JRA2

Adaptation and optimization of codes and tools for HPC

Brussels

30 March 2011

Partners

- Barcelona Supercomputing Center (BSC), Spain
- High Performance Computing Centre at the University of Edinburgh (EPCC), United Kingdom
- Åbo Akademi University (ABO), Finland

Scope

- Capture application requirements for terascale/petascale systems
- Port, optimize and improve the scalability of the selected applications
- Evaluate applications on the HPC infrastructure
- Join HPC applications with grid applications

Deliverables and Milestones

- MJRA2.1 First set of codes optimized and parallelized
- MJRA2.2 Complex workflow involving at least 2 codes
- MJRA2.3 Second set of codes optimized and parallelized
- MJRA2.4 Subset of remaining codes
- MJRA2.5 Complex workflow involving additional codes

Deliverables and Milestones

- DJRA2.1 Parallellization/optimization management report for first set of codes
  - Due M12. Status: delivered on time
- DJRA2.2 Establishing a complex workflow in a single HPC
  - Due M20. Status: delivered on time
- DJRA2.3 Parallellization/optimization management report for second set of codes
  - Due M24. Status: delivered on time
- DJRA2.4 Final report on Parallellization/optimization management report and establishment of complex workflows
  - Due M24. Status: delivered on time

Implementation

- Application requirements analysis
- Parallelizing, optimizing and scaling
- Complex workflows and interfaces with grid applications

Results from Optimization and Scaling

Code	Date assigned	Initial scaling	Present scaling	Code-opt Speed-up	EUFORIA status	Partner
BIT1	2008	64-128	512-1024	20 % faster	finished	BSC
CENTORI	2010	128	128	Improved I/O functionality	finished	BSC
EIRENE	2009	1024	1024	no speed up	finished	BSC
ELMFIRE	2008	512	2048	700%	finished	EPCC
ERO	2010	128	128	29% faster	finished	BSC
ESEL	2009	1	32	8 % faster	finished	ÅBO
GEM	2009	128	512	275%	finished	EPCC
GENE	2009	16384	16384	3% computing, 400% for I/O	finished	EPCC
ISDEP	2010	1	1	42% faster	finished	ÅBO
SOLPS	2010	1	24	51% faster on 8 threads	finished	EPCC
TYR	2008	128	2048	15% faster	finished	ÅBO

