High Performance Computing as a Combination of Machines and Methods and Programming

Soutenance en vue de l'obtention du Diplôme d'Habilitation à Diriger les Recherches

Claude Tadonki

Mines-ParisTech

Université Paris-Sud / 16 mai 2013

"A vouloir toujours aller au fond des choses, on court le risque d'y rester."

- **Fundamental Aspects of Algorithms and Complexity** *(Formalism, proof, quantification, classification, ...)*
- **Graph Theory and Applications** (characterisation, modeling, scheduling)

Claude Berge

- **Discrete Dynamical Systems** (Chip Firing Game, Periodicity, Garden of Eden, Invariants, ...)

Quinton

Charles Leiserson

Richard Karp

- **Parallel Scheduling** (Theory, SPMD, SIMD, Systolic, ...)
- **Polyhedral Model** (Recurrences Equations and Scheduling)
- **Efficient Parallel Programming** (Hybrid Parallel Computing, Scalability, Load Balancing, Data Transfers & Exchanges)
- **Automatic Code Generation and Transformations** (High-Level Specification $\Rightarrow$ Program; Program + Annotations $\Rightarrow$ Program)
- **Power Aware Computing** (Energy Efficient Programming: Models and Methods, Cloud Computing)

Schrijver

Vial

Maculan

- **Applied Mathematics and Operation Research** (Mathematical modeling, Optimization, Linear Algebra, Matrix Computation)

Saad

Places where I have been (study/work/conference/cooperation/teaching)

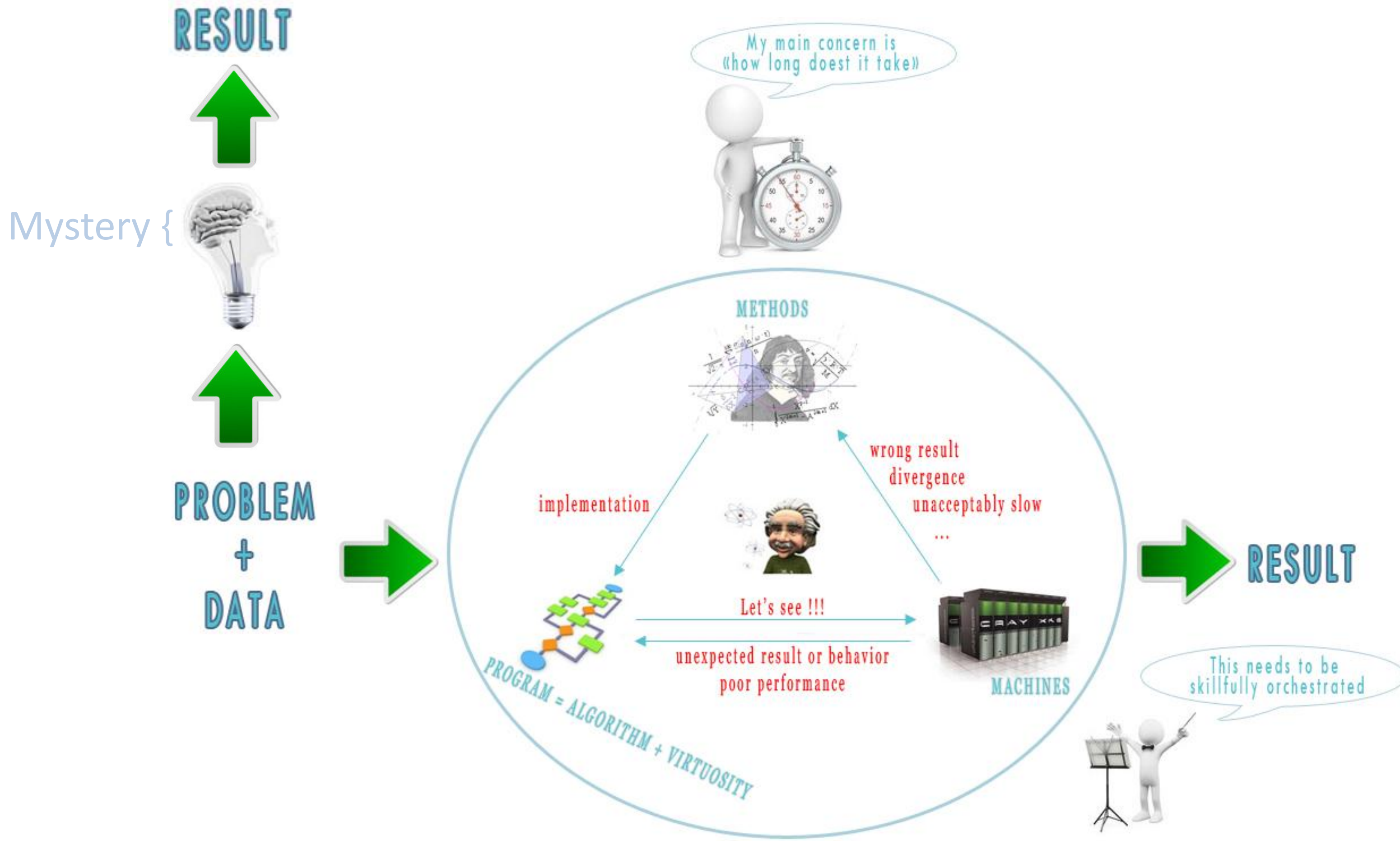
University of Yaoundé (Cameroon) → University of Rennes/IRISA → University of Geneva (HEC/CUI) → European Laboratory of Molecular Biology → University Paris-Sud → Ecole des Mines de Paris

92: CARI
97:CEPAMOQ

«science without border »
is the hallmark of my route.

France	Italy	Spain	Argentina	Austria
Switzerland	Germany	Hungary	Madagascar	Brazil
Romania	Portugal	Finland	Senegal	UK
Danmark	Greece	USA	Morroco	Ireland
Canada	Mexico	Cuba	Egypt	India
Madagascar	China	Japan	Cameroon	

The HPC Symphony



- **Significant advances** have been achieved in each of the aforementioned aspects.
- A **skillfull combination** of all HPC components is really the key to **absolute efficiency**.
- This **expected pluridiscipline interaction** should be better done at the **earliest**.
- This is the **main point of my defense** and the motivation behind **my future plans**.

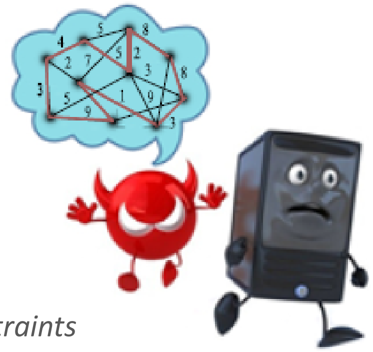
Computer vs Difficult Problem ≡ David & Goliath

- **TRAVELING SALESMAN PROBLEM**

Given a list of cities and their pairwise distances, the task is to find a **shortest tour** that visits each city exactly once.

$$\begin{cases} \text{minimize} & \sum_{i=1}^{n}\sum_{j=1}^{n} c_{ij}x_{ij} \\ \text{subject to} & \\ & \sum_{j=1}^{n} x_{ij} = 1 \quad i = 1,\cdots,n, i \neq j \qquad (2.4) \\ & \sum_{i=1}^{n} x_{ij} = 1 \quad j = 1,\cdots,n, i \neq j \\ & x \in \{0,1\}^{n\times n} \end{cases}$$

+ Subtour breaking constraints



The TSP has an a priori $n!$ complexity. Solving any instance with $n = 25$ using the current world fastest supercomputer (TITAN-CRAY XK7) might require 25 years of calculations.

TITAN in 1min ≈ 6 billion people calculating
24h/24h during 300 years

"Tu quieres celeste que te cueste."

Applications:

Transportation, logistic, genome sequencing, benchmark for optimization methods, ...

