE v0	DIRAC	WPE v2
Test 1	6471	3617	5580	10238	8925	13129	22031	13326	20145
Test 2	4825	4434	7938	9956	5823	7855	21180	11039	21603
Test 3	5190	3593	5657	13825	10238	8410	19922	11631	19378
Test 4	7033	3785	5308	13016	6643	7558	23496	11647	20769
Test 5	6775	2899	8088	8993	5900	7267	20341	11226	20644
Test 6	7209	3325	6227	8640	8723	11209	20084	12157	21154
Test 7	5717	3869	5477	13200	6904	12874	20835	11607	19937
Test 8	6289	4349	5509	9728	6207	10509	19100	15071	20700
Test 9	5136	5244	6256	9015	5897	7916	20101	12340	20249
Test 10	5471	6851	5544	7367	6695	8130	22399	14265	21509
Test 11	7210	5368	7385	7761	6357	10112	19538	11943	21338
Test 12	6200	6121	5263	10298	7120	8930	24193	11582	21826
Test 13	7713	4787	8473	14329	6718	12383	21402	12121	26116
Test 14	6845	3528	7962	13200	6037	10674	20335	13023	20993
Test 15	5491	5181	5554	12084	6294	11321	19372	14470	22224
Test 16	5516	5987	4977	13116	10256	7993	21186	11882	21782
Test 17	6113	4227	5360	9233	6260	8150	21008	12678	25417
Test 18	7004	4517	6403	9654	5072	9036	22109	13223	23024
Test 19	5238	5134	8042	12134	6399	13023	20019	13005	21092
Test 20	5359	4985	5667	8703	6822	7225	19919	11939	19987
Moyenne	6140	4590	6334	10725	6965	9685	20929	12509	21494
Ecart-type	854	1031	1170	2170	1433	2058	1358	1113	1698
	250000 séquences			500000 séquences			1000000 séquences		

Tableau 3-3 - Test de performance – comparaison sur WPE version originale, DIRAC et WPE

Tests	WPE v0	DIRAC	WPE v2	WPE v0	DIRAC	WPE v2	WPE v0	DIRAC	WPE v2
Test 1	31032	26938	41043	36930	47502	41900		57644	80210
Test 2	38401	30105	32302	39298	46039	45824		84418	75639
Test 3	28430	32853	29874	42043	47489	53569		48514	64481
Test 4	29337	32161	38456	41807	44353	52173		45469	68220
Test 5	37105	36350	34440	52156	39896	50184		72661	92303
Test 6	42045	38190	29350	38956	40935	44278		50526	65552
Test 7	36323	30656	40168	36733	36656	56371		71742	72899
Test 8	29345	47191	34852	40138	41629	40110		61994	78519
Test 9	32239	36941	39482	41286	51592	42138		63672	95672
Test 10	29675	36092	33452	38740	43488	50129		134580	101262
Test 11	39123	30943	28224	43006	50246	41050		51510	82144
Test 12	38741	36849	32856	41043	43105	43272		65883	63003
Test 13	34320	59774	39921	38841	60448	43970		71932	52670
Test 14	30102	56105	42239	38992	43854	51025		84940	63677
Test 15	28390	33536	43103	44022	42361	44473		64269	71349
Test 16	32012	40772	32109	36049	44209	50244		66234	81004
Test 17	29792	41290	35290	40129	41373	50124		62230	71292
Test 18	30239	30305	41203	36594	62100	45329		62173	57900
Test 19	31246	34952	33245	34014	41029	40873		58824	104092
Test 20	28450	42477	35619	38732	50663	41203		57489	55033
Moyenne	32817	37724	35861	39975	45948	46412		66835	74846
Ecart-type	4267	8491	4541	3794	6478	4877		19089	14741
	250000 séquences			500000 séquences			1000000 séquences		

Tableau 3-4 - Test de performance – comparaison sur WPE version originale, DIRAC et WPE (ePANAM avec la méthode NN)

3.4. Conclusion du chapitre 3

Nous avons présenté le WISDOM Protection Environment (WPE) dans ce chapitre. Les tests de performances ont permis de montrer l'efficacité du WPE mais aussi a mis en évidence certaines limitations. Nous avons proposé et

implémenté des modifications qui ont considérablement amélioré les performances du WPE notamment pour la métagénomique. Les résultats obtenus montrent l'impact du déploiement sur la grille qui vont faire l'objet des études de cas décrites dans les chapitres 4 et 5.

CHAPITRE 4 : G-INFO PLATE-FORME D'EPIDEMIOLOGIE MOLECULAIRE

Dans ce chapitre, nous discutons de la première étude de cas du WPE, un portail de surveillance de la grippe aviaire. g-INFO est une solution complète qui inclut un système basé sur l'environnement de production WISDOM et un portail pour aider les utilisateurs non experts de la grille à analyser et construire les arbres phylogénétiques sur les séquences du virus. Nous avons développé g-INFO dans le cadre du projet EuAsiaGrid avec 2 ingénieurs des instituts vietnamiens. La structure de ce chapitre est la suivante : la section 4.1 abordera les enjeux scientifiques dans le domaine de la grippe aviaire. Ensuite, nous proposons la solution g-INFO et discutons quelques résultats.

4.1. Introduction

Les maladies émergentes sont caractérisées par le fait qu'elles sont par définition peu connues. Il est donc essentiel de pouvoir accumuler rapidement un ensemble d'informations biologiques, épidémiologiques et géographiques sur les foyers identifiés. Ces informations sont souvent très éparpillées géographiquement mais elles doivent être diffusées le plus rapidement possible au sein de la communauté internationale pour alimenter les travaux des chercheurs et pour permettre aux états de prendre des mesures concertées de prévention et de surveillance. Depuis 10 ans, la technologie des grilles informatiques a permis le développement de véritables infrastructures distribuées fournissant de vastes ressources de calculs et de stockage aux communautés scientifiques. Plus récemment, elle met aussi à la disposition de ses utilisateurs des outils de gestion de données distribuées permettant la création de véritables fédérations de bases de données avec accès sécurisé dont les informations peuvent être exploitées à des fins d'analyses statistiques. Grâce à ces progrès récents, les grilles constituent aujourd'hui des infrastructures très prometteuses pour la mise en place de systèmes de surveillance épidémiologique à l'échelle internationale.

La pandémie de 2009 de la grippe H1N1 (ou encore appelée la « grippe porcine ») a confirmé que la prudence, la préparation et la forte compétence de recherche en santé publique sont essentielles pour faire face aux menaces émergentes pour la santé. Par ailleurs, le virus plus « ancien » H5N1 (ou la « grippe aviaire ») a continué à évoluer et à provoquer des foyers épidémiques. L'épidémiologie est l'une des bases pour détecter, identifier et analyser les problèmes de santé liés à la grippe causée par Influenza A. L'épidémiologie moléculaire s'intéresse à l'analyse des génomes, des séquences et des structures et utilise pour cela un vaste ensemble d'outils bioinformatiques qui nécessitent très souvent des ressources adéquates de calcul. En d'autres termes,

l'épidémiologie moléculaire considère les *données* moléculaires et utilise les *outils* bioinformatiques pour les analyser.

En termes de données, il y a plusieurs ressources de bases de données publiques disponibles telles que NCBI (National Center for Biotechnology Information)[1], BioHealthBase[2], Influenza Virus Database[3] ou Influenza Sequence Database[4]. Ces organismes collectent toutes les données de virus de la grippe dans le monde entier et offrent des services riches afin de renforcer la surveillance de la grippe et les capacités d'alerte précoce. Dans certains cas, comme le H5N1, les données ne sont pas toujours rendues publiques (pour une raison ou une autre) et restent désynchronisées avec les ressources de la base de données publique. Par conséquent, les systèmes sont non interopérables, ce qui conduit à la duplication du travail et des efforts coûteux d'intégration des données. La technologie de grille apporte des solutions pour traiter de tels problèmes en apportant des ressources de calcul et le partage sécurisé des données.

En termes d'outils, il est bien connu dans la bioinformatique que le traitement des données en mode batch permet d'économiser du temps et des efforts. Par conséquent, il y a de nombreux pipelines phylogénétiques qui ont été développés pour traiter les données en mode batch. Certains sont d'usage général, tel que Phylogenetic Diversity Analyzer, AIR-Appender, AIR-Identifier, ou AIR-Remover [89]; tandis que d'autres sont utilisés pour des analyses spécifiques, tel que PhyloGena [90]. Les analyses phylogénétiques consomment beaucoup de temps de calcul. C'est la raison pour laquelle la plupart des outils phylogénétiques utilisent des clusters ou la grille. PhyML-MPI [91], par exemple, est une version parallèle du logiciel bien connu PhyML [16] pour la construction d'arbres phylogénétiques. Ceci est également vrai pour l'analyse génomique comparative avec l'exemple de mpiBLAST [92], une version parallélisée de BLAST [22].

Profitant de la puissance de la grille, nous avons développé le projet g-INFO (grid-based International Network for Flu Observation) qui fournit un système complet pour surveiller la grippe du type A. Le but de g-INFO est d'intégrer des sources de données des virus de la grippe en une fédération de bases de données qui peut être interrogée à la demande pour traiter les données sélectionnées par des pipelines d'analyse. Les séquences, extraites de sources de données existantes fournis par la communauté scientifique, sont traitées de façon dynamique en utilisant quelques principes de l'architecture orientée services (Service Oriented Architecture) pour composer des pipelines phylogénétiques déployés sur la grille. g-INFO prend en charge les pipelines

---

[1] http://www.ncbi.nlm.nih.gov/genomes/FLU/FLU.html

[2] http://www.fludb.org

[3] http://influenza.big.ac.cn

[4] http://www.flu.lanl.gov

automatiques et les pipelines dynamiques. Le pipeline automatique est constitué de trois algorithmes bien connus, généralement utilisés pour l'analyse phylogénétique. Il peut fonctionner quotidiennement sur les données mises à jour sur des bases de données publiques. Le pipeline dynamique est configurable afin qu'un expert puisse choisir les outils qu'il veut ainsi que l'ordre d'exécution des outils pour son analyse spécifique.

Dans les sections suivantes, nous présentons le portail g-INFO qui facilite le processus de création et le fonctionnement des pipelines phylogénétiques dans le système g-INFO. Avec le portail g-INFO, l'utilisateur peut créer et enregistrer des modèles de flots pour exécuter les instances de pipeline. Les sorties peuvent être visualisées avec un outil de visualisation intégré.

4.2. Analyse des besoins

Le portail g-INFO qui a développé dans le cadre de cette thèse a été conçu pour répondre aux besoins spécifiques des spécialistes qui doivent intervenir sur les foyers de grippe aviaire au Vietnam. L'Influenza Aviaire (IA) est causée par des virus spécifiques qui sont membres de la famille des Orthomyxoviridae. Il y a trois genres de grippe - A, B et C : seuls les virus grippaux A sont connus pour infecter les oiseaux. Les infections chez les oiseaux peuvent générer une grande variété de signes cliniques selon l'hôte, la souche du virus, le statut immunitaire de l'hôte et les conditions environnementales. Le diagnostic de la grippe est fait par isolement du virus ou par détection et caractérisation des fragments de son génome.

Pour identifier le virus, des prélèvements faits sur des oiseaux vivants ou sur des déjections ou organes d'oiseaux morts sont inoculés dans le liquide allantoïque d'œufs embryonnés de 9 à 11 jours d'âge. Après une incubation de quelques jours (quatre à cinq jours), le liquide allantoïque contenant les virus est récolté. La présence du virus grippal est confirmée par des méthodes d'immunodiffusion.

L'isolement des embryons a été récemment remplacé, dans certaines circonstances, par la détection d'un ou plusieurs segments du génome du virus en utilisant rRT-PCR (*real-time reverse-transcription polymerase chain reaction*) ou d'autres techniques moléculaires.

Au Vietnam, des échantillons sont collectés chaque jour dans toutes les régions du pays et envoyés au Centre National de Diagnostic Vétérinaire (CNDV). Ils sont traités par les méthodes décrites ci-dessus. S'il est confirmé qu'il s'agit d'échantillons du virus de la grippe A, les résultats de traitement sont envoyés à des laboratoires étrangers ou au laboratoire de l'Institut National de Biotechnologie de l'Académie des Sciences et Technologies du Vietnam pour séquencer le virus incriminé. Les séquences du virus sont renvoyées au CNDV où les épidémiologistes construisent les arbres phylogénétiques à partir des séquences et les comparent au virus original. Le travail sur la grippe aviaire décrit dans cette thèse a fait l'objet d'une collaboration avec Dr. Le Thanh Hoa de l'Institut National de Biotechnologie. La section suivante explique son cahier des charges et nos choix d'implémentation.

4.3. Cahier des charges et choix d'implémentation

A chaque fois qu'il y a des nouvelles séquences de la grippe aviaire, Dr. Hoa suit le pipeline suivant :

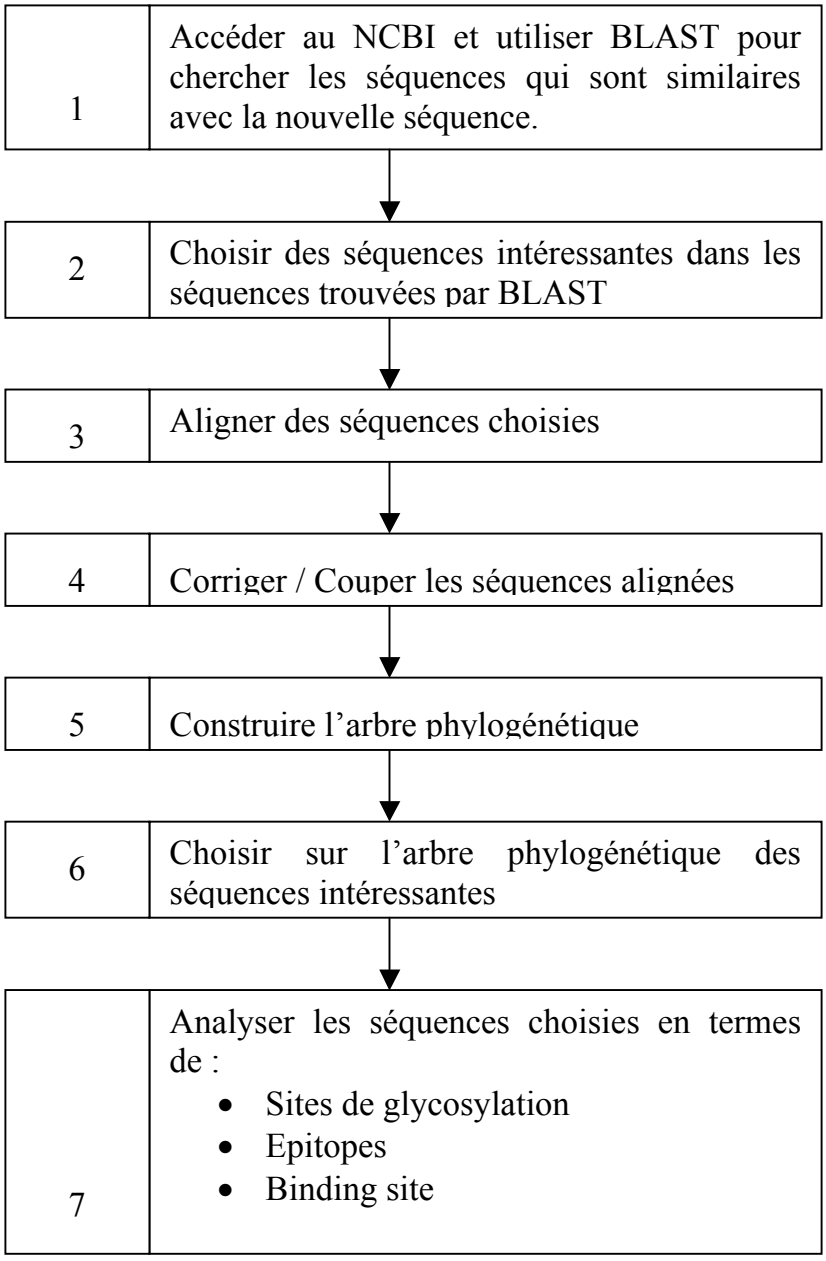


Schéma 1 – Pipeline d'analyse des séquences de la grippe aviaire

- Dans la phase 1, le nombre des séquences analysées par l'outil BLAST que Dr. Hoa utilise sur son ordinateur est limité à 500
- Dans la phase 2, Dr. Hoa doit choisir manuellement une séquence à chaque fois. Ce qu'il veut, c'est une façon de choisir plusieurs séquences à chaque fois
- La phase 4 est une phase manuelle
- Dans la phase 6, comme dans la phase 2, Dr. Hoa veut choisir plusieurs séquences sur l'arbre à chaque fois

Le système g-INFO a été développé pour répondre aux besoins du Dr. Hoa. Il lui donne surtout un nouveau moyen d'analyse de séquences utilisant un gestionnaire de flot. Ce qu'il fait manuellement peut être remplacé par des pipelines automatiques déployés dans g-INFO. Avec la puissance de la grille, la limitation à 500 du nombre des séquences à traiter avec l'outil BLAST peut être levée. L'interface du portail g-INFO lui permet de sélectionner des séquences aux phases 2 et 6. Nous avons choisi d'implémenter dans g-INFO des outils bioinformatiques très populaires et *open source* (BLAST [22], PHYML [16] et MUSCLE [94]). Le système g-INFO est conçu pour intégrer d'autres outils si nécessaire.

