Some examples of algorithmic contributions on high performance computing driven by highly demanding applications

Stéphane Lanteri
Inria Sophia Antipolis - Méditerranée, France

Forum ORAP : spécial 20 ans
October 14-15, 2014
Maison de l'UNESCO, Paris

Table of contents

Outline

Context and motivations

General context

Computational science (also scientific computing or scientific computation) is concerned with constructing mathematical models and quantitative analysis techniques and using computers to analyze and solve scientific problems (Wikipedia)

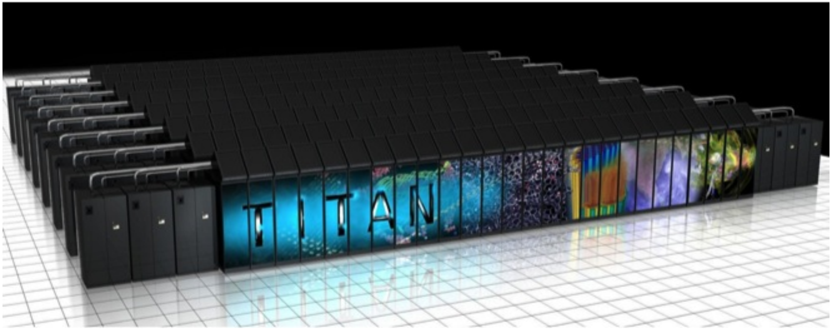
In practical use, it is typically the application of computer simulation (numerical simulation) tools to study problems in various scientific disciplines

- Computational fluid dynamics
- Computational electromagnetics
- Computational geoseismics
- Computational chemistry
- Etc.

Typical (minimal) scenario

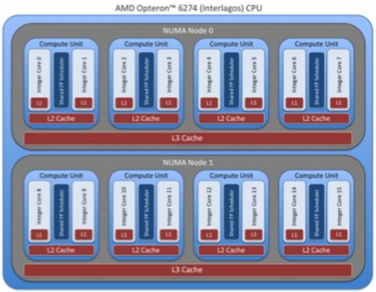
$$\boxed{\begin{array}{c}\text{Physical phenomena}\\ \Downarrow \\ \text{Mathematical model}\\ \Downarrow \\ \text{Numerical model}\\ \Downarrow \\ \text{Simulation tool}\end{array}}$$

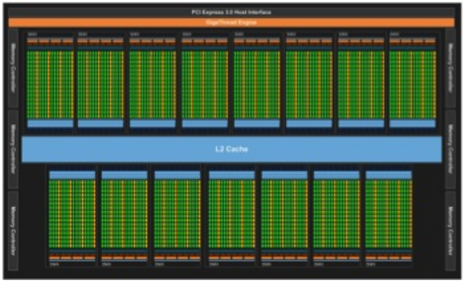
Context and motivations



Titan system, Oak Ridge National Laboratory (# 2 TOP 500 - June 2014)
Cray XK7 , AMD Opteron 6274 16C 2.2 GHz, Cray Gemini interconnect, NVIDIA K20x
560,640 cores, 27,112 TFlop/s peak

Context and motivations





Titan system, Oak Ridge National Laboratory (# 2 TOP 500 - June 2014)
Cray XK7 , AMD Opteron 6274 16C 2.2 GHz, Cray Gemini interconnect, NVIDIA K20x
560,640 cores, 27,112 TFlop/s peak

Context and motivations

Hardware trends to exascale

- In recent computer systems, parallelism spreads over many architecture levels including nodes, processors, cores, threads, registers, SIMD-like and vector units
- Several different levels of parallelism (from coarse to fine or very fine grain parallelism) need to be harnessed in order to maximize computational efficiency and scalability
- Moreover, heterogeneity of the memory is growing at the node as well as at the chip level
- The NUMA penalty in data accesses is one of the main critical issues for parallel performances and one must take care of locality access and memory affinity when distributing data on cores
- All these heterogeneous characteristics of hardware resources keep effective performance far from theoretical peak