TSP heros: Applegate, Bixby, Chvatal, and Cook

- As difficult as the HAMILTONIAN CYCLE PROBLEM, which is NP-COMPLETE (Karp)
- Please, forget about brute force approach! (**16 cities ⇒ 653 837 184 000 possibilities**)
- Modern optimization method have shown optimistic performances on practical instances
- Competitive solutions are **parallel implementation of powerful optimization methods**
- TSP (VLSI-Bell Labs) of size **85 900** solved in **1.5 year** (2004-2006) using a cluster of 96 2.8 GHz Intel Xeon and 32 2.4 GHz AMD Opteron connected with 100 MB ethernet.

Approximation algorithm is also an interesting pragmatic way to go

- **Quicksort** is a worst-case *n^2* algorithm, but is still preferred to the *nlogn* **heapsort**.

- LP can be solved in a polynomial time, but the (expontial) **SIMPLEX** is still preferred.
  - The simplex has led to the leading solver **CPLEX**

  - We have implemented a nice interface between **CPLEX** and **MATLAB** in cooperation with **David Musicant** (Carleton College) and **Travis Johnson.**

    Our interface (started in 2004) has been used and cited in several mathematical programming papers, including ours (*P-median, Portfolio, Energy minimization*).

    Can be dowloaded at http://www.omegacomputer.com/staff/tadonki/using_cplex_with_matlab.htm

General Linear Programming

[obj, x, lambda, status, colstat, it] = lp_cplex(c, A, b, l, u, le, ge, maxIter, optimizer)

min c' * x with linear constraints

optimizer is a variable for optimizer choice, which can be one of the following
0 : primal simplex
1 : dual simplex
2 : network
3 : primal - dual barrier

Mixed Integer Programming

[obj, x, lambda, status, colstat, it] = mip_cplex(q, c, A, b, l, u, le, ge, binvar, genvar, maxIter, verbose)

min c' * x with linear contraints and integer variables (binary or general)

Indices of binary variables are listed in binvar, and indices of general integer variables are listed in genvar

- **High-precision L**attice **Q**uantum **C**hromo**D**ynamics simulations.
- The ANR project **PetaQCD** was targeting **256×128^3** lattices.
- One evaluation of the *Dirac operator* on a **256×128^3** lattice involves

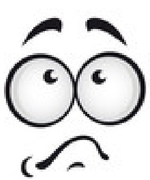
$256 \times 128^3 \times 1500 \approx 10^{12}$ (stencil) **floating-point operations**

With 10,000 cores, we can roughly perform $500 \times 10^3 \times 10^6 = 5\text{x}10^9$ fps

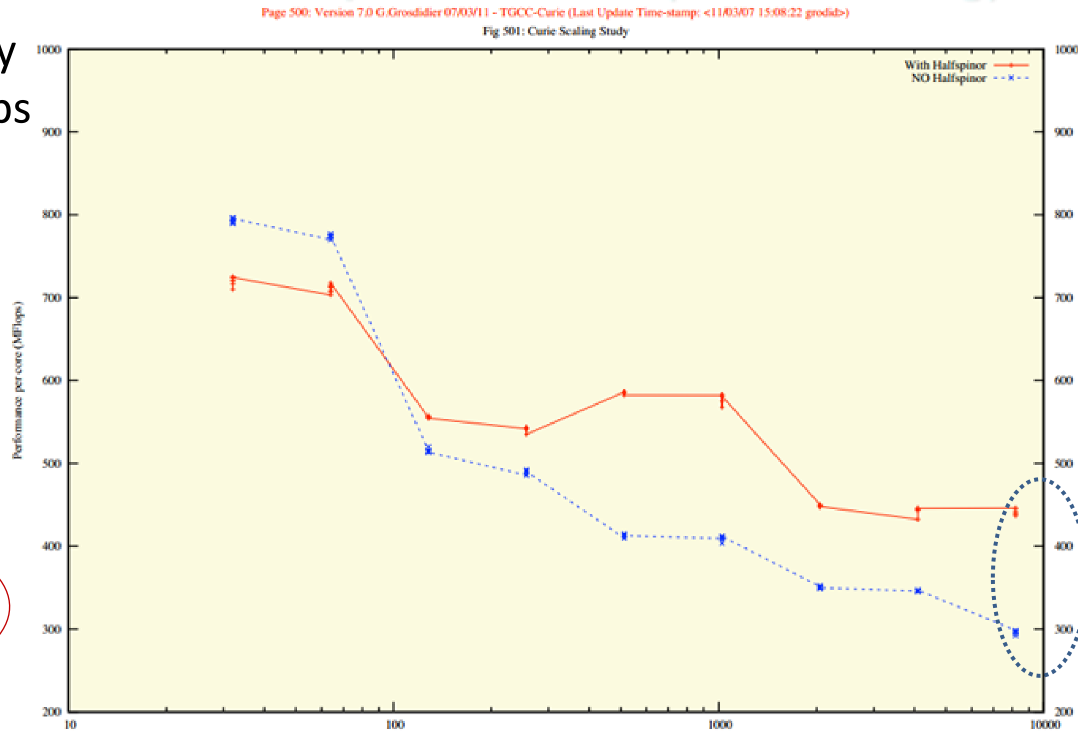
Our 256×128^3 lattice would then require 200 seconds ≈ **3 minutes** for each evaluation of the *Dirac operator*.

10 days !!!

Now, image that we have to do it 5000 times to solve One Dirac linear system !!!



Curie Fat performance (weak scaling)

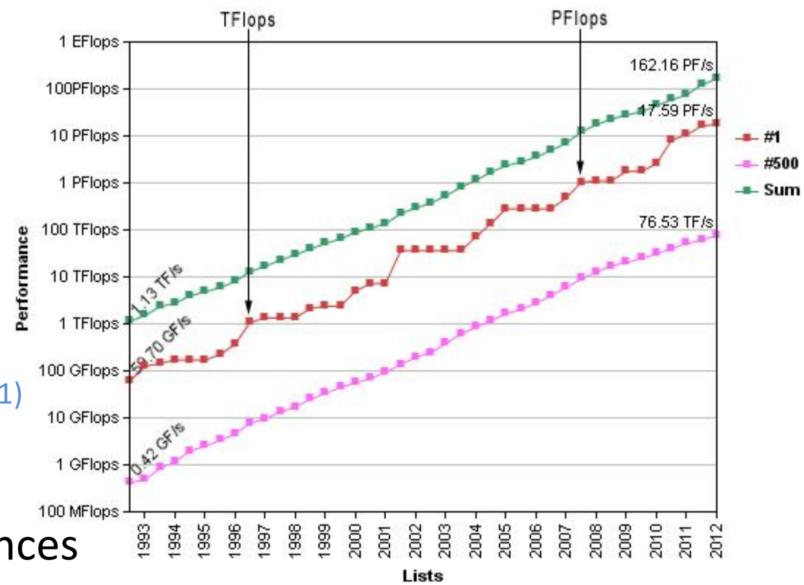


500 Mflops/core

G.Grosdidier, « *Scaling stories* », PetaQCD Final Review Meeting, Orsay, Sept. 27th – 28th 2012

Important Facts about Supercomputers

- The (peak/sustained) performance of supercomputers is increasing significantly (≈ $\times 10^5$ since 1993).
- The following characteristics are becoming a standard
  - Several cores (2012 list: 84% ≥ 6 cores and 46% ≥ 8 cores)
  - Vector units (with larger vector registers)
  - Accelerated (mainly GPU: 62 systems of the 2012 including the #1)



Performance evolution (Top500.org)

- The **gap between peak and sustained** performances on real-life applications is clearly questionable.
  HPC investigations should focus on this.
- When it comes to accelerators, **data transfer** is critical. Compromise and overlap (scheduling)

Chris Gregg and Kim Hazelwood, *Where is the Data? Why You Cannot Debate CPU vs. GPU Performance Without the Answer*, International Symposium on Performance Analysis of Systems and Software (ISPASS), Austin, TX. April 2011.

