DE LA RECHERCHE À L'INDUSTRIE

PROGRAMMING ENVIRONMENT FOR MULTI-PETASCALE AND EXASCALE SIMULATIONS IN THE FRAME OF LARGE SCALE SCIENTIFIC SIMULATION CODES IN REACTOR PHYSICS

Productive Programming Models for Exascale

C. Calvin – J. Dubois
CEA/DEN

AUGUST 2012

www.cea.fr

OUTLINE

HPC for Reactor Physics Simulation

Development constraints and software lifecycle

Trends in supercomputing and impacts on scientific codes

HPC in deterministic neutronics simulations using APOLLO3®

Some requirements and needs for developing large scale simulation codes at the post petascale and exascale era

HPC FOR REACTOR PHYSICS SIMULATION

APPLICATION FIELDS AND CHALLENGES

In core stay
Pool storage
Fuel transport, refining, production

Dismantling

Geological warehousing

radioactivity $\alpha, \beta, \gamma, n,$

0s **3 years** **50 years** **10^6 years** *time*

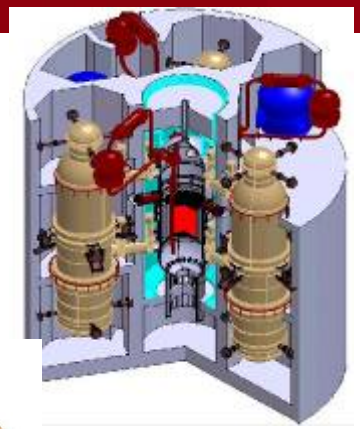
Core physics, criticality, radiation shielding, Instrumentation

. Particles propagation– neutrons, …-
. Isotopic depletion

. Energy range: 0. – 20 MeV

. Domain size: → 30 000 m^3 ≡ 3. 10^{10} cm^3

HPC FOR REACTOR PHYSICS: OBJECTIVES

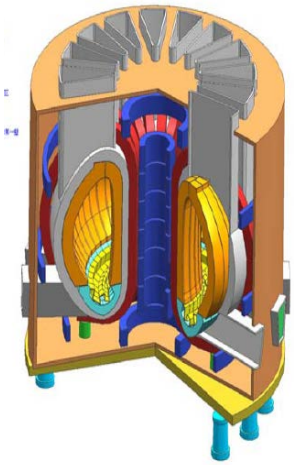


Decrease margin by decreasing uncertainties:

- Parametric calculations (statistic approach)
- Finer physics (CFD + transport / MC)
- More “realistic” physics → multiphysics coupling

Design optimization:

- Finer and more realistic physics
- “Numerical Nuclear Reactor”



Improve safety :

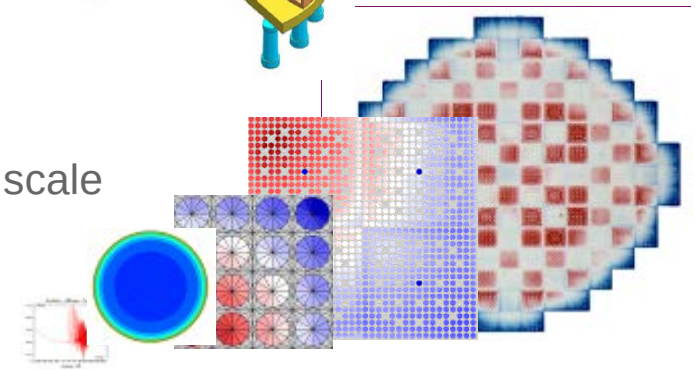
- Criticality and shielding: MC approach

Improve reactor management:

- Embedded calculator (real time)
- Nuclear fuel optimization (neural networks)

Progress in physics:

- Models improvements for a “coarser” physics → multi scale



HPC FOR REACTOR PHYSICS: SIMULATION TARGETS

To provide simulation capabilities in order to:

- To simulate specific situations for safety reasons
  - **Sensibility calculations, Transient simulations**
- Allow fast design of complex reactors
  - **Design simulations**
- Validate and qualify models and calculation routes when experiments are not possible or too expensive
  - **Reference simulations**
- Improve reactor management
  - **Real time simulation**

These simulations have to be as realistic as possible and this implies use of very fine **3D modeling**.