4.4. Architecture informatique

Le système g-INFO est implémenté et déployé avec l'environnement de production WPE décrit au chapitre 3. Dans le WPE, les jobs pilotes sont automatiquement soumis à la grille par le Job Manager de telle sorte que les utilisateurs de g-INFO qui n'ont pas l'expérience de grille peuvent facilement exécuter leurs pipelines sur la grille. Cependant, ils doivent parfois télécharger leurs propres données d'entrée (notamment celles qui n'ont pas encore été publiées) sur la grille avant l'exécution d'un pipeline. Pour un utilisateur non-expert de la grille, une opération simple comme cela n'est toujours pas une tâche facile. Le portail g-INFO permettra de simplifier ces tâches liées à la grille pour que les utilisateurs puissent concentrer leurs efforts sur leurs pipelines phylogénétiques.

Pour gérer la création et la manipulation des flots dans g-INFO, nous avons choisi d'utiliser le moteur de flot MOTEUR [93] développé par les laboratoires I3S et CREATIS. MOTEUR est un gestionnaire de flots optimisé pour traiter efficacement des applications manipulant de grandes masses de données sur des infrastructures de grille. MOTEUR exploite plusieurs niveaux de parallélisme et groupe les tâches à réaliser pour réduire le temps d'exécution des applications. De plus, MOTEUR utilise un service web d'encapsulation pour faciliter la réutilisation des codes développés sans prise en compte des spécificités des grilles de calcul. MOTEUR est très utilisé par la communauté de recherche en imagerie médicale sur la grille. Son utilisation s'est élargie aux neurosciences et à la bioinformatique.

L'architecture de g-INFO est illustrée dans la Figure 4-1 :

- Les tâches phylogénétiques sont définies dans le Task Manager du WPE.
- La base de données g-INFO contient les métadonnées de séquences des virus qui sont mises à jour quotidiennement à partir de la base de données NCBI.
- Les services web MOTEUR fournissent les services qui manipulent les tâches phylogénétiques (créer des tâches, vérifier le statut d'une tâche, etc.). Ces services ont pour but de construire un flot.
- Les services web intermédiaires aident le portail à appeler les services

web MOTEUR ou à accéder à la base de données g-INFO.

- Le portail fournit les interfaces web simples pour les fonctions comme : rechercher des séquences, créer des flots, etc. (voir section 4.4.1)

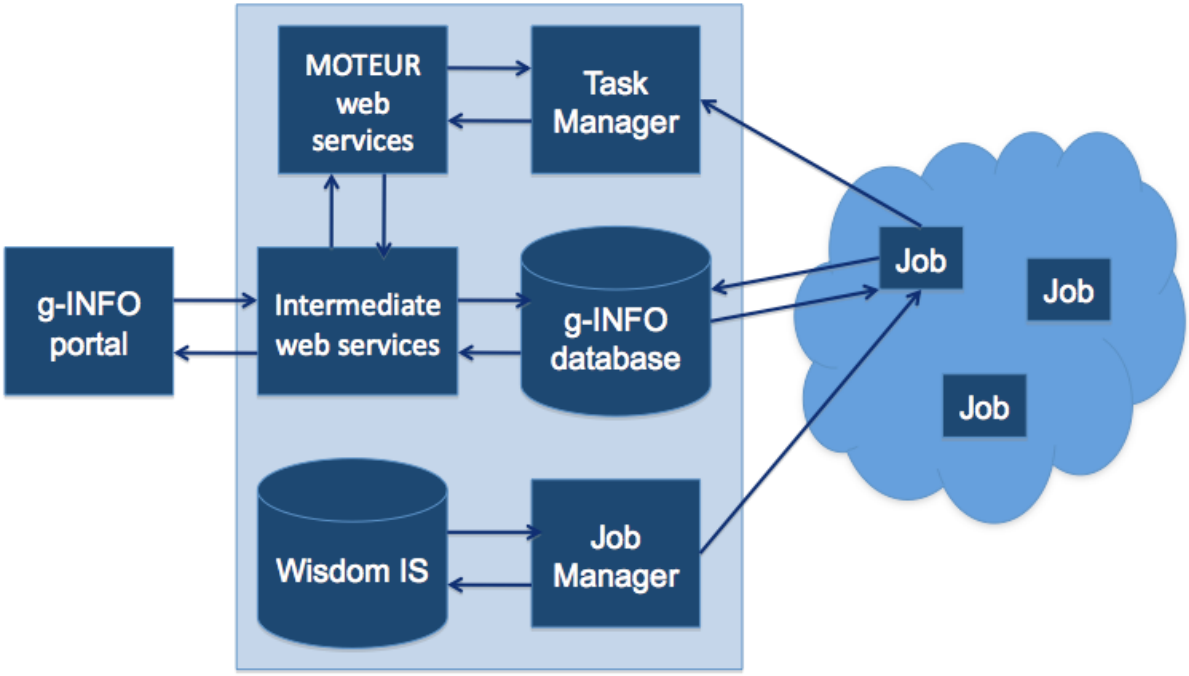


Figure 4-1 - L'architecture de système g-INFO

Le système de g-INFO stocke dans le Task Manager quatre services généraux pour l'analyse phylogénétique:

- *BLAST* pour chercher des régions de similarité entre séquences [22],
- *Muscle* pour aligner les sequences [94],
- *Gblocks* pour corriger les séquences alignées [95]
- et *PhyML* pour construire les arbres phylogénétiques [16]

Grâce aux services disponibles dans le WPE, nous pouvons mettre en œuvre des pipelines phylogénétiques statiques qui s'exécutent automatiquement (section 4.5.2).

Le portail g-INFO a été développé avec plusieurs technologies de web. Ce portail interagit avec le système g-INFO via des services web et aide l'utilisateur à créer des modèles de flots, à exécuter un flot existant et à visualiser des résultats avec une interface conviviale sur un navigateur de web.

Afin de créer un flot avec MOTEUR, nous fournissons pour chaque tâche phylogénétique (dans le Task Manager) un service web asynchrone correspondant. Selon la convention du MOTEUR, chaque service web contient au moins 4 fonctions principales suivantes:

- `submitSequence`: crée une tâche correspondante dans le Task Manager, par exemple le service web de BLAST va créer la tâche BLAST.

- `isFinished`: pour vérifier si une tâche est terminée afin de permettre au flot de se poursuivre avec la prochaine tâche. S'il y a plusieurs instances d'une tâche, le flot s'exécute de manière asynchrone: il n'attend pas pour que toutes les instances d'une tâche soient terminées avant de passer à la tâche suivante.
- `GetOutput`: renvoie le LFN (Logical File Name) d'un fichier sur un élément de stockage qui contient la liste de séquences résultat de `submitSequence` à utiliser pour l'entrée de l'étape suivante. Par exemple, BLAST renvoie une liste de hits, MUSCLE retourne une liste de séquences alignées, etc.
- `getOutputArchive`: retourne le LFN d'une archive pour les autres sorties d'une tâche donc les erreurs ou fichiers de log.

Parce que MOTEUR fonctionne sur la base de services web, il est très naturel d'importer ces services web dans le portail g-INFO afin que les experts puissent créer leurs flots via une interface web conviviale.

4.4.1. Portail

Le portail g-INFO est développé en utilisant plusieurs technologies du web comme : JSF 2.0 (JavaServer Faces Technology, http://www.oracle.com/technetwork/java/javaee/javaserverfaces-139869.html) pour le cadre de base, Ajax (http://www.w3schools.com/ajax/default.asp) pour construire une interface conviviale et JAX-WS (http://jax-ws.java.net/) pour interagir avec le système g-INFO. Le portail interagit avec le système g-INFO via des services web intermédiaires (Figure 4-1). Ces services intermédiaires appellent les services web prédéfinis de MOTEUR pour créer des tâches dans le Task Manager. Ces services web intermédiaires se connectent aussi à la base de données g-INFO pour chercher des séquences de virus ou pour afficher des informations du pipeline à l'utilisateur.

Le portail fournit une méthode d'authentification simple qui nécessite un nom d'utilisateur et un mot de passe pour s'assurer que seuls les utilisateurs enregistrés peuvent utiliser les ressources de la grille accessibles via le portail. Chaque fois qu'un utilisateur accède au portail, un filtre d'autorisation détecte la zone où il veut accéder. Si l'utilisateur souhaite accéder à une zone d'accès limité, comme «Chercher» ou « Gérer les sessions de travail », le filtre le redirige vers la page d'authentification s'il n'est pas encore authentifiée ou n'a pas de droits d'accès appropriés. Après s'être authentifié avec succès, l'utilisateur est redirigé vers la zone qui l'intéresse. Les informations d'identité de l'utilisateur sont stockées dans une variable de session pour que l'utilisateur ne soit pas authentifié à nouveau dans la même session.

La version courante du portail g-INFO supporte les fonctions suivantes:

- **Recherche de séquences** importées de la base de données publique NCBI : l'utilisateur fournit des paramètres de recherche dans un formulaire de requête. Les services web sont lancés et retournent le

résultat. La technique de pagination est utilisée lorsque les résultats contiennent beaucoup des séquences. Ajax est utilisé pour compter rapidement le nombre de séquences trouvées. Si l'utilisateur voit qu'il y a un nombre de séquences trop grand, il peut ajouter des paramètres de recherches supplémentaires pour réduire la taille du résultat de recherche. Les utilisateurs peuvent sélectionner une séquence pour afficher ses informations détaillées ou sélectionner quelques séquences pour télécharger sur sa machine locale au format FASTA

- **Créer et gérer des modèles de flot** : avant de lancer une nouvelle session de travail, un utilisateur doit sélectionner un modèle de flot et fournir des entrées pour les pipelines. Le portail cherche dans sa base de données tous les modèles prédéfinis de l'utilisateur puis en fait une liste que l'utilisateur peut consulter pour en sélectionner un. L'utilisateur peut aussi créer un nouveau modèle de flot et ce modèle peut être réutilisé. Un modèle de flot contient un ensemble d'outils avec leurs paramètres et les relations d'ordre entre ces outils. L'utilisateur sélectionne chaque outil et définit ses paramètres. La technique d'Ajax est utilisée pour accélérer la phase de sélection d'outil. Après avoir été ainsi créé, le nouveau modèle est stocké dans la base de données pour les utilisations futures.
- **Afficher les statuts des sessions de l'utilisateur** : Les utilisateurs peuvent voir le statut de leur session de travail. Chaque session de travail contient des informations sur les pipelines exécutés par un modèle de flot. Chaque pipeline contient des informations de ses composants.
- **Créer et exécuter des flots** dans une session : Après avoir choisi un modèle de flot, l'utilisateur doit fournir des entrées pour l'exécuter. L'utilisateur peut copier-coller des données de séquences dans une zone de texte, ou les télécharger à partir de sa machine locale
- **Visualiser** les sorties (voir section 4.4.2)

4.4.2. Outils de visualisation

De nombreux outils de visualisation ont été développés pour représenter et manipuler des arbres phylogénétiques. Ils permettent de définir des étiquettes, définir la couleur (pour un nœud ou une branche), agrandir et sélectionner (un nœud seulement ou toute la branche). Ici, la tâche essentielle est de fournir un outil de visualisation intégré dans le portail pour que les spécialistes puissent manipuler les résultats du pipeline. Parmi les outils libres disponibles pour la visualisation, l'outil PhyloWidget [96] a été choisi.

PhyloWidget est un logiciel libre pour visualiser, éditer et publier en ligne des arbres phylogénétiques. PhyloWidget contient une interface utilisateur simple mais efficace et quelques fonctionnalités utiles qui ne sont pas disponibles dans les autres outils de visualisation phylogénétique (Tableau 4-1). Outre PhyloWidget, il existe d'autres logiciels comme Archaeopteryx [97],

TreeViewJ [98], et Baobab [99], qui sont aussi utilisés largement par la communauté des épidémiologistes moléculaires. Le Tableau 4-1 présente une comparaison des caractéristiques de ces outils de visualisation.

Caractéristiques	(1)	(2)	(3)	(4)
Supporter le format Newick	oui	oui	oui	oui
Étiqueter et ressortir des nœuds	oui	oui	oui	non
Zoomer	oui	oui	oui	oui
Ajouter / effacer des nœuds ou des branches	oui	oui	non	oui
Supporter plusieurs types de visualiser un arbre	oui	oui	oui	oui
Interface simple	oui	non	non	non
Code source simple	oui	non	oui	oui

Tableau 4-1 - Comparaison des outils de visualisation. (1) – PhyloWidget (2) – Archaeopteryx (3) – TreeViewJ (4) - Baobab

PhyloWidget répond très bien au cahier des charges de la visualisation des arbres phylogénétiques. Cet outil fonctionne efficacement et supporte les raccourcis de clavier, ce qui aide l'utilisateur à manipuler des arbres rapidement et efficacement. Les opérations sur PhyloWidget sont assez flexibles et personnalisables. PhyloWidget possède un support complet pour la manipulation de l'arbre.

Quelques nouvelles fonctions ont été ajoutées à PhyloWidget afin qu'il puisse travailler avec le portail g-INFO:

- Recevoir des informations de sortie d'un flot phylogénétique via des services web intermédiaires.
- Télécharger la sortie d'un pipeline à partir d'un Elément de Stockage pour la visualiser.
- Enregistrer l'arbre phylogénétique édité au format FASTA. Les arbres au format Newick (un format populaire des arbres phylogénétiques) contiennent des identifications de séquences seulement. Pour obtenir des données de séquence, PhyloWidget appelle les services web intermédiaires pour interroger la base de données de séquence de g-INFO. Cette nouvelle fonction permet aux utilisateurs d'utiliser la sortie

d'un arbre phylogénétique pour d'autres analyses et de sélectionner nœuds et branches sur l'arbre pour éditer (colorer, enlever ou sauvegarder). Cette fonction est très utile pour les chercheurs, mais elle est absente dans la plupart des outils de visualisation.

4.5. Résultats

4.5.1. Fonctions du portail g-INFO

Le portail g-INFO est actuellement dans la dernière phase de tests internes et sera bientôt mis en mode de production. Le portail actuel fournit la plupart des fonctions nécessaires pour aider les épidémiologistes à surveiller des virus de la grippe de type A.

