SPICE2d – SPEC – BIT1 efficiency analysis

Alejandro Soba

HPC ACHs | 27-28 November 2024

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.

SPICE2-SPEC-BIT1 efficiency analysis

- **SPICE** (SheathParticle In CEll) package includes two codes: SPICE2 (2D3V) and SPICE3 (3D3V).
- PIC code for simulations of particles in a fixed magnetic and self-consistent electric field
- Written in Fortran 90, outputs in the Matlab MAT binary format.
- Parallelization implemented using domain decomposition principles and message passing interface (MPI).
- All internal routines are parallel except for the Poisson solver. The Poisson solver is serial (taking 3% of the overall calculation time). It operates with global matrices of potential and charge density.
- Best solver are SPARCE, based in UMFPACK.

SPICE2

Initial Work required into ACH

- Implementation of 2D parallel Poisson solver with good scaling and speed so that the number of cores in simulations can be increased to at least 128 (current practical limit is around 32) and the grid size can be increased (UMFPACK has a limit of ~4000 cells in one dimension);

- Implementation of parallel routine for E-field calculation (later task).

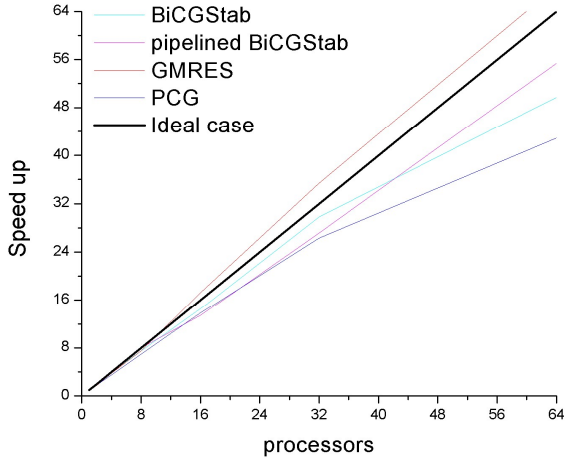
New requirements for 2024/2025

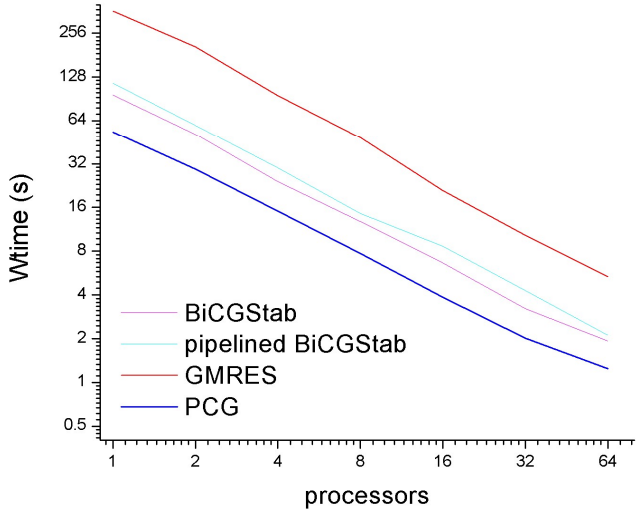
- Continue the solver development;
- Create a git branch for BSC version;
- Create parallel Outputs;
- Reduce the memory consumption;
- Go to a complete parallel SPICE2d.

SPICE2

SPICE2

- We included PETSC solver, based in KSP linear solver context and we explored several options
- From all the solvers tested, we identify 4 of them that possess the better CPU time and scalability. The best is KSP^{-1}