With "controlled" uncertainties

New reactor generation
→ **new approach for design and safety**
→ **new code generation**

DEVELOPMENT CONSTRAINTS AND SOFTWARE LIFECYCLE

APOLLO LIFE STORY

1970's 1980's 1990's 2000 2005 2012 2015

APOLLO1

EDF industrial release

Maintenance

APOLLO2

AP 2.0 1987

AP 2.5 1999

AP 2.7 2004

AP 2.8 2006

Evolutive Maintenance

EDF (GAB)

AREVA (SCIENCE V2)

CRISTAL V1

AREVA (ARCADIA)

AREVA (NRC agreement)

APOLLO3

AP3 1.0 2012

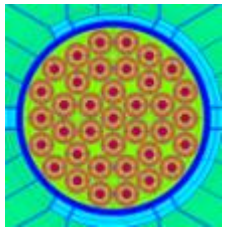
AP3 2.0 2015

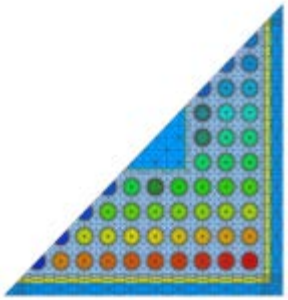
CEA release - FBR

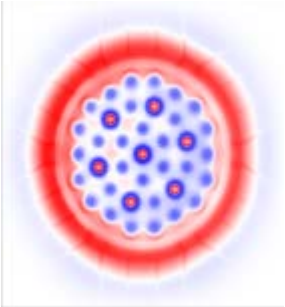
1st pre-release for EDF - AREVA

1st industrial release

Development

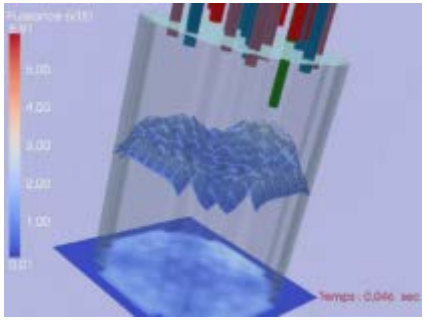




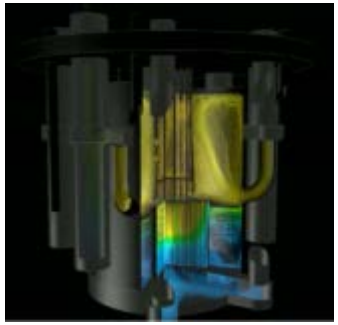


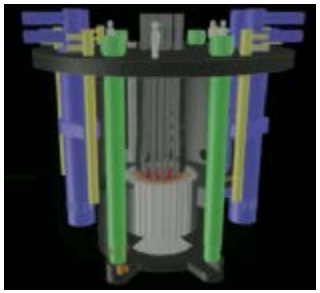
DEVELOPMENT CONSTRAINTS

- Developed codes are used by
  - Industrial partners
  - Regulation authority
  - CEA



- For
  - Safety
  - Design
  - Reference calculations





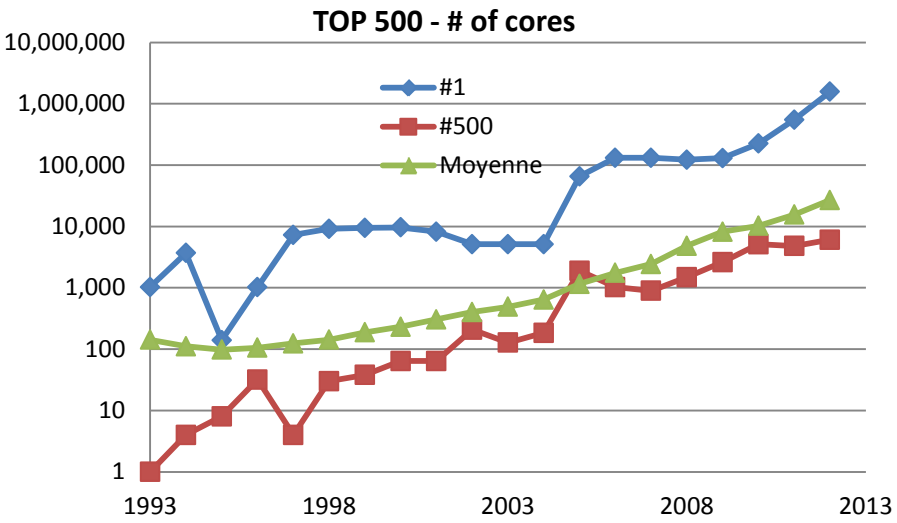
- Imply many constraints:
  - Portability (different architectures, OS, from PC to supercomputers)
  - Robustness
  - Maintainability
  - Performances