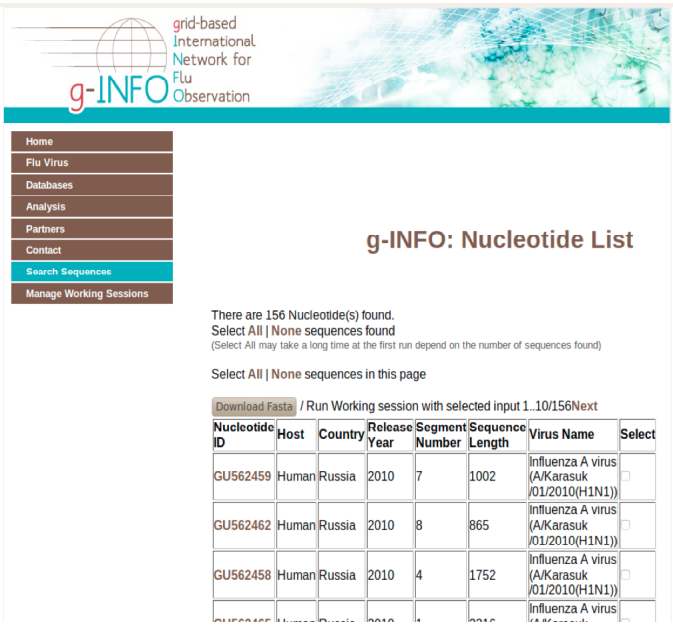


Figure 4-2 - Chercher les séquences H5N1 sur le portail g-INFO

Les utilisateurs du portail peuvent chercher des données de séquences avec des critères communs tels que : l'hôte, le nom, le segment, la protéine, etc. Les résultats peuvent être téléchargés sur l'ordinateur local de l'utilisateur ou utilisés pour exécuter des pipelines sur la grille. La Figure 4-2 illustre une page de résultat retournée par une requête.

Les utilisateurs du portail peuvent définir des modèles de flots réutilisables. Ces modèles de flots sont re-configurables avec plusieurs outils intégrés et leurs paramètres.

A chaque exécution d'un flot, les utilisateurs du portail peuvent fournir de nombreuses entrées à la fois. Chaque entrée va créer un pipeline correspondant et tous les pipelines seront exécutés de façon asynchrone sur des éléments de Calcul de la grille. Un exemple de gestion d'un flot est montré dans la Figure 4-3. Quand un utilisateur clique sur le bouton "Finish", le flot s'exécutera avec deux entrées fournies par l'étape

précédente.

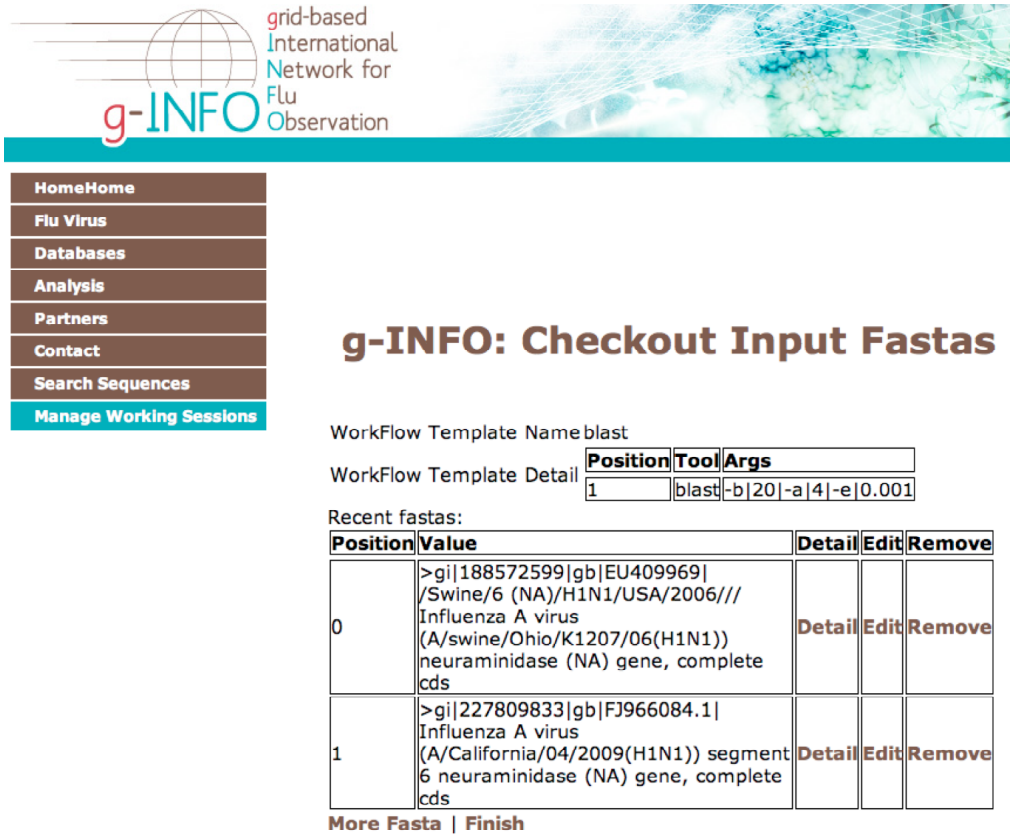


Figure 4-3 - Lancer un flot sur le portail g-INFO

Les paramètres d'une session de travail, y compris les paramètres de flot, l'état du pipeline, l'entrée et la sortie sont enregistrés afin que les utilisateurs puissent les consulter plus tard.

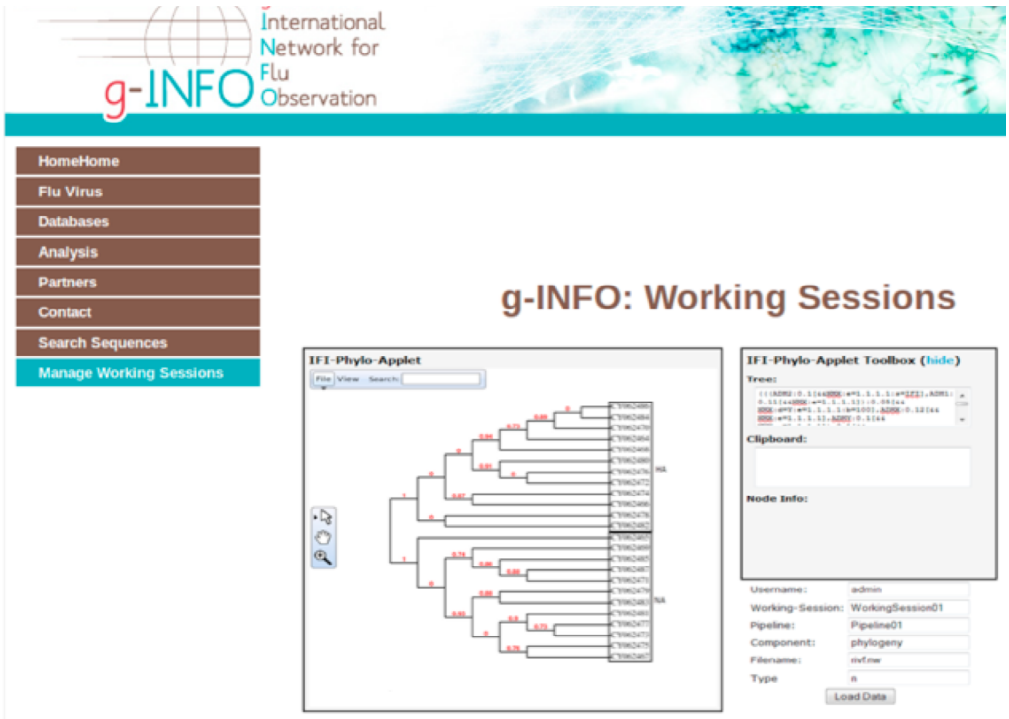


Figure 4-4 - Outil de visualisation dans le portail

L'outil de visualisation modifié à partir de PhyloWidget est intégré dans le portail sous forme d'un applet pour visualiser la sortie des arbres phylogénétiques. Les utilisateurs peuvent manipuler les branches et les nœuds dans les arbres de résultats, puis enregistrer l'arbre édité en format FASTA ou Newick sur leur machine local. La Figure 4-4 illustre l'utilisation de l'outil de visualisation dans le portail g-INFO.

4.5.2. Exemples de la surveillance de H5N1 avec g-INFO

Nous avons fait un test en utilisant un flot créé sur le portail g-INFO avec 3 composantes (Muscle, Gblocks et PhyML). En entrée du test, les séquences des souches H5N1 de l'année 2010 ont été utilisées. Le résultat est montré sur la Figure 4-5 où les séquences sont groupées dans deux branches correspondant aux séquences HA et NA.

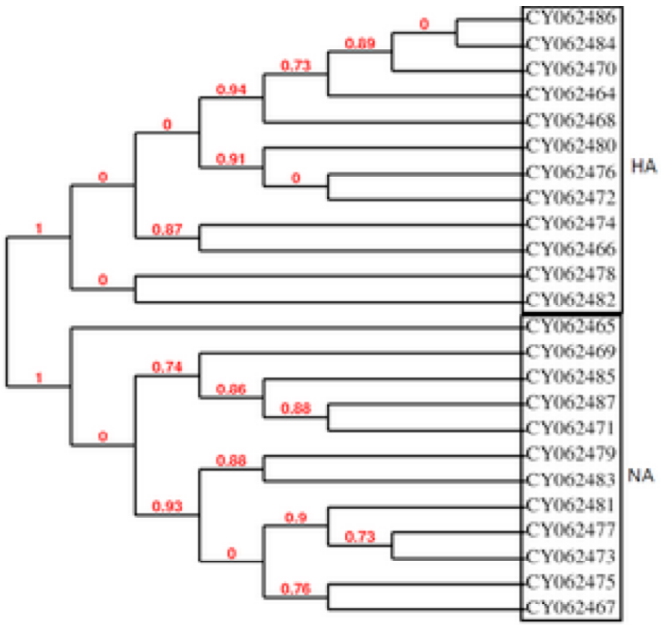


Figure 4-5 - Groupes de HA et NA de séquences de H5N1

Dans le deuxième test, nous étudions les différences sur les segments HA / NA pour les souches de virus H5N1 entre l'année 2009 et 2010. Dans la figure 2, nous pouvons voir que la plupart des séquences virales de l'année 2010 (en couleur rouge) sont regroupées.

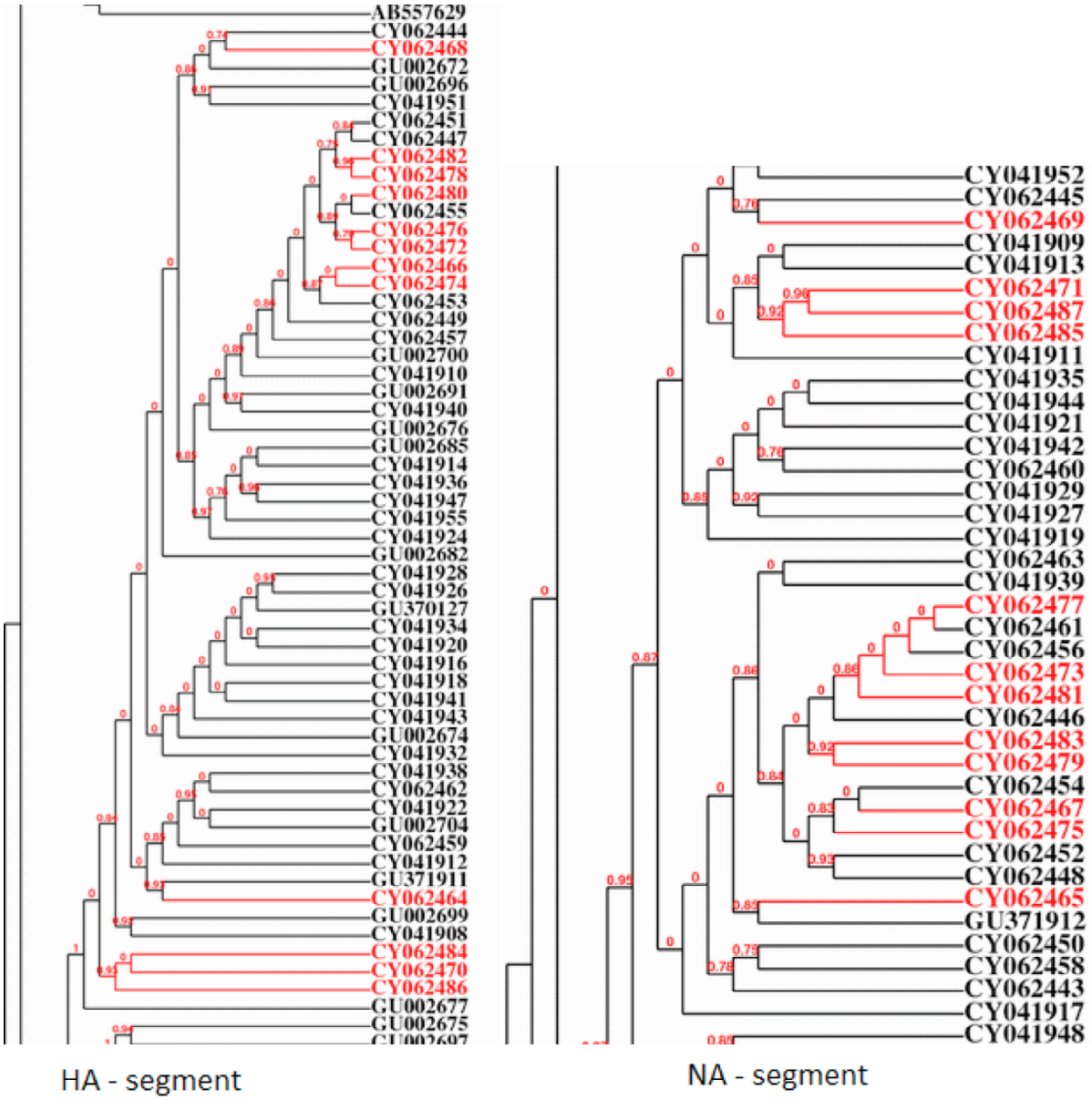


Figure 4-6 - La différence de segment HA / NA de souches de virus H5N1 entre l'année 2009 et 2010

Dans une session de travail, nous pouvons lancer simultanément plusieurs instances (pipelines) d'un flot pour profiter la puissance de la grille. En fournissant de multiples entrées à un flot, le système g-INFO va générer plusieurs instances de ce flot et les laisser s'exécuter en parallèle.

Un autre cas de figure est celui où un flot peut contenir plusieurs branches parallèles. Considérons le test suivant : dans ce test, nous ajoutons une composante BLAST à la tête du flot dans les exemples précédents. Pour chaque séquence dans le fichier d'entrée, BLAST cherchera *n* séquences dans la base de donnée qui sont similaires à cette séquence d'entrée. Le reste du flot sera de construire un arbre phylogénétique à partir de ces *n + 1* séquences (la séquence d'entrée et *n* séquences similaires). Si l'on suppose maintenant que le fichier d'entrée contient *N* séquences, le flot va générer *N* arbres phylogénétiques de *n+1* séquences, soit N instances du même flot. Dans le système g-INFO,

chaque job pilote déployé sur la grille peut effectuer une tâche. Dès que la tâche a été terminée, le job est libre de gérer une autre tâche. En conséquence, nous pouvons conclure que nous devons avoir au moins N jobs pour exécuter simultanément les N instances du flot. Ceci est illustré dans la Figure 4-7 pour le cas N=3. La plupart du temps, le flot nécessite 3 jobs pour exécuter les 3 instances I_1, I_2 et I_3.