TITAN CRAY-XK7 the (2012) world fastest supercomputer

- 299 008 CPU cores (16-cores AMD Opteron 6274)
- 18 688 NVIDIA Tesla K20 GPUs
- Peak: **27.11** PFlop/s.
- Sustained: **17.59** PFlop/s (Linpack)

- Taking advantage of all aspects of a computing nodes requires a complex **hybrid/heterogeneous programming** . Code generation and transformation
- **Energy** is major concern.
  Power aware programming and scheduling.

Some of my achievements

Let’s have a look on some of my achievements

"Only those who attempt the absurd...will achieve the impossible." M. C. Escher

Given a graph $G = (X, A)$ and a transition function φ such that $\varphi(G) = (X, A')$ with $card(A) = card(A')$. We study the evolution $G_0 = G,\ G_1 = \varphi(G_0),\ \cdots,\ G_{n+1} = \varphi(G_n)$.

Remark: on a finite graph, such an evolution is ultimately periodic. Any state of the graph before the period is called transcient. Interesting questions are

- the set of period lengths for a given graph
- bounds of period lengths
- garden of eden (graph with no transcient length)
- relation between the structure of the graph and periodicity
- applications (games, simulation, graph scheduling)

This work gave an inspiration to derive a parallel scheduling methodology for acyclic graphs with some recurrent structure.

René Ndoundam, Claude Tadonki, Maurice Tchuente: ***Parallel Chip Firing Game Associated with n- cube Edges Orientations***. International Conference on Computational Science (2004). , CoRR abs/1007.0381 (2010)

A **system of recurrence equations** (SRE) defining a variable X has the following form

$$X(z) = \begin{cases} & \vdots \\ D_i^X & : \quad g_i(\ldots X(f(z))\ldots) \\ & \vdots \end{cases} \qquad F(i,j,k) = \begin{cases} \{i,j,k \mid k=0\} & : a_{i,j} \\ \{i,j,k \mid i=j=k\} & : F(i,j,k-1)* \\ \{i,j,k \mid i=k\neq j\} & : F(k,k,k)\otimes F(i,j,k) \\ \{i,j,k \mid j=k\neq i\} & : F(i,j,k-1)\otimes F(k,k,k) \\ \{i,j,k \mid i\neq k; j\neq k\} & : F(i,j,k-1)\oplus \\ & \quad (F(i,k,k)\otimes F(k,j,k-1)) \end{cases}$$

- ≻ Algebraic and syntaxic transformations (polyhedral model framework, compiler, scheduler, symbolic analysis, ...)
- ≻ Graph based approaches easily apply to SRE through the underlying dependence graph ⇒ systematic synthesis !

> Scheduling a system of recurrence equations onto p processors is the task of finding a valid *timing function* T (i.e. $T(z) > T(f(z))$) and an allocation function A such that $[T(z_1) = T(z_2)] \Rightarrow [A(z_1) \neq A(z_2)]$.

Theorem 1. *Let $G_1 = (X_1, \Gamma_1)$ and $G_1 = (X_2, \Gamma_2)$ two isomorphic graphs, and let φ an isomorphism from G_2 to G_1. If (t, a) is a valid schedule of G_1, where t is a timing function and a an allocation function, then $(t\circ\varphi, a\circ\varphi)$ is a valid schedule of G_2.*

Theorem 2. *Let $G = (X, A)$ be a directed acyclic graph. If G is self isomorphic with the decomposition $X = X_1 \cup X_2 \cup \cdots \cup X_\gamma$ and the set of isomorphisms $\{\varphi_k, k = 2, \cdots, \gamma\}$, then the schedule specified by the timing function t and the allocation function a given by (6-7) is valid.*

Applications: sorting, APP, Cholesky, tensor product of matrices

Our scheduling method (already published) opens the following perspectives

- Implementation as a scheduling module into a polyhedral model based framework.
- Study of an appropriate graph clustering (hybrid parallelism or modularity)
- Inclusion of hardware parameters

LOGILAB (University of Geneva)

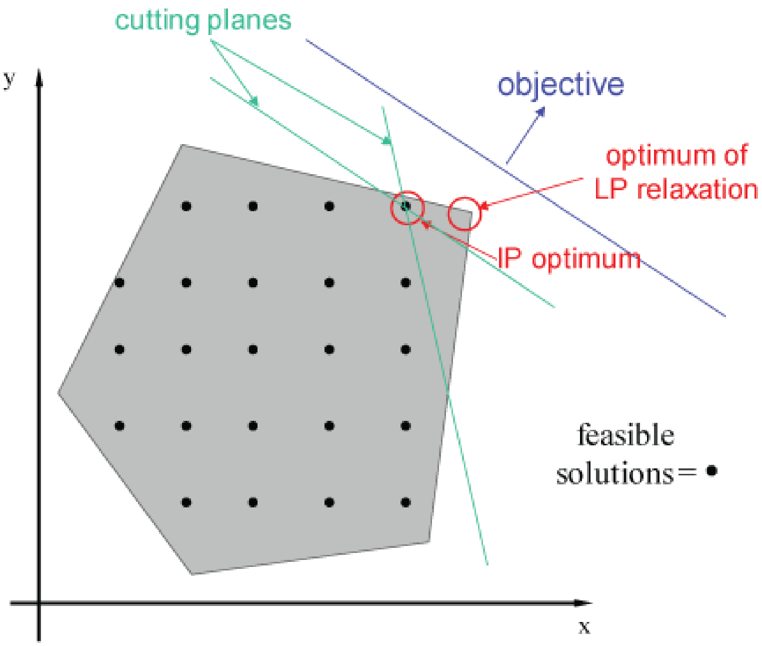




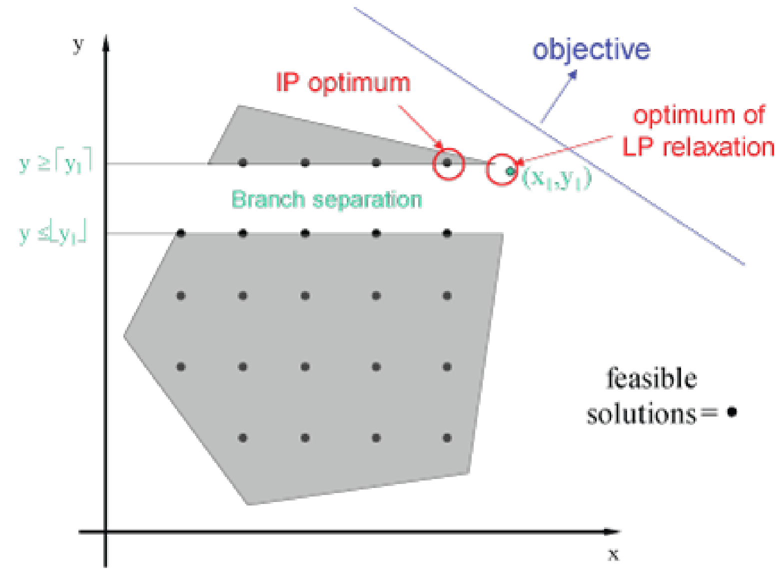
Inside the LOGILAB, Operation Research Laboratory
founded by Pr Jean-Philippe Vial and Pr Alain Haurie
at the University of Geneva (Switzerland)

Dealing with modern, efficient, and somehow pragmatic approaches for modeling and solving difficult real-world problems