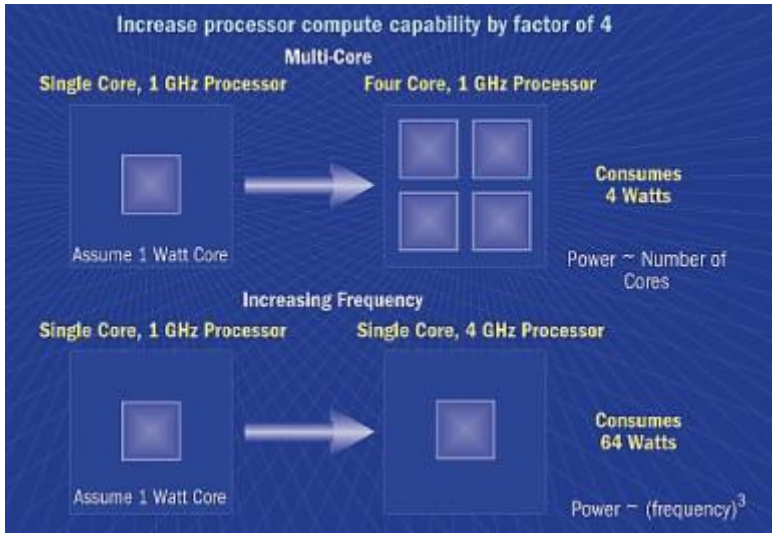
TRENDS IN SUPERCOMPUTING AND IMPACTS ON SCIENTIFIC CODES

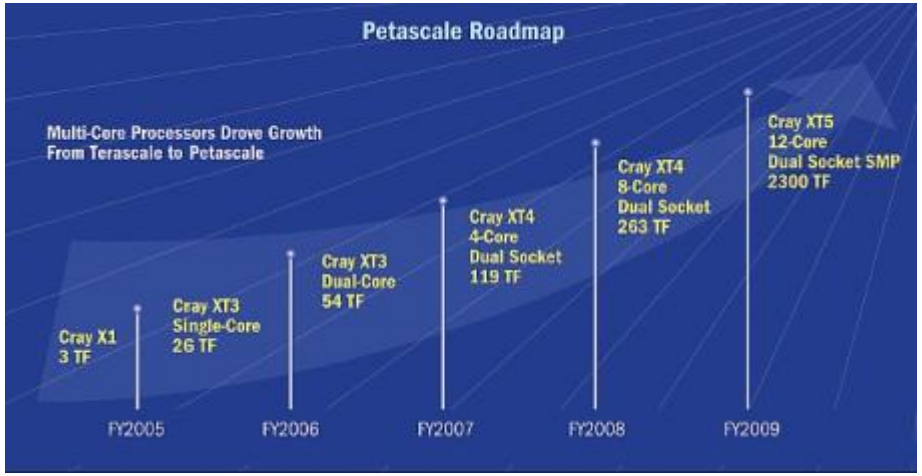
TRENDS IN SUPERCOMPUTING

Petascale and post-petascale

- The degree of parallelism is increasing
- The number of levels of parallelism is increasing
- The number of levels of memory is increasing
- Heterogeneous architectures with accelerators





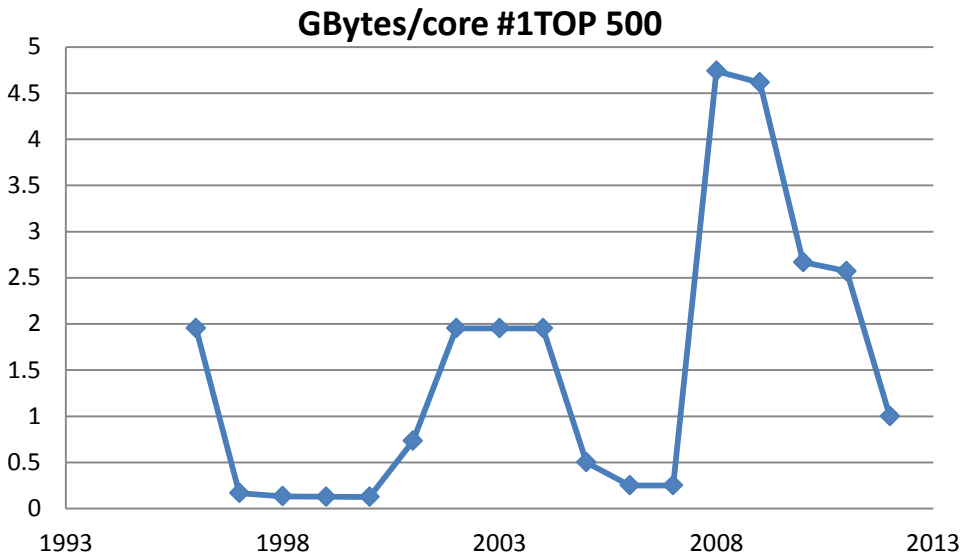


TRENDS IN SUPERCOMPUTING

Systems	2009	2011	2015	2018
System Peak Flops/s	2 Peta	1x	100-200 Peta	50x Exa
System Memory	0.3 PB		5 PB	10x 1(
Node Performance	125 GF		400 GF	5-50x F
Node Memory BW	25 GB/s		100 GB/s	5-10x 20(
Node Concurrency	12		O(100)	30x)00)
Interconnect BW	1.5 GB/s		25 GB/s	5x 50 GB
System Size (Nodes)	18,700		500,000	10x llion)
Total Concurrency	225,000		50 Million	300X n)
Storage	15 PB		150 PB	10x 30(
I/O	0.2 TB/s		10 TB/s	10x 20 T
MTTI	Days	Days	Days	O(1Day)
Power	6 MW	~10 MW	~10 MW	~20 MW

