EXA2CT

Intel's European Research and Innovation Conference

ENES HPC workshop, Hamburg,

March 17-19th, 2014

Marie-Christine.sawley@intel.com

SEVENTH FRAMEWORK PROGRAMME

e-infrastructure

EXA2CT

CREATING BREAKTHROUGH *INNOVATIVE ALGORITHMS* AND NEW *PROGRAMMING MODELS*, PAVING THE WAY TO EXASCALE FOR *LEADERSHIP CLASS CODES*.

The EXA2CT project brings together experts at the cutting edge of the development of solvers, related algorithmic techniques, and HPC software architects for programming models and communication.

It will take a revolutionary approach to exascale solvers and programming models, rather than the incremental approach of other projects.

We will produce modular open source proto-applications that demonstrate the algorithms and programming techniques developed in the project, to help boot-strap the creation of genuine exascale codes.

Strategy

Start from existing, leadership class applications running on Tier-0 systems today and extract knowledge about them to create the appropriate Proto-Applications.

Enhanced Numerical Algorithms

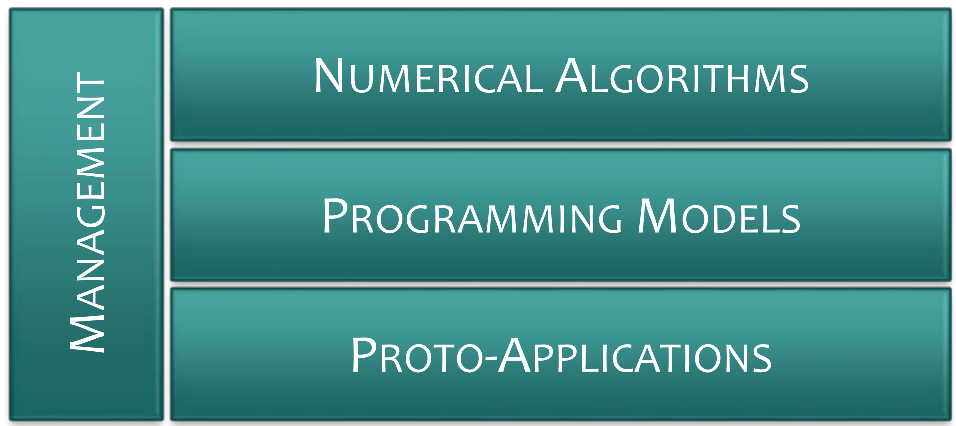
Scalable, Pipelined, Robust Numerical Solvers