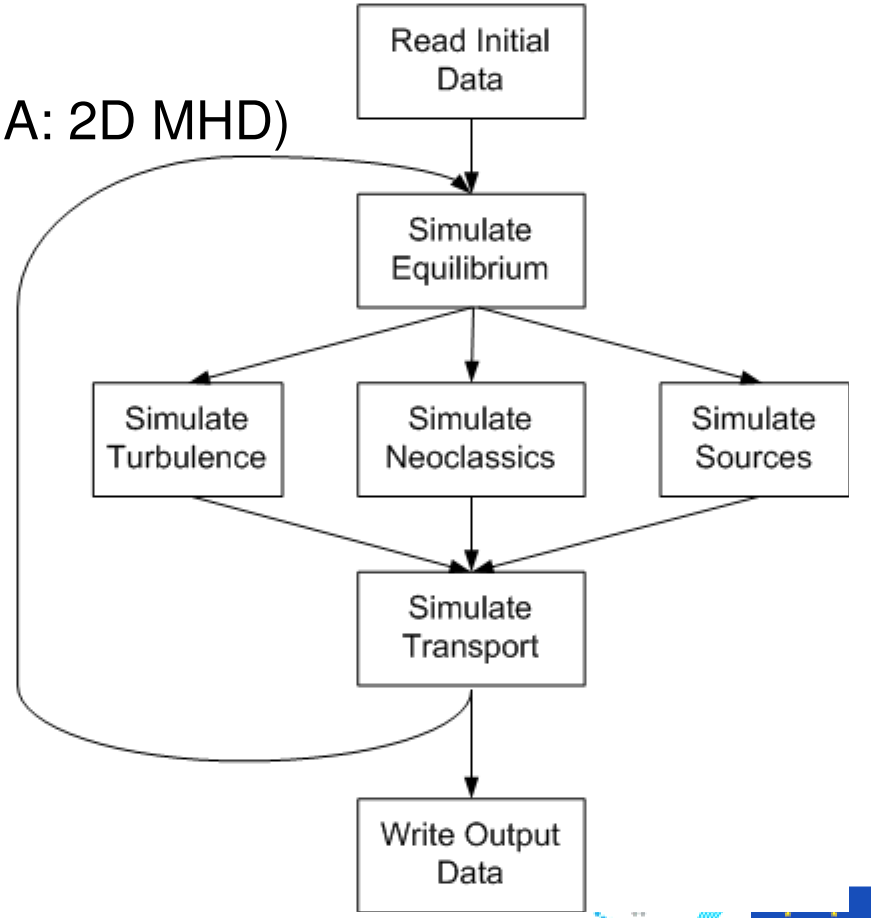
Results from Optimization and Scaling

- Non-performance related work was necessary before optimizations were done
  - JRA2 has written better makefiles for the programs, making portability possible
  - JRA2 has exchanged proprietary numerical subroutines with high performance open source libraries (PETSc, Gnu Scientific Library)
  - JRA2 has located and fixed programming errors which stopped the programs or corrupted internal data (Valgrind, performance monitors)

Workflows

- Data communication using CPO (Consistent Physical Objects)

1. Read initial data
2. Simulate plasma equilibrium (HELENA: 2D MHD)
3. Simulate plasma turbulence (GEM)
4. Simulate plasma neoclassic (bespoke codes)
5. Simulate plasma sources (bespoke codes)
6. Simulate plasma transport (ETS)
7. Write output data



Uses and Impact on Community

- Change in program use
  - codes are no longer designed to run on a single system by very few users
  - portability is acknowledged: programs should run or be easily modified to run on any CPU-based HPC-system
  - programs may even be run by another group of people
- Change in use cases
  - codes are designed to run on very many cores simulating bigger systems

Uses and Impact on Community

- Code developers of fusion simulations have become more aware of developments in hardware and software from which they can benefit
  - importance of scalability throughout program design
  - vectorization using SSE (Streaming SIMD Extensions)
  - parallel file I/O

Sustainability Path

- Results have been communicated to code developers personally: transfer of knowledge
- Clear benefits from improvements on any existing CPU-based HPC system
- Workflows have been shown to be feasible
- Cooperation with High Level Support Team at Garching/EFDA, Max Planck Institute providing them with all our internal reports to help them with their work eg on parallel file I/O.