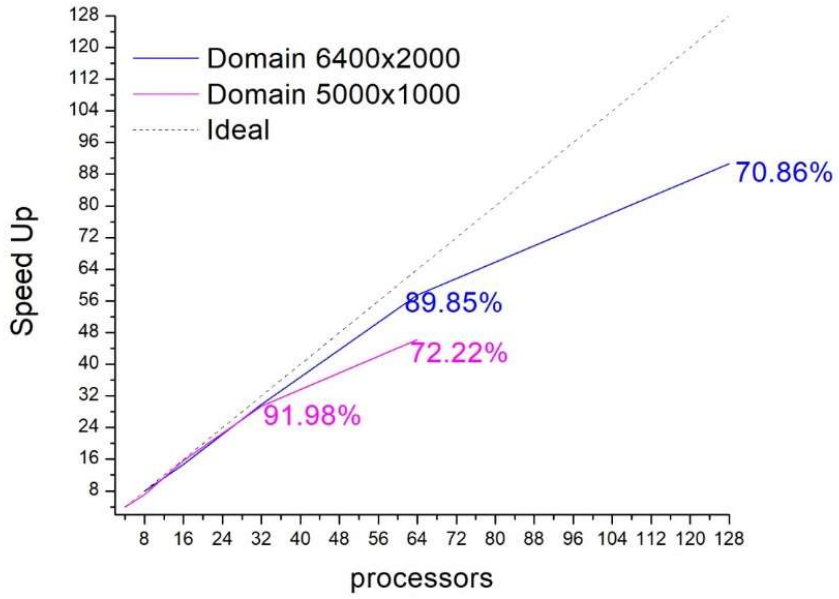


Scalability of selected KSP Petsc solver analized (left); Wall time consumed by each KSP solver (right).

SPICE2

We included several options in the input files to choose variables for PETSC.

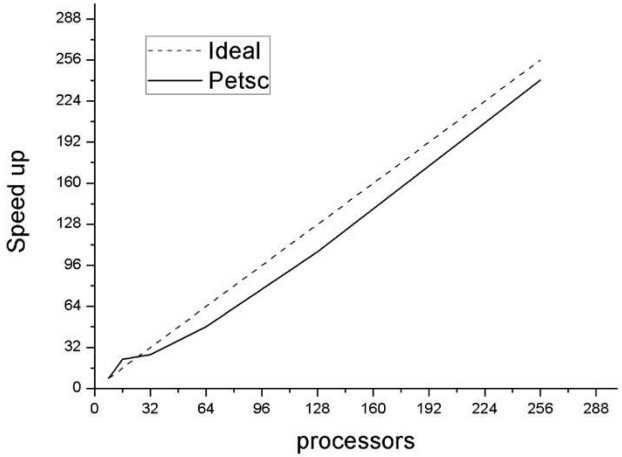
SPICE2 is able to run in big domains up to 128 processors with good scalability