- that scale up to exascale performance
- that survive hardware failures

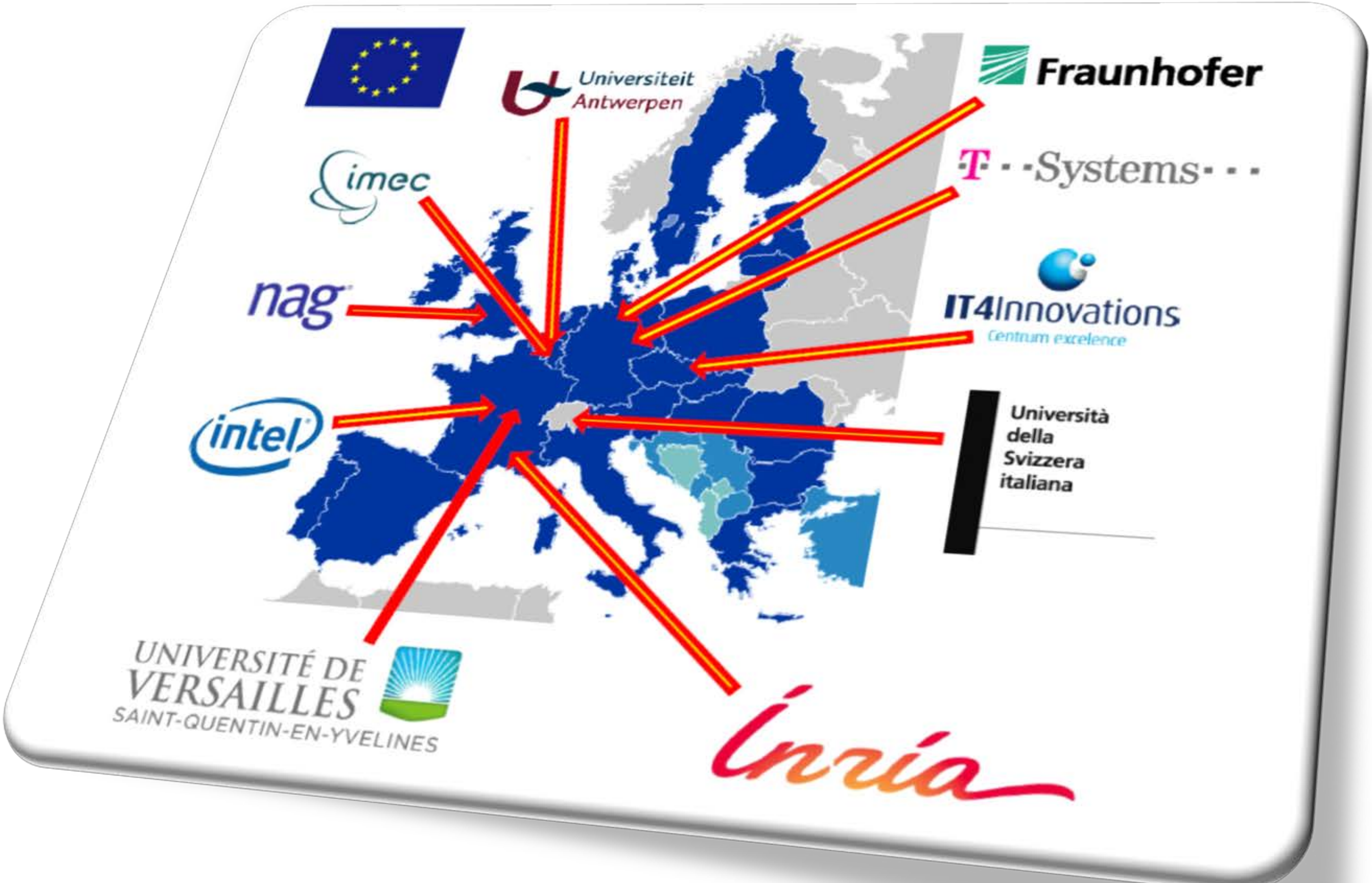
Enhanced Programming Models

GPI/GASPI, Shark,
Irregular Stencil Code, PATUS

- communication model
- to deal with platform heterogeneity
- to deal with resilience



Partners


Universiteit Antwerpen
Fraunhofer
imec
T-Systems
nag
IT4Innovations
Centrum excelence
intel
Università della Svizzera italiana
UNIVERSITÉ DE VERSAILLES
SAINT-QUENTIN-EN-YVELINES
Inria

Numerical Algorithms

Current linear solvers will not scale well to exascale, because of

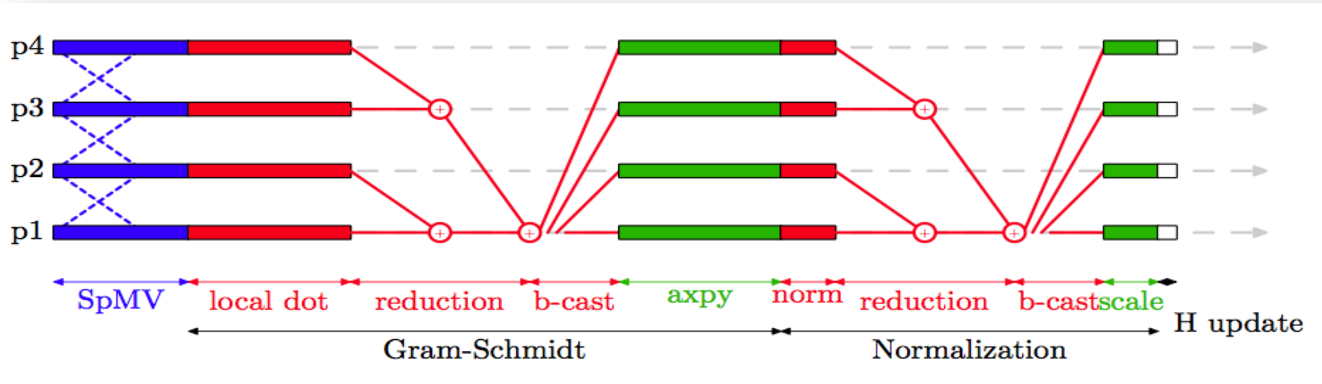
- Global communication
- Bandwidth : Insufficient arithmetic intensity
- Lack of Resilience