Exascale: Main questions

- Power
  - Low power processor (ARM)
  - Maximize Flops/watt (GPU, MIC)
  - Minimize the memory per core
- Communication
  - Avoid global communication
- Fault tolerance
  - MTBF <1h

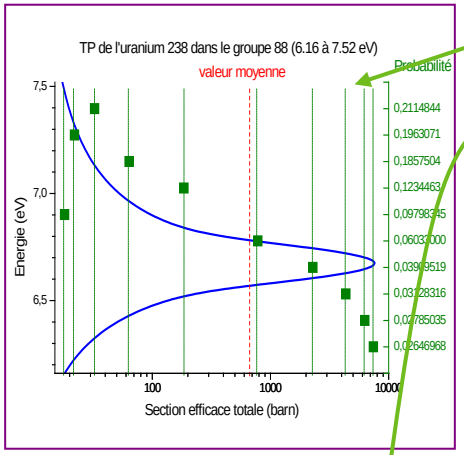


HPC IN DETERMINISTIC NEUTRONICS SIMULATIONS USING APOLLO3®

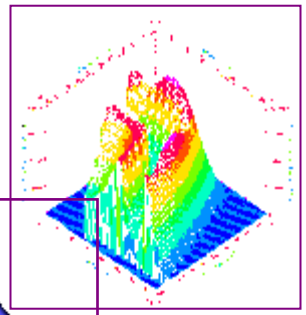
REACTOR PHYSICS SIMULATION : APOLLO3® KEYWORDS

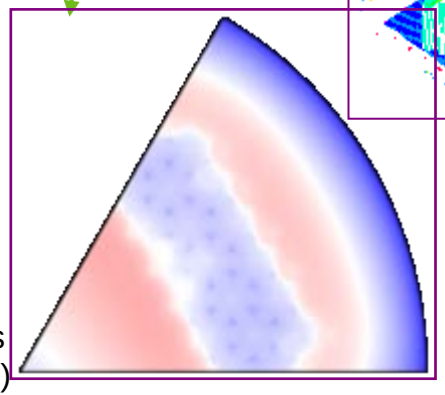
- **Continued integration and knowledge capitalization**
- **Unique calculation system for all-types reactors and cycle calculation**

- **R&D frame for physics and mathematics modelization**
- **R&D tool for nuclear systems design**
- **Industrial or easy-to-industrialize tools**

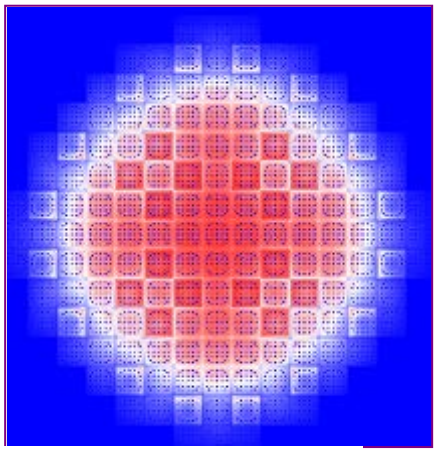


Probability tables σ representation

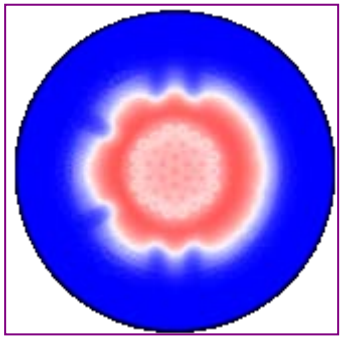




Neutron flux in a gas cooled reactor (1/6°)