Sustainability Path

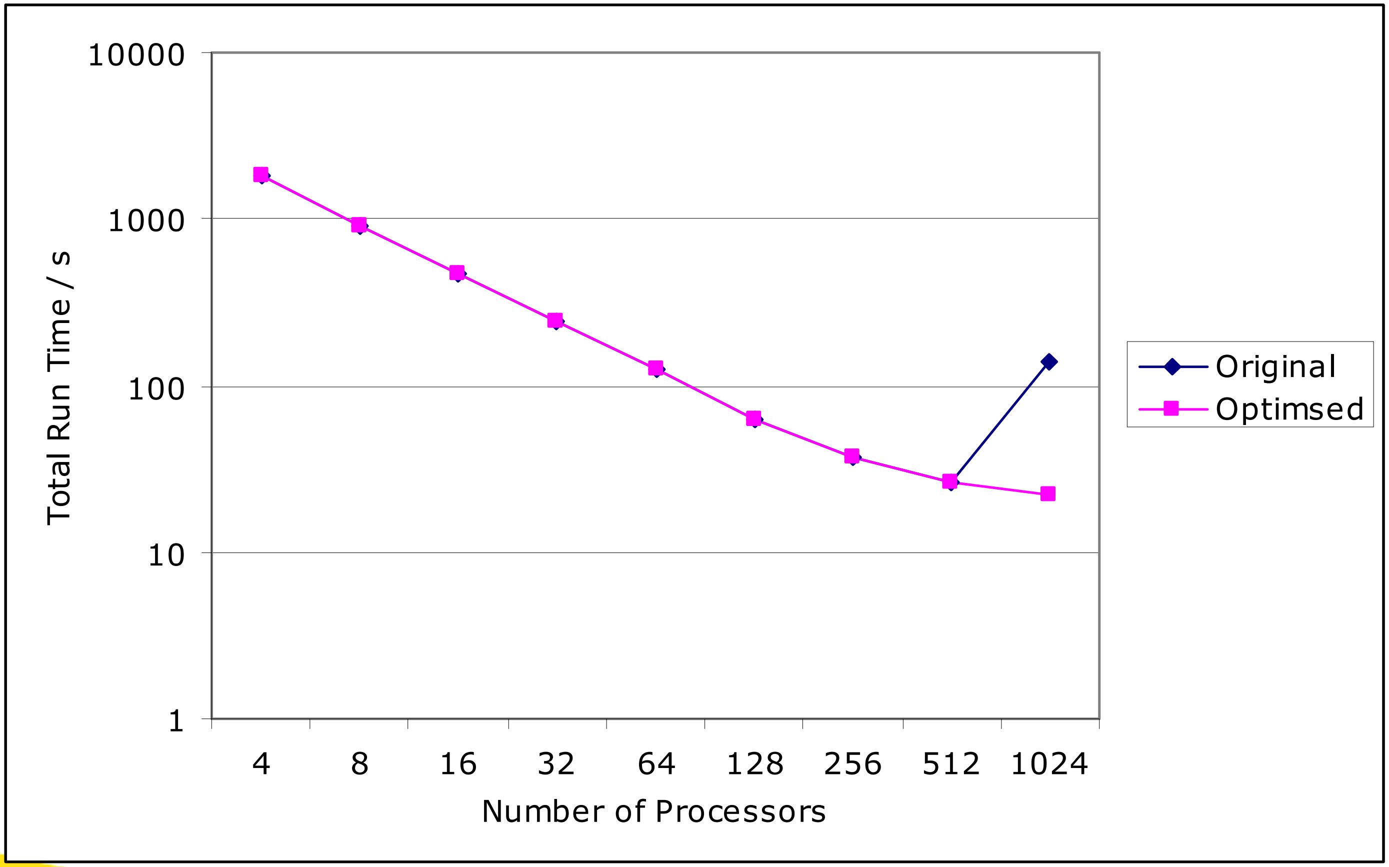
- *Continuous developments* in hardware (eg GPGPU) and software (file I/O, MPI) require *continuous efforts* in programming and even program redesign to benefit from those new resources.

End of Presentation

- Thank you!