Context and motivations

Challenges on the application side

- Most applications and algorithms are not yet ready to utilize these available architecture capabilities
- Developing large-scale scientific computing tools that efficiently exploit this processing power (currently, in the petascale range) is a very complicated task and will be an even more challenging one with future exascale systems
- Heterogeneity characteristic and hierarchical organization of modern massively parallel computing systems are recognized as central features that impact all the layers from the hardware to the software with issues related to computer science and numerical mathematics as well
- At the current state of the art in technologies and methodologies, a multi-disciplinary approach is required to tackle the obstacles in manycore computing, with contributions from computer science, applied mathematics, high performance computing, and engineering disciplines

Context and motivations

C2S@Exa Inria Project Lab

Establishment of a continuum of skills in the applied mathematics and computer science fields for a multidisciplinary approach to the development of numerical simulation tools that will take full benefits of the processing capabilities of emerging high performance massively parallel architectures

Launched in January 2013, 4 years duration

Supported by Inria's Reseach Department and Technological Development Department

Multi-site, multi-teams project, involving permanent researchers, PhDs, postdocs and fixed-term engineers ($\approx$ 30 people)

Activities and contributions are organized along a three-level structure from generic building-blocks to large-scale applications

- Level 1 - Towards generic and scalable algorithms
- Level 2 - Towards robust, accurate and highly scalable numerical schemes for complex physical problems
- Level 3 - Towards exascale computing for the simulation of frontier problems

Context and motivations

C2S@Exa Inria Project Lab

- Pole 1: Numerical linear algebra
  Core numerical kernels, sparse direct solvers, preconditioned iterative solvers and continuous solvers
- Pole 2: Numerical schemes for PDE models
  High order finite element and finite volume schemes optimized for single core computational performances and scalability
- Pole 3: Strategies and tools for performance optimization of numerical solvers
  Scheduling strategies, runtimes for resource virtualization, parallel mesh (re)partitioning
- Pole 4: Programming models
  Component models and high level programming models
- Pole 5: Resilience for exascale computing
  Efficient fault tolerant protocols and algorithm-based fault tolerance, performance execution models for fault-tolerant applications, resilience for sparse linear algebra

Context and motivations

C2S@Exa Inria Project Lab
Core project-teams: numerical mathematicians

- Bacchus [Inria Bordeaux - Sud-Ouest]
  Parallel tools for numerical algorithms and resolution of essentially hyperbolic problems
- Hiepacs [Inria Bordeaux - Sud-Ouest]
  High-end parallel algorithms for challenging numerical simulations
- Nachos [Inria Sophia Antipolis - Méditerranée]
  Numerical modeling and high performance computing for evolution problems in complex domains and heterogeneous media
- Sage [Inria Rennes - Bretagne Atlantique]
  Simulations and algorithms on Grids for environment
- Tonus [Inria Nancy - Grand-Est]
  Tokamak numerical simulations

Context and motivations

C2S@Exa Inria Project Lab
Core project-teams: computer scientists

- Avalon [Inria Grenoble - Rhône-Alpes]
  Large algorithms and software architectures for service oriented platforms
- Moais [Inria Grenoble - Rhône-Alpes]
  Programming and scheduling design for applications in interactive simulation
- Roma [Inria Grenoble - Rhône-Alpes]
  Resource optimization: models, algorithms, and scheduling
- Runtime [Inria Bordeaux - Sud-Ouest]
  Efficient runtime systems for parallel architectures

C2S@Exa Inria Project Lab
Core project-teams: numerical mathematicians

- Castor [Inria Sophia Antipolis - Méditerranée]
  Control, analysis and Simulations for tokamak research
- Pomdapi [Inria Paris - Rocquencourt]
  Environmental modeling, optimization and programming models

Context and motivations

C2S@Exa Inria Project Lab - Driving applications and external partners

- Nuclear waste management
  ANDRA (National Radioactive Waste Management Agency)
  Reactive transport of radionuclide in porous media (TRACES software)
- Nuclear fusion
  CEA-IRFM (Research Institute on Magnetic Fusion)
  Numerical study of edge localized modes and disruptions (JOREK software)
  Simulation of plasma turbulence (GYSELA software)

