Characterizing the Performance of the Implicit Massively Parallel Particle-in-Cell iPIC3D Code

SC23 Denver, CO | i am hpc.

Jeremy J. Williams
Department of Computer Science, EECS, KTH, Stockholm, Sweden
jjwil@kth.se

Daniel Medeiros
Department of Computer Science, EECS, KTH, Stockholm, Sweden
dadm@kth.se

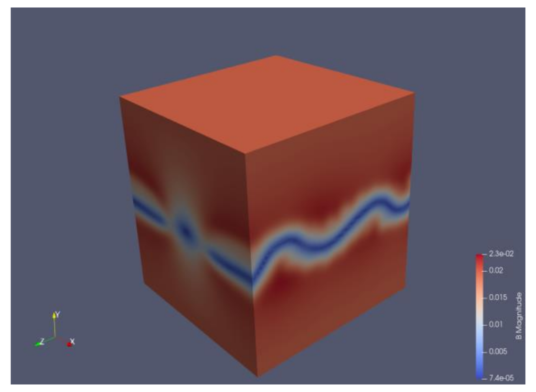
Ivy B. Peng
Department of Computer Science, EECS, KTH, Stockholm, Sweden
bopeng@kth.se

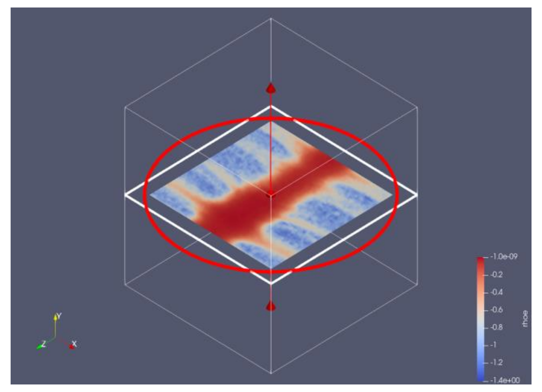
Stefano Markidis
Department of Computer Science, EECS, KTH, Stockholm, Sweden
markidis@kth.se

Funded by the European Union

iPIC3D

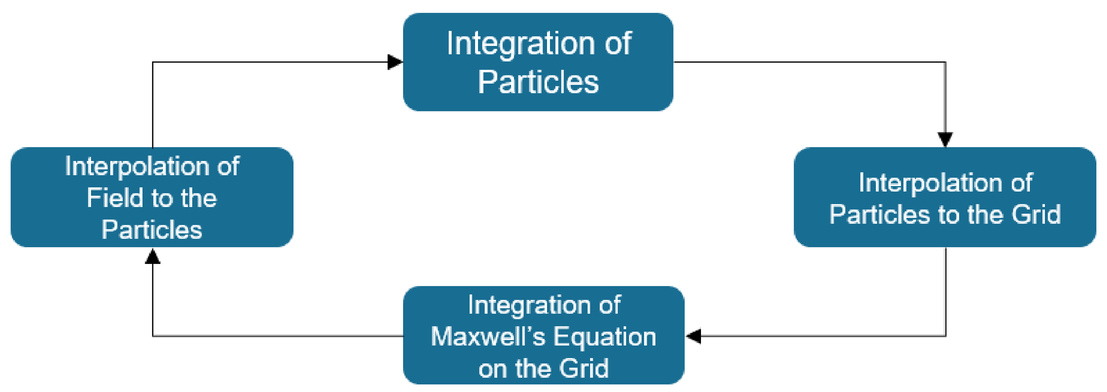
iPIC3D, a highly respected Particle-in-Cell (PIC) code, is known for its ability to simulate plasma phenomena in three dimensions. It serves as a powerful tool designed to uncover the secrets of plasma dynamics and the complex interactions between electromagnetic fields and charged particles [**2**].