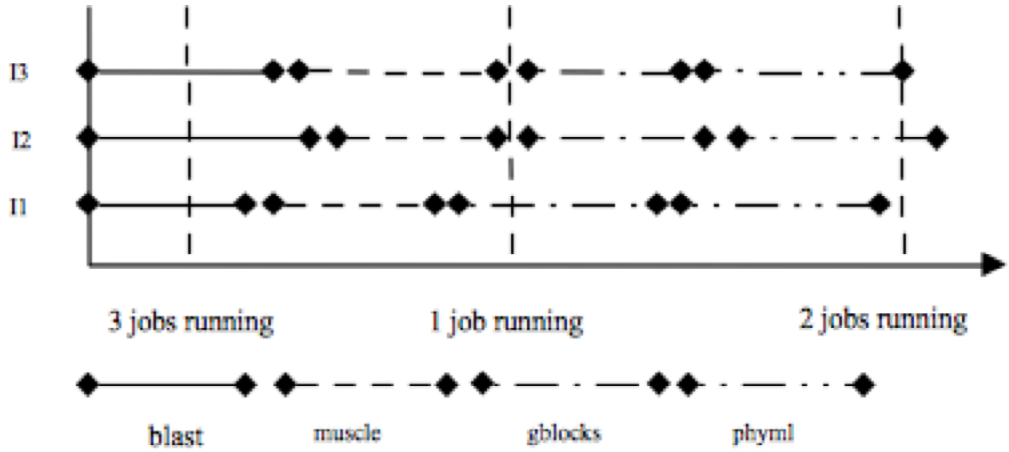
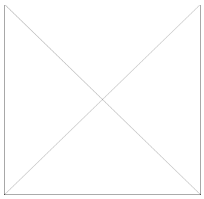


Figure 4-7 - Le nombre des jobs nécessaires pour exécuter un flot avec 3 instances

Supposons que le WPE ait soumis M jobs et M > N, si T est le temps total du flot, T_i est le temps de chaque instance, nous avons :



Le Tableau 4-2 montre le temps d'exécution d'un flot de 12 séquences de H5N1 (segment HA, année 2010, hôte humain). Le temps d'exécuter 12 instances du flot est de 897 (s) alors que le maximum de temps d'exécution d'une instance est de 866 (s). Nous n'utilisons que 20 jobs pour ce test. Dans le mode de production réel, le WPE a approximativement 3000 jobs disponibles sur la grille.

Séquence	**CY062476**	**CY062478**	**CY062480**	**CY062482**	**CY062484**	**CY062486**
T_i (s)	**792**	**709**	**830**	**866**	**727**	**710**
Séquence	**CY062466**	**CY062464**	**CY062468**	**CY062472**	**CY062470**	**CY062470**
T_i (s)	**749**	**851**	**785**	**771**	**708**	**728**
Tmin = 708s Tmax = 866s Tmean = 769s T = 897s						

Tableau 4-2 - Temps d'exécuter d'un flot

En bref, les exemples ci-dessus montrent la capacité de g-INFO à gérer la surveillance de la grippe aviaire. Le portail g-INFO fournit des outils visuels pour aider les épidémiologistes à créer et exécuter facilement les flots phylogénétiques pour l'analyse des séquences virales. Grâce à la puissance de la grille, les experts peuvent exécuter simultanément les flots pour plusieurs séquences en entrée ou pour une seule séquence et plusieurs jeux de paramètres. Ainsi le temps d'analyse est diminué considérablement, les experts peuvent fournir des alertes précoces en cas d'une épidémie qui est susceptible de se produire.

4.6. Conclusion

La plate-forme g-INFO apporte une solution innovante pour la surveillance des maladies émergentes au Vietnam. g-INFO fournit non seulement une plate-forme mais aussi des outils extensibles et configurables.

Le déploiement de g-INFO sur la grille au Vietnam a rencontré des gros problèmes avec l'infrastructure réseau. Toutes les connexions de grille au Vietnam sont faites sur TEIN2, un projet international dont les objectifs sont les suivants :

- Augmenter la connectivité directe à Internet pour la recherche et l'éducation entre l'Europe et l'Asie
- Améliorer la connectivité intra-régionale en Asie
- Agir comme un catalyseur pour le développement des réseaux de recherche nationaux dans les pays en développement dans la région Asie-Pacifique.

Bien que TEIN2 offre en principe une infrastructure de connexion à haute vitesse qui remplit les conditions requises pour l'opération de sites de la grille, de nombreux problèmes ont été rencontrés dans le processus de déploiement des nœuds de la grille au Vietnam : instabilité de la connexion, les capacités limitées de gestion, etc.

CHAPITRE 5 : EPANAM

Le chapitre 4 a présenté une étude de cas de l´utilisation du WPE pour la construction d´un réseau de surveillance de la grippe aviaire. Ce chapitre montrera une autre application du WPE dans un autre domaine : le séquençage haut débit de nouvelle génération. La structure du chapitre est comme suit : la section 5.1 présente d´abord les enjeux scientifiques dans ce domaine. Ensuite, dans la section 5.2, nous allons proposer une solution pour le séquençage de nouvelle génération basée sur le WPE. Les tests de performances seront abordés dans la section 5.3.

5.1. Enjeux scientifiques

La métagénomique est définie comme « l'application des techniques génomiques modernes pour l'étude des communautés d'organismes microbiens directement dans leurs milieux naturels, en contournant la nécessité pour l'isolement et la culture en laboratoire des espèces individuelles » [100]. L'impact potentiel d'approches métagénomiques pour l'étude des écosystèmes a été évoqué depuis la fin des années 1990 [101] mais ce champ de recherche a explosé avec l'arrivée du séquençage à haut débit et notamment du séquenceur Roche/454 [102]. Avec l'avènement d'un échantillonnage véritablement aléatoire des génomes (l'approche *shotgun* du séquençage) et de nouveaux protocoles [103], les sciences des estuaires et l'étude des environnements côtiers expérimentent une renaissance par la génomique. Par définition, la métagénomique est l'étude du matériel génétique à partir d'un échantillon de l'environnement. Bien qu'une grande partie de la métagénomique se concentre sur les bactéries, le domaine est en pleine expansion pour englober toute la diversité microbienne.

Le séquençage à haut débit (High Throughput Sequencing – HTS) offre aux chercheurs et aux scientifiques un outil révolutionnaire pour obtenir rapidement et à moindre coût des séquences d'ADN. Le génome humain a été séquencé initialement par un ensemble de laboratoires équipés avec les instruments de séquençage Sanger. Il a fallu 13 ans et 3 milliards USD ont été dépensés du prélèvement de l'échantillon à la publication du génome humain complet [104]. Aujourd'hui, un génome humain peut être séquencé de novo en 2 mois pour seulement 1 million de dollars US [105].

Les données générées par les expériences de métagénomique sont à la fois massives et intrinsèquement bruitées, contenant des informations génomiques fragmentées d'environ 10.000 espèces [106]. Le séquençage du métagénome

du rumen de la vache, par exemple, a généré 279 milliards de paires de base de données de séquences nucléotidiques [107], tandis que le catalogue de gènes microbiens de l'intestin humain a identifié 3,3 millions de gènes assemblés à partir de 567,7 gigabases de données de sequences [108]. La collection, la conservation et l'extraction des informations biologiques utiles des ensembles de données de cette taille représentent d'importants défis pour les chercheurs et requièrent l'accès à des ressources importantes de calcul pour lesquelles la grille est une solution.

Les progrès récents dans les méthodes de séquençage avec le développement rapide des technologies du séquençage de nouvelle génération (*next generation sequencing*) permettent aux écologistes microbiens de faire des expériences avec une grande quantité d'informations pour analyser des marqueurs phylogénétiques comme la petite sous-unité (PSU) du ribosome. Ces nouvelles possibilités méthodologiques permettent d'étudier la diversité des écosystèmes, par exemple la biosphère rare [109]. Le pyroséquençage de la région hypervariable PSU de l'ARNr devient une approche standard pour étudier la diversité microbienne du sol [110], de l'océan [111] ou de l'intestin humain [112].

En raison du volume, jusqu'à 1,5 million de séquences, et de la complexité des données produites par une expérience de pyroséquençage, le choix du protocole d'analyse bioinformatique est primordial à l'extraction d'une information biologique pertinente. Certains développements ont été proposés pour traiter les aspects particuliers de cette nouvelle technologie, tels que la détection d'erreur de séquençage, ou la correction de biais dans la détermination de la richesse en espèces dans un écosystème [113].

PANAM [114] est un pipeline développé par l'équipe Microbiologie de l'Environnement et Bioinformatique du laboratoire LMGE, dirigée par le professeur Didier Debroas. PANAM est l'objet de la thèse de Najwa Taib. Ce pipeline est dédié à l'analyse de séquences environnementales issues du séquençage de nouvelle génération (*New Generation Sequencing - NGS*). Il permet l'annotation phylogénétique automatisée de séquences d'amplicons de la petite sous-unité du ribosome. Ce pipeline doit être proposé à la communauté scientifique et intégré à ePANAM, une ressource web pour l'analyse de la diversité microbienne. ePANAM permet de faire un contrôle sur la qualité des séquences, de calculer des indices de richesse et de la diversité, de comparer les écosystèmes, d'attribuer une taxonomie aux séquences expérimentales et de décrire les clades environnementaux d'intérêt. Ce pipeline est basé sur des outils disponibles publiquement comme UCLUST [115], HMMER [15] et FASTTREE [116]. Il comporte trois étapes :

- une phase de comparaison des séquences à celles contenues dans une base de données de référence
- une phase de clusterisation, au cours de laquelle les séquences vont être rassemblées en fonction des différentes espèces présentes dans l'écosystème

- une phase de traitement phylogénétique pour caractériser les relations de parenté entre les espèces présentes

PANAM est aujourd'hui un outil très original et très en avance sur le plan international car il permet une étude exhaustive de la diversité microbienne et une exploitation phylogénétique des données issues du pyroséquençage. Par contre, le prix à payer pour la richesse du traitement effectué est son coût en temps de calcul.

Sur un PC, processeur Intel (R) Xeon (R) CPU 32 bits à 2 GHz et 24 Go de RAM, PANAM permet de réaliser la phylogénie de 1000 séquences de 200 pb (paires de base) en environ 20 minutes. Pour des séquences de 400 pb générées par les pyroséquenceurs actuels, le temps d'exécution varie de 24 minutes pour 5.000 séquences à 6 jours et 14 heures pour 1.000.000 de séquences. Si ces temps d'exécution sont acceptables pour une utilisation individuelle, il sont prohibitifs pour que cette approche devienne une ressource web utilisable par la communauté mondiale des microbiologistes. Ainsi, les performances de PANAM seraient améliorées en utilisant des approches de parallélisation et des technologies comme la grille de calcul.

Nous présentons dans le reste de ce chapitre notre expérience de porter PANAM sur la grille [117].

5.2. Architecture informatique

Le traitement des séquences par PANAM est réalisé en deux étapes. La première consiste à comparer les séquences expérimentales d'entrée à une base de référence constituée de séquences extraites de SILVA [118] avec USEARCH [115]. Ensuite, ces séquences expérimentales sont associées à des groupes phylétiques prédéfinis, constitués à partir des séquences de référence.

Dans la deuxième étape, les séquences expérimentales sont alignées contre les profils d'alignement des séquences de références auxquelles elles sont associées. Un arbre phylogénétique contenant les séquences de référence et les séquences expérimentales est généré pour chaque groupe phylétique, et une affiliation taxonomique est inférée pour chaque séquence expérimentale selon sa position dans l'arbre.

Les profils d'alignement ont été construits selon le critère de la monophylie d'un «arbre guide » phylogénétique fourni par SILVA [118] avec la base de données de la PSU. Les phylogénies ont été obtenues en utilisant l'outil FASTTREE (les arbres produits par FASTTREE sont considérés de la même qualité que ceux de PhyML [16]). Pour construire les arbres phylogénétiques, deux méthodes de parcours sont mises en œuvre : le plus petit ancêtre commun (LCA – *Lowest Common Ancestor*) et le voisin le plus proche (NN – *Nearest Neigbour*) :

- PANAM-NN : Pour chaque séquence, tous les nœuds sont scannés du nœud le plus récent au nœud le plus profond. Le plus proche voisin est défini comme la première séquence référée à partir du nœud le plus bas. La séquence va acquérir la taxonomie complète de son voisin le plus

proche. C'est aussi la méthode d'affiliation taxonomique mise en œuvre dans STAP [119].

- PANAM-LCA : chaque nœud ne contient que la taxonomie commune entre tous ses descendants et peut être donc incomplet. Chaque séquence hérite de la taxonomie de son nœud le plus bas.

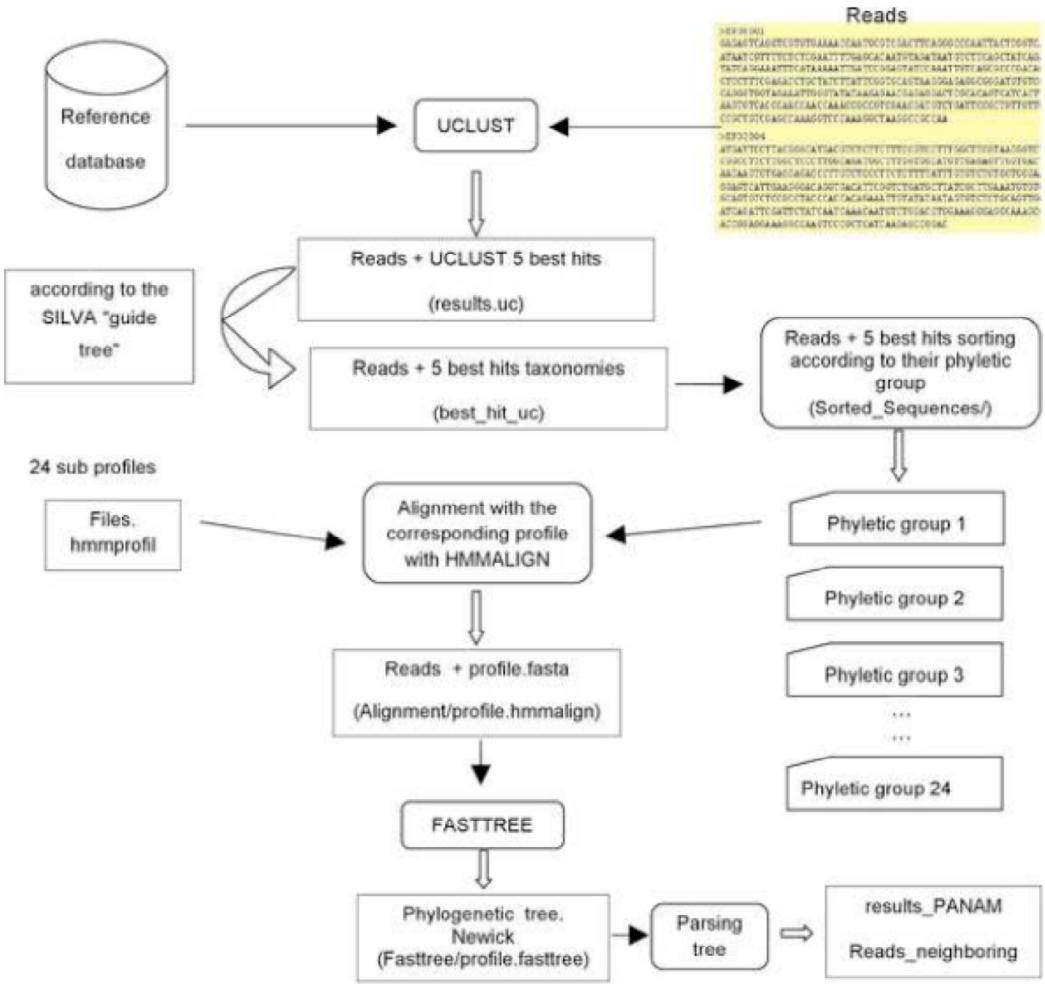


Figure 5-1 - Le pipeline de ePANAM

Pour déployer le pipeline PANAM sur la grille avec le WPE, 4 services ont été créés (voir la Figure 5-1 - Le pipeline de ePANAM) :

- Le service `g-panam-trim` adapte les profils d'alignement des séquences de références à la région amplifiée des séquences expérimentales avant de lancer le traitement phylogénétique.
- Le service `g-panam-split` trie les séquences selon le groupe phylétique inféré par USEARCH et les affecte aux différents groupes phylétiques.
- Pour chaque groupe phylétique non vide, le service `g-panam-alnphy` aligne des séquences par un outil d'alignement (HMMER), avant de construire les phylogénies par l'outil FASTTREE. Cette phase est parallélisée selon le nombre des fichiers récupérés à la fin du service

précédent (`g-panam-split`).