Beyond state of the art

- Extend communication avoiding/hiding
  - Preconditioning, aug'd/blocked/deflated
- Exploit arithmetic intensity using *tasking models* as backend
- Resilience through interpolation/ABFT
- Application to FETI methods

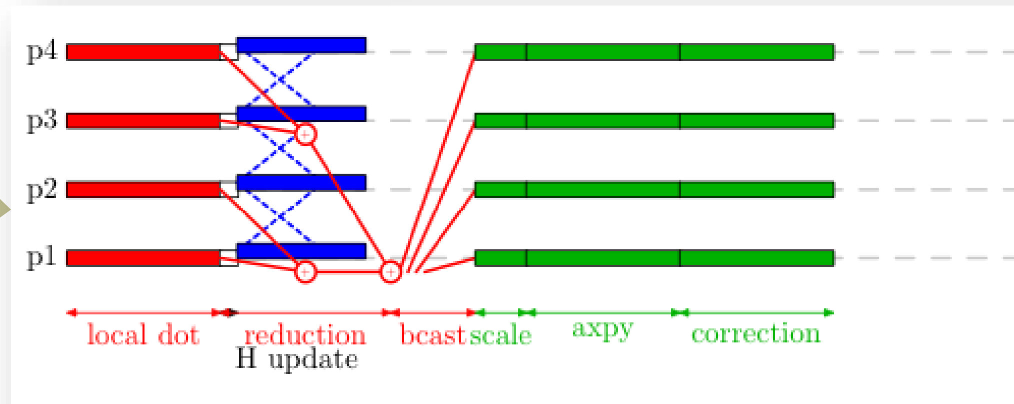
Numerical Algorithms, an example

P. GHYSELS, T.J. ASHBY, K. MEERBERGEN AND W. VANROOSE



Pipelined GMRES overlaps dot-product global communication latency with SpMV

Next generation solvers can hide these latencies and boost performance of a wide range applications in complex systems.