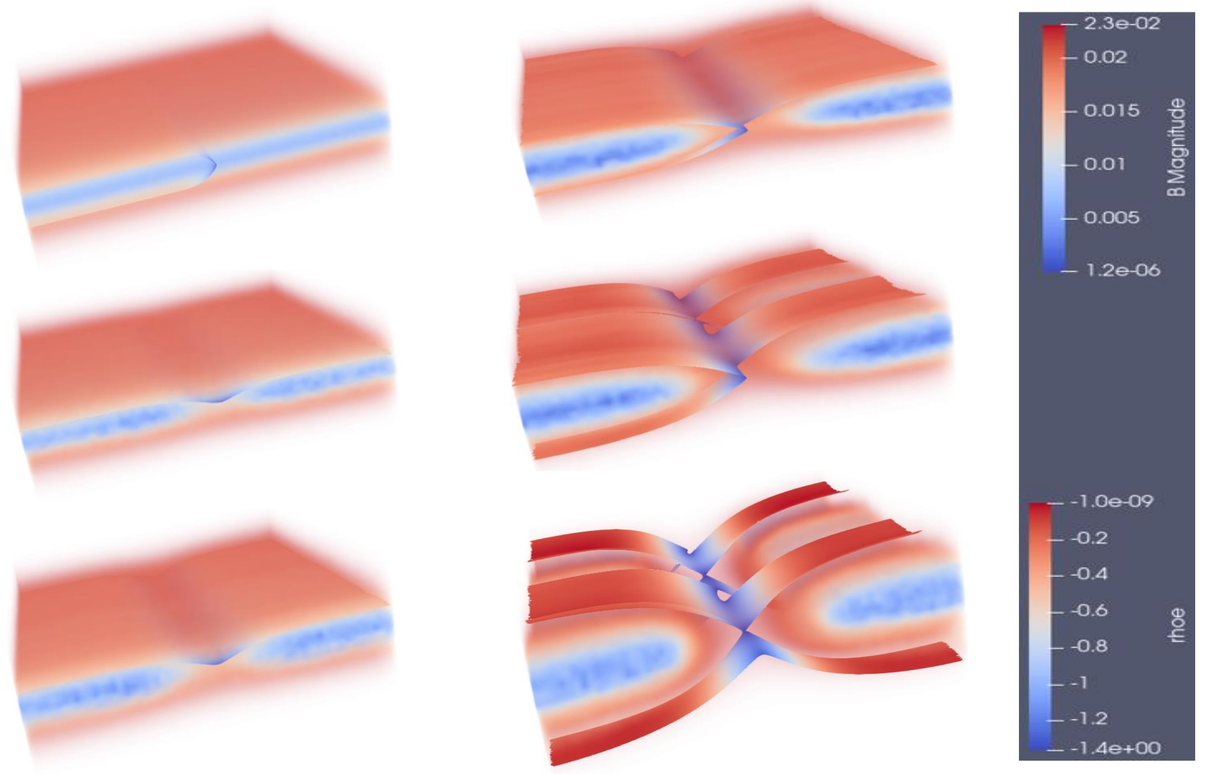
Efficient parallelization is critical in **Implicit PIC (iPIC)** simulations where the costs of particle movement and field solving are of the same order (unlike **Explicit PIC** where particle-related computations dominate). To achieved optimal performance, both field solving and particle movement must be effectively parallelized [**2**].

The **iPIC** method follows a typical cycle consisting of four primary steps:



In this work, our aim is to gain a comprehensive understanding of the performance and activity dynamics of 3D magnetic reconnections in **iPIC3D** plasma simulations. We will leverage advanced profiling and tracing tools to identify areas that can be optimized and enhanced in the field of plasma physics [**1**].

GEM Challenge Test Evolution



Advanced Tools

CRAY APPRENTICE2

BSC

DARSHAN
HPC I/O Characterization Tool

Environments

- **Greendog,** a workstation with an i7-7820X processor (8 cores), 32 GB DRAM, and one NVIDIA RTX2060 SUPER GPU.

- **Dardel,** a HPE Cray EX supercomputer, with 1270 compute nodes. Each node used is equipped with 256GB DRAM and two AMD EPYC Zen2 2.25 GHz 64 core processors per node, for a total of 128 cores per node. The GPU partition consists of 56 nodes, each with a specialized node architecture.