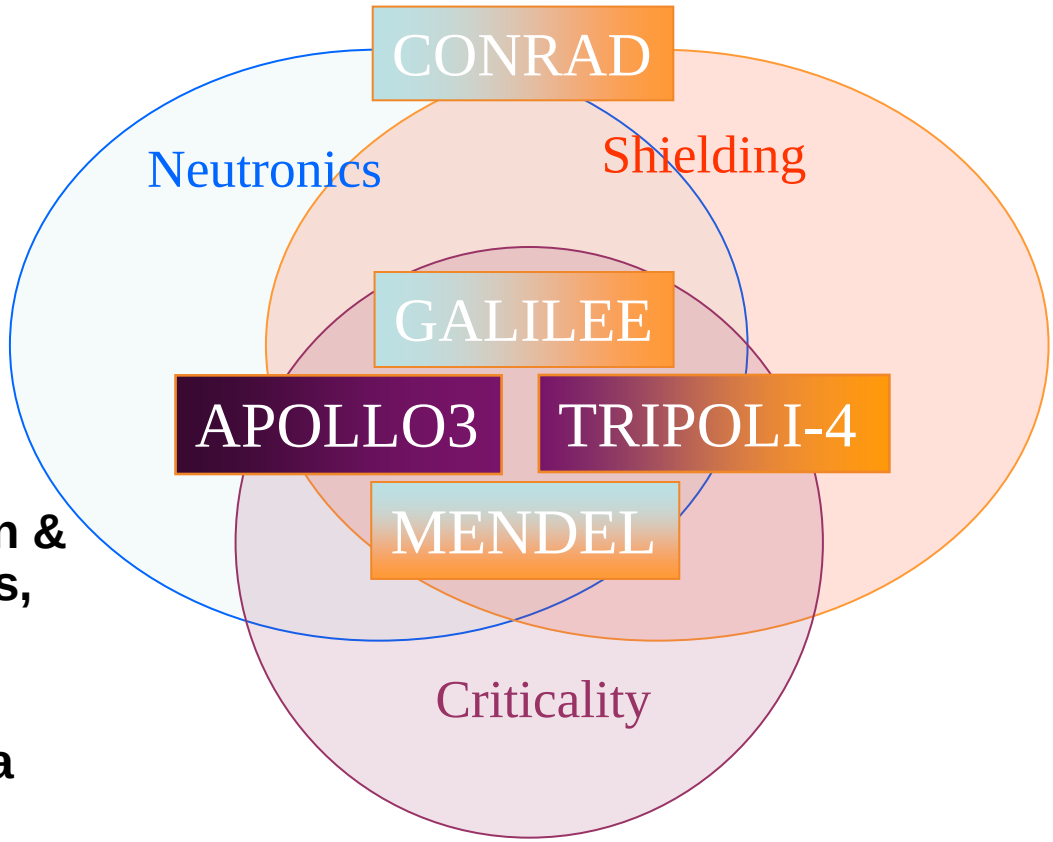
Neutron flux in a GEN3 reactor



APOLLO3® CEA/AREVA/EDF registered mark

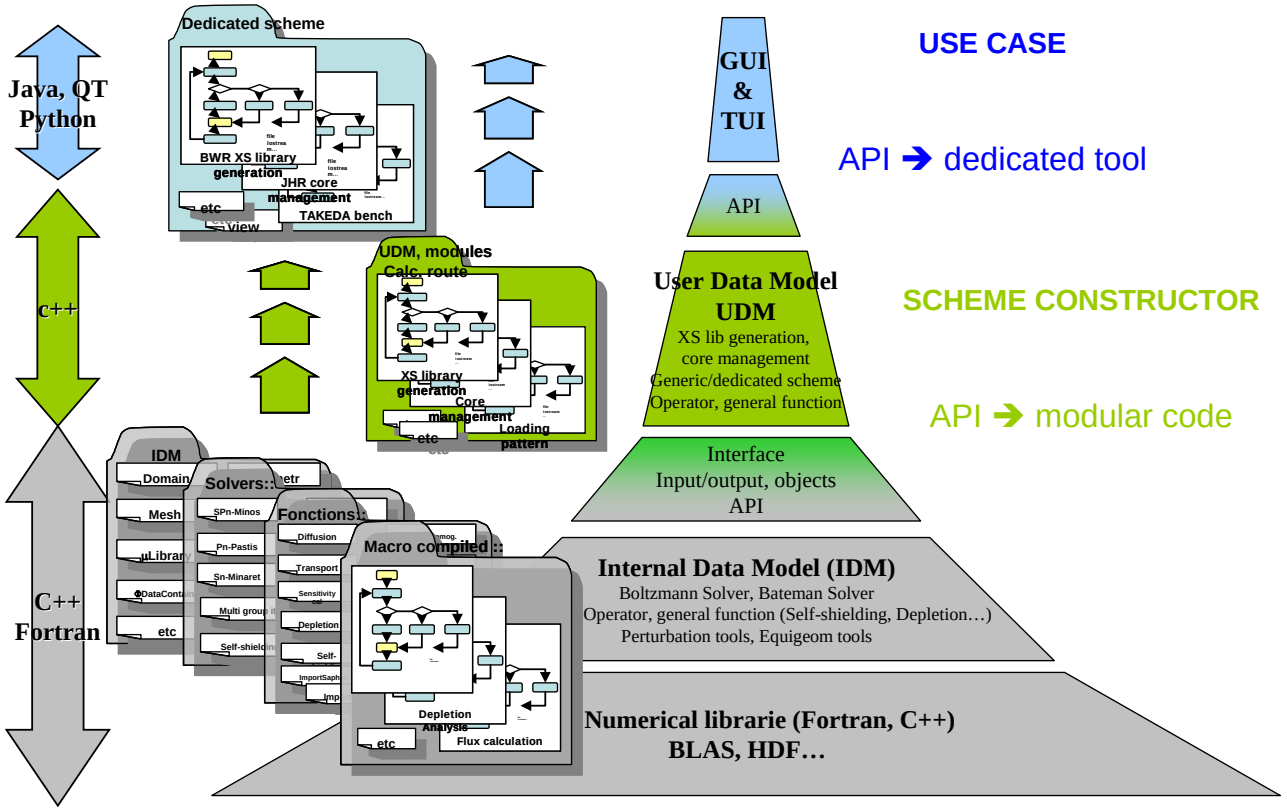
THE CEA NEW SOFTWARE SEQUENCE FOR NUCLEAR ENGINEERING SIMULATION

- **CONRAD/GALILEE**
  - **Nuclear data treatment and evaluation**
- **APOLLO3®**
  - **3D deterministic multipurpose transport code**
- **MENDEL**
  - **Isotopic depletion & derived quantities,**
- **TRIPOLI-4**
  - **Monte Carlo & simplified gamma transport**



