27–28 Nov. 2024

STELLA

Aina Gomez Piñol and Valentin Seitz

2nd Annual Meeting of EUROfusion HPC ACHs

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.

Improve LU decomposition of response matrix

INIT

Compute LU

TIMESTEP

back substitution

back substitution

. . .

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.

LU Decomposition on Marenostrum 4

Scaling-results obtained on Marenostrum 4 of the global LU decomposition

	48	96	192
Global efficiency	83.93	50.65	26.08
-- Parallel efficiency	83.93	75.17	66.86
-- Load balance	97.43	96.97	95.88
-- Communication efficiency	86.14	77.52	69.74
-- Serialization efficiency	95.79	92.78	87.96
-- Transfer efficiency	89.92	83.55	79.28
-- Computation scalability	100.00	67.38	39.01
-- IPC scalability	100.00	103.76	105.45
-- Instruction scalability	100.00	66.81	40.02
-- Frequency scalability	100.00	97.21	92.43

Input options: vmec_filename = 'wout_ref_003.nc', nzed=60, nx = 10, ny = 50

Porting to ScaLAPACK on Marenostrum 5

Scaling-results obtained on Marenostrum 5 of the ScaLAPACK LU decomp.

	112	224	448	896
Global efficiency	84.50	65.62	81.11	57.97
-- Parallel efficiency	84.50	66.17	81.80	58.50
-- Load balance	88.75	86.13	89.52	69.45
-- Communication efficiency	95.22	76.82	91.38	84.24
-- Serialization efficiency	97.40	97.28	93.04	85.50
-- Transfer efficiency	97.76	78.97	98.22	98.52
-- Computation scalability	100.00	99.17	99.15	99.09
-- IPC scalability	100.00	100.20	100.90	101.00
-- Instruction scalability	100.00	100.79	101.33	101.47
-- Frequency scalability	100.00	98.20	96.98	96.69

Input options: vmec_filename = 'wout_ref_003.nc', nzed=60, nx = 10, ny = 50

Summary & Future Work

- ScaLAPACK LU outperforms current implementation (on Marenostrum 5)
- Order of magnitudes less MPI Collectives in ScaLAPACK implementation
- **Tradeoff:** Keep the matrices distributed and add additional communication in timestep **OR** memory overhead of storing them in every process.

- Integrate back substitution and verify correctness (currently only partially done)
- Integrate our work into new code-base (major update released)
- Try SLATE (as a ScaLAPACK successor) to allow hybrid parallelism
- Analyze the newly released version.

27–28 Nov. 2024

ERO2.0

Joan Vinyals Ylla-Català

2nd Annual Meeting of EUROfusion HPC ACHs

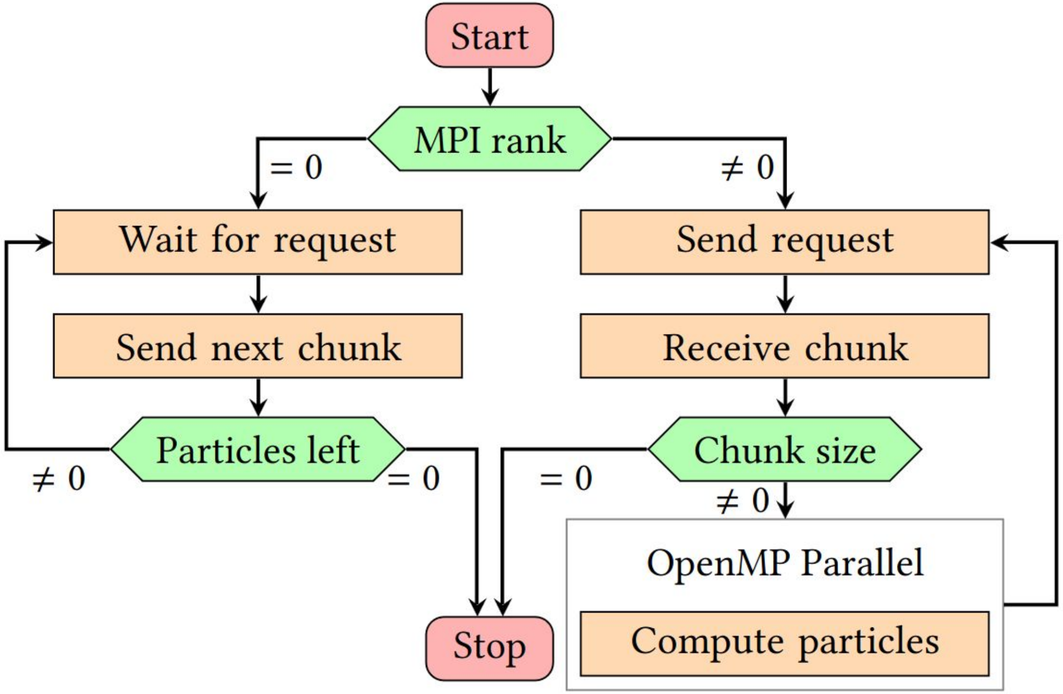
This work has been carried out within the framework of the EUROfusion Consortium, funded by the European Union via the Euratom Research and Training Programme (Grant Agreement No 101052200 — EUROfusion). Views and opinions expressed are however those of the author(s) only and do not necessarily reflect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