Cache Usage and Tracing

50% Increase Size (6 6 6 6)	Baseline Size (4 4 4 4)	50% Reduction Size (2 2 2 2)	CacheTest Size
L1 Ddcache Load Misses	L1 Ddcache Load Misses	L1 Ddcache Load Misses	L1 Ddcache Load Misses
1.99%	2.22%	3.79%	5.53%
LLC load misses	LLC load misses	LLC load misses	LLC load misses
54.75%	58.03%	47.95%	18.95%

Table 1: Using **perf** to reveal and extract cache load misses in percentages

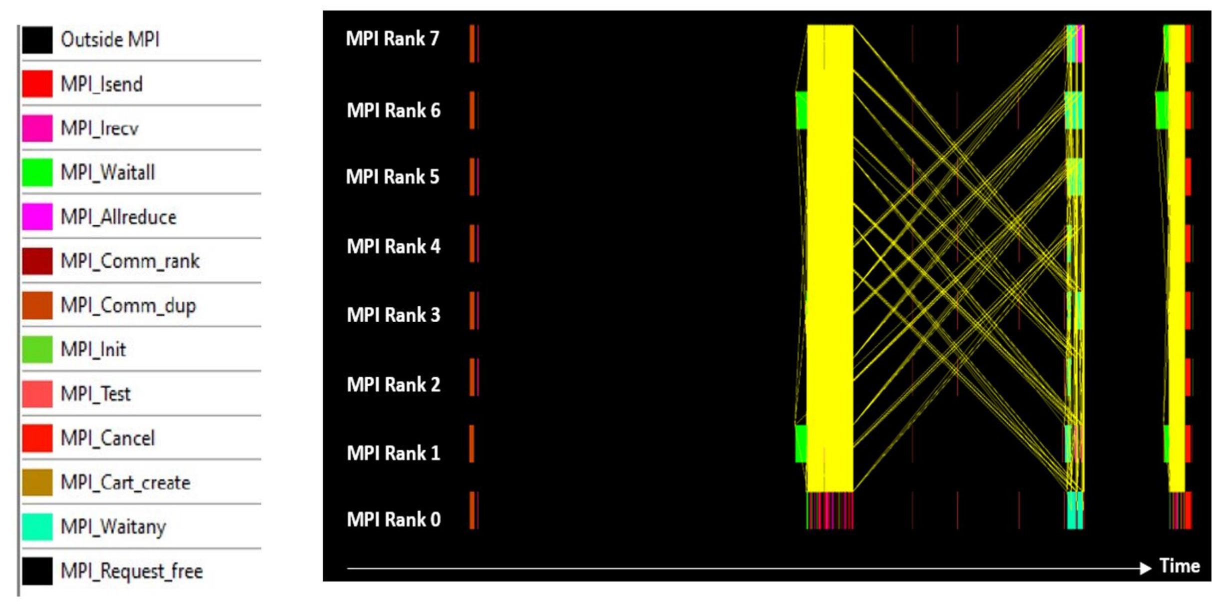


Fig 1: Using **Extrae & Paraver** to reveal **iPIC3D's** full simulation (up to 8 ranks) from one cycle with communication lines (yellow) and MPI functions used.