APOLLO3® SOFTWARE

- Object Oriented conception
- Mainly C++ + some F90 modules
- Parallelism support:
  - MPI
  - OpenMP
  - CUDA



A3: THE DIFFERENT LEVELS OF PARALLELISM

Level 1: multi parameters (distributed computing)

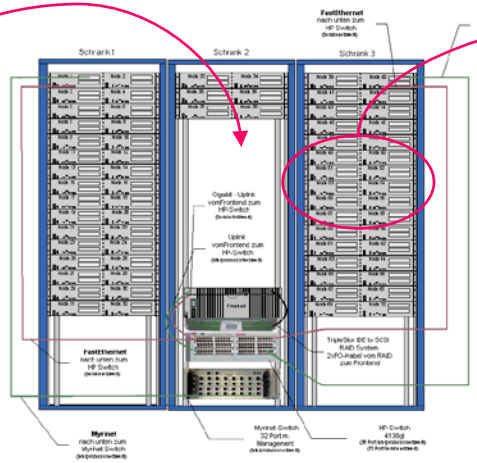
- multi parameters assemblies calculations

Level 2: multi domains (massive parallel computing)

- σ, Σ
- Depletion, Neutronic Feedback,
- DDM for flux computation

Level 3: flux computation on a sub-domain (SMP)

Level 3': hybrid computation using accelerator



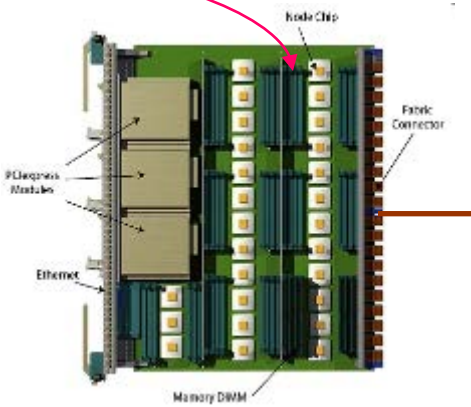
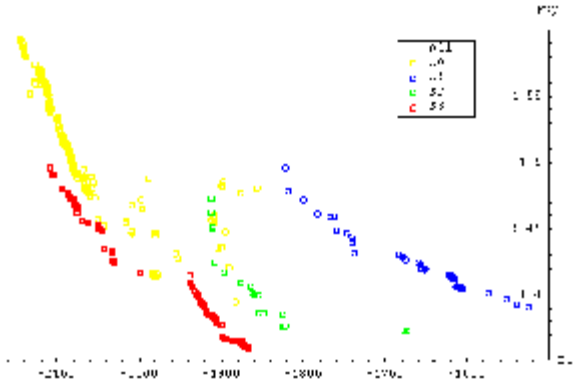
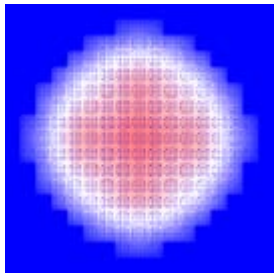


ILLUSTRATION OF HPC IN APOLLO3

- Level 1: Loading Pattern optimization using genetic algorithms
  - 10 millions of independent full 3D core computations with AP3 in 8h using 4k cores



- Level 2-3': Domain decomposition mixed with hybrid GPGU computing
  - Simplified transport solver – FEM method
  - Additive Schwartz DD method
  - GPU acceleration on each sub domain