Think about gaussian pivoting vs iterative methods

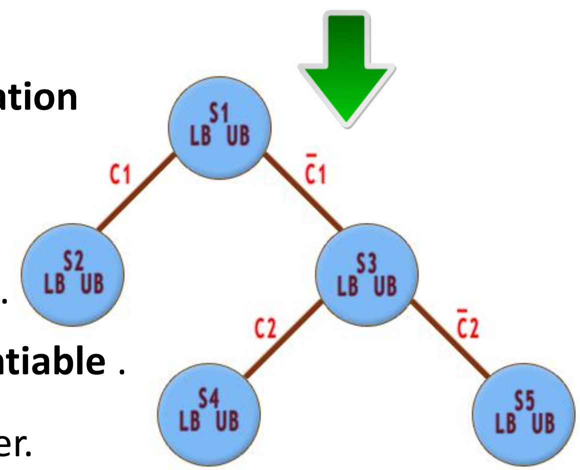


Integer Program: Cutting Planes



Integer Program: Branch and Bound
(or Divide and Conquer)

- **Practical instances** of **discrete** (pure or mixed) **optimization problems** are better solved though a skillfull combination of **continuous optimization** techniques and **branch&bound-like mechanisms**.
- For a pure discrete problem, a **relaxation** is used.
  For a mixed formulation, a **decomposition approach** can be considered.
- In number of cases, the objective function is (or becomes) **non differentiable** .
- We then need a good non differentiable optimization method and solver.



$$\min\{f(y) = f_1(y) + b^T y + f_2(y) \mid y \in C\}, \tag{1}$$

where

- f_1 and f_2 are convex functions on R^n defined over $C \subset R^n$
- f_1 is nonsmooth and f_2 is twice differentiable
- C is a convex domain
- $b \in R^n$ is a constant vector

Remark 1 *The nonsmooth function f_1 is often a positively weighted sum of p nonsmooth functions*

$$f_1(y) = \sum_{i=1}^{p} \pi_i f_{1i}(y). \tag{2}$$

This property can be exploited in the method.

Remark 2 *For a given convex function f, there are many combinations of the form $f(y) = f_1(y) + b^T y + f_2(y)$. The appropriate choice is left to the user convenience.*

*Project funded by the **S**wiss **N**ational Science **F**oundation*

In order to accomplish the optimization task, *ProxAccpm* runs in conjunction with user defined *oracles*.

An *oracle* is a user defined routine (black box for ProxAccpm), which computes and returns informations about feasibility and/or optimality.

We consider tree types of oracle:

Feasibility oracle: Check if the current point y belongs to the optimization domain. Otherwise, its returns a so-called *feasibility cut*, we denote $(\xi, \tau) \in R^n \times R$, which satisfies

$$\xi^T(y' - y) + \tau \leq 0, \forall y' \in C. \quad (3)$$

First order oracle: Given a feasible point y, it returns $f_1(y)$ and one subgradient $u \in \partial f_1$. This yields the following relation:

$$u^T(y' - y) + (f_1(y') - f_1(y)) \leq 0, \forall y' \in C. \quad (4)$$

Second order oracle: Given a feasible point y, it returns $f_2(y)$, $f_2'(y)$, and $f_2''(y)$.

Localization Set: At a given step, the *localization set* is defined by

$$\mathcal{L}_{\bar{\theta}} = \{(y, z, \zeta) \mid A^T y - E^T z \leq c,\ \pi^T z + b^T y + \zeta \leq \bar{\theta},\ f_2(y) \leq \zeta,\ y \in Y_2\}, \qquad (7)$$

where

- $\bar{\theta}$ is the current upper bound
- A is the matrix of cuts (columnwise)
- E is a binary banded matrix

Methods differ in

- the management of the localization set (objective cut, updating, spatial transformations, etc...)
- the selection of the query point from the localization set
- the lower bound and the termination criterion

Accpm and Proximal Accpm

The Analytic Center Method: The query point is obtained by minimizing (over the localization set) the weighted logarithmic barrier given by

$$F(\bar{s}) = -\sum w_i \log s_i - \omega \log \sigma, \tag{8}$$

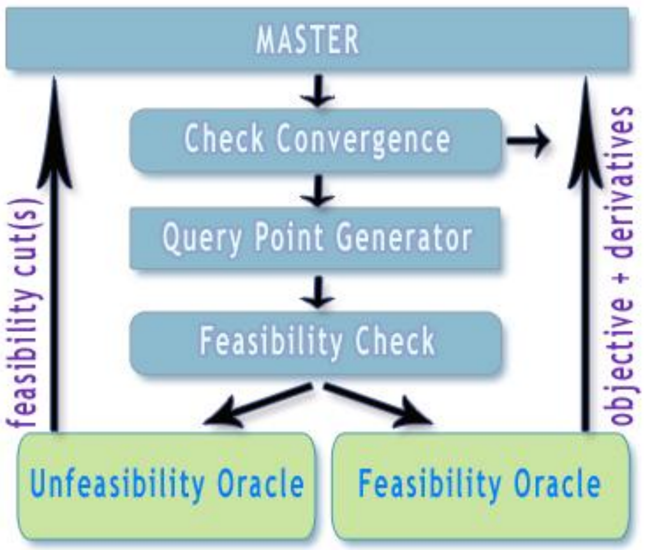
with $\bar{s} = (s_0, s, \sigma) > 0$ defined by

$$\begin{aligned} s_0 &= \bar{\theta} - (\pi^T z + b^T y + \zeta) \\ s_i &= c_i - (A^T y - E^T z)_j, \qquad i \in I = \{1, \ldots, m\}, \\ \sigma &= \zeta - f_2(y). \end{aligned}$$

The Proximal Analytic Center Method: The barrier function is augmented with a proximal term to yield the augmented barrier

$$\frac{1}{2}(y - \bar{y})^T Q (y - \bar{y}) + F(\bar{s}) \tag{9}$$

where Q is a positive definite matrix and $\bar{y}$ the so-called *proximal center*..



The main concerns are

- force the global convergence
- minimize the number of oracle calls
- optimize the computation time and memory space
- improve the computation accuracy
- take care of numerical stability

Babonneau, F., Beltran, C., Haurie, A., Tadonki, C. and Vial, J.-P., *Proximal-ACCPM: a versatile oracle based optimization method*, In Optimisation, Econometric and Financial Analysis, E. J. Kontoghiorghes editor, vol. 9 of Advances in Computational Management Science**,** 2006.

Achievements

- Implementation of the method in a **complete framework** (https://projects.coin-or.org/OBOE).
- Implementation as a query point generator for **connection with other packages** (*branch and bound*)
- Several case studies **published in journal and conferences**

Some perspectives

- Deep investigation of the **parallelization** on supercomputers (*scheduling, scalability, load balance,…*)
- Study how to deal with **updates** instead of performing matrix computations from scratch at each step *Sherman-Morrison formula*
- Investigate on numerical issues with large-scale ill-conditioned systems
- Approximation algorithms

Power consumption is a crucial concern with embedded systems and supercomputers

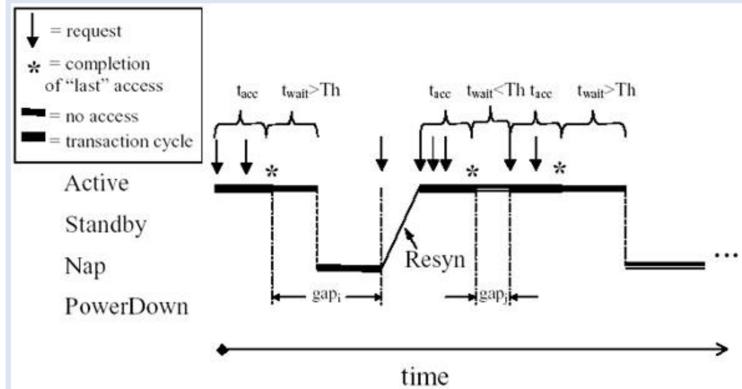
Fujistsu K-Computer ≈ US$10 millions/year for electricity

Pr Jose Rolim
Theoretical Computer Science Laboratory (TCS Lab)
Centre **U**niversitaire **I**nformatique
University of Geneva (Switzerland)

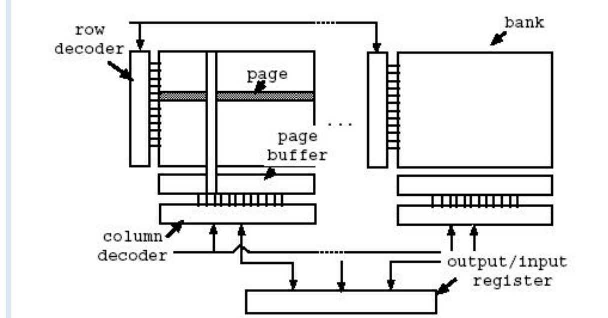
My work on Power Aware Computing and Distributed Algorithms were initiated from my stay at the TCS-Lab.