Background

ERO2.0 is a 3D Monte-Carlo code for simulating wall erosion and impurity transport through plasma and subsequent redeposition. [1]

[1] J Romazanov, D Borodin, A Kirschner, S Brezinsek, S Silburn, A Huber, V Huber, H Bufferand, M Firdaouss, D Brömmel, et al . 2017. First ERO2. 0 modeling of Be erosion and non-local transport in JET ITER-like wall. Physica scripta 2017, T170(2017), 014018

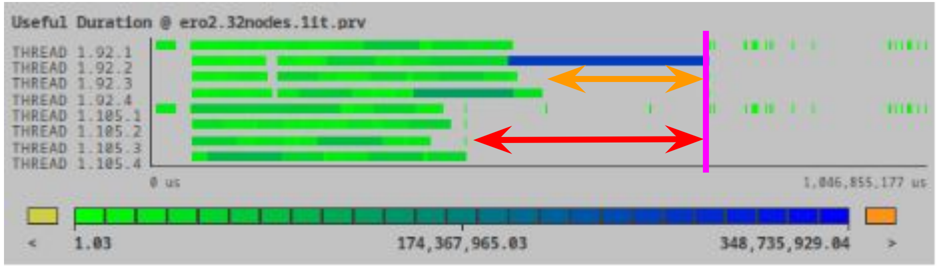
Background



Start
MPI rank
= 0
≠ 0
Wait for request
Send next chunk
Particles left
≠ 0
= 0
Send request
Receive chunk
Chunk size
= 0
≠ 0
OpenMP Parallel
Compute particles
Stop

