

15/12/2022

Fusion Group



EXCELENCIA SEVERO OCHOA

The Advanced Computing Hub at BSC:

Improving fusion codes following the principles of EUROfusion standard software

X. Sáez¹, M. Garcia-Gasulla¹, J. Vinyals¹, C. Morales¹, D. Vicente¹, A. Soba^{1,2}, I. Gasilova¹, M. J. Mantsinen^{1,3}

¹ Barcelona Supercomputing Center, Spain

² National Scientific and Technical Research Council (CONICET), Argentina

³ ICREA, Barcelona, Spain



This work has been carried out within the framework of the EUROfusion Consortium, funded by the European Union via the Euratom Research and Training Programme (Grant Agreement No 101052200 — EUROfusion). Views and opinions expressed are however those of the author(s) only and do not necessarily reflect those of the European Union or the European Commission. Neither the European Union nor the European Commission can be held responsible for them.

European Research Roadmap

- Provides a **clear and structured way toward commercial fusion energy**
- Is **the basis of the EUROfusion and Fusion for Energy programmes** for the realisation of Fusion Energy
- Includes DEMO design studies that are being conducted in Europe
- The DEMO project follows ITER with the goal of designing a commercial fusion power plant





Motivation

- **The step from ITER to DEMO is challenging**. DEMO design is complex in terms of the amount of systems needed to produce and control the plasma
- Experimental data from ITER and IFMIF-DONES are essential, but not sufficient to design DEMO with confidence in an unexplored environment to predict plasma and materials performance
- There is a need to create a high-quality suite of research codes (**EUROfusion-standard software**) to model data from existing EUROfusion facilities and to reliably extrapolate to future devices



EUROfusion ACHs

- The goal is to bring together fusion physicists, materials scientists and engineers with computer scientists within the same organisational framework to make use of high-performance computers (HPC) and accelerate the development of fusion energy
- EUROfusion has initiated coordination among theory and advanced simulation creating:
 - Theory, Simulation, Validation and Verification tasks (TSVV) that perform fundamental research and channel science
 - Advanced Computing Hubs (ACHs) that provide advanced simulation expertise and knowledge to TSVVs





BSC-ACH

- Barcelona Supercomputing Center (BSC) has been awarded one of these ACH that will offer advanced simulation to TSVVs
- BSC-ACH provides expert support to users regarding High-Performance Computing (HPC):





Codes assigned to BSC-ACH

Code	Work required	Code Coordinator	Groups in charge	Start
ERO 2.0	GPU porting	Dmitry Matveev, Juri Romazanov	CASE-Fusion, Best performance	2022
SPICE2	Implementing a parallel Poisson Solver and a Parallel electric field calculations.	Michael Komm	CASE-Fusion, Operations	2022
KNOSOS	Optimization.	José Luis Velasco	Operations	2022
STELLA	Optimization.	Michael Barnes	Best performance	2022
SOLPS	 Check the correctness of the OpenMP multi-threading Improve compiler options and/or job submission script Consider vectorization of the code profitable for use on Cray platforms 	David Coster	Operations, Best performance	2023
STELLA	Optimization.	Michael Barnes	Best performance	2022
XTOR-K	GPU porting	Hinrich Lutjens	CASE-Fusion, Operations	2023
BIT1	GPU porting	David Tskhakaya	CASE-Fusion, Operations	2023
GENE-X	 Implementing the ability to access the unstructured computational grid in arbitrary order Testing the performance of different reordering strategies 	Philipp Ulbl	CASE-Fusion, Operations	2023
JOREK	Reduce memory consumption and improve performance	Matthias Hoelzl	CASE-Fusion, Operations	2023



ERO2

- ERO2.0 is a code for modelling plasma-wall interaction and global material migration in fusion devices. The migration is simulated by following 3D trajectories of Monte-Carlo test particles
- Language: C++
- Parallelization: MPI + OpenMP

ACH tasks

- Performance analysis
- Porting to GPUs







Tracing

• BSC Tools: Extrae and Paraver (http://tools.bsc.es)