C2S@Exa Inria Project Lab - Satellite projects

- DEEP-ER (Dynamic Exascale Entry Platform - Extended Reach)
  EU FP7 (Exascale Programme) project
- EXA2CT (Exascale Algorithms and Advanced Computational Techniques)
  EU FP7 (Exascale Programme) project
- TECSER (Novel high performance numerical solution techniques for RCS computations)
  ANR ASTRID 2013 project
- HOSCAR (High performance cOmputing and SCientific dAta management dRiven by highly demanding applications)
  CNPq-Inria (International Relation Department) Brazil-France project
  Brazilian coordinator: LNCC

Outline

Numerical schemes for PDEs

Problem characteristics abd algorithmic challenges

- Future exascale supercomputers will enable the numerical treatment of more challenging problems involving, on one hand, discretized models with higher spatial resolution and, on the other hand, more complex physical models possibly encompassing multiple space and time scales
- However, the simulation of such complex physical phenomena will require very accurate and efficient numerical schemes, that will ideally be able to automatically switch between arbitrary high order accuracy in regions where the problem solution is smooth, and low order accuracy combined to local adaptivity of the discretization mesh in less regular regions
- From the algorithmic point of view, the implementation of the proposed numerical schemes will have to be optimized for maximizing both the single core computational performance and the scalability in view of exploiting massive parallelism

Numerical schemes for PDEs

Application context: computational electromagnetics DEEP-ER FP7 project

- Development of a flexible and efficient finite element simulation tool adapted to hybrid MIMD-SIMD parallel systems for the study of 3D ElectroMagnetic (EM) wave propagation problems in complex domains and heterogeneous media
- Application to the numerical modeling of human exposure to EM fields
- Numerical ingredients
  - Unstructured meshes (tetrahedra in 3D)
  - Discontinuous Galerkin Time-Domain method with polynomial interpolation (DGTD-$\mathbb{P}_p$ method)
    - Can easily deal with discontinuous coefficients and solutions
    - Can handle unstructured, non-conforming meshes
    - Yield local finite element mass matrices
    - High order accurate methods with compact stencils
    - Naturally lead to discretization (h-) and interpolation order (p-) adaptivity
    - Amenable to efficient parallelization
  - Explicit time stepping

Numerical schemes for PDEs

Application context: computational electromagnetics
SC'10 - Hybrid CPU-GPU computing

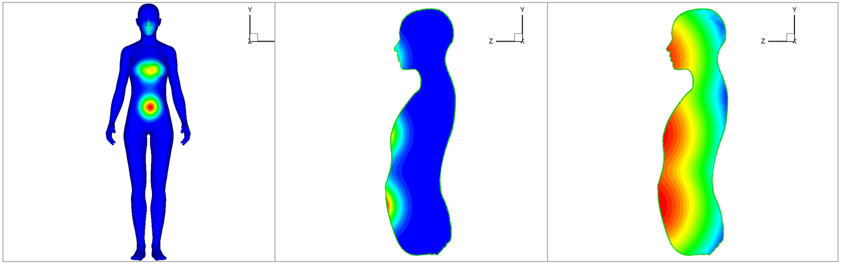
- HPC resource made available by GENCI (allocation 2010-t2010065004)
- Hybrid CPU-GPU Bull cluster of the CCRT
- 1068 Intel CPU nodes with two quad-core Intel Xeon X5570 Nehalem processors operating at 2.93 GHz each
- 48 Teslas S1070 GPU systems with four GT200 GPUs and two PCI Express-2 buses each
- Network is a non-blocking, symmetric, full duplex Voltaire InfiniBand double data rate organized as a fat tree
- Hybrid MPI-CUDA programming model
- Simulations are performed in single precision arithmetic

Numerical schemes for PDEs

Application context: computational electromagnetics
SC'10 - Hybrid CPU-GPU computing

- Mesh: # elements = 5,536,852
- Total # dof is 132,884,448 (DGTD-$\mathbb{P}_1$ method) and 332,211,120 (DGTD-$\mathbb{P}_2$ method)
- Time on 64 CPU cores for the DGTD-$\mathbb{P}_1$ method: 7 h 10 mn