Background

We observe load imbalance in MPI and OpenMP

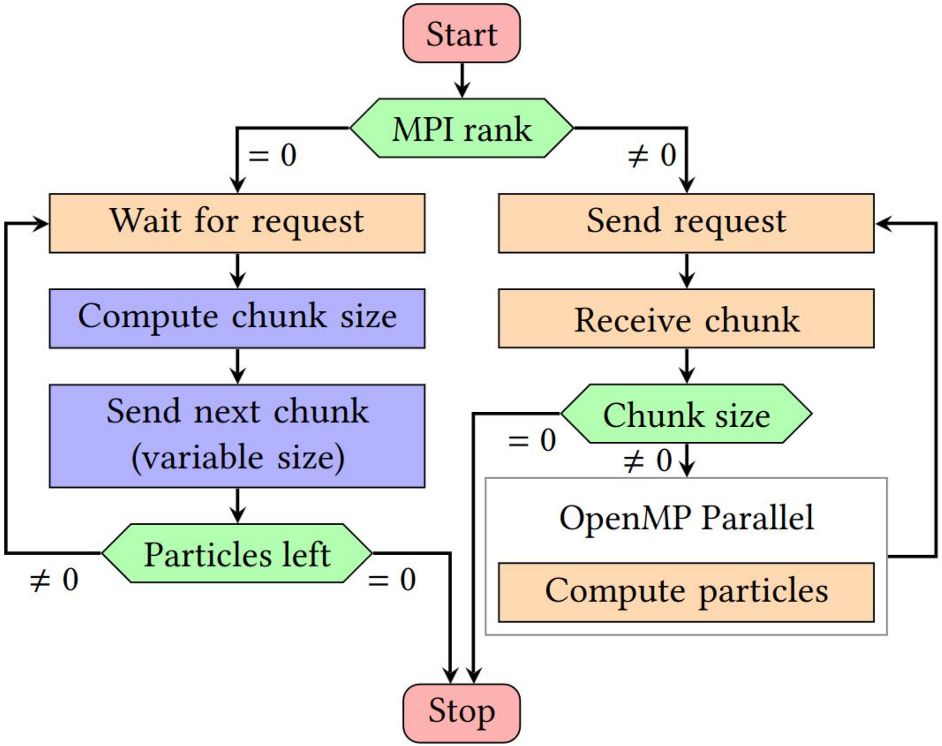


The amount of instruction per computational burst correlates with the useful duration.

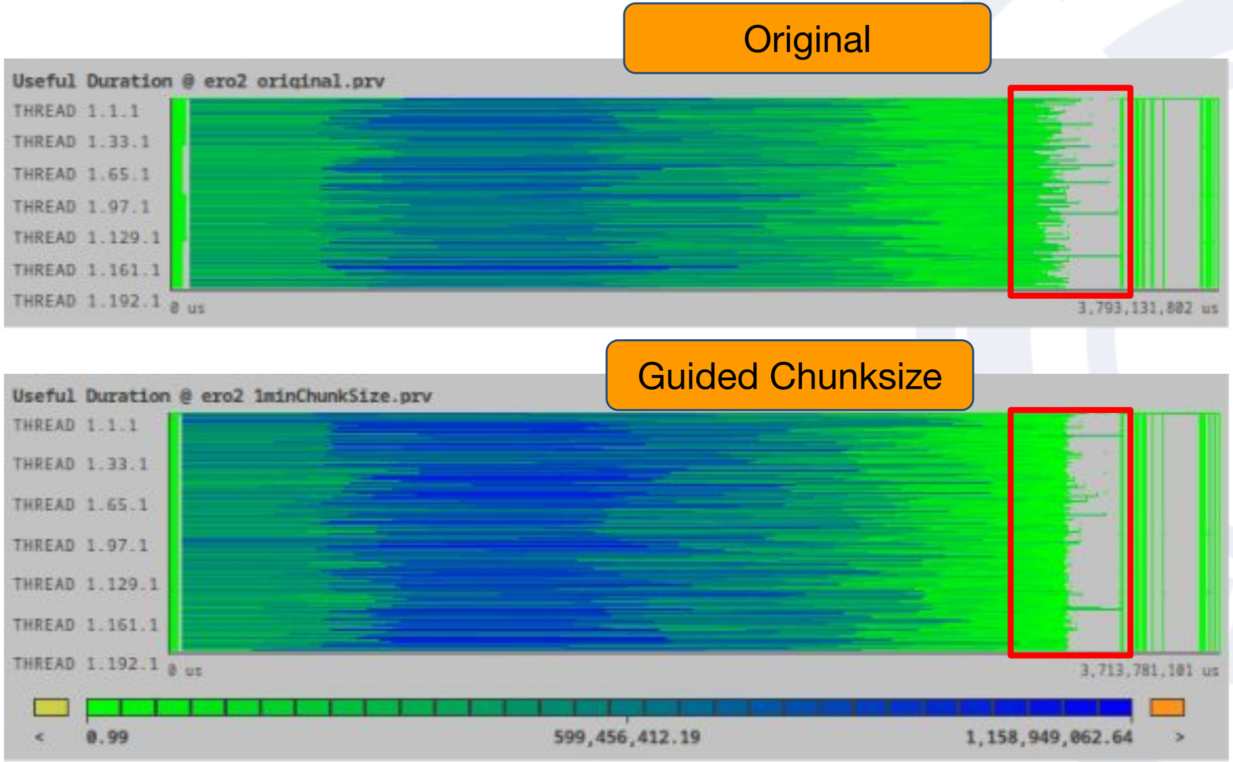
Guided Chunksize

Objectives:

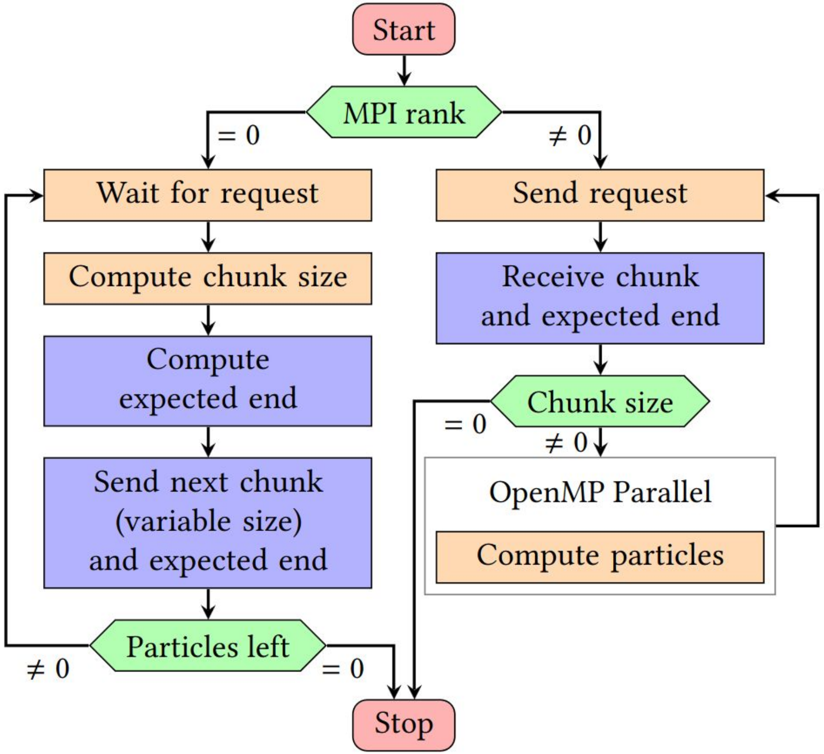
- Reduce grain size to improve load balance.
- Don't increase MPI communication overhead.