- We focus on memory energy
- The memory is partitionned into several banks
- Each bank can be put into a specific power mode
- We assume regular (uniform) power state transitions
- We formulate and solve the optimization problem

$$\min\ xH^TQy^T + RQy^T$$
subject to
1. $x \in \mathbb{N}^q$,
2. $y \in \mathbb{R}^q$,
3. $y_1 + y_2 + \cdots + y_q = p$,
4. $y_0 \geq \mu C$,
5. $x_1 + x_2 + \cdots + x_q = \rho C$,
6. $y \leq \varphi x$.
7. $y \geq \eta x$.



Multi-states Memory Architecture



Multi-banked SDRAM architecture

*Good collaboration with **Mitali Singh** and **Viktor Prasanna** (University of Southern California)*

Energy Minimization

Input	Model and Optimization	Output	
Hardware parameters : *W, Q, H, p, q, and* δ Program parameters : *R, C* Memory management parameters : ρ, φ, η		Optimal Energy *E* Optimal transition repartition *X* Optimal time repartition *Y*	Power aware program design and monitoring

Some perspectives are:

- Design a **methodology** that will use the output of our work for a systematic synthesis of **energy efficient policies**
- Extend and adapt our model to **current and future memory systems** (*multilevel* and *shared*)
- Use a similar formal approach to analyse the energy issue on the **cloud systems** (user and provider)
- Investigate on other approaches (dynamic scheduling, compilation, …)

One of our proposal on this topic received an important grant from the *__S__wiss __N__ational Science __F__oundation*

We have been also investigating on:

- Distributed algorithm in sensors networks (*localization and information retrieval*)
- Dual-power management problem (*mathematical programming approach and heuristics*)
- Algorithms for the web (*search engines and social networks*)

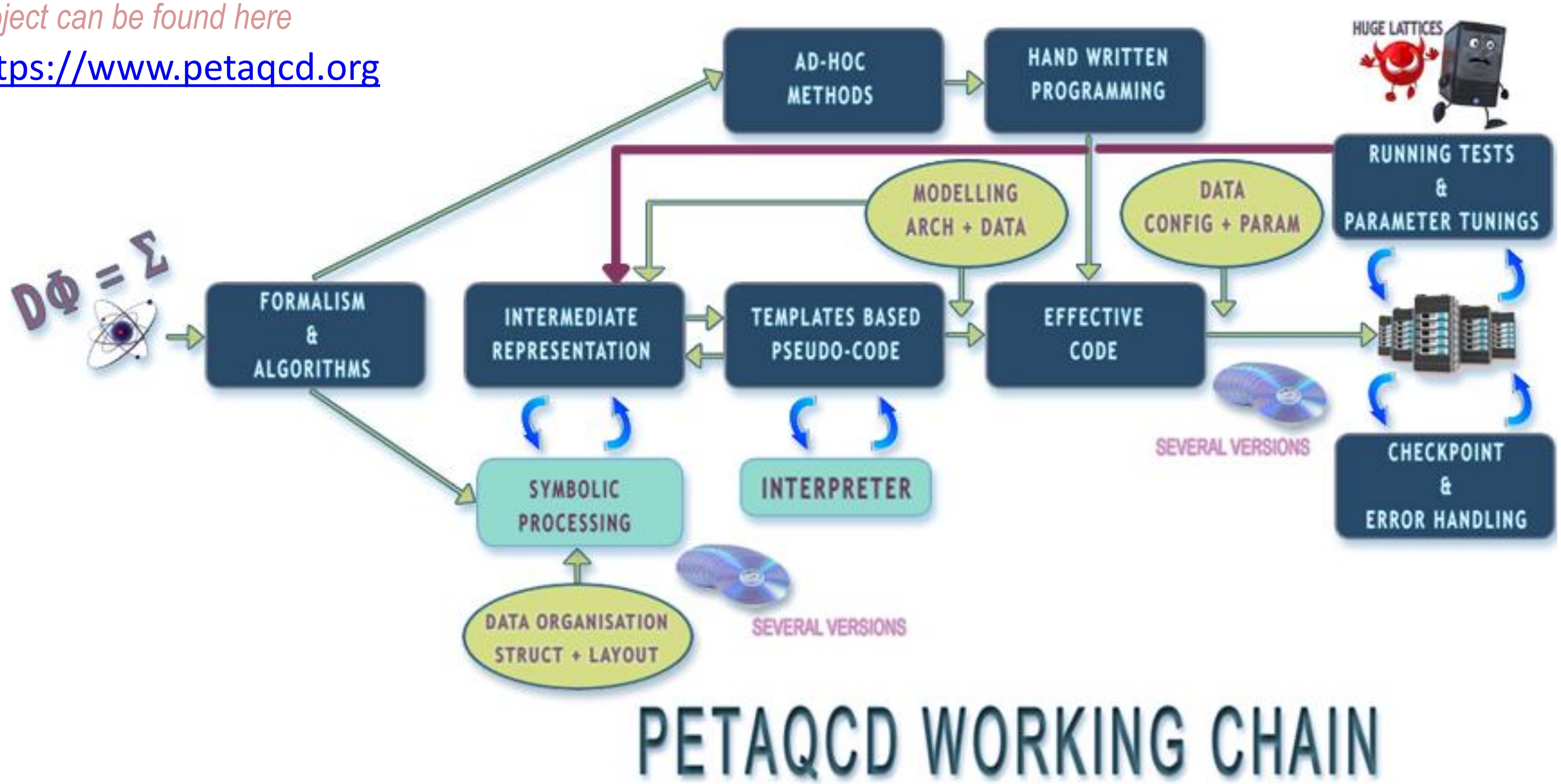
- ANR Project (HPC & Particles Physics)
- Origin of the universe (matter)
- Good cooperative effort
- Multidisciplinary collaboration
- HPC & numerical challenges

National: G. Grosdidier (*coordinator*), P. Roudeau, O. Pène, C. Tadonki, K. Petrov, D. Barthou, M. Kruse, C. Eisenbeis, B. Blossier, L. Morin, F. Bodin, F. Touze, O. Brand-Foissac, J. C. Angles d'Auriac, A. Seznec, A. Cohen, C. Bastoul, and more.

Abroad: K. Urbach, K. Jansen, L. Scorzatto, D. Pleiter, R. Tripiccione

Lot of materials and outputs of this project can be found here

https://www.petaqcd.org



PetaQCD – Contributions

- Hybrid extension of the reference package (tmLQCD) using *Pthread* library
- Accelerated algorithm & implementation of the ***Dirac operator*** and other linear algebra kernels

Illustrative results

- A 32×16^3 configuration solved using the CGR algorithm on the **PPU in 138 seconds**
- The same configuration and algorithm on the **(PPU + 8 SPEs) double precision in 4.58 seconds**

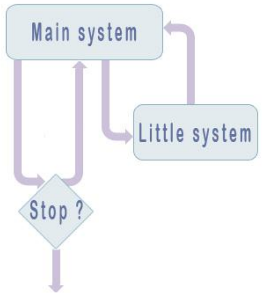
CGR:	57 iterations in	4.58 s (QS20)	3.68 s (QS22)	28.34s s (Intel 2.83Ghz)
CG:	685 iterations in	51.70 s (QS20)	38.80 s (QS22)	362.45 s (Intel 2.83Ghz)

- Compare to the PPU, a speedup around **30** was obtained (also valid per iteration)

C. Tadonki, G. Grosdidier, and O. Pene, « *An efficient CELL library for Lattice Quantum Chromodynamics* », ACM SIGARCH Computer Architecture News, vol 38(4) 2011.