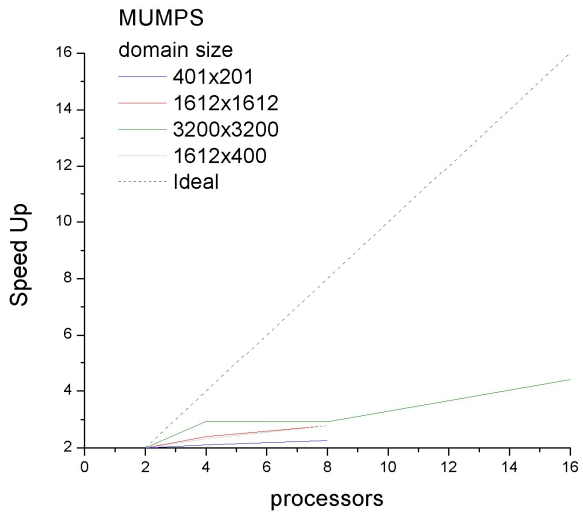


SPICE2

SPICE2

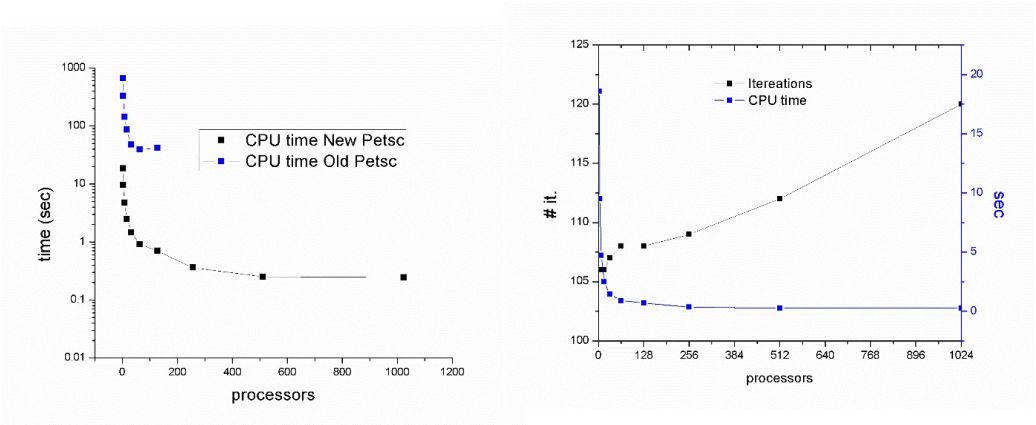
- The developers don’t like that the solver were slower than the serial one.
- We implement other version in PETSC and in MUMPS



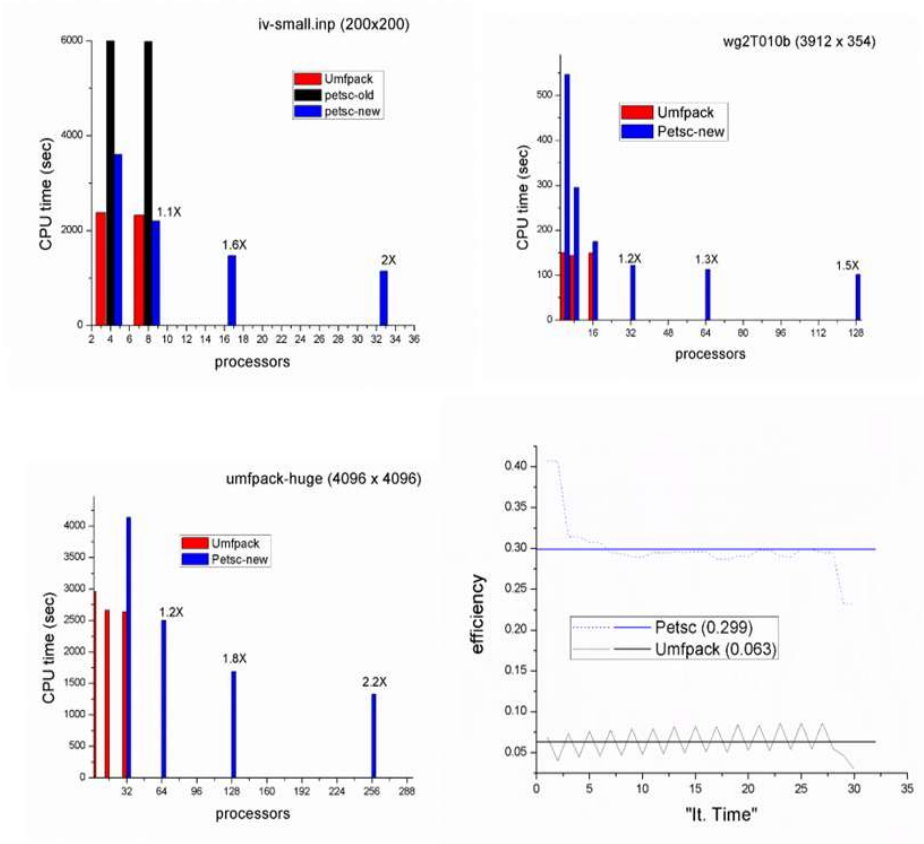


Petsc (Left) and MUMPS (right) scalability in MN4.