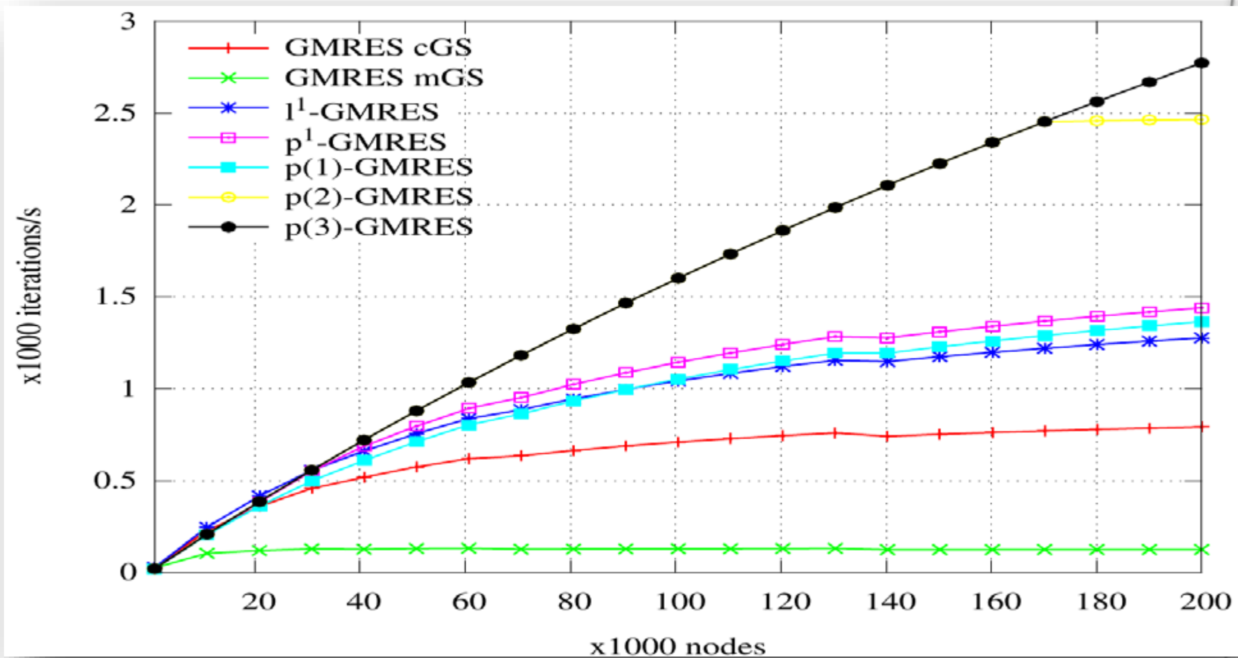


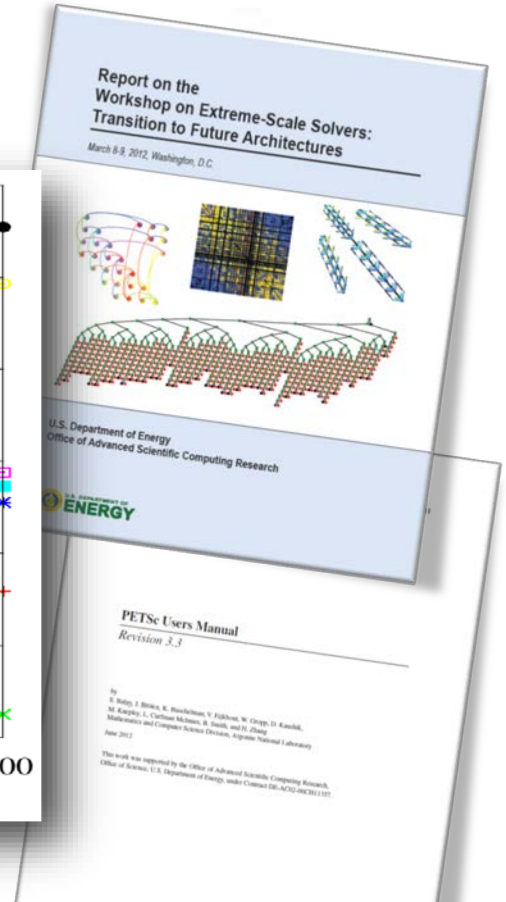
Universiteit Antwerpen
IT4Innovations Centrum excelence
inria
Università della Svizzera italiana

Numerical Algorithms, an example

P. GHYSELS, T.J. ASHBY, K. MEERBERGEN AND W. VANROOSE

Pipelined GMRES shows significant speedups compared to standard GMRES for strong scaling experiments




Report on the Workshop on Extreme-Scale Solvers: Transition to Future Architectures

March 8-9, 2012, Washington, D.C.