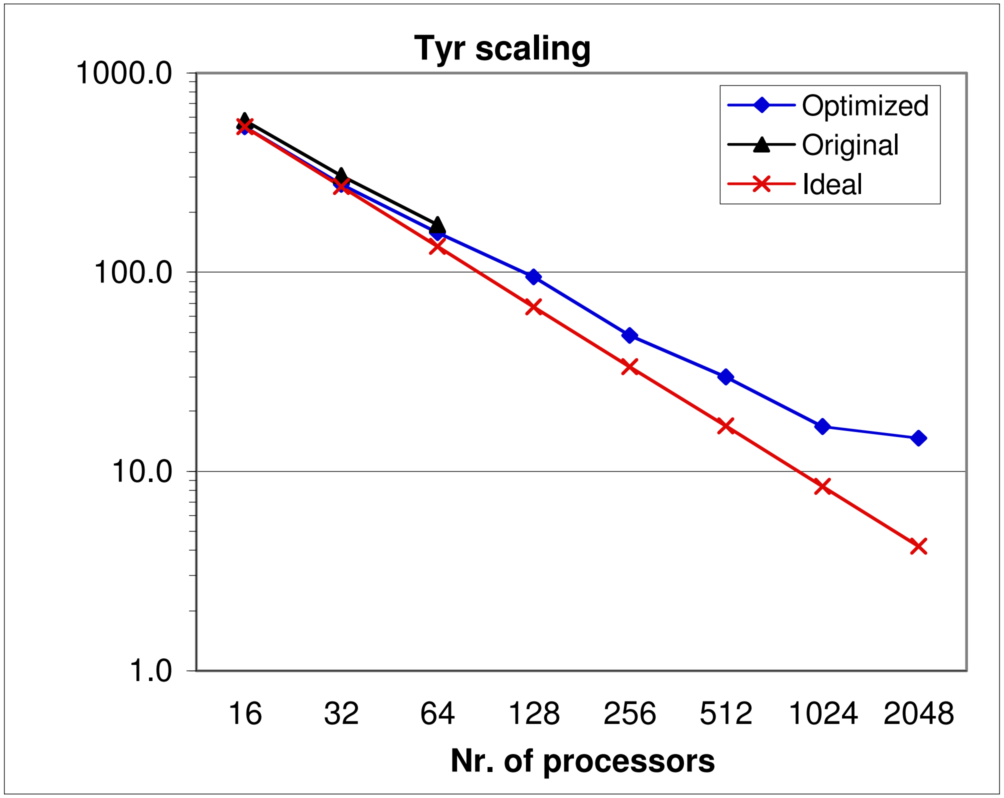
Supplementary material: Sample case

- Elmfire, a gyrokinetic particle-in-cell code
- Bottleneck: file I/O, now made parallel



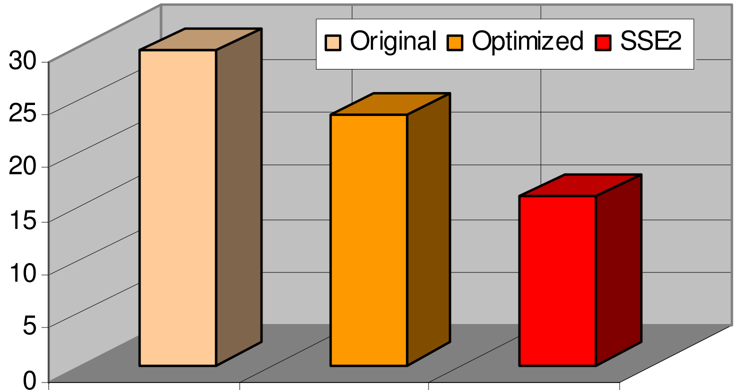
Supplementary material: TYR

- Originally contained a Helmholtz solver in 1D using FFT to reduce dimensions
- Rewritten utilizing a 2D Helmholtz solver
- Scales beyond 2048 processors for bigger problems