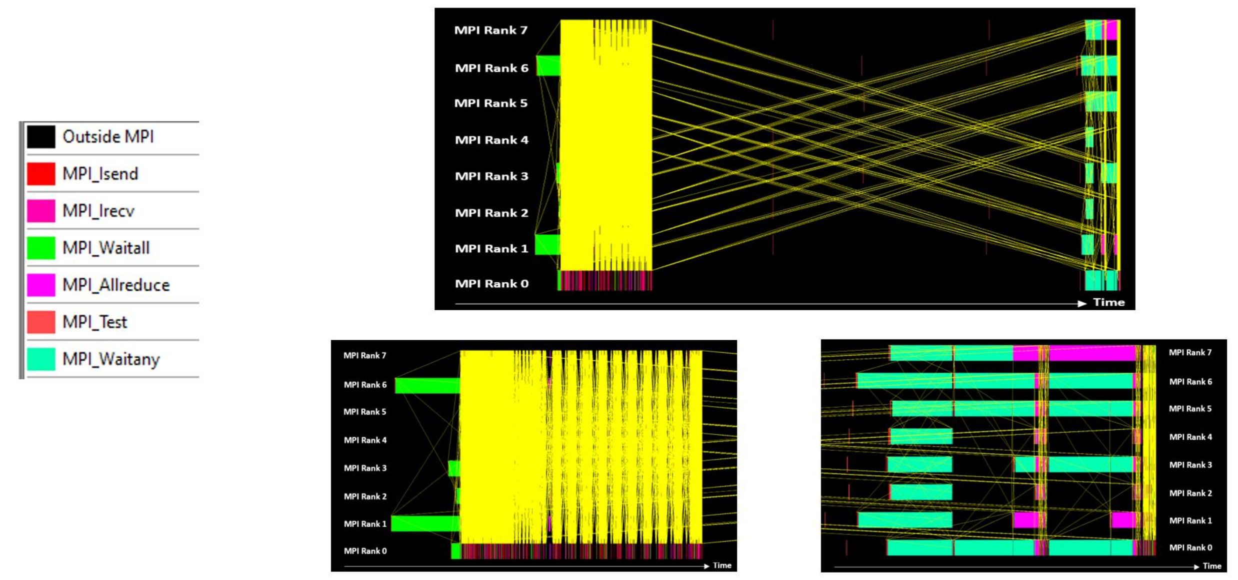


Fig 2: Using **Extrae & Paraver** to reveal a close-up view of workload imbalance significantly impacting **iPIC3D's** full simulation communication efficiency.