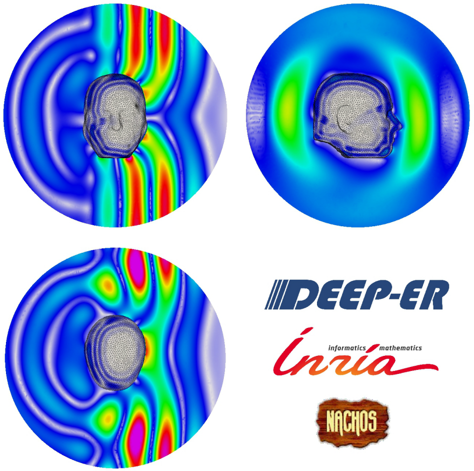
# GPU	DGTD-$\mathbb{P}_1$			DGTD-$\mathbb{P}_2$		
	Time	GFlops	Speedup	Time	GFlops	Speedup
64	12 mn	2762	-	59 mn	4525	-
128	7 mn	4643	1.7	30 mn	8865	1.95



Numerical schemes for PDEs

Application context: computational electromagnetics DEEP-ER FP7 project

- Hybrid MPI-OpenMP programming model
- Parallel I/O SIONlib library
- OmpSs runtime

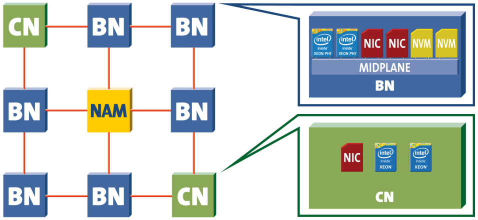


Numerical schemes for PDEs

Application context: computational electromagnetics DEEP-ER FP7 project

In the DEEP project, an innovative architecture for heterogeneous HPC systems has been developed based on the combination of a standard HPC Cluster and a tightly connected HPC Booster built of manycore processors

DEEP-ER will use second-generation Intel Xeon Phi manycore CPUs that boot without the help of an attached Intel Xeon processor for the Booster part



Numerical schemes for PDEs

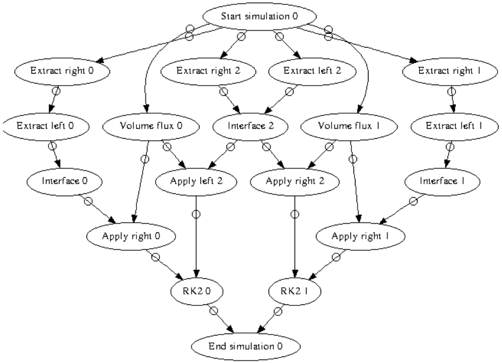
Application context: computational plasma dynamics

- CLAC (Conservation Laws Approximation on manyCores) is a C++ library developed by Inria Nancy - Grand-Est (Tonus project-team) and AxesSim (SME in Strasbourg)
- Discontinus Galerkin methods for systems of conservations laws
- Abstract physical model: the user provides the numerical flux and the source term (Maxwell, Vlasov-Maxwell, MHD, Euler, Navier-Stokes, etc.)
- Actually used for electromagnetics/plasms dynamics simulations
- Hybrid MPI-OpenCL programming model
- Subdomain decomposition: each domain is associated to a MPI process, and each MPI process is associated to an OpenCL device (CPU or GPU)
- Zone decomposition: each subdomain is split into volume zones and interface zones
- A zone possess identical elements (same order, same geometry, same physical model)
- A computation kernel is compiled for each zone (for avoiding branch tests)
- Fine grain parallelization based on a task dependency graph

Numerical schemes for PDEs

Task-based programming model

- Well adapted to (massive) fine grain parallelism
- A set of inter-dependent tasks can be modelled as a Directed Acyclic Graph (DAG)
- The DAG is traversed breadth-first by a task scheduler assigning the tasks to the cores
- Software infrastructure for such task-based parallelism: QUARK, Cilk and StarPU
- Programming models: OpenCL, OpenM 3.0 and Intel's TBB

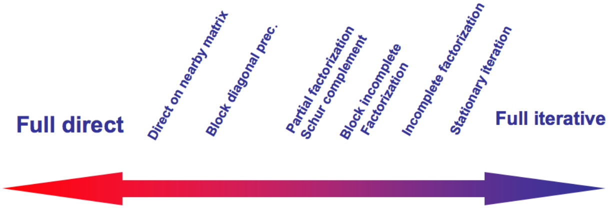


CLAC task graph
By courtesy of P. Helluy, Tonus project-team

Outline

Scalable linear solvers

Goal: solving $\mathcal{A}x = b$, where $\mathcal{A}$ is sparse



Usual trades off

Direct

- Robust/accurate for general problems
- BLAS3 based implementations
- Memory/CPU prohibitive for large 3D problems
- Limited weak scalability