- Finalement, le service `g-panam-clade` parcourt les arbres, assigne une taxonomie pour chaque séquence et infère les clades.

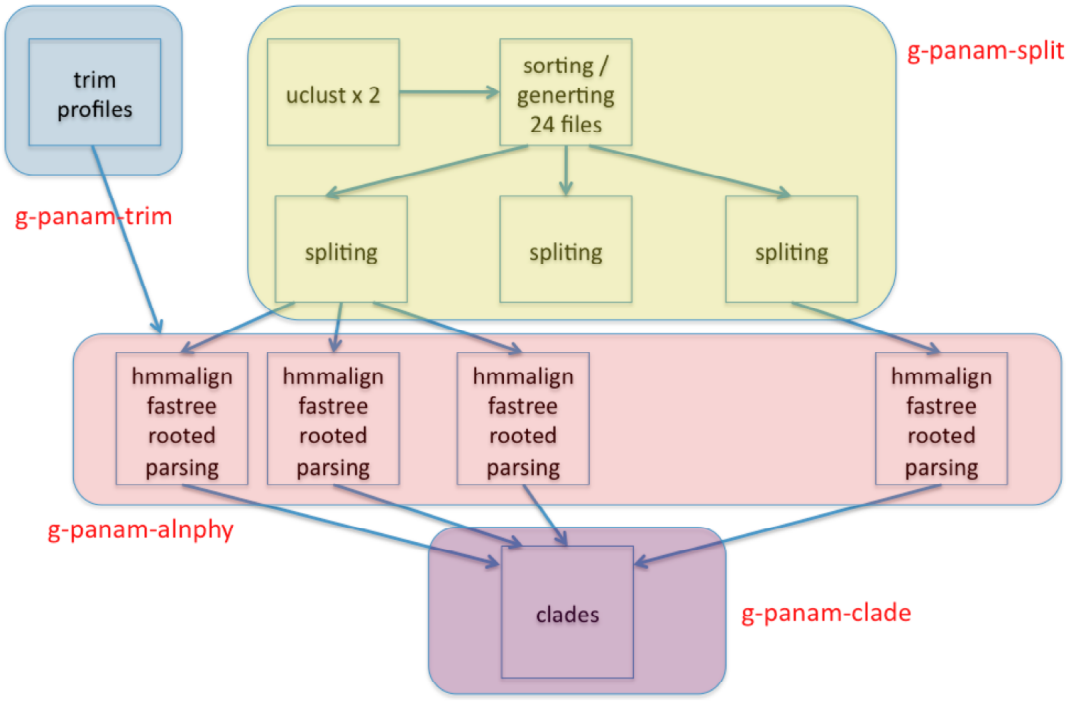


Figure 5-2 – Gridification du pipeline ePANAM

5.3. Résultats

5.3.1. Comparaison avec un déploiement sur une machine locale

Le tableau Tableau 5-1 présente les temps de calcul mesurés par Najwa Taib pour des jeux de séquences de taille variable sur un seul ordinateur local (processeur Intel (R) Xeon (R) CPU 32 bits à 2 GHz et 24 Go de RAM) en utilisant la méthode de parcours LCA. La méthode de parcours NN requérait un temps de calcul bien supérieur à la méthode LCA lorsque les premiers tests sur la grille ont été réalisés. Elle a été optimisée depuis par l'équipe du LMGE et ses temps d'exécution sont maintenant comparables à ceux de la méthode LCA.

Nombre de séquences / longueur des séquences	100 bp	200 bp	400 bp	800 bp	>1200 bp
5000	0,33	0,37	0,4	0,53	2,2
50000	0,82	0,97	1,08	1,65	12,17
100000	2,03	2,17	2,63	3,48	24,12
250000	9,77	11	11,82	13,98	64,68
500000	37,27	37,42	38,75	43,28	154,77
750000	82,68	86,25	90,58	103,32	263,27
1000000	143,35	143,58	158,73	172,52	382,58

Tableau 5-1 - Temps de calcul mesurés par Najwa Taib pour des jeux de séquences de taille variable sur un seul ordinateur local (processeur Intel (R) Xeon (R) CPU 32 bits à 2 GHz et 24 Go de RAM) en utilisant la méthode de parcours LCA.

A des fins de comparaison, nous avons concentré nos tests sur des échantillons de séquences de longueur 400bp.

Les tableaux ci-dessous présentent pour différents nombres de séquences les performances pour le déploiement du pipeline PANAM sur la grille pour les deux méthodes de parcours (LCA et NN). Les temps de calculs sont indiqués en heure pour les trois services g-panam-split, g-panam-alnphy et g-panam-clade. Le service `g-panam-alnphy` est parallélisé sur la grille : nous avons fait figurer dans le tableau le temps moyen d'exécution sur la grille des tâches (Mean AlnPhy) et non le temps d'exécution de la tâche la plus lente (Max AlnPhy). Comme la dernière étape `g-panam-clade` ne peut démarrer qu'une fois toutes les tâches de l'étape précédente finie, le temps total est donc toujours supérieur à la somme des temps (Split, Mean AlnPhy, Clade).

Nous avons répété plusieurs fois les tests à des moments différents de la journée et/ou de la semaine pour évaluer de façon réaliste l'impact de la charge de la grille au moment de la soumission.

Workflow	Total	Split	Mean AlnPhy	Clade
Test70s25LCA	3h10	2h39	0h10	0h14
Test107s25LCA	5h32	3h15	0h20	2h17
Test109s25LCA	2h36	1h46	0h4	0h50
Test80s25NN	9h7	1h44	1h8	3h26
Test81s25NN	8h13	2h13	1h13	5h59
Test79s25NN	9h43	1h53	1h19	7h50

Tableau 5-2 - temps de calcul sur la grille pour une échantillon de 250.000 séquences et pour les deux méthodes de parcours (LCA, NN). Sur une machine locale, le temps d'exécution est de 11 heures et 49 minutes pour la méthode LCA.

Workflow	Total	Split	Mean AlnPhy	Clade
Test88s50LCA	6h22	2h33	0h22	3h48
Test86s50NN	7h52	2h11	4h7	5h39
Test98s50NN	9h49	2h15	3h58	6h29
Test100s50NN	10h17	1h42	2h33	7h34

Tableau 5-3 - temps de calcul sur la grille pour une échantillon de 500.000 séquences et pour les deux méthodes de parcours (LCA, NN). Sur une machine locale, le temps d'exécution est de 38h25m pour la méthode LCA.

Workflow	Total	Split	Mean AlnPhy	Clade
T40s750LCA	8:49	5:51	0:28	2:41
T53s750LCA	9:23	8:6	0:23	0:57
T54s750LCA	12:2	5:22	0:29	0:32
T56s750NN	16:47	10:35	3:15	0:27
T58s750NN	13:58	8:46	3:18	0:39
T92s750NN	17:51	5:15	3:15	0:14

Tableau 5-4 - temps de calcul sur la grille pour un échantillon de 750.000 séquences et pour les deux méthodes de parcours (LCA, NN). Sur une machine locale, le temps d'exécution est de 90h36m pour la méthode LCA.

Workflow	Total	Split	Mean AlnPhy	Clade
T69s1mLCA	8:58	5:42	0:22	0:23
T73s1mLCA	11:24	7:15	0:50	0:46
T81s1mLCA	12:43	7:18	0:27	0:25
T86s1mLCA	14:21	8:57	0:17	0:31
T101s1mLCA	11:8	9:45	0:32	0:42

Tableau 5-5 - temps de calcul sur la grille pour un échantillon de 750.000 séquences et pour les deux méthodes de parcours (LCA, NN). Sur une machine locale, le temps d'exécution est de 158h43m pour la méthode LCA.

La comparaison des tableaux met en évidence les conséquences positives et négatives de l'utilisation de la grille :

- parmi les points très positifs, il faut noter que la parallélisation du service g-panam-alnphy permet de garder en moyenne la durée de cette étape à moins d'une heure pour la méthode LCA, quelque soit le nombre de séquences en entrée.
- Par contre, l'utilisation de la grille introduit des fluctuations significatives dans les temps totaux d'exécution.

5.3.2. Variabilité des performances sur la grille

Pour mieux quantifier l'impact de ces fluctuations sur les temps d'exécution, nous avons refait 20 fois chacun des tests avec un nombre de séquences égal à 250.000, 500.000 et 1.000.000. Nous avons utilisé pour ces tests la plate-forme DIRAC et les deux versions de WISDOM, avant (v0) et après (v2) les modifications apportées à la plate-forme qui sont décrites dans le chapitre 3.

Ces modifications ont deux impacts principaux sur les performances :

- la première est un accroissement considérable du taux de succès. Grâce à l'amélioration du mécanisme de resoumission automatique des tâches, la batterie de 20 tests a été beaucoup plus rapide à mettre en œuvre. Par exemple, pour le fichier d'entrée de 250.000 séquences, il a fallu soumettre plus de 30 tests pour avoir 20 tests réussis avec la version v0, tandis que moins de 25 tests ont été nécessaires pour la version modifiée v2.
- la seconde est l'accroissement du temps d'exécution lié à la resoumission automatique des jobs. Cette resoumissions accroît le taux de succès mais s'accompagne d'une augmentation du temps de calcul.

5.3.2.1. Résultats obtenus pour la méthode de parcours LCA

Les tableaux suivants montrent les temps de calcul de 20 tests réussites de ePANAM avec la méthode de parcours LCA.

Temps d'exécution en secondes (méthode LCA)	WPE v0	DIRAC	WPE v2
Test 1	6471	3617	5580
Test 2	4825	4434	7938
Test 3	5190	3593	5657
Test 4	7033	3785	5308
Test 5	6775	2899	8088
Test 6	7209	3325	6227
Test 7	5717	3869	5477
Test 8	6289	4349	5509
Test 9	5136	5244	6256
Test 10	5471	6851	5544
Test 11	7210	5368	7385
Test 12	6200	6121	5263
Test 13	7713	4787	8473
Test 14	6845	3528	7962
Test 15	5491	5181	5554
Test 16	5516	5987	4977
Test 17	6113	4227	5360
Test 18	7004	4517	6403
Test 19	5238	5134	8042
Test 20	5359	4985	5667
Moyenne	6140	4590	6334
Ecart-type	854	1031	1170

Tableau 5-6 - Comparaison des temps d'exécution de 20 tests effectués sur la grille avec un fichier d'entrée de 250.000 séquences

Temps d'exécution en secondes (méthode LCA)	WPE v0	DIRAC	WPE v2
Test 1	10238	8925	13129
Test 2	9956	5823	7855
Test 3	13825	10238	8410
Test 4	13016	6643	7558
Test 5	8993	5900	7267
Test 6	8640	8723	11209
Test 7	13200	6904	12874
Test 8	9728	6207	10509
Test 9	9015	5897	7916
Test 10	7367	6695	8130
Test 11	7761	6357	10112
Test 12	10298	7120	8930
Test 13	14329	6718	12383
Test 14	13200	6037	10674
Test 15	12084	6294	11321
Test 16	13116	10256	7993
Test 17	9233	6260	8150
Test 18	9654	5072	9036
Test 19	12134	6399	13023
Test 20	8703	6822	7225
Moyenne	10725	6965	9685
Ecart-type	2170	1433	2058

Tableau 5-7 - Comparaison des temps d'exécution en secondes de 20 tests effectués sur la grille avec un fichier d'entrée de 500.000 séquences

Temps d'exécution en secondes (méthode LCA)	WPE v0	DIRAC	WPE v2
Test 1	22031	13326	20145
Test 2	21180	11039	21603
Test 3	19922	11631	19378
Test 4	23496	11647	20769
Test 5	20341	11226	20644
Test 6	20084	12157	21154
Test 7	20835	11607	19937
Test 8	19100	15071	20700
Test 9	20101	12340	20249
Test 10	22399	14265	21509
Test 11	19538	11943	21338
Test 12	24193	11582	21826
Test 13	21402	12121	26116
Test 14	20335	13023	20993
Test 15	19372	14470	22224
Test 16	21186	11882	21782
Test 17	21008	12678	25417
Test 18	22109	13223	23024
Test 19	20019	13005	21092
Test 20	19919	11939	19987
Moyenne	20929	12509	21494
Ecart-type	1358	1113	1698

Tableau 5-8 - Comparaison des temps d'exécution de 20 tests effectués sur la grille avec un fichier d'entrée de 1.000.000 séquences

5.3.2.2. Résultats obtenus pour la méthode de parcours NN

Comme nous l'avons souligné précédemment, la méthode de parcours NN était 3 à 5 fois plus lente que la méthode LCA lorsque nous avons effectué nos tests, au point qu'elle n'était pas utilisable sur les fichiers d'entrée de 1 million de séquences. Pour que les tests soient possibles sur la grille, il a fallu accroître la durée de vie du proxy (voir chapitre 3). Même si la méthode a été optimisée depuis que les tests ont été effectués, il n'en demeure pas moins que la parallélisation sur la grille a rendu possible des analyses qui n'étaient pas envisageables sur une machine individuelle, même pour les concepteurs du

service PANAM.

Temps d'exécution en secondes (méthode NN)	WPE v0	DIRAC	WPE v2
Test 1	31032	26938	41043
Test 2	38401	30105	32302
Test 3	28430	32853	29874
Test 4	29337	32161	38456
Test 5	37105	36350	34440
Test 6	42045	38190	29350
Test 7	36323	30656	40168
Test 8	29345	47191	34852
Test 9	32239	36941	39482
Test 10	29675	36092	33452
Test 11	39123	30943	28224
Test 12	38741	36849	32856
Test 13	34320	59774	39921
Test 14	30102	56105	42239
Test 15	28390	33536	43103
Test 16	32012	40772	32109
Test 17	29792	41290	35290
Test 18	30239	30305	41203
Test 19	31246	34952	33245
Test 20	28450	42477	35619
Moyenne	32817	37724	35861
Ecart-type	4267	8491	4541

Tableau 5-9 - Comparaison des temps d'exécution de 20 tests effectués sur la grille avec un fichier d'entrée de 250.000 séquences

Temps d'exécution en secondes (méthode NN)	WPE v0	DIRAC	WPE v2
Test 1	36930	47502	41900
Test 2	39298	46039	45824
Test 3	42043	47489	53569
Test 4	41807	44353	52173
Test 5	52156	39896	50184
Test 6	38956	40935	44278
Test 7	36733	36656	56371
Test 8	40138	41629	40110
Test 9	41286	51592	42138
Test 10	38740	43488	50129
Test 11	43006	50246	41050
Test 12	41043	43105	43272
Test 13	38841	60448	43970
Test 14	38992	43854	51025
Test 15	44022	42361	44473
Test 16	36049	44209	50244
Test 17	40129	41373	50124
Test 18	36594	62100	45329
Test 19	34014	41029	40873
Test 20	38732	50663	41203
Moyenne	39975	45948	46412
Ecart-type	3794	6478	4877

Tableau 5-10 - Comparaison des temps d'exécution de 20 tests effectués sur la grille avec un fichier d'entrée de 500.000 séquences

Temps d'exécution en secondes (méthode NN)	WPE v0	DIRAC	WPE v2
Test 1		57644	80210
Test 2		84418	75639
Test 3		48514	64481
Test 4		45469	68220
Test 5		72661	92303
Test 6		50526	65552
Test 7		71742	72899
Test 8		61994	78519
Test 9		63672	95672
Test 10		134580	101262
Test 11		51510	82144
Test 12		65883	63003
Test 13		71932	52670
Test 14		84940	63677
Test 15		64269	71349
Test 16		66234	81004
Test 17		62230	71292
Test 18		62173	57900
Test 19		58824	104092
Test 20		57489	55033
Moyenne		66835	74846
Ecart-type		19089	14741

Tableau 5-11 - Comparaison des temps d'exécution de 20 tests effectués sur la grille avec un fichier d'entrée de 1.000.000 séquences

5.3.2.3. Analyse des résultats

La lecture des différents tableaux fait apparaître de façon très consistante que les performances de déploiement avec DIRAC sont meilleures qu'avec WISDOM. Elle montre aussi que la version révisée de l'environnement de production WISDOM, dont le taux de succès est très significativement meilleur, a des performances comparables à la version de départ. La nouvelle version permet aussi l'extension de la durée des calculs possibles sur la grille.