Call Functions and Profiling

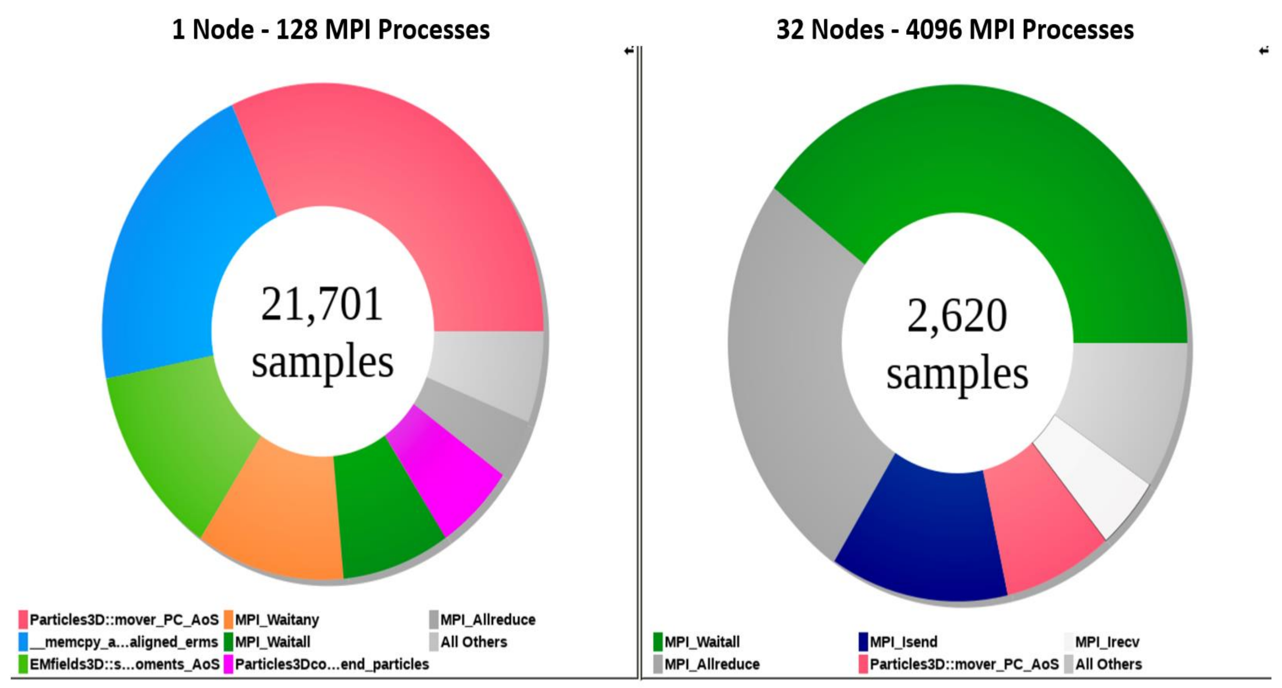


Fig 3: Using **CrayPAT & Apprentice2** to reveal **iPIC3D's** profile by call functions that have significant exclusive hits, averaged across ranks.

Scalability and Efficiency

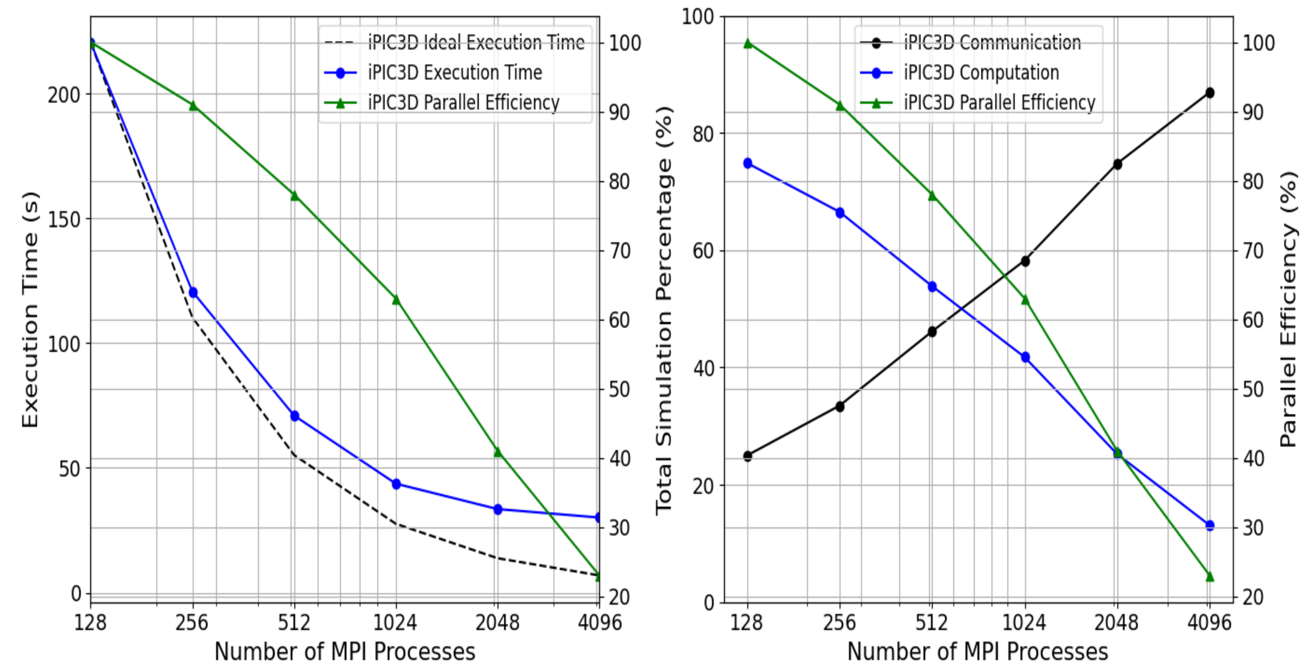


Fig 4: Using **CrayPAT** to reveal **iPIC3D's** execution time, communication, computation and parallel efficiency.