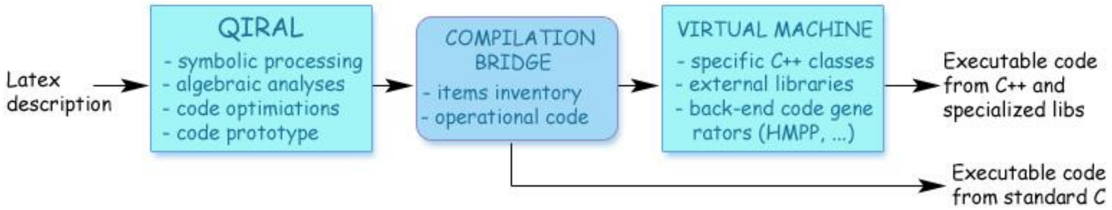
- Block decomposition

C. Tadonki, «Strengthening deflation implementation» for large scale LQCD inversions, Orsay, Sept. 27th – 28th 2012

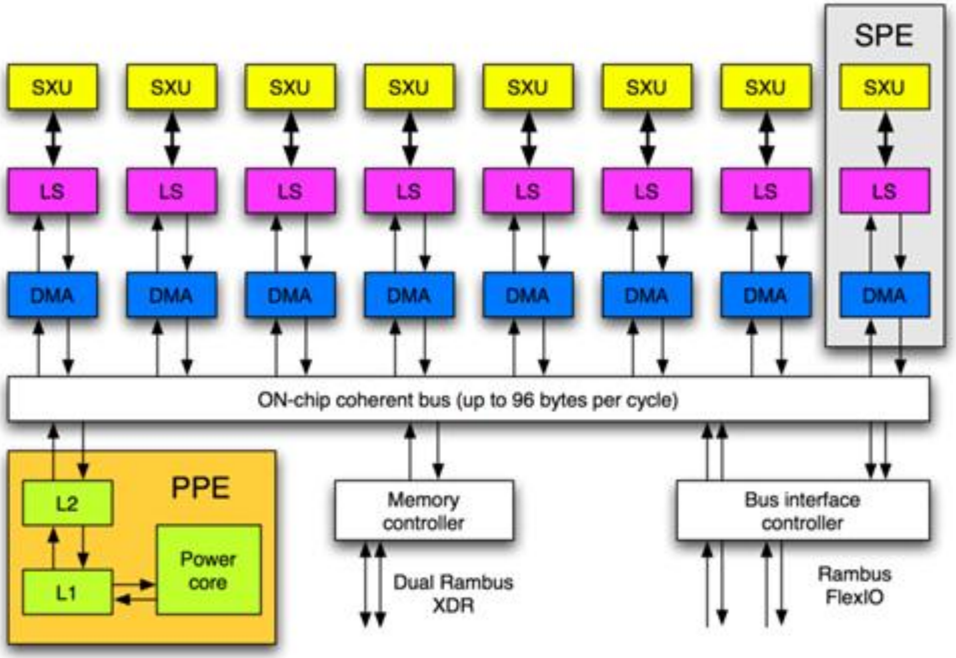


We have implemented a generic block decomposition (**multidimensional** and with no restriction on the **number of blocks per axis)** within the tmLQCD package. The aim is to increase the basis of the deflation method (to solve large ill-conditioned systems) without

- Code generation



D. Barthou, G. Grosdidier, M. Kruse, O. Pene and C. Tadonki, « QIRAL: A High Level Language for Lattice QCD Code Generation », Programming Language Approaches to Concurrency and Communication-cEntric Software (PLACES'12) in conjunction with the European joint Conference on Theory & Practice of Software (ETAPS), Tallinn, Estonia, March 24-April 1, 2012.

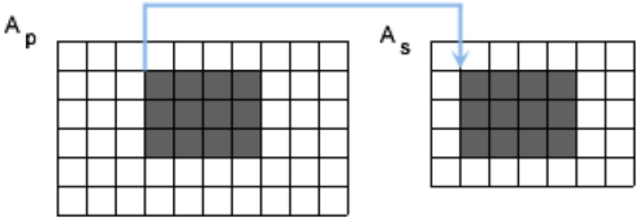


*I start investigating on this topic from my collaboration with Dr **Lionel Lacassagne** (ANR project Ocelle 2007-2009), also SIMD and Image Processing stuffs.*

*Good collaboration with Dr **Joel Falcou, Tarik Saidani, Khaled Hamidouche**, and **Pr Daniel Etiemble***

T. Saidani, J. Falcou, C. Tadonki, L. Lacassagne, and D. Etiemble,
« Algorithmic Skeletons within an Embedded Domain Specific Language for the CELL Processor »,
Parallel Architectures and Compilation Techniques (PACT), PACT09, Raleigh, North Carolina (USA), September 12-16, 2009.

DMA issues related to tiling



Main memory: $n_p = 6,\ m_p = 10,\ i_p = 2,\ j_p = 4$
Local store: $n_s = 5,\ m_s = 7,\ \ i_s = 2,\ j_s = 2$
$a = 3,\quad b = 4$

Fig. 4. Generic DMA pattern

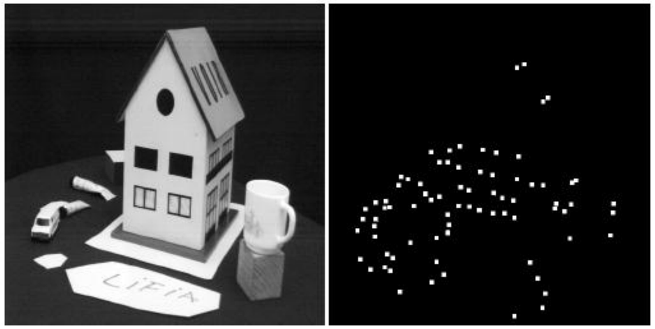
Performing the transfer expressed in figure 4 raises number of problems:

- *the region to be transferred is not contiguous on memory, thus list DMAs are considered*
- *the address of one given row is not aligned, thus the global list DMA is not possible*
- *the (address, volume) pair of a row does not match the basic DMA rules (the above two* ones), thus the entire list DMA cannot be carried out
- *misalignment could come from both sides (main memory and/or local store)*
- *the target region on the local store might be out of the container limits*

We have designed and implemented a routine which performs this task very efficiently

The Harris-Stephen algorithm is

- a corner (point of interest) detection algorithm
- an improved variant of the original algorithm by Moravec
- used in computer vision for feature extraction like
  - motion detection
  - image matching
  - tracking
  - 3D reconstruction
  - object recognition



Technically, the Harris algorithm is based on a pixelwise autocorrelation S given by

$$S(x,y) = \sum_{u,v} w(u,v)[I(x,y) - I(x-u,y-v)]^2$$

where *(x, y)* is the location of the pixel and *I(x, y)* its intensity (grayscale mode).

$tile_h$	$tile_w$	total time(s)
8	512	0.0494
16	256	0.0598
32	128	0.0485
64	64	0.0345
128	32	0.0517
256	16	0.0699
512	8	0.0734

Table 1. Timings on a 512× 512 image

$tile_h$	$tile_w$	total time(s)
8	512	0.198
16	256	0.238
32	128	0.187
64	64	0.110
128	32	0.180
256	16	0.218
512	8	0.352

Table 2. Timings on a 2048× 512 image

$tile_h$	$tile_w$	total time(s)
5	1200	0.494
10	600	0.360
20	300	0.264
40	150	0.235
80	75	0.183
160	37	0.247
320	18	0.275

Table 3. Timings on a 1200× 1200 image

$tile_h$	$tile_w$	total time(s)
8	512	0.985
16	256	0.726
32	128	0.643
64	64	0.438
128	32	0.692
256	16	0.866
512	8	1.422

Table 4. Timings on a 2048× 2048 image

We observe 50% improvement between square tiles and full row tiles.

