DE LA RECHERCHE À L'INDUSTRIE

www.cea.fr

JOREK, modelling non linear MHD

GYSELA, modelling transport, ITG turbulence

JOREK: M. Becoulet, G. Dif-Pradalier, A. Fil, V. Grandgirard, G. Latu, E. Nardon, F. Orain, C. Passeron, A. Ratnani, C. Reux

Collaborations with: INRIA, IPP, ITER Org., french Universities

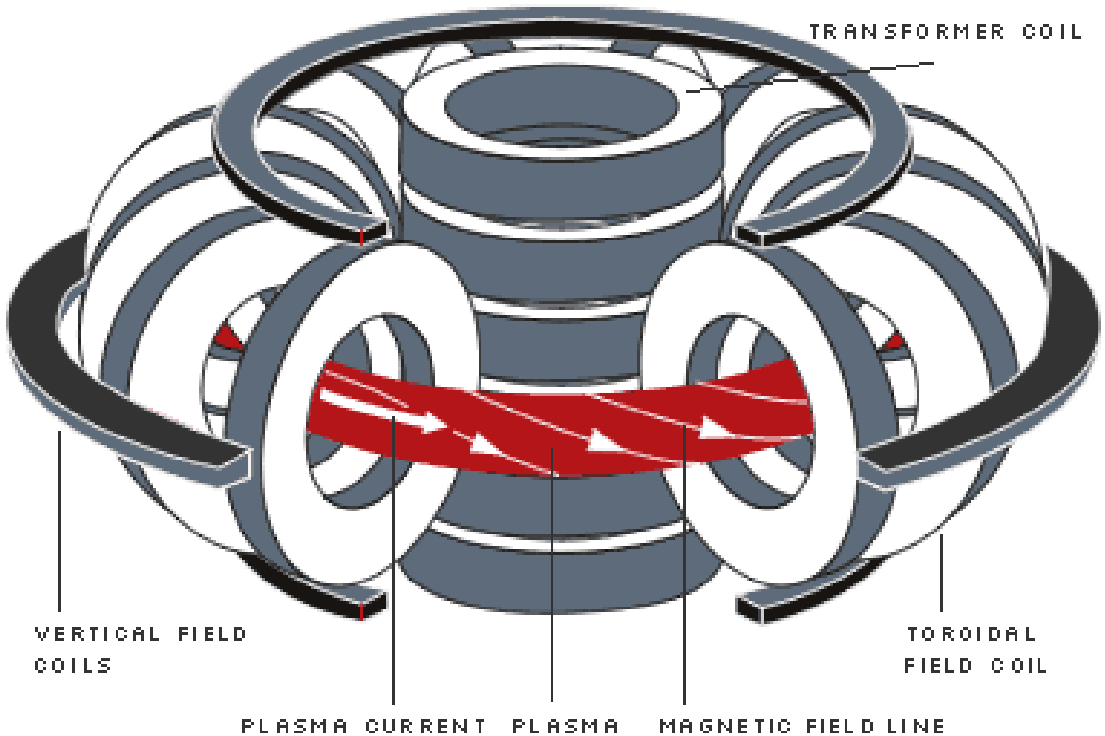
GYSELA: J. Bigot, G. Dif-Pradalier, D. Estève, V. Grandgirard, X. Garbet, Ph. Ghendrih, G. Latu, C. Passeron, Y. Sarazin, F. Rozar

Collaborations with: INRIA, IPP, french Universities

09/12/13

Outline

- JOREK code (non-linear MHD)
  - Work in progress, HPC bottlenecks
- GYSELA code (gyrokinetic, ITG turbulence)
  - Work in progress, HPC bottlenecks



JOREK, Work in progress (HPC)
(prod. runs 512 cores, test runs 1500 cores)

- [ANR ANEMOS]

  Aim: Reduce memory peak, Improve mem. scalability

  - Using Pastix - distributed API (achieve memory scalability)
  - Adapting/Improving Murge interface [$\rightsquigarrow$ Hiepacs, X. Lacoste]
  - Reduce Jorek mem. consumption (compressed data struct.)

- [Collaboration with IPP (Garching, Germany)]

  Aim: Porting Jorek on MIC architecture

  - Helios machine - Rokkasho/Japan (24^{th} in top500), dedicated to Fusion community, upgraded with hundreds of MIC nodes (december 2013)
  - First port to MIC: not targetting perf., only correct behaviour

JOREK, HPC bottlenecks

- **Pb1: Solving a large sparse system: quality of the preconditioner (convergence pb), time to solution, memory scalability**
- **Pb2: How to schedule threads**, to manage core/memory affinity (performance issues on INTEL Sandy Bridge) ?
  - Pastix uses POSIX threads, specific thread scheduling
  - Jorek uses OpenMP, thread affinity is managed with OMP_AFFINITY or KMP_AFFINITY
  - Underlying MPI implementation possibly uses threads ...
- **Possible goals through C2S@Exa:**
  - Find tools to visualize+understand dynamically the history of threads, get a trace for post-mortem analysis
  - Control threads affinity in Pastix/Jorek/MPI all together
  - Tools to show/control memory locality of data structures ...

GYSELA, Work in progress (HPC) (prod. runs 8k cores, test runs 128k cores)

- [G8-Exascale NUFUSE]

  Aim: Enhance performance/scalability

  - Designing a library for tracing/modelling memory consumption. Memory scalability improved. (PhD work of F. Rozar, Mds-CEA) [⇝ Hiepacs, J. Roman]
  - FTI (Fault Tolerance Interface) middleware is investigated in Gysela. Four levels of check-pointing offering increasing levels of resilience at the cost of higher execution time each. (Postdoc J. Bigot, MdS-CEA)
  - GYSELA scalability improved. Weak scaling 91% of relative efficiency on 460k cores of Juqueen (8^{th} in top500).
  - Data compression on large scale communication schemes

- [Collaboration with IPP + INTEL Exascale Lab]

  Aim: Optimizing a kernel of Gysela on MIC architecture

  - Work on a *new/light* version of Gysela (based on 4D advections)

GYSELA, new HPC targets

- **Possible goals through C2S@Exa:**
  - [Proposal ANR ADAM (F. Filbet)]
    - Designing a 6D parallel code and compare it to gyrokinetic model in 5D (Gysela)
  - [Considering the *new/light* version of Gysela]
    one computation kernel → a 4D advection
    - Evaluating OpenMP nested parallel for loops ?
    - Use StarPU to distribute tasks to MIC and CPU ?
  - [Resilience & IO & network]
    - Improve checkpointing (bottleneck to read/write large files)
    - Find portable methods to get high IO throughput
    - data compression for large scale comm.
  - [Memory trace]
    - Help for designing an efficient memory tracing library