ERO2





ERO2

Global efficiency metrics

Nodes:	4	8	16	32		
	192(48x4)[1]	384(96x4)[2]	768(192x4)[3]	1536(384x4)[4]		New low debal off due to had
Global efficiency -	80.20	70.95	57.81	39.51	- 100	parallel efficiency.
Parallel efficiency -	80.20	70.59	58.33	40.43	- 80	
Load balance -	91.52	86.73	76.55	62.27		
Communication efficiency -	87.64	81.39	76.21	64.93	ge of	
Computation scalability -	100.00	100.52	99.10	97.73	cent;	Manuar ad a supervisition
IPC scalability	100.00	99.94	100.23	100.35	æ	scalability.
Instruction scalability	100.00	100.58	98.91	97.67	- 20	
Frequency scalability	100.00	100.00	99.96	99.71		
	192(48x4)[1]	384(96x4)[2]	768(192x4)[3]	1536(384x4)[4]		Vanulau MDI Darallal Eff. Dua ta
Hybrid Parallel efficiency	80.20	70.59	58.33	40.43	- 100	Load Balance and Serialization.
MPI Parallel efficiency	- 88.95	79.11	66.42	47.71	- 80	
MPI Load balance	- 95.46	90.80	84.69	68.30		
MPI Communication efficiency	- 93.18	87.13	78.42	69.86	e(%)	
Serialization efficiency	93.20	87.18	78.53	70.03	ntag	
Transfer efficiency	99.98	99.94	99.86	99.75	- 40 2	
OpenMP Parallel efficiency	90.17	89.23	87.83	84.74	<u>م</u>	OpenMP Parallel efficiency
OpenMP Load Balance	95.87	95.53	90.39	91.17	- 20	shows a tendency to decrease.
OpenMP Communication efficiency	- 94.05	93.41	97.17	92.94		







Optimization



• Include MPI communication inside the OpenMP parallelization, so new work can be started when threads start idle



ERO2

Porting to GPU

- OpenACC
 - The compiler is responsible for generating kernel and managing data transfers
- Find top timeconsuming routine ⇒ octree distance search

Intel VTune (https://www.intel.com)

Symbol Name	Self, %	Total, % 🔺
• [Broken backtraces]		100,00
* 0x7fff7e9652f4		99,92
+ 0x7fff7e965100	-	99,91
* main		99.91
reo2::Ero2Simulation::run(int, char**)	,	99,91
reo2::Ero2Simulation::transportLoop()		99,45
*kmpc_fork_call	3	99,45
✓ runNewTeam		99,45
✓ launchTeam		99,45
✓ launchinternal		99,45
+ hxiEmulateHostThreadLaunch		99,45
targetFuncHostTrampoline_1		99,45
*		99,45
reo2::Ero2Simulation::transportParticleLoop(ero2::Particle&, ero2::Plasma&, ero2::DensityManager&, ero2::Sheath&, ero2::Particle&)		99,44
ro2::Ero2Simulation::transportParticleStep(ero2::Particle&, ero2::Plasma&, ero2::DensityManager&, ero2::Sheath&, ero2::ParticleStepData&, ero2::Particle*&)	0.00	99,33
y g3d::Octree::getDistance(g3d::Vector const&, g3d::Vector&, double&, g3d::Polygon const*&) const	0,00	69,91
v q3d::Octree::getDistanceInNodeSq(g3d::Vector const&, double&, g3d::Polygon const*&, std::set <std::pair<unsigned long="" long,="" unsigned="">, std::less<std::< p=""></std::<></std::pair<unsigned>	0,01	58,75

- Issues:
 - C++-style loops are not supported
 - Polymorphism is not supported
 - Routine calls within parallel loops can be problematic

- ...

- However, it is not as ideal as it was initially expected
 - OpenACC has lacks with C++
 - The feedback from error messages is not very clear





• NVIDIA Nsight (https://developer.nvidia.com/nsight-visual-studio-edition)



- Next steps
 - OpenACC + CUDA interoperability
 - CUDA Dynamic Parallelism: Recursive search into Octree







- Particle-In-Cell (PIC) code dedicated to performing simulations of particles in a fixed magnetic and self-consistent electric field for the study of plasma deposition near castellated plasma facing components
- Language: Fortran
- **Parallelization:** MPI

ACH tasks

- Implementation of 2D parallel Poisson solver that the number of cores in simulations can be increased to at least 128
- Implementation of a parallel routine for E-field calculation
- General improvement of the code



Profiling

• Intel Tools : VTune and Advisor

(https://www.intel.com/content/www/us/en/developer/tools/oneapi/vtune-profiler.html)

~	Program Metrics				
	Program Elapsed Time	429,30s		* GFLOPS	0,77
	Vector Instruction Set	AVX512, AVX2, AVX, SSE2, SSE		GFLOP Count	332.000
	Number of CPU Threads	2		FP Arithmetic Intensity 🗇	<0.001
				 GINTOPS 	0,81
	✓ Performance Char	racteristics			
	Metrics		Total		
	CPU Time		426,715		100%
	Time in 88 Vectorize	ed Loops	203,50s		47.7%
	Time in scalar code		223,215		52.3%
	INT+FLOAT Giga O	P Count	680,64		100%
	INT+FLOAT Giga O	PS	1,59		
	GFLOP Count		332,49		48.9%
	GFLOPS		0,77		
	GINTOP Count		348,14		51.1%
	GINTOPS		0,81		