U.S. Department of Energy
Office of Advanced Scientific Computing Research

ENERGY

PETSc Users Manual
Revision 3.3

Universiteit Antwerpen

IT4Innovations
Centrum excelence

Inria

Università della Svizzera italiana

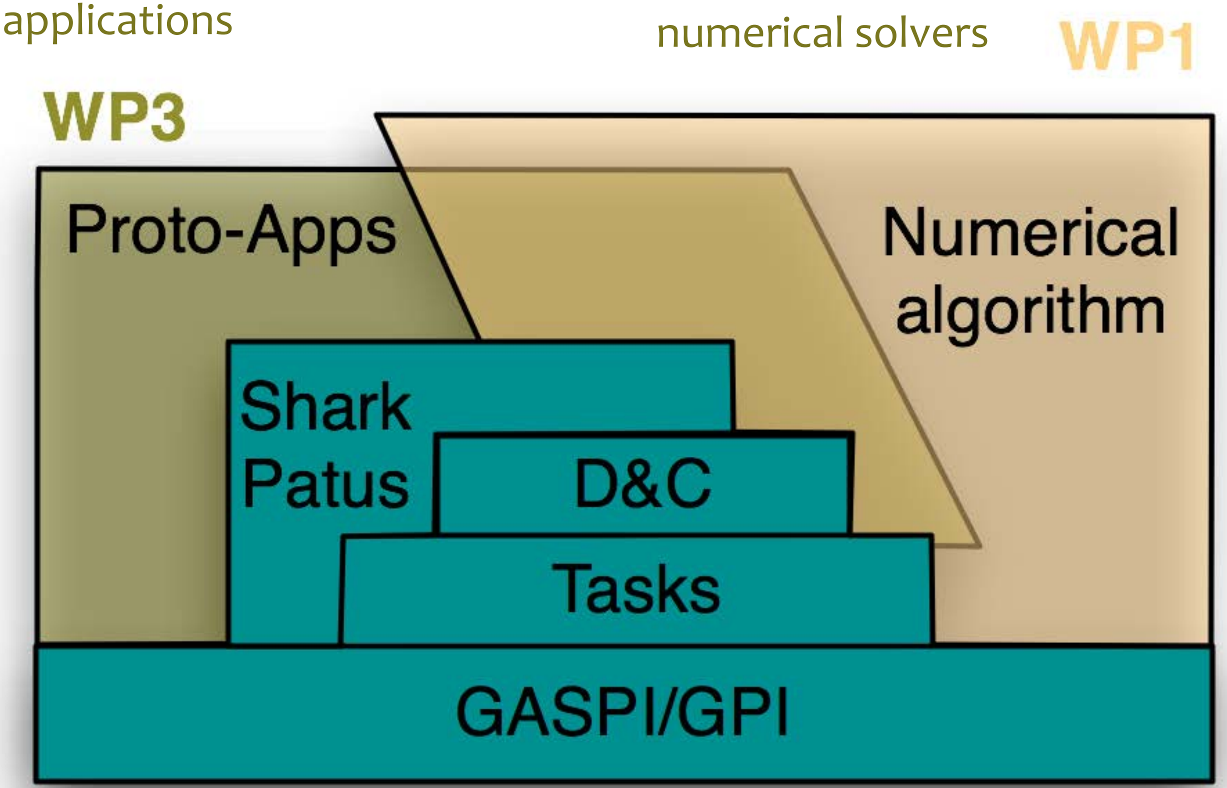
WP1 in a nutshell

- Communication hiding, pipelining global reduction
  - Wim Van Rose, University of Antwerpen
- Communication avoiding, increasing arithmetic intensity using matrix power kernel based on stencil compiler (PATUS)
  - Olaf Schenk, USI
- Resilience by interpolating missing data
  - Luc Giraud and Laura Grigori, INRIA
- FETI benchmarks with coarse operators
  - Tomas Kozubek, IT4Innovations
- Communication avoiding at the level of the preconditioner

WP2 Organization

Proto-Applications
Extract representative proto-applications from HPC applications

Enhance Numerical Algorithms
Scalable and robust numerical solvers



Enhance Programming Models
Improve efficiency, robustness and scalability of GPI/GASPI , Shark , PATUS and leverage task parallelism

WP2-Programming Models

Current HPC programming models don't scale very well

- *MPI*, *OpenMP*, Vectorization

  ... too complex
- *PGAS* Languages

  ... are a partial solution

Beyond State of the Art

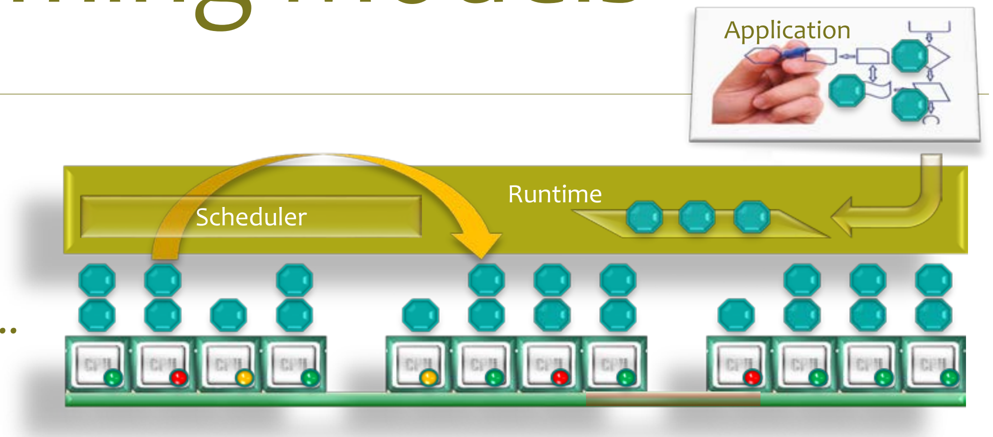
- Extend PGAS with Tasking Models

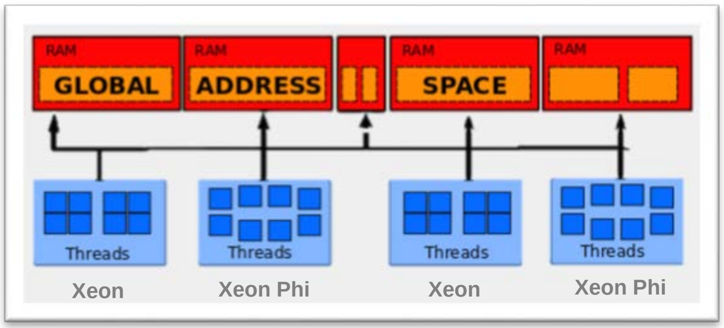
  ... starting from *GASPI*
- High Level Libs : *Patus*, *Shark*
- Resilience : Software that survives failures

WP2-Programming Models

Tasks

Formulate your program in terms of logical tasks, instead of threads
Threading Building Blocks , Cilk™ Plus, ...