SPICE2



left) Petsc computing time comparison; right) CPU time and iteration in the new version of the Petsc solver



Speed up for different cases. Efficiency measures of SPICE2 in MARCONI cluster using a complete single node (48 processors case)

SPICE2

Table I: Changes in output and restarts files

Old-subroutine	file	New-subroutine	file
LANG_MONITOR	Write tfile t-input.mat	LANG_MONITOR_lc	Write tfilep #p-t-t-input.mat
SAVE_RESULTS	Write ofile input.mat	SAVE_RESULTS_lc	Write ofilep #p_input.mat
restore_9_avdiag	Read ofile input.mat	restore_9_avdiag_lc	Read ofilep #p_input.mat
restore_run_9_histories	Read tfile t-input.mat	restore_run_9_histories_lc	Read tfilep #p-t-t-input.mat

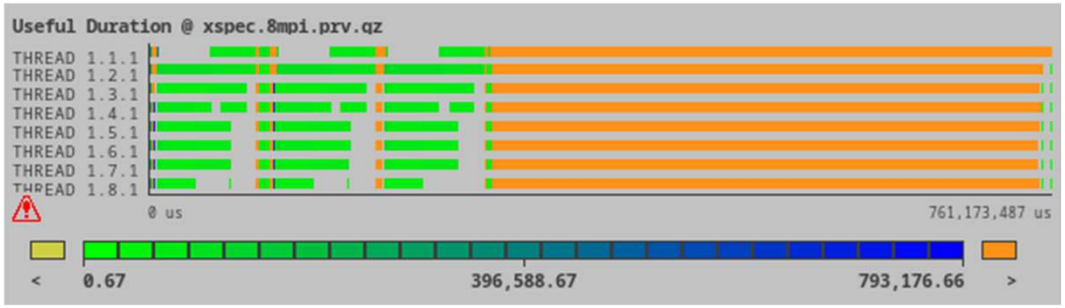
SPICE2

SPICE2, problems remaining from the past

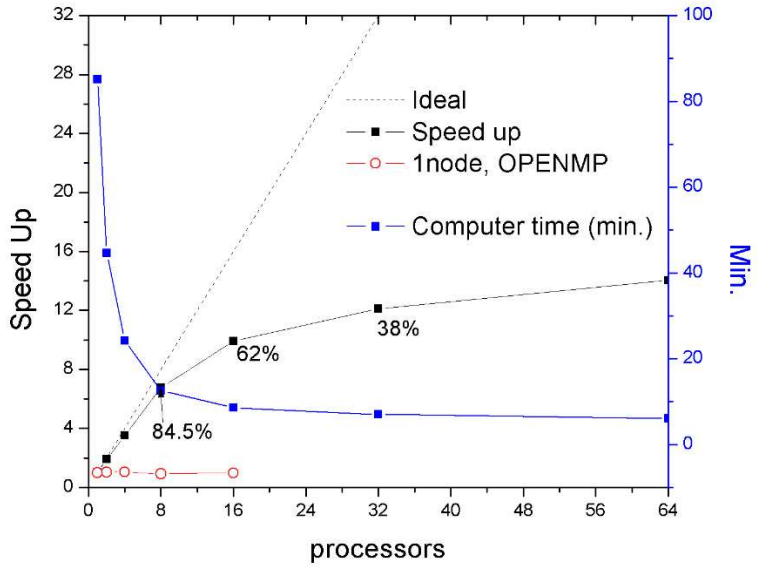
- During this work we identified several issues in the memory management.
- Bad distribution of the domain and several arrays that consume memory until making it starving.
- A compromise exists: It is not possible to run SPICE2 faster than the serial version using UMFPACK, therefore it is not possible to run the code in bigger domains.
- Now this issue is fixed, and we can proceed to a complete parallel version of SPICE2d.

SPEC

MHD equilibrium for Stellerators, quasi-axisymmetric configuration.
Written in Fortran 90 + hdf5 + fttw



PARAVER shows two separately regions. The first portion preset low dense communications patterns, with many communications of small size (that's degraded the efficiency). The other computational metrics show good performance.
We observed that only half of the choose processors participate in the communications zone. But in the second portion of the code, all the processors scale linearly.



scalability curve of SPEC code, case ACH_longrun.sp.
In 16 proc reach 62% but the computer time is 8 minutes.

Conclusions

- After the understanding of the different zones of the code and the meaning of this two regions, several improvements in the communications patterns can be do it.
- OPENMP can be a possible paradigm to use in order to improve efficiency. At the moment the introduction of a number of threads greater than 1 not introduce significative improvements.
- The non-convergence case is a “solver” problem (the solver can’t treat this system). Could be a good idea extract the system and try to explore other alternatives of solution.