Top Time-Consumir		
O loop in umfdl_lhsolve		
O loop in umfdl_uhsolve		
O loop in umfdl_lhsolve		
O loop in leapfrog_		
O loop in mainloop_		
Function	Module	CPU Time ③
weight_particles_	spice-2.14-debug.bin	204.805s
umfdl_lhsolve	spice-2.14-debug.bin	68.761s
leapfrog_	spice-2.14-debug.bin	58.497s
calc_e_field_general_	spice-2.14-debug.bin	47.722s
psolver_direct2_periodic_	spice-2.14-debug.bin	45.066s
[Others]	NI/Q*	131.634s

• By modifying the most consuming parts of the code, we achieve small improvements



Tracing

• BSC Tools: Extrae and Paraver (http://tools.bsc.es)





- All processes write to the same file at the beginning of the simulation
- Each process waits 6*their_rank seconds to start writing
- We updated the code to only do one writing on the process with rank 0 and the others processes wait on a Barrier



Optimizing

ſ	MPI call @ spice-2.14-release.bin.prv #4 (on s23r2b18)		×
messo 1.1 messo 1.2.1 messo 1.2.1 messo 1.2.1 messo 1.2.1 messo 1.2.1 messo 1.3.1 messo 1.4.1 messo 1.4.1 messo 1.1.1 messo 1.2.1 messo 2.1.1			
0 us	MPI_Recv	44,577	,065 us

	MPI call @ spice-2.14-release.bin.prv #6 (on s23r2b18)		×
0 us	MPI_Barrier	44,577,	065 ι



- The potentials are sent to all the processes from rank 0 with Point-to-Point (P2P) calls
- The density vector is calculated on process 0 and sent to all the processes with P2P
- Code updated to send the potentials with the MPI_Broadcast collective

• Code updated using MPI_Reduce collective to compute the density vector



Implementation (Poisson solver)

- **Current implementation:** Direct solver based on LU decomposition using the UMFPACK and BLAS libraries
- Limits: good scaling up to 32 cores (memory limit of the system) and a grid limit of ~ 4000 cells in one dimension
- Solution: use a KSP linear solver based on PETSc library (https://petsc.org). Several options of solvers were explored
- The best time is obtained for Conjugate Gradient with Jacobi preconditioner solver





• Results:

- Speed up of the whole code with the new solver included
- Good performance up to 128 processors
- Loss of scalability for more than 256 processor, but the code can now run over 512 processors, solving domains of 10000x4000 cells (Marconi)





KNOSOS

- Code that calculates neoclassical transport in low-collisionality plasmas of 3D magnetic confinement devices by solving the radially local drift-kinetic and quasineutrality equation
- Language: Fortran
- **Parallelization:** MPI

ACH tasks

- Overall performance assessment
- Optimize application



KNOSOS

Profiling

Supercomputing

Center

UROfusion

Ciemot

- There is a high load imbalance between MPI ranks, which seems to be produced by • different number of iterations in subroutines \Rightarrow feedback from developers
- Discover that parallelization relies on launching "independent" calculations \Rightarrow ٠ improving sequential version would translate to an overall improvement in the **MPI** version
- calc low collisionality nanl 90 91 % 97.14 % Firstly, identify the ٠ 400 x 400 x 2 x main most time consuming 99,91 % fill matrix petsc integrate g coefficients dke parts of the code 24.86 % 27.51 % 40.55 % MAIN 400 x 8 134 65... **2**7 944 x with Valgrind 99.91 % **1**x 1 x lagrange_ kspsetup bounces (https://valgrind.org) olve dke gn amb 22.43 % 40.55 % 8.32 % 62.49 9 400 x 7 944 x calc database 55 x 35.87 % KSPSetUp bounce integral 38.34 % calc fluxes 22.43 % **1**× 400 x 15 354 7... **399 x** PCSetUp bounce integrand 22.43 % 28.81 % 400 Barcelona 21

X.Sáez - 3rd Fusion HPC Workshop - 15/12/2022

KNOSOS

Optimization

- Introduce a new parallelization level using OpenMP in coefficients_dke routine
- It was found that different instances of the BOUNCES subroutine could be performed in parallel without producing data conflicts between iterations
- Result: Execution time was reduced from 15 sec (sequential) to 10 sec (using 4 OpenMP threads)
- We repeat the process with integrate_g routine. However, the improvement was negligible





Conclusions

- The Advanced Computing Hubs have an ambitious mission to accomplish in a small time window
- The creation of EUROfusion-standard software needs close collaboration between developers and computer science professionals
- We have shown some examples of our work, with different degrees of success to build trustable software capable to run efficiently in HPC systems
- In some cases, the best solution does not accomplish the objectives or the wishes of the developers



Fusion Group



BSC BSC Supercomputing Center Centro Nacional de Supercomputa EXCELENCIA SEVERO OCHOA

Thank you



fusion.bsc.es



@Fusion_BSC

fusion@bsc.es

xavier.saez@bsc.es