5.4. Synthèse

Nous avons étudié dans ce chapitre le déploiement sur la grille du pipeline PANAM dédié à l'analyse de séquences environnementales issues de

séquenceurs de nouvelle génération. Le Tableau 5-12 - Temps moyens de PANAM sur la grille résume les résultats obtenus pour la méthode de parcours LCA, à savoir une réduction significative du temps de traitement. Cette réduction croît avec le nombre de séquences en entrée.

Nombre de séquences	Temps de calcul sur un PC local (s)	Temps de calcul sur la grille (s)
250000	42540	6140
500000	138300	10725
1000000	571428	20929

Tableau 5-12 - Temps moyens de PANAM sur la grille

La Figure 5-3 - Comparaison de la performance de ePANAM entre la grille et un PC compare les performances de PANAM en utilisant deux méthodes de parcours. Nous ne pouvons pas faire des tests de PANAM sur un PC avec la méthode NN parce qu'elle prend trop de temps de calcul. Si nous ne considérons que les versions sur le WPE, PANAM avec la méthode NN est plus lente que PANAM avec la méthode LCA (de 3 à 6 fois approximativement). Cependant, ces deux versions sont toujours nettement meilleures que la version sur une machine locale en utilisant la méthode LCA.

Le déploiement de PANAM sur la grille a donc deux impacts significatifs :

- réduire le temps de calcul
- ouvrir la possibilité de traiter plusieurs jeux de séquence en parallèle.

Il ouvre donc la voie au service web ePANAM.

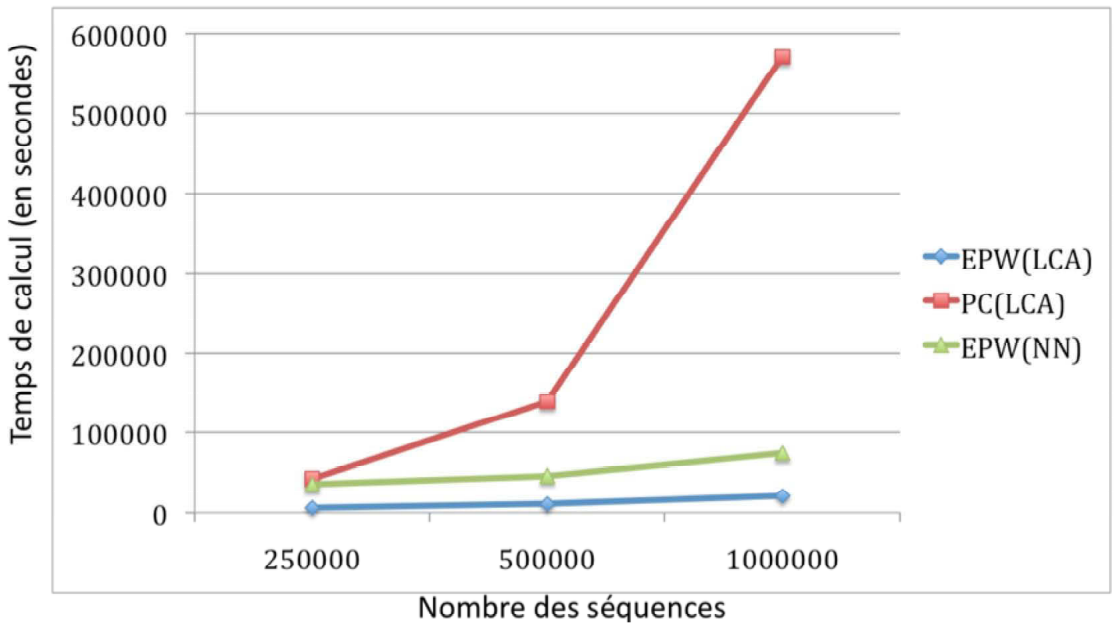


Figure 5-3 - Comparaison de la performance de ePANAM entre la grille et un PC

Une autre voie pour enrichir le service est d'implémenter un moteur de flot comme MOTEUR pour modifier certains paramètres du pipeline avant son exécution.

CHAPITRE 6 : CONCLUSIONS ET PERSPECTIVES

6.1. Conclusions

Les « grilles informatiques », infrastructures virtuelles constituées d'ordinateurs géographiquement éloignés fonctionnant en réseau pour fournir une puissance globale, constituent un environnement privilégié pour les collaborations scientifiques entre pays développés et pays en développement. Le travail décrit dans ce manuscrit s'est concentré sur l'utilisation de la grille en sciences du vivant dans deux nouveaux domaines scientifiques d'intérêt spécifique pour le Vietnam.

Le premier domaine est celui de la santé publique internationale. Le Vietnam a été l'un des premiers pays exposés au SRAS (Syndrome Respiratoire Aigu Sévère) et à la grippe H5N1. L'objectif était donc de donner aux virologues qui isolent les souches impliquées sur les nouveaux foyers des outils très performants permettant de diagnostiquer *in silico* par épidémiologie moléculaire l'émergence de nouveaux virus et de caractériser leur pathogénicité. Il s'agissait donc de proposer un ensemble d'outils d'analyse d'épidémiologie moléculaire intégrés dans une plate-forme permettant une prise de décision rapide sur les nouvelles souches isolées et séquencées.

Le deuxième domaine est celui de la métagénomique, pour l'exploration de la biodiversité au Vietnam. La biodiversité est à la base de notre alimentation et devient une source majeure d'innovation médicale. L'Institut de Chimie des Produits Naturels de l'Académie des Sciences et Technologies du Vietnam a ainsi un programme pour isoler des nouveaux composés chimiques issus de la biodiversité. Les progrès dans la technologie de séquençage ont ouvert la voie au développement d'approches dites de métagénomique pour caractériser et explorer au niveau moléculaire les microorganismes présents dans les différents écosystèmes. Les séquenceurs de nouvelle génération sont en passe de produire des volumes de données comparables à ceux de la physique des particules à l'échelle mondiale.

L'utilisation de la grille pour la métagénomique constitue donc une voie prometteuse pour l'analyse de la biodiversité au niveau mondial. La grille est aussi prometteuse pour la surveillance et la gestion des pandémies car elle rend possible des approches globales et systématiques grâce aux ressources qu'elle peut rendre disponible à quiconque aux 4 coins du monde.

Malheureusement, la grille n'a pas toujours tenu ses promesses et l'obstacle principal à son adoption par les communautés scientifiques en sciences du vivant a été la difficulté de son utilisation par des personnes peu expérimentées en informatique.

Les objectifs de notre travail étaient donc d'explorer la pertinence de la grille pour répondre aux besoins de nouvelles communautés scientifiques, de concevoir et/ou d'offrir des services faciles et agréables d'utilisation notamment par des chercheurs actifs dans un pays en développement.

Pour la surveillance des maladies émergentes, nous avons essayé de proposer un ensemble d'outils d'analyse d'épidémiologie moléculaire permettant une prise de décision rapide sur les nouvelles souches isolées et séquencées.

Pour la métagénomique, nous avons étudié comment l'utilisation de la grille pouvait accélérer et délocaliser le traitement de plusieurs centaines de milliers de séquences courtes dans un pipeline d'analyse complexe développé par une équipe de chercheurs spécialisés dans l'étude de l'écologie microbienne.

Nous nous sommes appuyés pour cela sur le catalogue de métadonnées AMGA, sur l'outil de flot MOTEUR ainsi que sur les environnements de production DIRAC et WISDOM aujourd'hui disponibles sur la grille EGI. Nous avons privilégié l'utilisation de l'environnement de production WISDOM (WISDOM Production Environment) conçu en 2005 et développé depuis pour la recherche *in silico* de nouveaux médicaments. Malgré plusieurs améliorations significatives que nous lui avons apportées, cet environnement s'est avéré moins performant que la plate-forme DIRAC pour le type d'applications étudiées dans cette thèse.

Avec ces outils, nous avons exploré pendant les deux premières années de la thèse l'utilisation de la grille pour la surveillance des maladies émergentes. Ce travail a débouché sur le développement de la plate-forme g-INFO. Mais l'utilisation de la plate-forme g-INFO au Vietnam s'est heurtée à une difficulté d'ordre technique : bien que l'accès à l'internet commercial se généralise au Vietnam, le réseau dédié à la recherche et l'éducation n'est pas encore pleinement opérationnel, loin s'en faut. Seuls quelques laboratoires dans les plus grandes universités et centres de recherche y sont raccordés et bien souvent, les autres laboratoires sur les campus ne peuvent en bénéficier du fait de l'absence d'un réseau local performant. Or, ce réseau pour l'éducation et la recherche est la clef de l'installation et du déploiement de sites de grille au Vietnam ainsi que de l'accès aux services centraux de grille en Europe à travers le réseau TEIN2 financé par la Commission Européenne. En pratique, le réseau VINAREN a été coupé pendant plus d'un an ce qui a rendu particulièrement difficile la prise en mains de la plate-forme g-INFO par ses utilisateurs potentiels et donc son adaptation aux besoins spécifiques des acteurs vietnamiens de la surveillance

Au-delà du problème technique localisé au Vietnam, s'ajoute un problème humain beaucoup plus complexe à gérer. Dans le contexte général de la santé publique internationale, il existe une certaine réticence, voire une méfiance, à l'utilisation de solutions informatiques intégrées. Le recours à des approches éprouvées et ayant déjà fait leurs preuves pour gérer les crises précédentes est systématiquement privilégié. Ainsi l'initiative de Global Public Health Grid n'a pas vraiment connu la dynamique que l'on pouvait espérer depuis 2009. Le

jour où les centres régionaux de l'OMS hébergeront des nœuds d'une grille mondiale de santé publique semble encore très éloigné.

Malgré tout, la plate-forme g-INFO est aujourd'hui opérationnelle sur l'organisation virtuelle Biomed d'EGI. Elle expose les algorithmes sous forme de services web chainés grâce au moteur de flot MOTEUR développé par les laboratoires CREATIS et I3S.

Au cours de la troisième année de thèse, nous nous sommes appuyés sur les développements logiciels réalisés pour g-INFO pour le déploiement sur la grille du pipeline PANAM de traitements de données de pyroséquenceurs en utilisant les environnements de production DIRAC et WISDOM. Les résultats obtenus montrent que la grille permet d'une part de distribuer efficacement la charge de calcul et d'autre part d'accélérer significativement le traitement des séquences. Cette accélération croît avec la taille des séquences. Ces résultats ouvrent des perspectives très encourageantes pour le déploiement d'un service de métagénomique ePANAM adossé à la grille de calcul.

6.2. Perspectives

6.2.1. Perspectives d'amélioration de l'Environnement de Production WISDOM

Au-delà des améliorations déjà apportées pendant la thèse, l'une des faiblesses de l'Environnement de Production WISDOM est le manque d'outils de surveillance visuelle. Observer l'état des jobs et des tâches est nécessaire pour l'administrateur et aussi que pour les utilisateurs. En principe, l'information sur l'état des jobs et des tâches est enregistrée de façon continue dans le WIS, et les commandes de AMGA permettent de les lire, mais cela n'est pas facile pour tous les utilisateurs. Une interface intuitive sur le Web comme la solution déployée par DIRAC améliorerait la convivialité de l'outil.

Dans la nouvelle version de WPE, le taux de succès des tâches s'est considérablement amélioré mais il reste inférieur à celui de DIRAC. Conçu à l'origine pour le déploiement d'applications de criblage virtuel qui nécessitent l'exécution simultanée d'un grand nombre de jobs courts, le WPE doit être encore amélioré pour un fonctionnement quotidien et la gestion de problèmes bioinformatiques divers.

Une autre piste d'amélioration est la sécurité : utiliser un certificat administrateur ou robot pour soumettre des jobs simplifie la gestion centralisée mais il serait préférable d'avoir un mécanisme pour soumettre des jobs avec les certificats des utilisateurs et éviter l'utilisation d'un certificat au nom de tous les utilisateurs.

Dans le contexte général d'évolution des ressources informatiques vers le *cloud computing*, la migration du WPE mais aussi de DIRAC sur le cloud pourrait faciliter son déploiement et son utilisation au Vietnam. Les difficultés rencontrées dans le déploiement de la grille au Vietnam ont été mentionnées et elles ne peuvent pas être résolues rapidement dans un proche avenir. Par conséquent, le *cloud computing* est une bonne solution dans la situation

actuelle.

6.2.2. Perspectives d'utilisation en épidémiologie moléculaire

Tel qu'il est aujourd'hui, le portail g-INFO peut aider les experts à créer des flots bioinformatiques dynamiques et à les exécuter pour surveiller l'évolution des virus Influenza A et l'apparition de nouvelles souches. Plusieurs perspectives existent pour améliorer le portail :

- Le flot phylogénétique de traitement des données n'est certainement qu'un point de départ. Le portail est conçu pour permettre une intégration facile de nouveaux algorithmes, leur chainage et leur exécution sur la grille grâce à MOTEUR.
- Le portail peut être enrichi pour importer des séquences issues de plusieurs bases de données
- Le cadre de sécurité doit être amélioré pour permettre aux propriétaires des données de mieux protéger la confidentialité des informations
- De nouveaux services peuvent être ajoutés pour enrichir l'offre aux utilisateurs :
    - Traitement phylogéographique des données pour reconstituer l'évolution de la distribution géographique des souches virales
    - Criblage virtuel à partir des séquences ou des structures tridimensionnelles des souches pour caractériser *in silico* l'effet des mutations sur l'activité des médicaments actuels, sur le modèle des études faites en 2007 sur l'impact de certaines mutations du virus H5N1

6.2.3. Perspectives en métagénomique

Le premier portage de PANAM sur la grille décrit dans ce manuscrit a montré l'efficacité de la grille pour distribuer une chaîne complexe de traitements de données métagénomiques de pyroséquençage. Dans le cadre de la thèse de Najwa Taib, PANAM ne cesse d'évoluer et une nouvelle version traitant aussi les bactéries et les archaea est en cours de développement. Le portage de cette nouvelle version sur la grille est important dans la perspective d'offrir le service web ePANAM à la communauté scientifique. Sur le modèle de g-INFO, il pourrait être intéressant d'implémenter un moteur de flot sur ePANAM apportant une flexibilité dans le choix et la définition des chaines de traitement.

Au-delà du cas spécifique de PANAM, les technologies de séquençage à haut debit font entrer la génomique dans l'ère du Big Data et la grille constitue clairement un élèment de réponse à l'avalanche de données qu'elles vont générer.

6.2.4. Perspectives pour la grille et le cloud au Vietnam

Comme nous l'avons indiqué précédemment, la migration technologique de la grille vers le cloud devrait faciliter le déploiement d'une offre de ressources académiques au Vietnam. Grâce aussi à l'amélioration des réseaux pour la

recherche et l'éducation, l'ensemble des travaux présentés dans ce manuscrit devrait trouver un environnement de plus en plus favorable à leur déploiement.

ANNEXE I – LISTE DE PUBLICATIONS

Conférences internationales (avec comité de lecture)

Trung-Tung DOAN, Aurélien BERNARD, Ana Lucia DA-COSTA, Vincent BLOCH, Thanh-Hoa LE, Yannick LEGRE, Lydia MAIGNE, Jean SALZEMANN, David SARRAMIA, Hong-Quang NGUYEN, Vincent BRETON. *Grid-based International Network for Flu Observation (g-INFO)*. Studies in Health Technology and Informatics, vol. 159, pp. 215-226, 2010.