Supplementary material: SSE and GPU

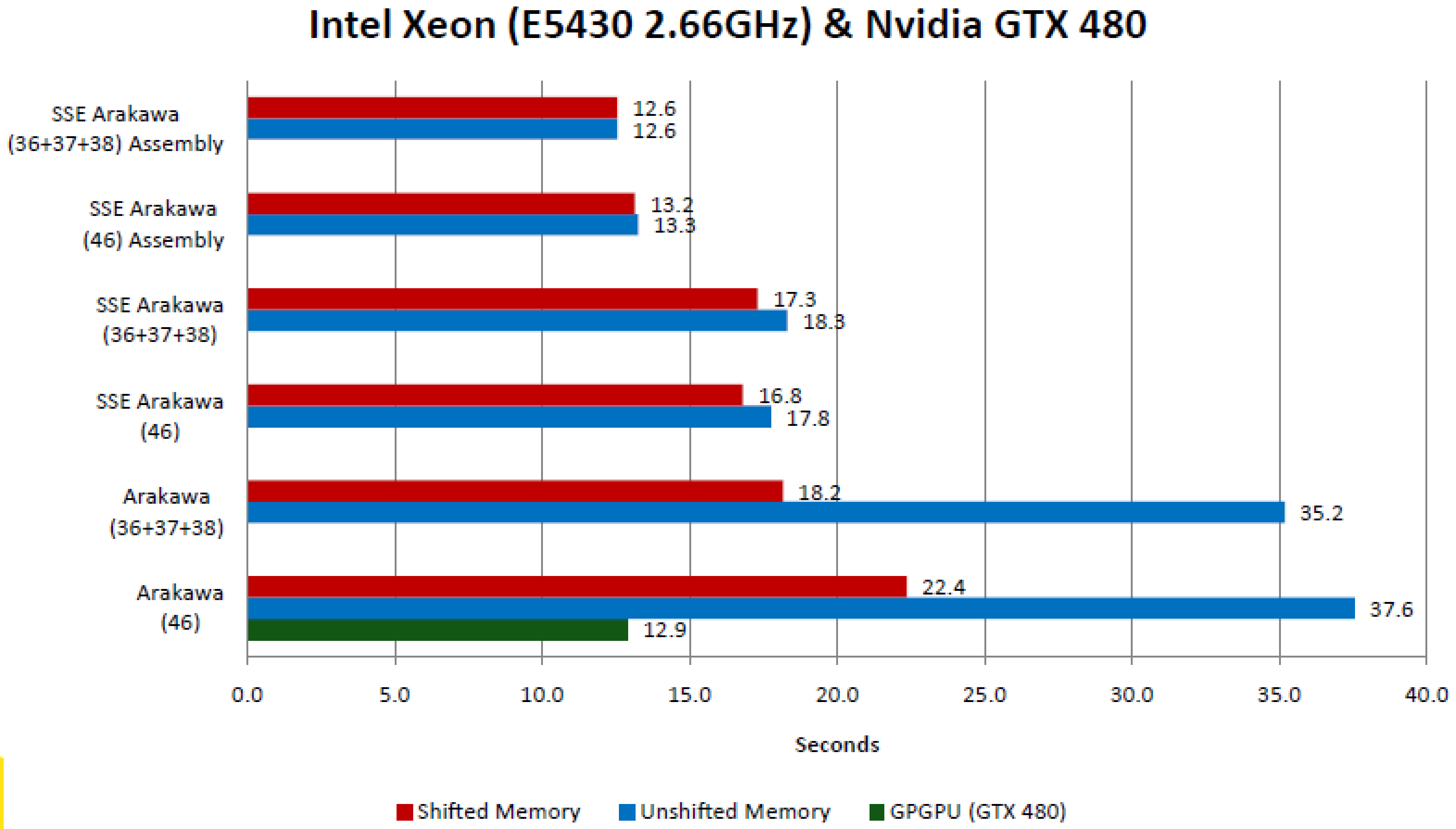
- None of the codes utilized SSE-instructions or graphics processors GPU
- JRA2 vectorized Arakawa’s formula (used in several codes) using SSE instructions and GPU



	Original	Optimized	SSE2
Seconds	29.4	23.4	15.8

Supplementary material: SSE and GPU

- SSE speed up by a factor of 2
- GPU speed up by a factor of 3


Intel Xeon (E5430 2.66GHz) & Nvidia GTX 480
SSE Arakawa (36+37+38) Assembly
SSE Arakawa (46) Assembly
SSE Arakawa (36+37+38)
SSE Arakawa (46)
Arakawa (36+37+38)
Arakawa (46)
Seconds
Shifted Memory
Unshifted Memory
GPGPU (GTX 480)