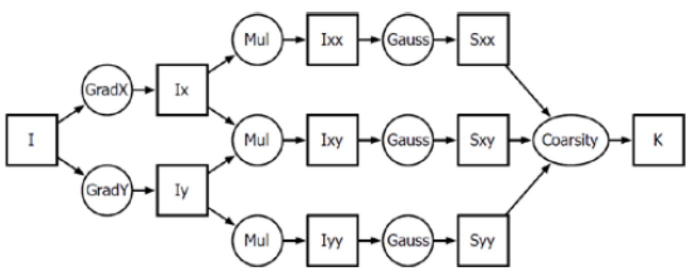
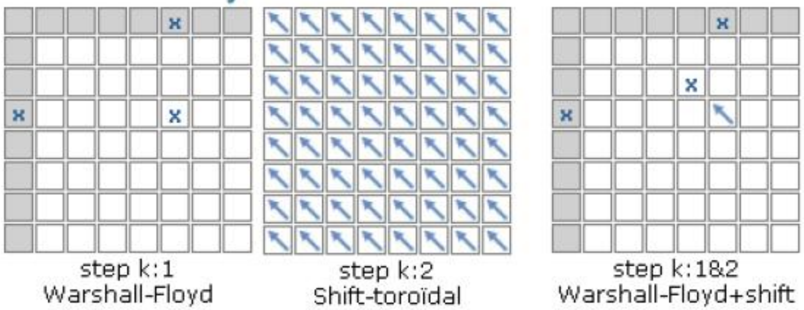


Fig. 3. Harris algorithm diagram

C. Tadonki, L. Lacassagne, E. Dadi, M. Daoudi, «Accelerator-based implementation of the Harris algorithm», 5th International Conference on Image Processing (ICISP 2012), June 28-30, 2012.

Warshall-Floyd + Shift-toroïdal



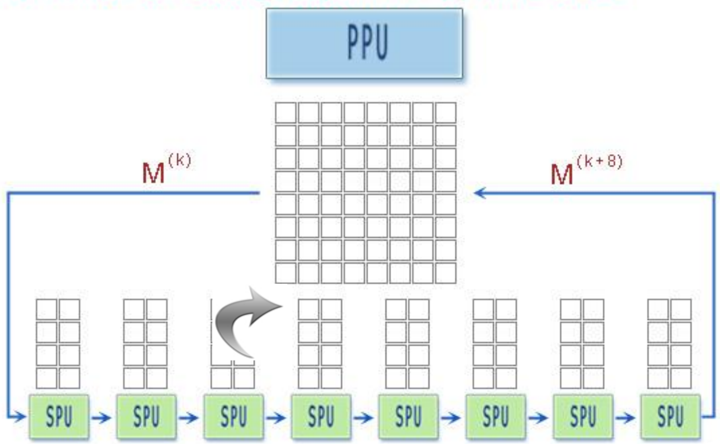
Analytical expression of the shift-toroïdal

$$m^{(k)}_{i-1,j-1} = m_{ij} \oplus (m^{(k-1)}_{ik} \otimes m^{(k-1)}_{kj})$$

Benefits of the shift-toroïdal

- *The pivot row (colum) remains on top*
- *Shift + computation can be done simultaneously*
- *All steps become rigourously identical*
- *The overhead of the shift can be hidden*
- *Does not change the O(n3) complexity*

MAPPING OF OUR ALGORITHM ON THE CELL



- PPE-DMA is issued only by the first and the last processor
- Inner SPEs communicate and synchronize locally
- Computation-communication overlap occurs for all communications
- Can run on more SPEs or CELL Blades by natural extension

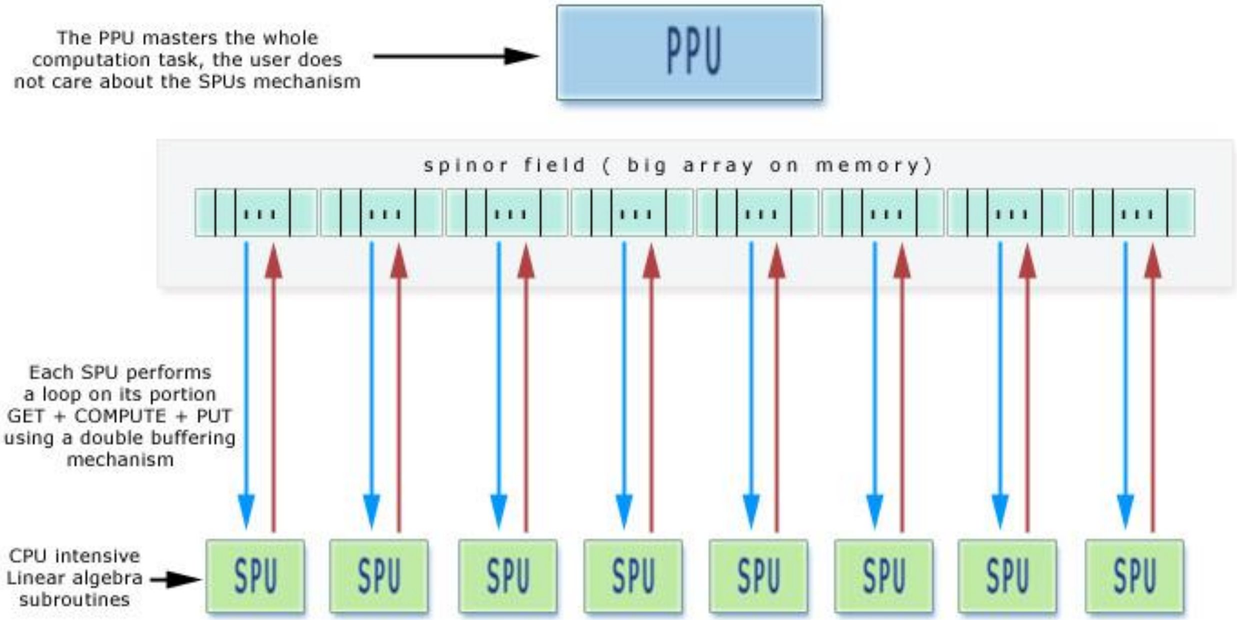
	1 SPE	2 SPEs		8 SPEs	
Tile	t(s)	t(s)	σ	t(s)	σ
1	6.67	3.28	2.03	1.60	4.16
4	5.01	2.50	2.00	0.62	7.99
8	4.79	2.39	2.00	0.60	7.95
12	4.70	2.32	2.02	0.58	7.98
16	4.72	2.36	2.00	0.60	7.79

(c) Performance with a 1024×1024 matrix

DMA timings (1024×1024 matrix)

	1 SPE	2 SPEs		8 SPEs	
Tile	t(s)	t(s)	σ	t(s)	σ
1	1.87	1.20	1.56	0.53	3.49
4	0.39	0.47	0.83	0.11	3.70
8	0.19	0.12	1.64	0.06	3.78
12	0.12	0.08	1.65	0.03	4.02
16	0.09	0.06	1.63	0.03	3.78

C. Tadonki, « Ring pipelined algorithm for the algebraic path problem on the CELL Broadband Engine », WAMMCA 2010 - SBAC PAD 2010, Petropolis, Rio de Janeiro, Brazil, October 27-30, 2010. (IEEE digital library)



GENERIC LQCD COMPUTATION MECHANISM ON THE CELL

- SIMD implementation of the basic operators (intensive use of SPU intrinsics)
- double buffering technique to overlap DMA and computation (ideal in double precision)
- optimal list DMA organisation by an algorithm similar that used for the *bin packing* problem (significant latency reduction)