Trung-Tung DOAN, Hong-Quang NGUYEN, Ana Lucia DA-COSTA, Yannick LEGRE, Aurélien BERNARD, Lydia Maigne, Jean SALZEMANN, David Sarramia, Vincent BRETON, Thanh-Hoa LE, Duc-Hung LE, Johan MONTAGNAT. *A new flexible workflow on the Grid for monitoring H5N1*. Proceedings of the International Symposium on Grids and Clouds, in conjunction with Open Grid Forum 31, March 19 - 25, 2011

Trung-Tung DOAN, Quang-Minh DAO, Trong-Hieu VU, Hong-Phong PHAM, Vincent BRETON, Hong-Quang NGUYEN, Jean SALZEMANN, Yannick LEGRE, Thanh-Hoa LE, Johan MONTAGNAT. *g-INFO portal: a solution to monitor Influenza A on the Grid for non-grid users*., Proceedings of HealthGrid conference 2011, in conjunction with IEEE CBMS2011, to be published in Studies in Health Technology and Informatics

T.Q. BUI, T.T. DOAN, H.T. NGUYEN, Q.M. PHAM, S.V. NGUYEN, V. BRETON, L.Q. PHAM, H.M. Le, H.Q. NGUYEN and E. MEDERNACH, On the Performance Enhancement of WISDOM Production Environment, Proceedings of International Symposium on Grids and Clouds, PoS(ISGC 2012)001, http://pos.sissa.it/cgi-bin/reader/conf.cgi?confid=153

Conférences nationales (avec comité de lecture)

T. Tung DOAN, Najwa TAIB, H.Quang NGUYEN, Jean-Christophe CHARVY, Gisele BRONNER, Vincent BRETON, Didier DEBROAS, Engelbert MEPHU. *Portage de ePANAM sur la grille de calcul.* Rencontres Scientifiques France Grilles. 19 September 2011.

T. Tung DOAN, T. Hieu VU, D. Hung LE, Q. Minh DAO, H. Quang NGUYEN, Vincent BRETON, Yannick LEGRE. *Le portail g-INFO pour surveiller la grippe Influenza A*. Rencontres Scientifiques France Grilles. 19 September 2011.

Conférences nationales

Trung-Tung DOAN. *Towards a global surveillance network on avian influenza using grid technology.* Doctoral presentation at IEEE-RIVF 2009 conference.

Trung-Tung Doan, Ana Lucia DA COSTA, Aurélien BERNARD, Thanh-Hoa LE, Hong-Quang NGUYEN, Yanick LEGRE, Vincent BRETON. *Grid-based International Network for Flu Observation.* Oral presentation at EGEE User Forum 2010. 12-16 April 2010.

Trung-Tung DOAN, Quang-Minh DAO, Ana Lucia DA-COSTA, Trong-Hieu VU, Thanh-Hoa LE, Yannick LEGRE, Lydia MAIGNE, Jean SALZEMANN, Hong-Quang NGUYEN, Vincent BRETON. *g-INFO portal for monitoring Influenza A on the Grid.* Demonstration at EGI User Forum 2011. 11-14 April 2011.

REFERENCES

[1]. I. Foster, What is the Grid? A Three Point Checklist, July 20, 2002. Available: http://www.mcs.anl.gov/~itf/Articles/ WhatIsTheGrid.pdf.

[2]. M. Rabb and C. Doninger, Grid computing and SAS, SAS Institute Inc.US, 2004, Available: http://support.sas.com/rnd/ scalability/papers/101948_1204.pdf.

[3]. J. H. Kaufman, T. J. Lehman, and J. Thomas, "Grid computing made simple", The Industrial Physicist, pp 32- 33, Aug- Sept 2003.

[4]. R. Gupta, Bio-Informatics, Bonding genes with IT, in INDIACOM 2010, Jaipur, Feb 2010.

[5]. E. Laure, S. M. Fisher, A. Frohner, C. Grandi, P. Kunszt, A. Krenek, O. Mulmo, F. Pacini, F. Prelz, J. White, M. Barroso, P. Buncic, F. Hemmer, A. Di Meglio, A. Edlund, Programming the Grid with gLite, Computational Methods in Science and Technology 12 (1) 33-45. 2006.

[6]. M. Romberg, The UNICORE Grid Infrastructure, Special Issue on Grid Computing of Scientifc Programming Journal, 10, pages 149 -157. 2002.

[7]. I. Foster, Globus Toolkit Version 4: Software for Service-Oriented Systems, IFIP International Conference on Network and Parallel Computing, Springer-Verlag LNCS 3779, pp 2-13, 2006.

[8]. Ellert,M. et al., Advanced resource connector middleware for lightweight computational grids, Future Generation Computer Systems, 23, 219–240. 2007.

[9]. Raphael Bolze, Analyse et déploiement de solutions algorithmes et logicielles pour des applications bioinformatiques à grande échelle sur la grille, Thèse de l'Ecole Normale Supérieuse de Lyon, 2008.

[10]. Venkat, Jikku, Grid Computing in the Enterprise with the UD MetaProcessor, P2P '02 Second International Conference on Peer-to-Peer Computing. pp. 4. 2002.

[11]. P. d'Anfray, R. Bolze, and F. Desprez. Le programme Décrypthon. Journée Réseaux JRES, Strasbourg, 2007.

[12]. F. Hupfeld, T. Cortes, B. Kolbeck, E. Focht, M. Hess, J. Malo, J. Marti, J. Stender, E. Cesario, XtreemFS - a case for object-based file systems in Grids, Concurrency and Computation: Practice and Experience. Volume 20 Issue 8 June 2008.

[13]. J.D. Thompson, D.G. Higgins, and T.J. Gibson, ClustalW: improving the sensitivity of progressive multiple sequence alignment through sequence weighting, positions-specific gap penalties and weight matrix choice, Nucleic Acids Research 22, vol. 22, no. 22, pp. 4673–4680, 1994.

[14]. Lipman DJ, Pearson WR, Rapid and sensitive protein similarity searches, Science 227 (4693): 1435–41. 1985.

[15]. Eddy SR., Profile hidden Markov models, Bioinformatics 14: 755- 63. 1998.

[16]. Guindon S., Dufayard J.F., Lefort V., Anisimova M., Hordijk W., Gascuel O., New Algorithms and Methods to Estimate Maximum-Likelihood Phylogenies: Assessing the Performance of PhyML 3.0, Systematic Biology, 59(3):307-21, 2010.

[17]. Tiphaine Martin, Florence Maurier, Alexis Groppi, Aurelien Barre, Delphine Naquin, Aurelien Roult, Clement Gauthey, Christophe Blanchet, Utilisation d'une Grille de Calcul (GRISBI) pour le Traitement de Données NGS (Next Generation Sequencing), Rencontres Scientifiques France Grilles. 2011.

[18]. P. Bala, J. Pytlinski, M. Nazaruk, V. Alessandrini, D. Girou, D. Erwin, D. Mallmann, J. MacLaren, J. Brooke, and J.-F. Myklebust, BioGRID - An European Grid for Molecular Biology, Proceedings of 11th International Symposium on High Performance Distributed Computing (HPDC), Edinburgh, Scotland, IEEE Computer Society Press, page 412. 2002.

[19]. W. Van Gunsteren and H. J. C. Berendsen. GROMOS (Groningen Molecular Simulation Computer Program Package). Biomos, Laboratory of Physical Chemistry, ETH Zentrum, Zurich, 1996.

[20]. D.A. Case, T.E. Cheatham, III, T. Darden, H. Gohlke, R. Luo, K.M. Merz, Jr., A. Onufriev, C. Simmerling, B. Wang and R. Woods. The Amber biomolecular simulation programs. J. Computat. Chem. 26, 1668-1688. 2005.

[21]. B. R. Brooks, R. E. Bruccoleri, B. D. Olafson, D. J. States, S. Swaminathan, and M. Karplus. A program for macromolecular energy, minimization, and dynamics calculations. J. Comp. Chem., 4:187–217, 1983.

[22]. Altschul, Stephen F., Thomas L. Madden, Alejandro A. Schaffer, Jinghui Zhang, Zheng Zhang, Webb Miller, and David J. Lipman (1997), Gapped BLAST and PSI-BLAST: a new generation of protein database search programs, Nucleic Acids Res. 25:3389-3402.

[23]. Houstis, E. N.; Rice, J. R., Computer as thinker/doer: Problem solving environments for computational science, IEEE Computational Science & Engineering 1 (2): 11–23. 1994.

[24]. Grillo, G., Licciulli, F., Liuni S., Sbisà, E. & Pesole G., PatSearch: a program for the detection of patterns and structural motifs in nucleotide sequences, Nucleic Acids Res., 31, 3608–3612. 2003.

[25]. Gisel A, Panetta M, Grillo G, Licciulli VF, Liuni S, Saccone C, Pesole G., DNAfan: a software tool for automated extraction and analysis of user-defined sequence regions, Bioinformatics 20(18):3676-9. 2004.

[26]. Vassura, M., Margara, L., Di Lena, P., Medri, F., Fariselli, P. & Casadio, R., Reconstruction of 3D Structures From Protein Contact Maps, Computational Biology and Bioinformatics, 5(3)357-367. 2008.

[27]. Blanchet C, Lefort V, Combet C, Deléage G., GPS@ bioinformatics portal: from network to EGEE grid, Stud Health Technol Inform. 120:187-93. 2006.

[28]. Combet C, Blanchet C, Geourjon C, Deléage G., NPS@: network protein sequence analysis, Trends Biochem Sci. 25(3):147-50. 2000.

[29]. T. F. Smith, M. S. Waterman, Identification of common molecular subsequences, J Mol Biol, 147, 195-197. 1981

[30]. Bairoch, A. and Apweiler, R., The SWISS-PROT protein sequence database and its supplement TrEMBL in 2000, Nucleic Acids Res., Vol. 28, pp. 45-48. 2000.

[31]. Sigrist CJA, Cerutti L, de Castro E, Langendijk-Genevaux PS, Bulliard V, Bairoch A, Hulo N., PROSITE a protein domain database for functional characterization and annotation, Nucleic Acids Research vol. 38 Database issue pp. 161-166. 2010.

[32]. Ian Taylor, Ian Wang, Matthew Shields, and Shalil Majithia, Distributed computing with Triana on the Grid, In Concurrency and Computation:Practice and Experience , 17(1-18), 2005.

[33]. D. Hull, K. Wolstencroft, R. Stevens, C. Goble, M. Pocock, P. Li, and T. Oinn, Taverna: a tool for building and running workflows of services, Nucleic Acids Research, vol. 34, Web Server issue, pp. 729-732, 2006.

[34]. Jay Alameda et al., The Open Grid Computing Environments Collaboration: Portlets and Services for Science Gateways, Concurrency and Computation: Practice and Experience vol 19 issue 6 pp. 921-942. 2006.

[35]. P.A. Knoop, J. Hardin, T.Killen and D. Middleton, The CompreHensive collaborativE Framework (CHEF), AGU Fall Meeting Abstracts 2002.

[36]. I. Foster et al., A Security Architecture for Computa-tional Grids, Proc. ACM Conf. Computers and Security, ACM Press, New York, pp. 83-91. 1998.

[37]. K. Czajkowski, Globus GRAM, Globus Toolkit Developer's Forum. Globus Alliance. 2006.

[38]. Taylor, Ian J. From P2P to Web Services and Grids - Peers in a Client/Server World. Springer, 2005.

[39]. J. Novotny, S. Tuecke and V. Welch, An Online Credential Repository for the Grid: MyProxy, Tenth International Symposium on High Performance Distributed Computing (HPDC- 10), 2001.

[40]. WeiLong Ueng, Introduction for Grid Application Platform, Online : http://www.euasiagrid.org/training/EAGSS28_GAP_introduction.pdf

[41]. Hsin-Yen Chen, GVSS : A new virtual screening service against the epidemic disease in Asia, ISGC Conference. 2010.

[42]. E. Caron. Contribution to the management of large scale platforms: the Diet experience , HDR thesis, ENS Lyon, 2010

[43]. The DIET Team. Diet : Distributed interactive engineering toolbox. http://graal.ens-lyon.fr/DIET.

[44]. K. Seymour, C. Lee, F. Desprez, H. Nakada, and Y. Tanaka. The end-user and middleware apis for gridrpc. In Workshop on Grid Application Programming Interfaces, In conjunction with GGF12, Brussels, Belgium, September 2004.

[45]. The Grid'5000 Team. Grid'5000 web page. http ://www.grid5000.org/.

[46]. E. Caron, A. Chis, F. Desprez, and A. Su. Design of plug-in schedulers for a gridrpc environment. Future Generation Computer Systems, 24(1) :46–57, January 2008.

[47]. Tsaregorodtsev, A. Garonne, V., Stokes-Rees, I., DIRAC: a scalable lightweight architecture for high throughput computing, Fifth IEEE/ACM International Workshop on Grid Computing, 2004.

[48]. T.Q. Bui, T.T. Doan, H.T. Nguyen, Q.M. Pham, S.V. Nguyen, V. Breton, L.Q. Pham, H.M. Le, H.Q. Nguyen and E. Medernach, On the Performance Enhancement of WISDOM Production Environment, Proceedings of International Symposium on Grids and Clouds, PoS(ISGC 2012)001, http://pos.sissa.it/cgi-bin/reader/conf.cgi?confid=153

[49]. S. Sild, U. Maran, M. Romberg, B. Schuller, and E. Benfenati, OpenMolGRID: Using Automated Workflows in GRID Computing Environment, Proceedings of European Grid Conference, Amsterdam, The Netherlands, Springer, LNCS 3470, pages 464-473. 2005.

[50]. H. Gjermundrød, M. D. Dikaiakos, M. Stümpert, P. Wolniewicz, and H. Kornmayer: g-Eclipse - An Integrated Framework to Access and Maintain Grid Resources, The 9th IEEE/ACM International Conference on Grid Computing (Grid 2008), Tsukuba, Japan, September 29 - October 1, 2008.

[51]. K. Harrison, C.L. Tan, D. Liko, A. Maier, J. Mościcki, U. Egede, R.W.L. Jones, A. Soroko, G.N. Patrick, Ganga: a Grid user interface for distributed analysis, Proc. Fifth UK e-Science All-Hands Meeting, Nottingham, UK, 18th-21st September, Ed. Simon J. Cox (National e-Science Centre, 2006) pages 518-525. 2006.

[52]. Sylvain Reynaud, Uniform Access to Heterogeneous Grid Infrastructures with JSAGA, Production Grids in Asia, Volume . ISBN 978-1-4419-0102-6. Springer-Verlag US, p. 185-196. 2010.

[53]. Vladimir V. Korkhov, Jakub T. Moscicki, Valeria V.

Krzhizhanovskaya, Dynamic workload balancing of parallel applications with user-level scheduling on the Grid, Future Generation Computer Systems Volume 25, Issue 1, p. 28-34. 2009.

[54]. J.T. Moscicki, M.G.Pia, A.Mantero, S.Guatelli: Distributed Geant4 Simulation in Medical and Space Science Applications using DIANE framework and the GRID, Innovative Particle and Radiation Detec- tors, Nuclear Physics B (Proc. Suppl.) 123 (2003).

[55]. Stone, James M.; Gardiner, Thomas A.; Teuben, Peter; Hawley, John F.; Simon, Jacob B., Athena: A New Code for Astrophysical MHD, The Astrophysical Journal Supplement Series, Volume 178, Issue 1, pp. 137-177. 2008.

[56]. Morris, G. M., Huey, R., Lindstrom, W., Sanner, M. F., Belew, R. K., Goodsell, D. S. and Olson, A. J., Autodock4 and AutoDockTools4: automated docking with selective receptor flexiblity, J. Computational Chemistry 2009, 16: 2785-91. 2009.