Iterative

- Problem dependent efficiency/accuracy
- Sparse computational kernels
- Less memory requirements and possibly faster
- Possible high weak scalability

Scalable linear solvers

Parallel sparse direct solver - PaStiX main features

Numerical features

- LL^T, LDL^T, LU factorization with supernodal implementation
- Static pivoting + refinement (CG/GMRES)
- 1D/2D block distribution + full BLAS3
- Simple/double precision + float/complex operations

Implementation features

- MPI/Threads implementation (SMP/Cluster/Multicore/NUMA)
- Dynamic scheduling inside SMP nodes (static mapping)
- Support external ordering library (PT-Scotch/METIS)

Additional information

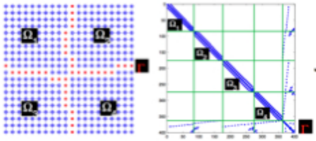
- Multiple RHS (direct factorization)
- Incomplete factorization with ILU(k) preconditionner
- Schur complement computation
- Out-of Core implementation (in SMP mode only)

Parallel hybrid iterative/direct sparse solver - MaPHyS

- Algebraic additive Schwarz preconditioners for the Schurs complement system
- Uses PaStiX as the (parallel) subdomain solver

Algebraic sparse linear solvers - HiePACS project-team

Partition the adjacency graph of the sparse matrix (Scotch)



$$\mathcal{A}^{(i)} = \begin{pmatrix} \mathcal{A}_{\mathcal{I}_i\mathcal{I}_i} & \mathcal{A}_{\mathcal{I}_i\Gamma_i} \\ \mathcal{A}_{\Gamma_i\mathcal{I}_i} & \mathcal{A}^{(i)}_{\Gamma\Gamma} \end{pmatrix}$$

Local calculation of Schur complements (MUMPS, PaStiX) and precondtioning operator (Magma, MUMPS, PaStiX)

$$\mathcal{S}^{(i)} = \mathcal{A}^{(i)}_{\Gamma\Gamma} - \mathcal{A}_{\Gamma_i\mathcal{I}_i}\mathcal{A}^{-1}_{\mathcal{I}_i\mathcal{I}_i}\mathcal{A}_{\mathcal{I}_i\Gamma_i} \qquad \mathcal{M} = \sum_{i=1}^{N} \mathcal{R}^T_{\Gamma_i}(\bar{\mathcal{S}}^{(i)})^{-1}\mathcal{R}_{\Gamma_i}$$

Solving $Ax = b$ with A a large sparse matrix

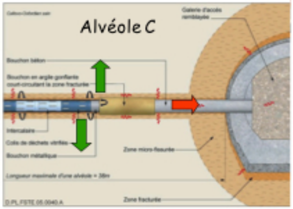
Scalable linear solvers

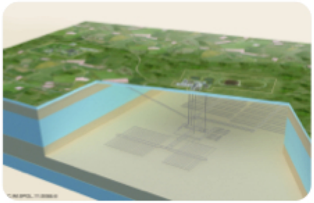
Parallel hybrid iterative/direct sparse solver - MaPHyS

Application context: nuclear waste management - ANDRA

- Distributed memory parallelization of the TRACES software
- Improvement of numerical linear algebra capabilities

Reactive transport of radionuclide in porous media
Modeling context of radioactive wave management


Alvéole C
Galerie d'accès remblayée
Bouchon béton
Bouchon en argile gonflante court-circuitant la zone fracturée
Intercalaire
Colis de déchets vitrifiés
Bouchon métallique
Zone micro-fissurée
Longueur maximale d'une alvéole = 38m
Zone fracturée
D.PL.FSTE.05.0040.A



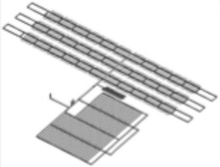
Hydraulic and migration of solute transfer from waste packages to the geological medium