1 Performance results

All our experimentations are performed with *double precision* data. We used a 32×16^3 lattice (hence 131 072 sites)

#SPEs	time(s)	speedup	GFlops
1	0.109	1.00	0.95
2	0.054	2.00	1.92
3	0.036	3.00	2.89
4	0.027	3.99	3.85
5	0.022	4.98	4.73
6	0.018	5.96	5.78
7	0.015	6.93	6.94
8	0.013	7.88	8.01

Figure 1: 32×16^3 Wilson-Dirac on a QS20

#SPEs	time(s)	speedup	GFlops
1	0.0374	1.00	2.76
2	0.0195	1.91	5.31
3	0.0134	2.79	7.76
4	0.0105	3.56	9.90
5	0.009	4.81	13.27
6	0.0081	5.75	15.87
7	0.0076	6.84	18.88
8	0.0075	7.82	21.59

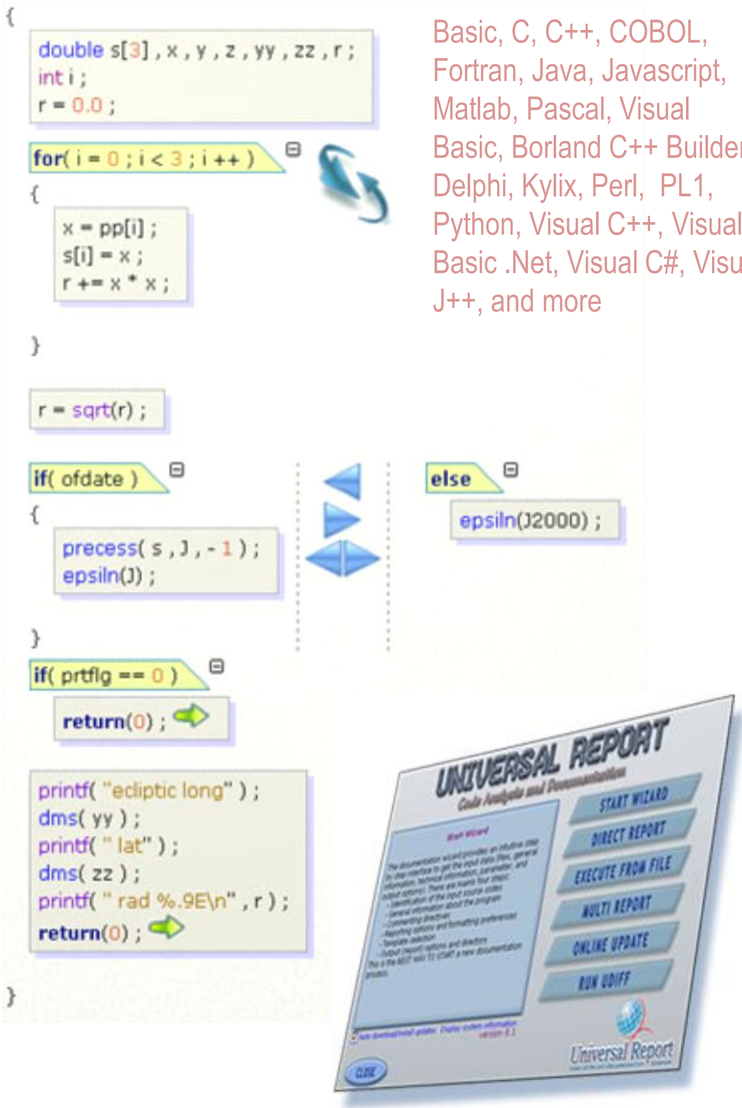
Figure 2: 32×16^3 Wilson-Dirac on a QS22

Without SSE		With SSE	
1 core	4 cores	1 core	4 cores
0.0820	0.0370	0.040	0.0280

Figure 3: 32×16^3 Wilson-Dirac timings (seconds) on an INTEL i7 quadcore 2.83 Ghz

C. Tadonki, G. Grosdidier, and O. Pene, «An efficient CELL library for Lattice Quantum Chromodynamics», HEART - ACM/ICS, Epochal Tsukuba, Tsukuba, Japan, June 1-4, 2010.

Basic, C, C++, COBOL, Fortran, Java, Javascript, Matlab, Pascal, Visual Basic, Borland C++ Builder, Delphi, Kylix, Perl, PL1, Python, Visual C++, Visual Basic .Net, Visual C#, Visual J++, and more



Prestigious custumers

NASA - Northrop Grumman Corporation

www.omegacomputer.com

Claude Tadonki , « Universal Report: A Generic Reverse Engineering Tool »,
12th IEEE International Workshop on Program Comprehension, IWPC 2004 , University of Bari, Bari, Italy , June 2004 ,

- Cooperation with Brazil
  - **BioCloud** project of the STIC-AmSud
  - Sandwich PhDs (*Brazilia/Rio/Niteroi*)
  - Papers, visitings, events, courses
  - Other Latin America partners (*Chile, …*)
  - + INRIA and Paris-Sud University
- Cooperation with Morocco (*Oujda*)
  - Sandwich PhDs
  - Papers, visitings, seminars
- Cooperation at Mines ParisTech
  - TIMC project (multi-target images)
  - CMM (Math Morphology) and CAOR

BioCloud Partners at Orsay

CPOCC

1st Workshop on Cost and Performance Optimization in Cloud Computing

in conjunction with the

25th International Symposium on Computer Architecture and High Performance Computing

October 23-26, 2013, Porto de Galinhas, Ipojuca, Pernambuco, BRAZIL

HOME
IMPORTANT DATES
SUBMISSION
COMMITTEES
REGISTRATION
CAMERA READY
PROGRAM
NEWS

Location of the conference and workshops at Porto de Galinhas

Cloud computing is coming into vogue as a convenient solution to share high performance computing resources (CPUs, storage medias and applications) among a widespread set of users. This configuration brings number of advantages related to the cost from both end-users and providers sides. It also offers a great opportunity to have a better control over power consumption and carbon emission. The concept of cloud computing itself is not so recent, but its implementation considering the various kinds of available devices is still on the way to maturity. The increasing audience and quality of service issues related to cloud systems stress to investigate on cost and energy at various levels. From the user side, the pay-as-you-go model together with the possibility to choose the resource configuration offer a way to reduce both cost and wastage. From the provider point of view, energy remains the main concern, which needs to be optimized for both profit and lifetime. Therefore, we need good models and methodologies that help to manage the cloud efficiently according the aforementioned keypoints, with a special attention to cloud federation.

CPOCC workshop aims at providing a platform to present and discuss both theoretical and experimental investigations related to Cloud computing performance and cost analyses. Authors are invited to submit original manuscripts on any of the topics described below.

TOPICS

Topics of interest for the workshop include but are not limited to:

- power consumption models for clouds and analysis
- cost models and optimization strategies for cloud computing
- federated cloud models and benchmarking
- successful case studies on existing clouds
- data and/or compute intensive applications on the cloud (*bioinformatics, ecology, weather and climate models, astrophysics, computational finance, computational chemistry, large databases, and others*)
- machine learning

WORKSHOP ORGANIZATION

- Claude Tadonki (Mines ParisTech / France)
- Christine Eisenbeis (INRIA / France)
- Lucia Maria Drummond (Federal Fluminense University / Brazil)
- Alba Cristina Magalhães Alves de Melo (University of Brasilia / Brazil)
- Maria Emilia Machado Telles Walter (University of Brasilia / Brazil)
- Mario Inostroza-Ponta (University of Santiago de Chile / Chile)
- Carolina Bonacic (University of Santiago de Chile / Chile)

Workshop on Cost and Performance Optimization in Cloud Computing

CLOUD COMPUTING

In memory of Jean Tadonki (1939-2001)
« You are the greatest »

Thanks to all of you

Family, friends, colleagues, collaborators, advisors, examiners, referees, administration, technicians, audience, you, …