Guided Chunksize



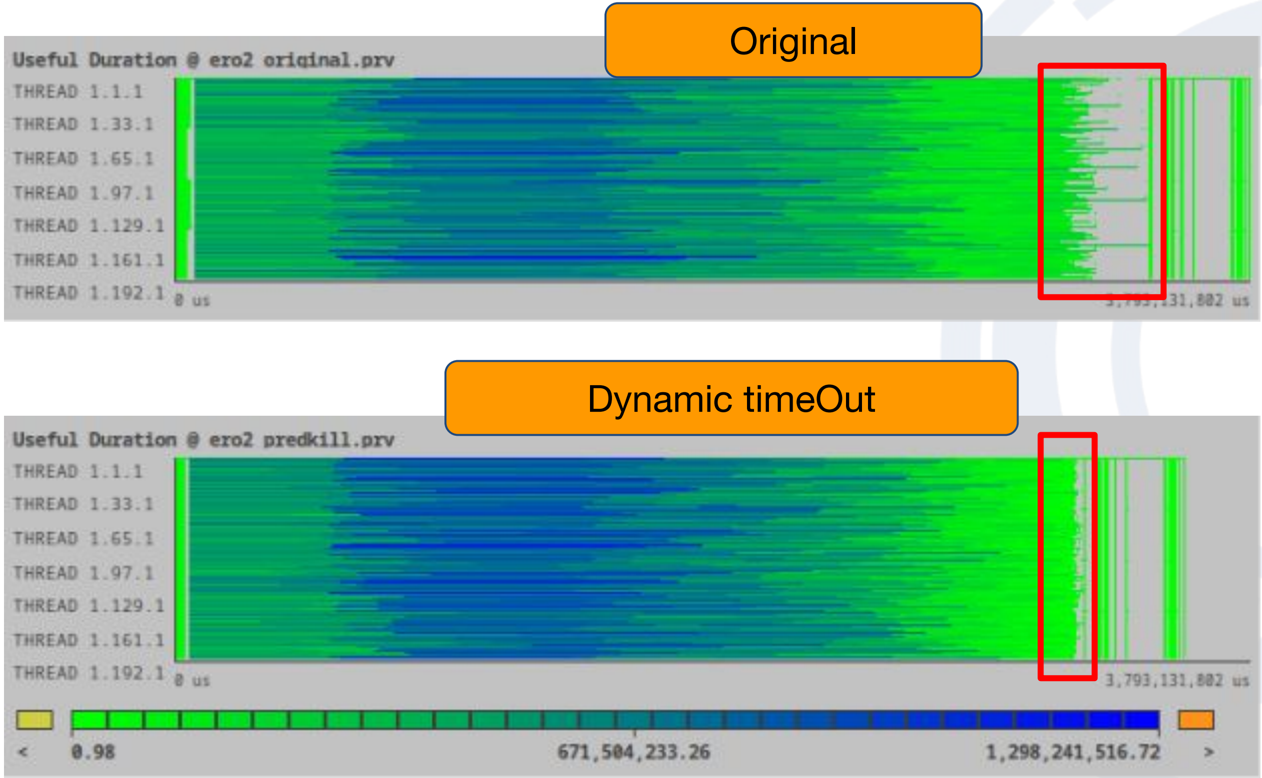
Dynamic timeOut



Objectives:

- Discard particle simulations that are in the critical path of the whole execution.

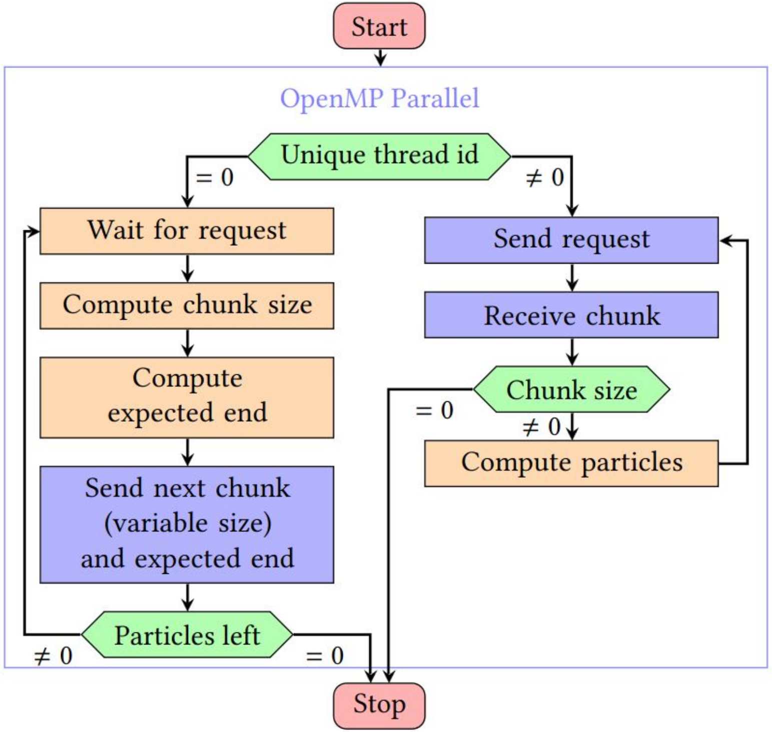
Dynamic timeOut



New OpenMP Schema

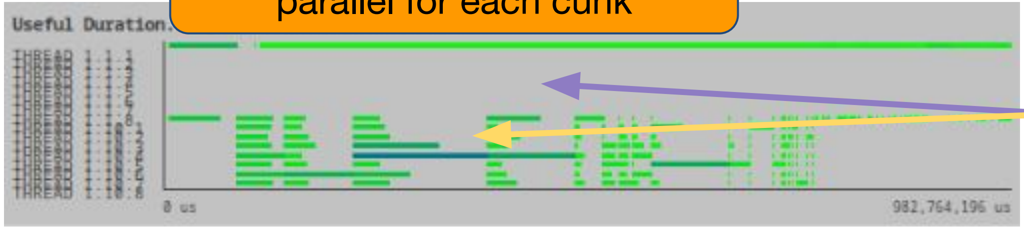
Objectives:

- Remove the implicit barrier in OpenMP parallels to improve load balance.