GASPI – a *PGAS API*

- is not a new language or a language extension, but complements existing languages *(library approach ~ MPI)*
- Support for resilience

  e.g. time-out mechanisms for all non-local procedures

GASPI + Tasks ➔ extreme scalability

Opportunities : Heterogeneous execution platforms for tasks, task/data migration, task/data resilience, ...

UNIVERSITÉ DE VERSAILLES SAINT-QUENTIN-EN-YVELINES · imec · Fraunhofer · T-Systems · Università della Svizzera italiana

EXA2CT-WP2 Contact

GASPI/GPI

- Christian Simindinger [1]
- Valeria Bartsch [2]
- Mirko Rahn [2]

D&C and Task programming

- Eric Petit [3] (WP leader)
- Christian Simmendinger [1]

Resilience

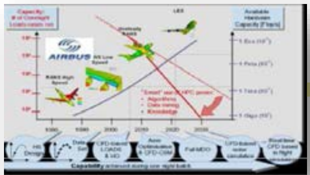
- Valeria Bartsch [2]
- Mirko Rahn [2]

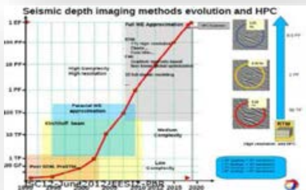
Shark and Patus

- Roel Wuyts [4]
- Olaf Shenk [5]

[1] @t-system-sfr.com
[2] @itwm.fraunhofer.de
[3] @prism.uvsq.fr
[4] @imec.be

Proto-Apps

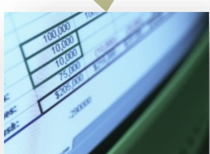


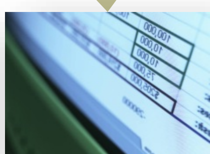


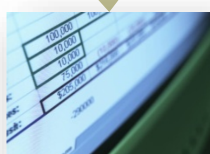
Sometimes considered identical to
Mini-apps, *Proxy-Apps*, *Skeleton Apps*

However proto-apps are not reduced version of real-world applications, they are specifically designed to be :

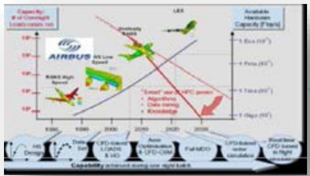
- demonstrating high scalability for re-factored applications
- encapsulating essential characteristics:
  - New exascale algorithms
  - Advanced parallel implementations

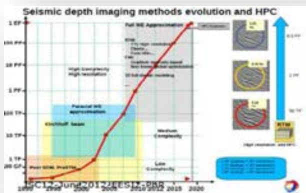






Proto-Apps



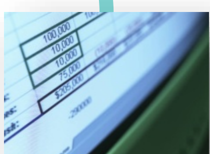


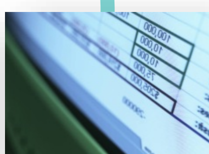
Scientific & Industrial Board

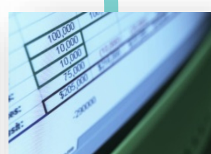
- Define the selection of the proto-apps
  - Have a say in the definition of the targets
    - Are preferred partners in the take-up of the results
      - Will get help in translating the proto-app (back) into the reference applications

Reference Application Areas

- Computational Fluid Dynamics for Aerospace, combustion
  - Structural Mechanics Analysis
    - Geophysics
      - Material sciences
        - And others …