	Time (s)
CPU solver	284
GPU solver	52
Speedup	**5**

Pin/pin 3D PWR core calculation using 2g diffusion approximation

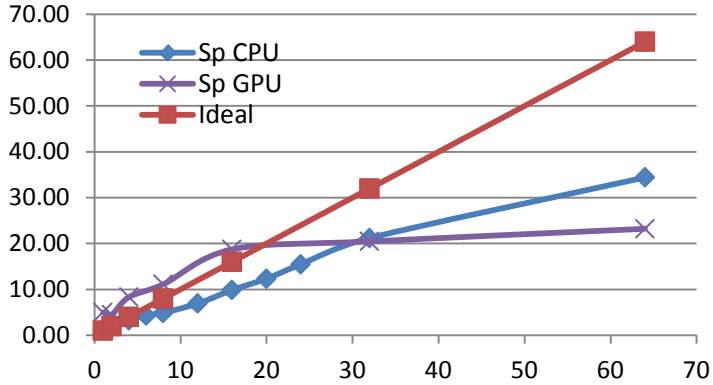
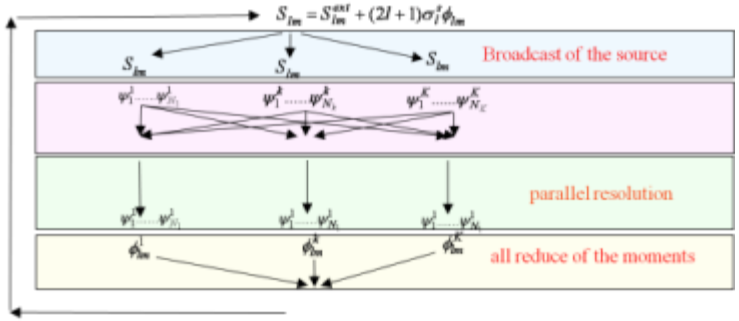
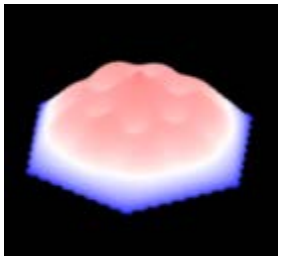
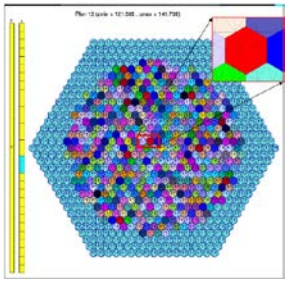


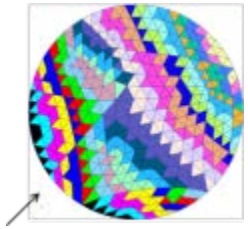
ILLUSTRATION OF HPC IN APOLLO3

- Level 1-2-3: Massively Parallel Computing 3D Sn Sodium Cooled Fast Reactor
  - Sn transport Solver 2D/3D
  - Unstructured mesh (triangle based)
  - Discontinuous Galerkin FEM
  - 2 parallelism levels
    - Through angular direction (MPI)
    - Along each angular direction (OpenMP)

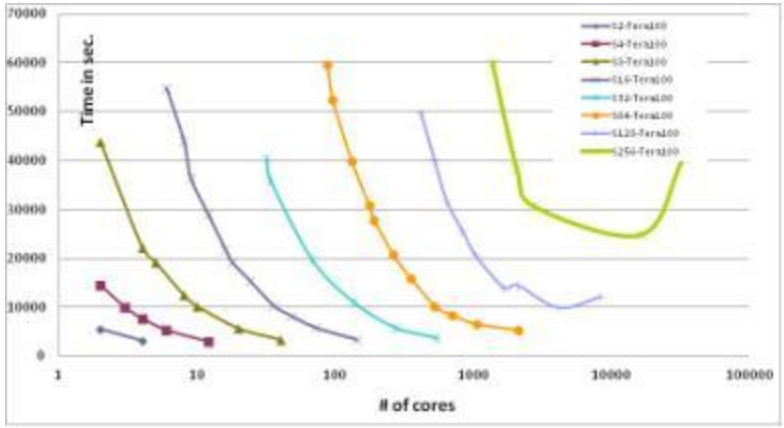








- 3D ref. calculation S256 using 33.000 cores
- Parametric study: result obtained in 12h using 5.000 cores instead of one month