I/O Characterization

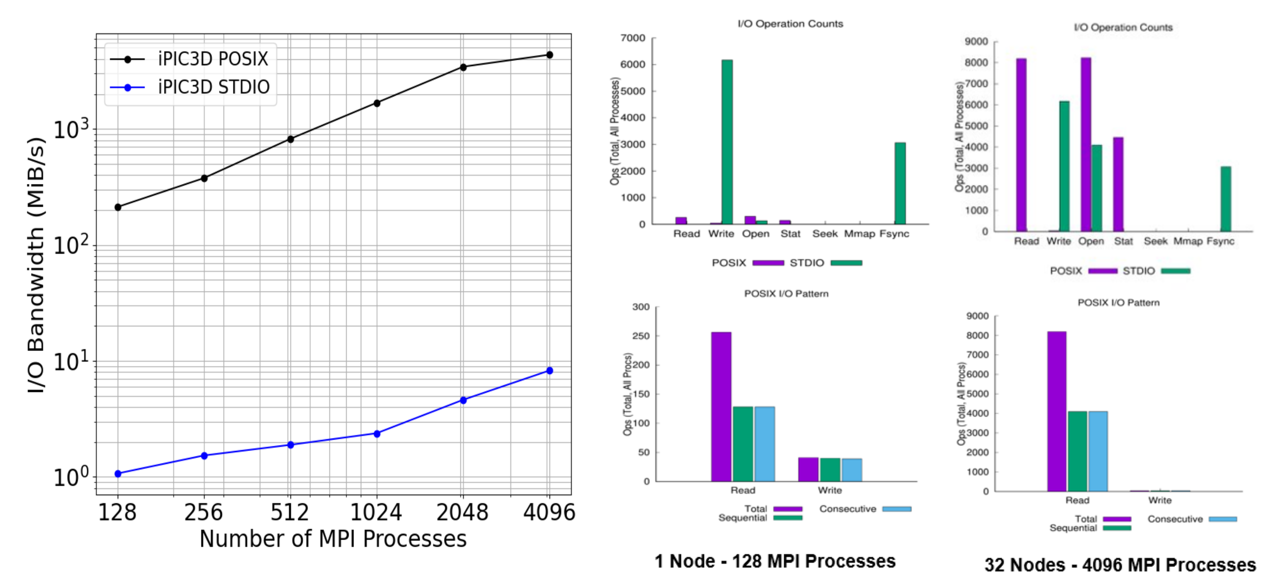


Fig 5: Using **Darshan** to reveal **iPIC3D's** I/O bandwidth and behavior.

Conclusions & Future Work

Conclusions:

- Communication identified as a critical factor impacting **iPIC3D's** efficiency on large runs
- The presence of MPI_Waitall was noted to hinder execution and slow progress, with file I/O operations (POSIX and logging) adding to **iPIC3D's** performance overhead.
- Suggested exploring alternative algorithms and data structures to reduce communication overhead in **iPIC3D's** plasma simulations.

Future Work:

- **Optimal Node Placement**: Minimizing communication delays and improving efficiency.
- **Communication and Computation Overlap**: Maximizing resource utilization and reducing idle time.
- **Load Balancing**: Optimizing performance and resource allocation

References

[**1**] Williams, J.J., Tskhakaya, D., Costea, S., Peng, I.B., Garcia-Gasulla, M., Markidis, S. (2023). Leveraging HPC Profiling Tracing tools to Understand the Performance of Particle-in-Cell Monte Carlo Simulations. arXiv:2306.16512 [cs.DC]

[**2**] Markidis, S., & Lapenta, G. (2010). Multi-scale simulations of plasma with iPIC3D. Mathematics and Computers in Simulation, 80(7), 1509-1519.

[**3**] Birn, J., Drake, J. F., Shay, M. A., Rogers, B. N., Denton, R. E., Hesse, M., ... & Pritchett, P. L. (2001). Geospace Environmental Modeling (GEM) magnetic reconnection challenge. Journal of Geophysical Research: Space Physics, 106(A3), 3715-3719.

This work has received funding from the European High Performance Computing Joint Undertaking (JU) and Sweden, Finland, Germany, Greece, France, Slovenia, Spain, and Czech Republic under grant agreement No 101093261

KTH VETENSKAP OCH KONST

KTH-HPC

iPIC3D GitHub