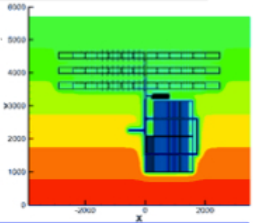
Scalable linear solvers

Parallel hybrid iterative/direct sparse solver - MaPHyS

Hydraulic and migration of solute transfer from waste packages to the geological medium

Up-to-date calculation
- ✓ 10 millions cells in structured mesh including all disposal cells and zone
- ✓ Many simplifications in the modeling (e.g. only one clay layer)
- ✓ Structured (Cartesian) grid approach
- ✓ High computing time (limit for sensitivity analysis)





C2S@Exa objective: improve scalability up to and beyond 1k cores
Requires to deal with applied mathematics (numerical schemes and solvers) and computer science issues (parallel programming model and runtime system for exploiting heterogeneous architectures)

Hexahedral mesh with 428,400 elements (i.e. coarse problem), PlaFRIM cluster

# procs	Total time Hypre (PCG/AMG)	Total time MaPHyS
32	1.2 s	6.6 s
64	1.9 s	3.1 s
128	4.7 s ($t_{32/128}=0.25$)	2.6 s ($t_{32/128}=2.5$)

Application context: EM simulation - Airbus Group Innovations

- Radar stealthiness (RCS computation)
- Electromagnetic compatibility
- Antenna design and sitting

Boundary Element Method

- Very accurate
- Surfacic mesh, no space truncation
- Frequency- and time-domain
- More than 20 years of experience: ASERIS BE, 2D, 3D, frequency- and time-domain

Remarks

- Meshing criteria: edge length $\simeq \lambda/10$
- $A \in \mathbb{C}^{N\times N}$, dense, symmetric or not, not positive and ill-conditionned
- Degrees of freedom : $N \propto 150\frac{S}{\lambda^2}$ and in practice, $N = 10^5 - 10^8$
- Computation of A is a costly operation
  - Numerical and analytical double integrals
  - Green kernel costly to compute
  - $N = 10^6 \Rightarrow 16$ TB

Dense linear systems

- Direct : factorize (LU ou LDL^T) A, then backward-forwars substitution

Complexity	$O(N^3)$ for the factorization, $O(N^2 n_{rhs})$ for the substitutions
Advantages	Accurate, many RHS
Issues	Computation time, storage

- Iterative + FMM: FMM $\sim Ax$

Complexity	$O(N \log_2(N) n_{rhs} n_{iter})$
Avantages	Fast !
Issues	Very hard to deploy, not generic, convergence issues, RHS count

- $\mathcal{H}$-matrix : fast direct solver

$\mathcal{H}$-matrix

Broadly speaking,

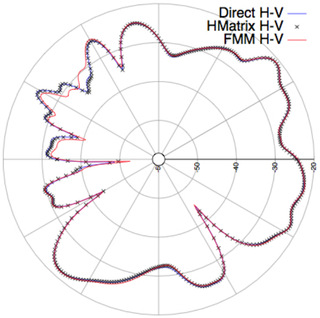
- Adaptative data-sparse representation of dense matrices
- Hierarchical, adaptative and approximate representation
- Broad range of operations on these compressed matrices (`GEMV`, `AXPY`, `GEMM`, *etc.*)

Two ingredients

- Data-sparse representation of dense matrices
- Approximate operations (e.g. matrix-matrix multiplication) that preserve this representation

A319neo: VHF Antenna

- VHF antenna on an A319neo
- 127 MHz
- $\lambda/20$ mesh
- 290,163 d.d.l, 1 RHS
- By courtesy of B. Lizé and G. Sylvand, Airbus Group Innovations



Scalable linear solvers

Parallel fast direct solver for BEM - $\mathcal{H}$-matrix

A319neo: VHF Antenna

- VHF antenna on an A319neo
- 127 MHz
- $\lambda/20$ mesh
- 290,163 d.d.l, 1 RHS
- By courtesy of G. Sylvand, Airbus Group Innovations

Solver	Total CPU time (s)	Ratio (/direct)
Direct	5,850,120	-
FMM	7728	757
$\mathcal{H}$-matrix	8608	679