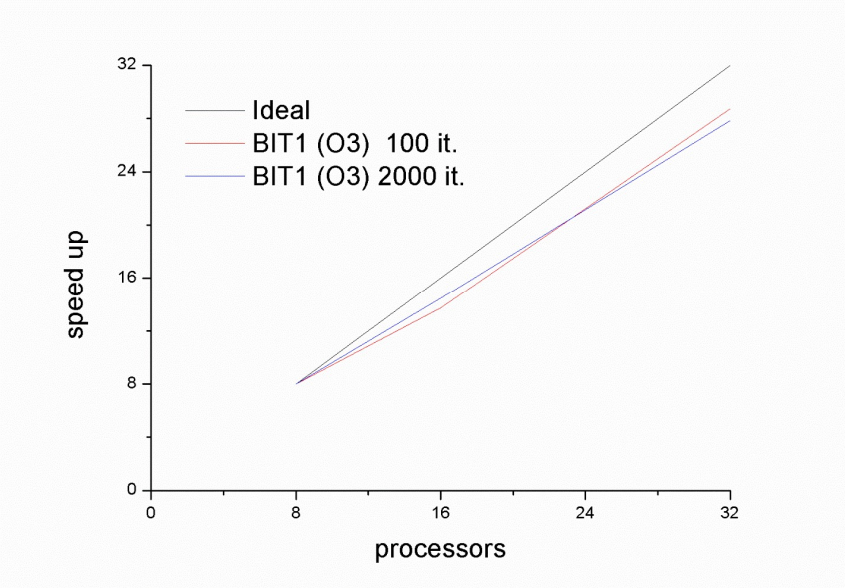
BIT1

BIT1

- BIT1 is an electrostatic 1D3V PIC direct Monte Carlo code for plasma simulations used to edge plasma simulations.

Main issues

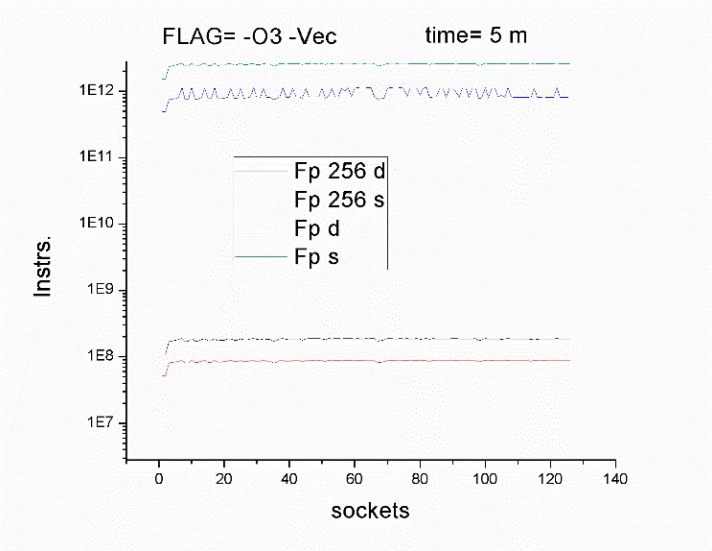
- Low efficiency
- No vectorization operation detected



BIT1

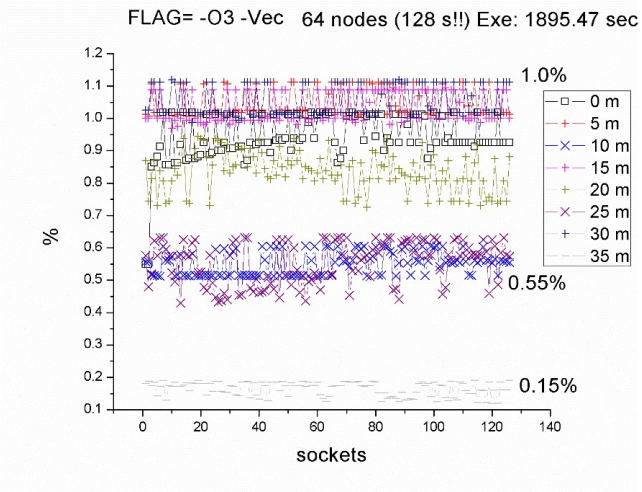
Vectorization flags issue: testing files –O3 -Vec