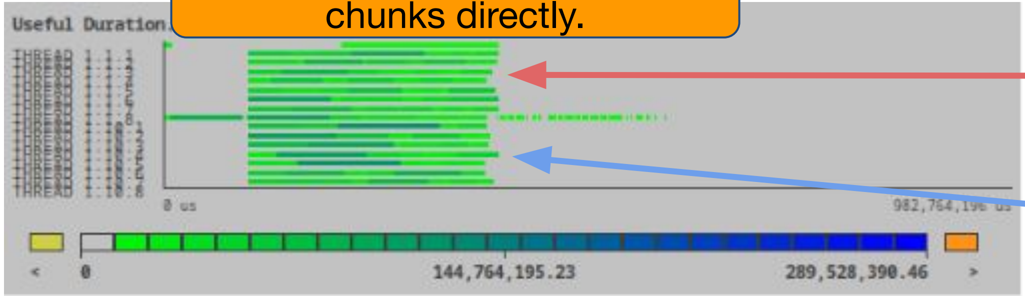
New OpenMP Schema

Worker spawns an OpenMP parallel for each cunk



We observe load imbalance for each parallel region, and none of the Rank 0's threads do any computation.

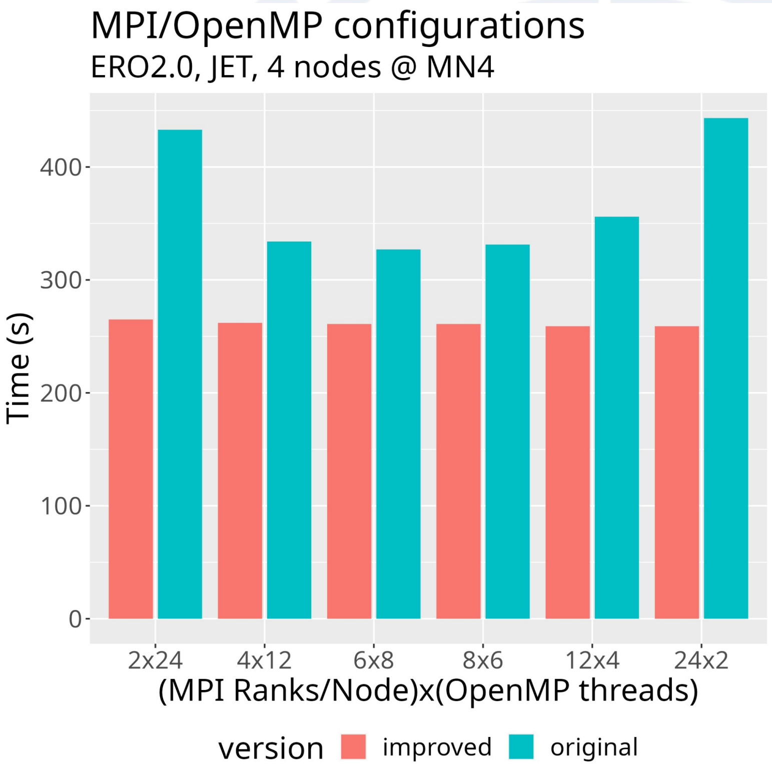
OpenMP threads receive chunks directly.



MPI Rank 0 does take part in the computation.

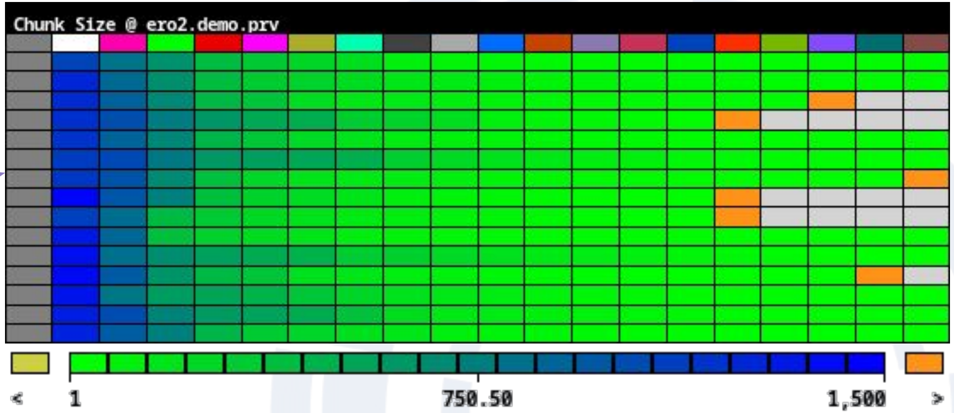
No load imbalance observed during the computation phase.