Total CPU time = matrix assembly + factorization + solve

Remarks

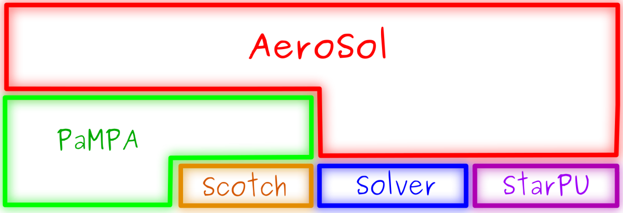
- Solve time is < 1 s with $\mathcal{H}$-matrix
- $\mathcal{H}$-matrix solver is now more accurate than the FMM solver

Outline

Mesh management tools

PaMPA

- AEROSOL is a high order finite element library for the parallel simulation of compressible flows in turbomachines
- PaMPA (Parallel Mesh Partitioning and Adaptation) is a middleware library managing the parallel repartitioning and remeshing of unstructured meshes modeled as interconnected valuated entities

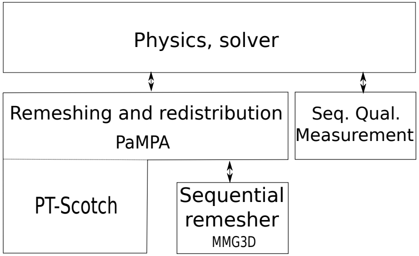


By courtesy of V. Perrier, Cagire team

Mesh management tools

PaMPA

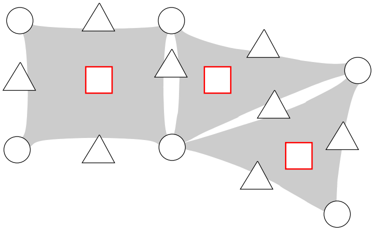
- The user can focus on his/her core business
  - Solver
  - Sequential remesher
- Relies on different external tools
  - MMG3: anisotropic tetrahedral remesher/moving mesh generation
  - PT-Scotch: parallel graph partitioning, static mapping and clustering, hypergraph partitioning sparse matrix block ordering
  - StarPU: task programming library for hybrid architectures



Mesh management tools

PaMPA: enriched graph

- Mesh
  - Element
  - Node
  - Edge (internal and boundary)
- PaMPA Mesh
  - Vertex
  - Relation
  - Entity
  - Sub-entity
  - Enriched graph



- Give the enriched graph to PaMPA
- Then PaMPA
  - Redistributes the mesh
  - Computes the overlap
  - Allocates the communications buffers
  - Performs communications
  - Gives accees to iterators compliant with FE way of thinking
  - Possibly adapts the mesh

i

Mesh management tools

PaMPA (Parallel Mesh Partitioning and Adaptation)

	PaMPA-MMG3D4 on 240 processors
Initial number of elements	27 044 943
Final number of elements	609 671 387
Elapsed time	00h34m59s
Elapsed time × number of PEs	139h56m
Smallest edge length	0.2911
Largest edge length	8.3451
Worst element quality	335.7041
% element quality between 1 and 2	98.92%
% edge length between 0.71 and 1.41	97.20%

By courtesy of C. Lachat and F. Pellegrini, Bacchus project-team

Outline

Virtualization of heterogeneous resources

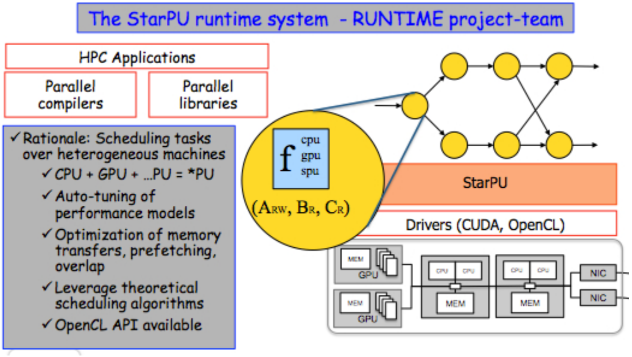
Context and objectives

- The increasing availability of different kinds of processing resources in heterogeneous system architectures has propelled an interest towards systems able to dynamically and autonomously adapt how computing resources are exploited to optimize a given goal
- The goal of resource virtualization is to create a layer of abstraction between actual physical hardware providing resources (CPu, GPU, RAM, network, etc.) and the logical or semantic activities which consume those resources
- A runtime system allows for deciding at runtime the mapping of computation kernels on the appropriate type of resource, based on the current system context and requirements