SOME REQUIREMENTS AND NEEDS FOR DEVELOPING LARGE SCALE SIMULATION CODES AT THE POST PETASCALE AND EXASCALE ERA

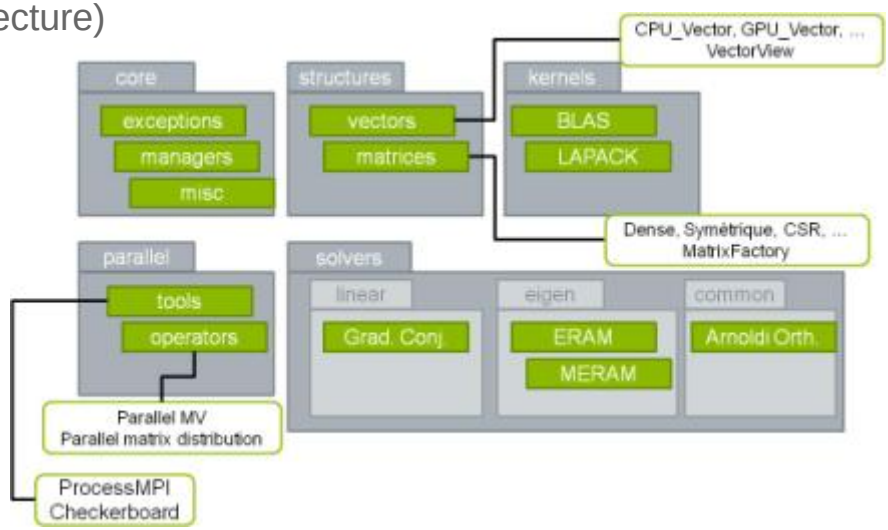
HOW TO DESIGN SCIENTIFIC CODES FOR POST PETASCALE AND EXASCALE

Need of high abstraction modeling environment

- Modeling language for parallel software design ("UML for Exascale")
- To design novel software architecture:
  - polymorphic algorithm (depending on the //ism degree) ,
  - use optimized components
  - auto tuning / smart tuning
- Unified productive programming model allowing us to express the different levels of parallelism and memory hierarchy:
  - Minimize languages interoperability problems (Fortran, C, C++, MPI, OpenMP, CUDA, OpenCL, ….)
  - Help to express in easier way the new parallel programming challenges
- Fault tolerance:
  - To be integrated now at the application level and perhaps at the algorithmic level

SOME EXAMPLES

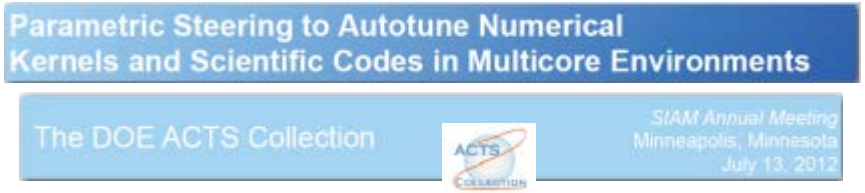
- Framework for heterogeneous architecture abstraction
  - KASH (Krylov bAsed Solvers for Hybrid architecture)
  - Trilinos/Kokkos



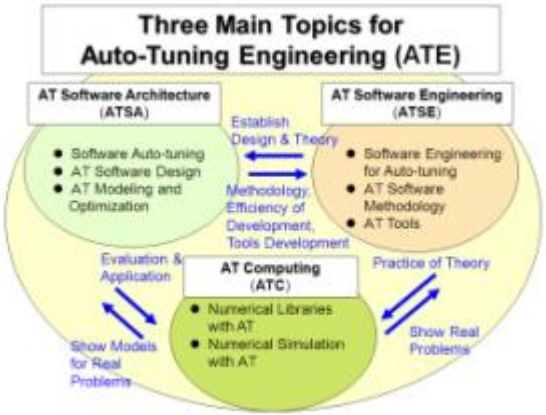
- Polymorphic algorithm
  - Petsc 3.3

"Mixed-mode implementation of PETSc for scalable linear algebra on multi-core processors"

- Auto tuning / smart tuning
  - DOE ACTS collection



  - ABCLib



SOME EXAMPLES

- Languages
  - PGAS and Data parallel languages
    - UPC, X10, CHAPEL, …
    - CAF, GlobalArrays, XcalableMP, …
  - Meta languages for distributed computing(YML, …)

- Programming environment for post-petascale (FP3C project)
  - "FP2C+" "Framework for Post-Petascale Computing".

- **A Multi-Level Parallel Programming Framework Implementation:**
  - **Top:YML a graph description language and framework**
  - **Intermediate: XMP a directive based programming**
  - **Low: Thread programming paradigm (StarPU)**
  - **Low: XMP-Dev for GPU**

SUMMARY

PETASCALE AND FURTHER FOR REACTOR PHYSICS

- HPC is a compulsory tool for Reactor Physics Simulations.
- As for many industrial and R&D numerical simulation fields, many constraints in code development:
  - Code lifecycle, Portability, performances, robustness, maintainability, …
- Meanwhile petascale and further architectures becomes more and more complex and hierarchical:
  - Multi-level parallelism, Memory hierarchy, Heterogeneity
  - …
- → Need of new programming environment in order to:
  - Take into account the evolving architectures and keep performances as best as possible: allow the programmer to express the different levels of parallelism, heterogeneity…
  - Respect the constraints of industrial scientific codes.

PETASCALE AND FURTHER FOR REACTOR PHYSICS

- Petascale and post-petascale machines are needed, and will be used in a useful way only if large scientific applications can be efficiently run on them.
  - Not only efficient architectures and/or some high end application
  - New programming environment in order to adapt large scientific applications on these new architectures.

DE LA RECHERCHE À L'INDUSTRIE

Thanks for your attention!

Time for questions and discussions.

C. CALVIN – J. DUBOIS
christophe.calvin@cea.fr – jerome.dubois@cea.fr
CEA/DEN/DANS/DM2S

COMMISSARIAT À L'ÉNERGIE ATOMIQUE ET AUX ÉNERGIES ALTERNATIVES
CENTRE DE SACLAY | 91191 GIF-SUR-YVETTE CEDEX
T. +33 (0)1 69 08 68 60 F. +33 (0)1 69 08 64 92

www.cea.fr