Scalability results

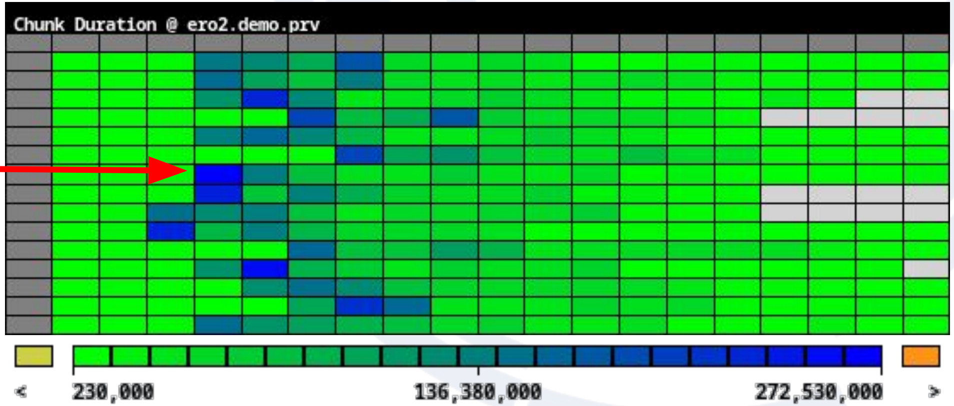


DEMO Input

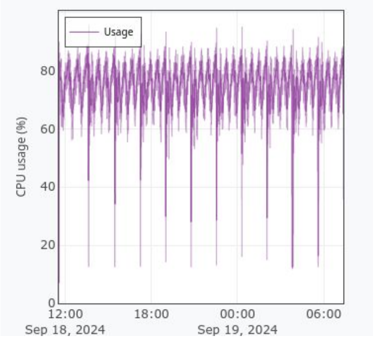
Number of particles sent per chunk decreases according to the guided policy.

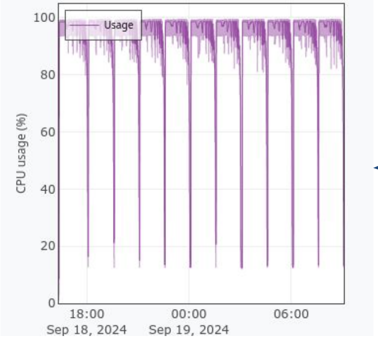


However, the chunks with higher load arrive later, where we don't have as much malleability.



Results





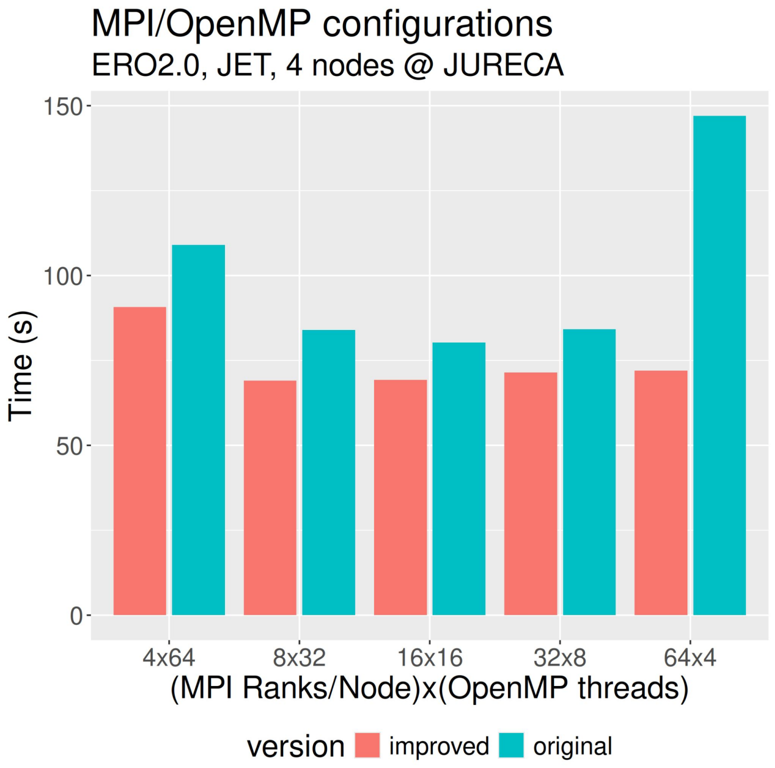
Load is evenly distributed within the step.