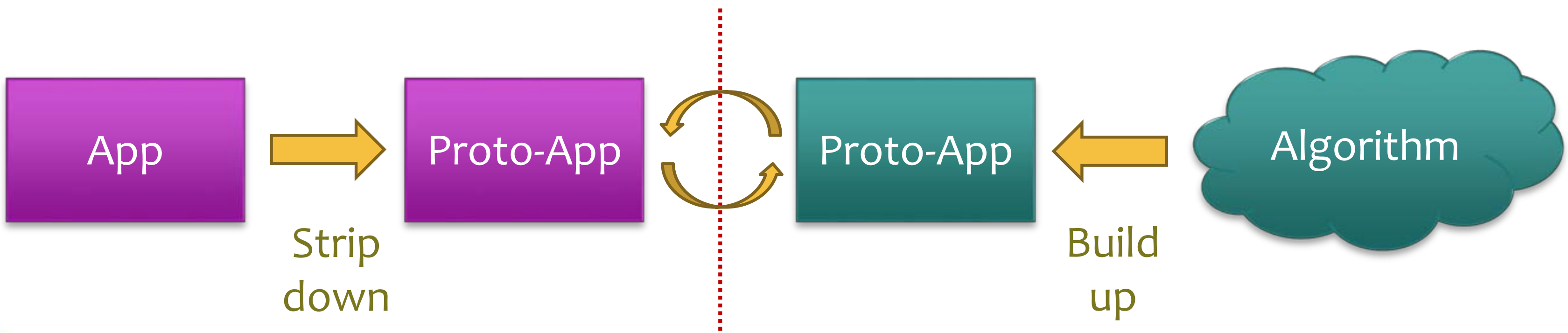


Proto-app development questions

- Discussion with WP1 & WP2: what proto-app description?
  - Numerics-oriented: actually functional (computes)
  - Dataflow-like description (tasks + dependencies)
  - Communication-oriented (profiling-parameterized « sleeps »)

Abstraction

Coupling to implementation



Presently 8 reference applications

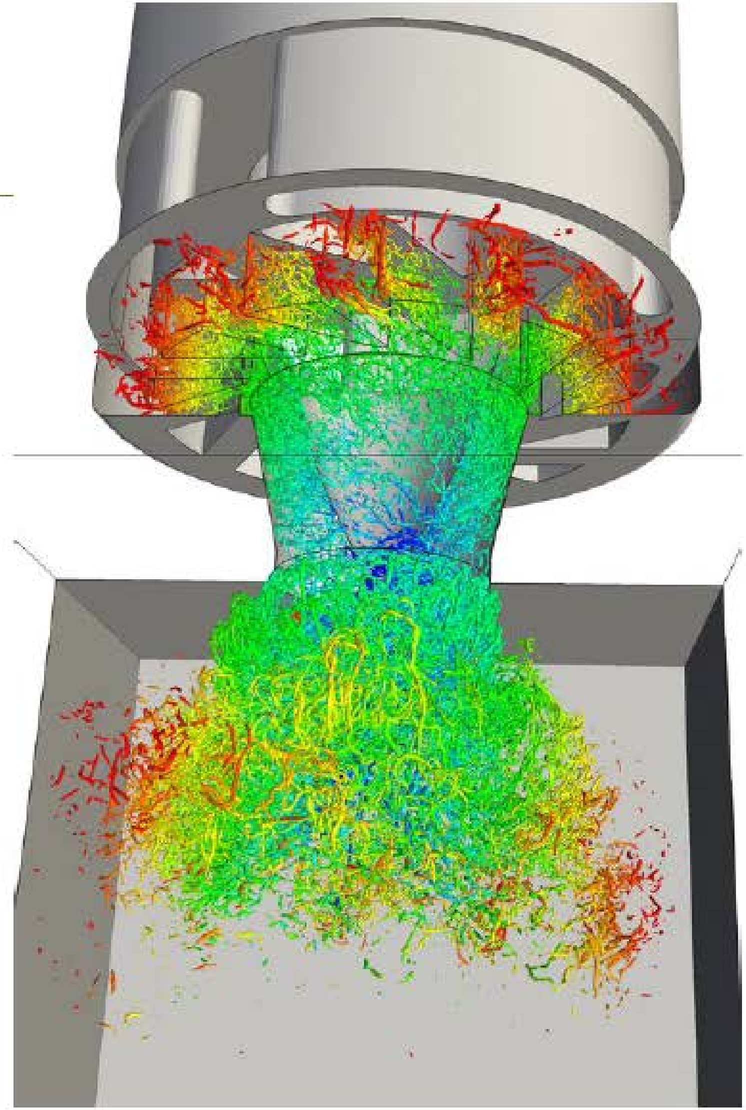
Code	Domain
YALES2	**CFD combustion**
TAU	**CFD**
SPECFEM3D	**Seismology**
elsA	**CFD**
MUPHY	**Fields/particles multiphysics**
Aeronautics code	**CFD**
GYSELA	Fusion
Quantum Espresso	Quantum chemistry
Climate	**Climate science**