[57]. S. Agostinelli et al., Geant4: A Simulation toolkit , Nucl. Instrum. Meth. A506 (3): 250-303, 2003.

[58]. V. Breton, K. Dean and T. Solomonides, editors on behalf of the Healthgrid White Paper collaboration,"The Healthgrid White Paper", Proceedings of Healthgrid conference, IOS Press, Vol 112, 2005

[59]. Sinnott, Richard O.; Stell, Anthony J., Towards a Virtual Research Environment for International Adrenal Cancer Research, Proceedings of The International Conference on Computational Science (ICCS) , 2011, Volume: 4 Pages: 1109-1118

[60]. J. Gonzales, S. Pomel, V. Breton, B. Clot, J.L. Gutknecht, B. Irrthum & Y. Legré, Empowering humanitarian medical development using grid technology, *Methods of Information in Medicine* 2005; 44: 186-189, ISSN 0026-1270, May 2005.

[61]. Segrelles, Damia; Blanquer, Ignacio; Salavert, Jose; et al., Exchanging Data for Breast Cancer Diagnosis on Heterogeneous Grid Platforms, Computing and Informatics (2012) Volume: 31 Issue: 1 Pages: 3-15

[62]. A Review of Grid Authentication and Authorization Technologies and Support for Federated Access Control, Jie, Wei; Arshad, Junaid; Sinnott, Richard; et al. ACM COMPUTING SURVEYS Volume: 43 Issue: 2 Article Number: 12 DOI: 10.1145/1883612.1883619

[63]. M. Mirza et al, Global Public Health Grid - WHO-CDC Public Health Informatics Initiative: Value Proposition and Pilot Projects , proceedings of PHIN2009 conference, https://cdc.confex.com/cdc/phin2009/webprogram/Paper21091.html

[64]. T. Meinnel et al, Conclusions du groupe de travail e-Infrastructures pour la Génomique et la Biologie à grande échelle, 2012.

[65]. T. Martin et al, Utilisation d'une Grille de Calcul (GRISBI) pour le Traitement de Données NGS (New Generation Sequencing), Proceedings of Rencontres Scientifiques France Grilles, http://france-grilles-2011.sciencesconf.org/1132.

[66]. Maltsev N, Glass E, Sulakhe D, Rodriguez A, Syed MH, Bompada T, Zhang Y, D'Souza M. PUMA2--grid-based high-throughput analysis of genomes and metabolic pathways. Nucleic Acids Res. 2006 Jan 1;34(Database issue):D369-72.

[67]. Kanehisa,M., Goto,S., Hattori,M., Aoki-Kinoshita,K.F., Itoh,M., Kawashima,S., Katayama,T., Araki,M. and Hirakawa,M, From genomics to chemical genomics: new developments in KEGG, Nucleic Acids Res., 34, D354–D357. 2006.

[68]. Caspi R, Foerster H, Fulcher CA, Hopkinson R, Ingraham J, Kaipa P, Krummenacker M, Paley S, Pick J, Rhee SY, Tissier C, Zhang P, Karp PD: MetaCyc: a multiorganism database of metabolic path- ways and enzymes. Nucleic Acids Res 2006, 34:D511-D516.

[69]. Overbeek,R., Larsen,N., Pusch,G.D., D'Souza,M., Selkov,E.Jr, Kyrpides,N., Fonstein,M., Maltsev,N. and Selkov,E. (2000) WIT: integrated system for high-throughput genome sequence analysis and metabolic reconstruction. Nucleic Acids Res., 28, 123–125.

[70]. Sulakhe D, Rodriguez A, D'Souza M, Wilde M, Nefedova V, Foster I, Maltsev N. GNARE: automated system for high-throughput genome analysis with grid computational backend. J Clin Monit Comput. 2005 Oct;19(4-5):361-9.

[71]. Christopher Hyland, John W. Pinney, Glenn A. McConkey, and David R. Westhead, metaSHARK: a WWW platform for interactive exploration of metabolic networks, Nucleic Acids Res. 2006 July 1; 34(Web Server issue): W725–W728. 2006.

[72]. Pinney,J.W., Shirley,M.W., McConkey,G.A. and Westhead,D.R. metaSHARK: software for automated metabolic network prediction from DNA sequence and its application to the genomes of Plasmodium falciparum and Eimeria tenella. Nucleic Acids Res., 33, 1399–1409. 2005.

[73]. Douglas Thain, Todd Tannenbaum, and Miron Livny, "Distributed Computing in Practice: The Condor Experience" Concurrency and Computation: Practice and Experience, Vol. 17, No. 2-4, pages 323-356, February-April, 2005.

[74]. Roux, Simon; Faubladier, Michael; Mahul, Antoine; et al., Metavir: a web server dedicated to virome analysis, BIOINFORMATICS (2011) Volume: 27 Issue: 21 Pages: 3074-3075.

[75]. L-M. Birkholtz, O. Bastien, G. Wells, D. Grando, F. Joubert, V. Kasam, M. Zimmermann, P. Ortet, N. Jacq, N. Saidani, S. Hofmann-Apitius, M. Hofmann-Apitius, V. Breton, A.I Louw and E. Marechal, Integration and

mining of malaria molecular, functional and pharmacological data: how far are we from a chemogenomic knowledge space?, Malaria Journal, Vol. 5, (2006) 110.

[76]. V. Kasam, A. Maaß, H. Schwichtenberg, M. Zimmermann, A. Wolf, N. Jacq, V. Breton, M. Hofmann, Design of Plasmepsine Inhibitors : A virtual high throughput screening approach on the EGEE Grid, J. Chem. Inf. Model., 47 (5), 1818 -1828, 2007

[77]. Jacq N., Breton V., Chen H.-Y., Ho L.-Y., Hofmann M., Kasam V., 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., Wu Y.-T., Zimmermann M., Virtual Screening on Large Scale Grids, Parallel Computing Volume 33, Issue 4-5 (2007) 289-301.

[78]. Sebastien Cipière, Paul De Vlieger, David Sarramia, David R. C. Hill, Lydia Maigne, Development of a metamodel for medical database management on a grid network, CCGRID '12 Proceedings of the 2012 12th IEEE/ACM International Symposium on Cluster, Cloud and Grid Computing, Pages 892-897. 2012.

[79]. Vinod kasam, Jean Salzemann, Marli Botha, Ana Dacosta, Gianluca Degliesposti, Raul Isea, Doman Kim, Astrid Maass, Colin Kenyon, Giulio Rastelli, Martin Hofmann-Apitius, Vincent Breton, WISDOM-II: Screening against multiple targets implicated in malaria using computational grid infrastructures, Malaria Journal 2009, 8:88 (1 May 2009).

[80]. H-C Lee, J. Salzemann, N. Jacq, H-Y Chen, L-Y Ho, I. Merelli, L. Milanesi, V. Breton, S.C. Lin and Y-T. Wu, Grid enabled high throughput in silico screening against Influenza A neuraminidase, IEEE transaction on nanobioscience, Vol.5, n°4 (2006) 288-295

[81]. T. T. H. Nguyen, H-J Ryu, S-H Lee, S. Hwang, V. Breton, J-H Rhee and D. Kim, Virtual screening identification of novel severe acute respiratory syndrome 3C-like protease inhibitors and in vitro confirmation, Bioorganic & Medicinal Chemistry Letters, Volume 21, Issue 10 (2011) 3088-3091

[82]. Thi Thanh Hanh Nguyen, Hwa-Ja Ryu, Se-Hoon Lee, Soonwook Hwang, Jaeho Cha, Vincent Breton et Doman Kim, Discovery of novel inhibitors 1 for human intestinal maltase: virtual screening in a WISDOM environment and in vitro évaluation, Biotechnology Letters 2011, Volume 33, 2185-2191

[83]. N. Jacq, J. Salzemann, Y. Legré, M. Reichstadt, F. Jacq, E. Medernach, M. Zimmermann, A. Maaß, M. Sridhar, K. Vinod-Kusam, J. Montagnat, H. Schwichtenberg, M. Hofmann, V. Breton, Grid enabled virtual screening against malaria, Journal of Grid Computing Vol 6 n°1, 29-43, 2008

[84]. Berman, H. M., Henrick, K. & Nakamura, H. (2003). Nat. Struct. Biol. 10, 980.

[85]. R.P. Joosten, J. Salzemann, V. Bloch, H. Stockinger, A-C Berglund, C.

Blanchet, E. Bongcam-Rudloff, C. Combet, A. Da Costa, G. Deleage, M. Diarena, R. Fabbretti, G. Fettahi, V. Flegel, A. Gisel, V. Kasam, T. Kervinen, E. Korpelainen, K. Mattila, M. Pagni, M. Reichstadt, V. Breton, I. Tickle and G. Vriend, PDB_REDO : automated re-refinement of X-Ray structure models in the PDB, Journal of Applied Cristallography, (2009) 42, 1-9.

[86]. Gagliardi, F., Jones, B., Grey, F., Bégin, M.E., Heikkurinen, M.: 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).

[87]. The LCG Editorial Board: LHC Computing Grid Technical Design Report, CERN-LHCC-2005-024, (2005).

[88]. S. Ahn, et. al., Performance analysis and optimization of AMGA for the large-scale virtual screening, Software Practice & Experience Volume 39 Issue 12, pp. 1055-1072. 2009.

[89]. S. Kumar, A. Skjaeveland, R. Orr, P. Enger, T. Ruden, B. H. Mevik, F. Burki, A. Botnen, and K. S. Tabrizi, Air: A batch-oriented web program package for construction of supermatrices ready for phylogenomic analyses, BMC Bioinformatics, vol. 10, no. 1, pp. 357+, October 2009.

[90]. K. Hanekamp, U. Bohnebeck, B. Beszteri, and K. Valentin, Phylogena-a user-friendly system for automated phylogenetic annotation of unknown sequences, Bioinformatics (Oxford, England), vol. 23, no. 7, pp. 793-801, April 2007.

[91]. Phylm-mpi, [Online], Available: http://atgc.lirmm.fr/phyml/versions.html.

[92]. A. Darling, L. Carey, and W. Feng, The Design, Implementation, and Evaluation of mpiBLAST, 4th International Conference on Linux Clusters: The HPC Revolution 2003 in conjunction with ClusterWorld Conference & Expo, June 2003.

[93]. T. Glatard, J. Montagnat, D. Lingrand, X. Pennec. “Flexible and efficient workflow deployement of data-intensive applications on grids with MOTEUR”, International Journal of High Performance Computing Applications (IJHPCA), 22 (3), pages 347-360, 2008.

[94]. Edgar, Robert C. (2004), MUSCLE: multiple sequence alignment with high accuracy and high throughput, Nucleic Acids Research 32(5), 1792-97.

[95]. Castresana, J. Selection of conserved blocks from multiple alignments for their use in phylogenetic analysis, Molecular Biology and Evolution 17 (2000), 540-552.

[96]. Gregory E. Jordan, William H. Piel. PhyloWidget: web-based visualizations for the tree of life. Bioinformatics (Oxford, England), Vol. 24, No. 14. (15 July 2008), pp. 1641-1642.

[97]. Han M.V. and Zmasek C.M. (2009). phyloXML: XML for evolutionary

biology and comparative genomics. BMC Bioinformatics, 10:356.

[98]. Peterson, M.W. and M.E. Colosimo, TreeViewJ: an application for viewing and analyzing phylogenetic trees. Source Code Biol Med, 2007. 2(1): p. 7.

[99]. J. Dutheil and N. Galtier, BAOBAB: a Java editor for large phylogenetic trees, Bioinformatics (2002) 18(6): 892-893

[100]. Chen, K. and L. Pachter, Bioinformatics for whole-genome shotgun sequencing of microbial communities, PLoS Computa- tional Biology 1: e24. doi:10.1371/journal.pcbi.0010024. 2005.

[101]. Handelsman, J., M.R. Rondon, S.F. Brady, J. Clardy, and R.M. Goodman, Molecular biological access to the chemistry of unknown soil microbes: a new frontier for natural products, Chemistry & Biology 5: R245–R249. doi:10.1016/S1074-5521 (98)90108-9. 1998.

[102]. Margulies et al., Genome sequencing in microfabricated high-density picolitre reactors, Nature 437: 376–380. 2005

[103]. Pernthaler, A., A.E. Dekas, C.T. Brown, S.K. Goffredi, T. Embaye, and V.J. Orphan. Diverse syntrophic partnerships from- deep-sea methane vents revealed by direct cell capture and metagenomics. Proceedings of the National Academy of Sciences of the United States of America 105: 7052–7057. 2008.

[104]. Venter et al., The sequence of the human genome, Science 291: 1304. 2001.

[105]. Wheeler et al., The complete genome of an individual by massively parallel DNA sequencing. Nature 452: 872–U875. 2008.

[106]. Wooley, J. C.; Godzik, A.; Friedberg, I., A Primer on Metagenomics. PLoS Computational Biology 6 (2): e1000667. 2010.

[107]. Hess M, Sczyrba A, Egan R, Kim TW, Chokhawala H, Schroth G, Luo S, Clark DS, Chen F, Zhang T, Mackie RI, Pennacchio LA, Tringe SG, Visel A, Woyke T, Wang Z, Rubin EM., Metagenomic discovery of biomass-degrading genes and genomes from cow rumen, Science 331 (6016): 463–467. 2011.

[108]. Junjie Qui et al., A human gut microbial gene catalogue established by metagenomic sequencing, Nature 464 (7285): 59–65. 2009.

[109]. Sogin ML, Morrison HG, Huber JA, Mark Welch D, Huse SM, et al. (2006). Microbial diversity in the deep sea and the underexplored "rare biosphere". Proc. Natl. Acad. Sci. USA 103: 12115-12120.

[110]. Roesch LFW, Fulthorpe RR, Riva A, Casella G, Hadwin AKM, et al. (2007). Pyrosequencing enumerates and contrasts soil microbial diversity. The ISME Journal 1: 283-290.

[111]. Brown MV, Philip GK, Bunge JA, Smith MC, Bissett A, et al. (2009). Microbial community structure in the North Pacific Ocean. The ISME Journal 3: 1374-1386.

[112]. Andersson AF, Lindberg M, Jakobsson H, Bäckhed F, Nyren P, et al. (2008). Comparative Analysis of Human Gut Microbiota by Barcoded Pyrosequencing. PloS ONE 3(7): e2836.

[113]. Quince C, Lanzen A, Davenport RJ, Turnbaugh PJ (2011). Removing nNoise from pyrosequenced amplicons. BMC Bioinformatics. 12:38

[114]. Taib N, Bronner G, Debroas D (2010). Annotation phylogénétique de sequences du gène codant pour l'ARNr 18S généré par pyrosequençage (PANAM). Rencontres ALPHY – 2-3 Février - Marseille.

[115]. Edgar RC. (2010). Search and clustering orders of magnitude faster than BLAST. Bioinformatics 26: 2460-2461.

[116]. Price MN, Dehal PS, Arkin PA (2009). FastTree: Computing Large Minimum Evolution Trees with Profiles instead of a Distance Matrix. Mol. Biol. Evol. 26(7): 1641-1650.

[117]. T. Tung DOAN, Najwa TAIB, H.Quang NGUYEN, Jean-Christophe CHARVY, Gisele BRONNER, Vincent BRETON, Didier DEBROAS, Engelbert MEPHU. *Portage de ePANAM sur la grille de calcul.* Rencontres Scientifiques France Grilles. 19 September 2011.

[118]. Pruesse E, Quast C, Knittel K, Fuchs B, Ludwig W, et al (2007). SILVA: a comprehensive online resource for quality checked and aligned ribosomal RNA sequence data compatible with ARB. Nucl. Acids Res. 35: 7188-7196.

[119]. Wu D, Hartman A, Ward N (2008). An Automated Phylogenetic Tree-Based Small Subunit rRNA Taxonomy and Alignment Pipeline (STAP). PLoS ONE 3(7): e2566.