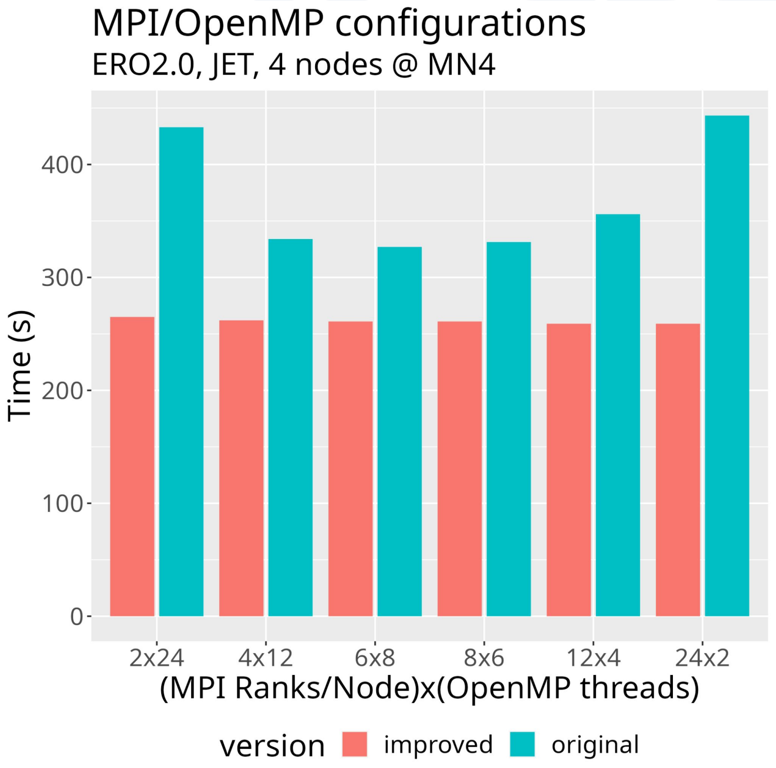
Execution with the master branch code.

With the optimized branch code, we achieve at least 18% speedup.

Runtime	#Nodes	Usage	#Cores	#PhysCores	#LogicCores	Load	MaxMem
filter	filter	filter	filter	filter	filter	filter	filter
19h49m	10	74.47	124.57	124.57	0.00	93.12	35.45
16h43m	10	93.62	124.18	124.18	0.00	117.14	37.39
16h26m	10	93.90	124.71	124.71	0.00	117.35	37.91

MPI/OpenMP Configuration Results





Conclusions

- Currently working on **verification** with bigger input sets (DEMO).
- Performance improvements
    - Achieve up to **3x speedup** with a JET input with Marenostrum4.
    - Preliminary results on the DEMO input achieve up to **18% speedup** in Jureca.
- **Analyzed** ERO2.0 code parallelization.
    - Identified main bottleneck - **Load Imbalance**.
    - Original code already considered the Load Imbalance.
    - Then our optimizations focussed on improving the load balancing strategy.
- **Improved** the code by
    - **Guided chunksize**: **Improving granularity** without compromising MPI Overhead!
    - **Dynamic timeout**: **Intelligently discarding particles** that become a bottleneck.
    - **New OpenMP parallelization**: Use MPI in OpenMP, to remove OpenMP load imbalance.

27–28 Nov. 2024

ERO2.0 & STELLA

Joan Vinyals Ylla-Català joan.vinyals@bsc.es
Valentin Seitz valentin.seitz@bsc.es
Marta Garcia Gasulla marta.garcia@bsc.es

2nd Annual Meeting of EUROfusion HPC ACHs

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.