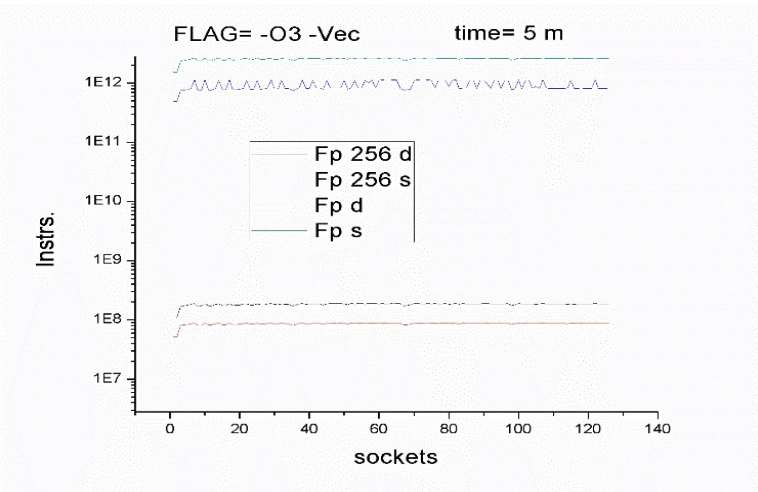
Compilation FLAGs	case	Nodes/sockets	Time (sec)
-O2	DCSM_0_e21_c8.inp	1 /2	~15200
-O3 - Vect	DCSM_0_e21_c8.inp	1 /2	~1600
-O3 - Vect	DCSM_0_e21_c8.inp	1 /2	~15400
-O2	BIT1_N_CU.inp	64/128	~16800
-O3	BIT1_N_CU.inp	64/128	~1890
-O3	BIT1_N_CU.inp	64/128	~17200
-O3 - Vect	BIT1_N_CU.inp	64/128	~1895
-O2	BIT1_N_CU.inp	128/256	~1075
-O3 - Vect	BIT1_N_CU.inp	128/256	~9700
-O3	BIT1_N_CU.inp	192/384	~1600