Virtualization of heterogeneous resources

StarPU execution model: task scheduling

- Mapping the graph of tasks (DAG) on the hardware
  - Allocating computing resources
  - Enforcing dependency constraints
  - Handling data transfers
- Adaptiveness
  - A single DAG enables multiple schedulings
  - A single DAG can be mapped on multiple platforms

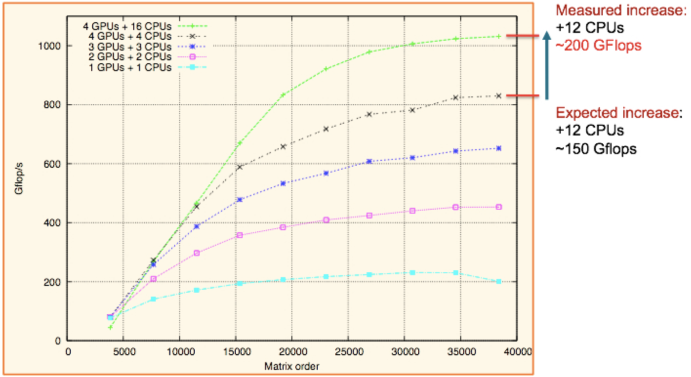


✓ Rationale: Scheduling tasks over heterogeneous machines
- ✓ CPU + GPU + ...PU = *PU
- ✓ Auto-tuning of performance models
- ✓ Optimization of memory transfers, prefetching, overlap
- ✓ Leverage theoretical scheduling algorithms
- ✓ OpenCL API available

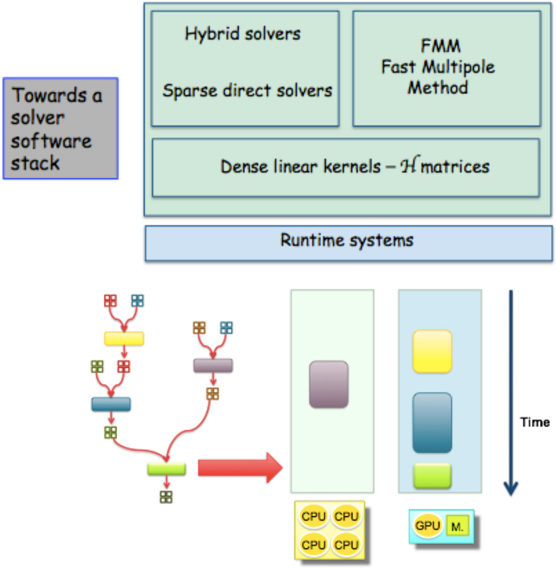
Virtualization of heterogeneous resources

Showcase with the MAGMA linear algebra library

- QR decomposition on 16 CPUs (AMD) + 4 GPUs (C1060)
- By courtesy of Emmanuel Agullo and Mathieu Faverge (Hiepacs project-team) Olivier Aumage and Samuel Thibault (Runtime project-team)



Virtualization of heterogeneous resources



Outline

Closure

Thank you for your attention !

Particular thanks to:

- Bacchus project-team (Inria Bordeaux - Sud-Ouest)
  - Cédric Lachat and François Pellegrini
- Cagire teamn (Inria Bordeaux - Sud-Ouest)
  - Vincent Perrier
- Hiepacs project-team (Inria Bordeaux - Sud-Ouest)
  - Emmanuel Agullo, Mathieu Faverge, Luc Giraud, Pierre Ramet and Jean Roman
- Nachos project-team (Inria Sophia Antipolis - Méditerranée)
  - Tristan Cabel, Rachid El Khaoulani and Raphaël Léger
- Runtime project-team (Inria Bordeaux - Sud-Ouest)
  - Olivier Aumage, Raymond Namyst and Samuel Thibault
- Tonus project-team (Inria Nancy - Grand-Est)
  - Philippe Helluy and Thomas Strub (AxesSim)
- Benoît Lizé and Guillaume Sylvand (Airbus Group Innovations)
- Laurent Loth (ANDRA)