Reference applications features

- Strong scaling is a popular scalability model
  - Time to solution drives most industrial codes
- Some use cases for weak scaling
  - Multi-scale physics (YALES2)
  - Higher resolution (Climate, muphy)
  - More complex physics (elsA)
- Some very precise answers about target problem sizes
  - YALES2
  - TAU
- Most science is done on 1k-10k cores
- I/O is a recurring problem
- Fault tolerance: not a lot of feedback beyond checkpoint/restart

Test Cases

- Getting suitable test cases is crucially important to us
  - Should show good scalability, up to a point
  - Should also show performance issues
  - Should be representative of genuine application usage



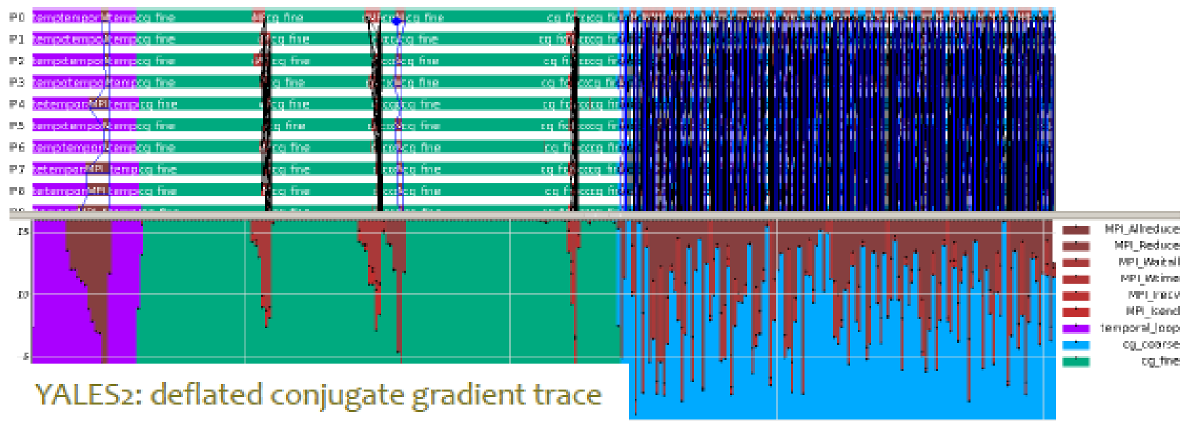
YALES2, PRECCINSTA 2.6B test case (V. Moureau)

YALES2 status update

- Questionnaire feedback received
- Two main development cases identified:
  - CFD linear solvers
    - Good scaling up to 16k cores
    - Want to reduce cell count/core, from 100k to 10k: strong scaling
    - Main identified issue: deflated preconditioned CG
  - Reactive code (combustion)
    - Large computational imbalance across MPI domains
    - MPI task scheduler (dynamic load balancing) scaling up to 1k cores
    - Performance issues beyond that

Application Profiling

- Understanding both node-level performance and parallel scalability
  - In-depth, **quantitative** knowledge of application runtime behaviour is vital for building proto-apps
- Detailed analysis greatly facilitated by access to relevant source code
  - Allows targeted instrumentation
  - When not possible, extensive work on code owner's part is to be expected



YALES2: deflated conjugate gradient trace

YALES2: deflated CG

- Conjugate gradient on 2 levels
  - Main grid
  - « Deflated » level (typically 200x coarser than main grid)
- Scalability issues
  - Extremely fine-grained MPI comms, esp. at deflated level
  - Global comms. (sychronizing) for norm and dot product computations

WP3 Roadmap

- NAG
  - Mike Dewar and Chris Goodyer
- Intel Exascale Lab, France
  - Omar Awile, Thomas Guillet, Marie-Christine Sawley
- Prepare the list of reference applications
- Profile and help refactoring reference applications
- Prepare the library of protoapplications for Opensource release

Dissemination

- EPCC workshop with all 6 Exascale EU projects
  - GASPI tutorial
- ISC booth and BOF
- TERATEC, July 1 and 2nd

www.exa2ct.eu

EXA2CT

Contacts and Workpackage Leaders

Coordinator

Wilfried Verachtert, Imec
wilfried.verachtert@imec.be

Programming Models

Eric Petit, UVSQ
eric.petit@prism.uvsq.fr

Numerical Algorithms

Wim Vanroose, U Antwerp
wim.vanroose@ua.ac.be

Proto-Applications

Marie-Christine Sawley, Intel
marie-christine.sawley@intel.com