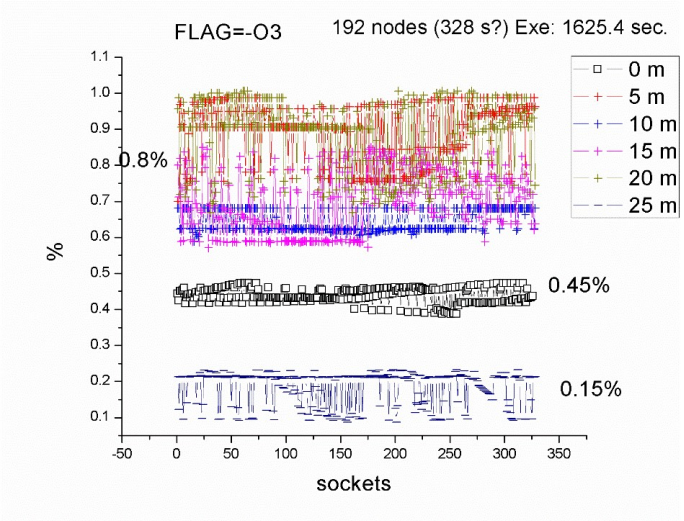


Number of Instruction of each type detected in the execution

BIT1

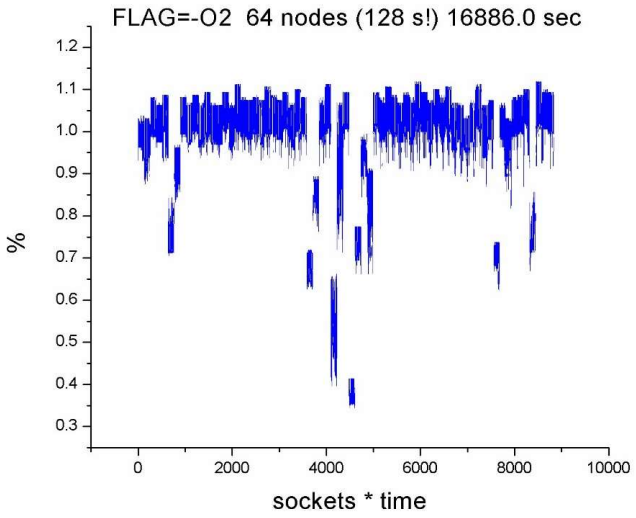






Cases with 64 nodes and 192 nodes.

BIT1

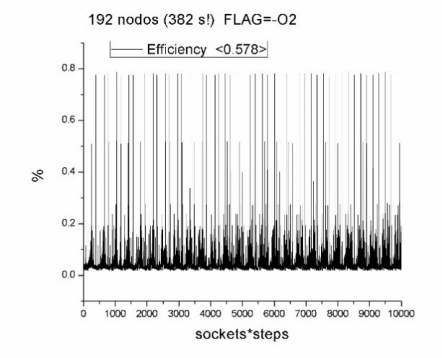


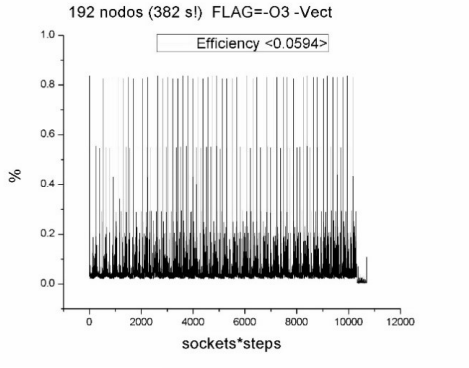
More time steps
Continues the
stable efficiency.

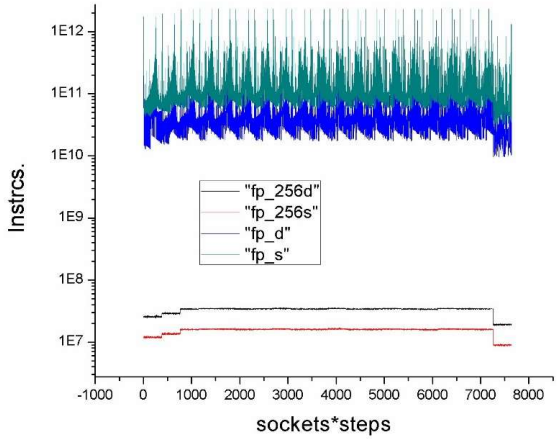
Not relevant
influence of the
O2 or O3 option.

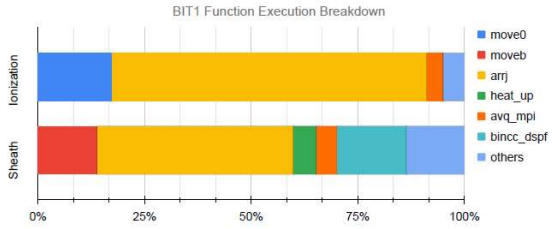
BIT1

Restart files! We reinitiate from time step>1 171 000 000









Influence of the reordering function grows with time steps

General Conclusions

- The first requirements for SPICE2 were completed accomplished. We will generate a real parallel version of the code.
- We complete the requirements for SPEC code and give some recommendation to improvements.
- The efficiency problems of BIT1 was deeply analyzed using HPCMI. New compilations FLAGS turn on those operations, but not relevant increments in efficiency measures.
- For “few” time steps (less than millions) the behavior is acceptable
- For high time steps (>1 170 000 000) very low efficiency. Disbalance of nonlinear plasmas.
- A new load balanced version of BIT1 is under development.

SPICE2d – SPEC – BIT1 efficiency analysis

Many Thanks!

HPC ACHs | 27-28 November 2024

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.