Leveraging HPC Tools to Understand the Performance and Impact of openPMD on BIT1, a Particle-in-Cell Monte Carlo Code, through Instrumentation, Monitoring, and In-Situ Analysis

J. J. Williams[1], S. Costea[2], D. Tskhakaya[3], L. Kos[2], A. Podolnik[3], J. Hromadka[3],
M. Garcia-Gasulla[4], A. D. Malony[5], K. Huck[5], E. Laure[6], I. B. Peng[1], and S. Markidis[1]

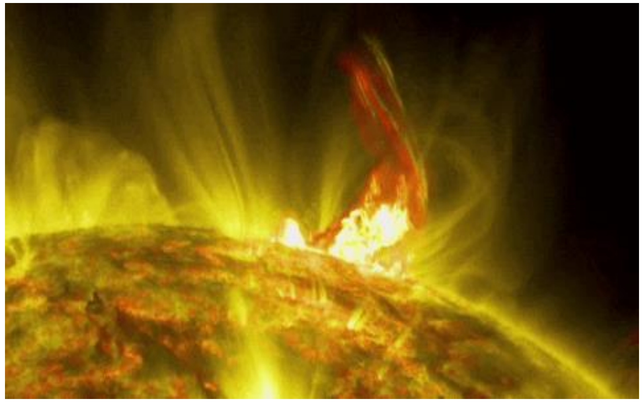
[1]*KTH Royal Institute of Technology, Stockholm, Sweden*
[2]*LeCAD, University of Ljubljana, Ljubljana, Slovenia*
[3]*Institute of Plasma Physics of the CAS, Prague, Czech Republic*
[4]*Barcelona Supercomputing Center, Barcelona, Spain*
[5]*University of Oregon, Eugene, Oregon, The United States of America*
[6]*Max Planck Computing and Data Facility, Garching and Greifswald, Germany*

Outline

- Motivation
- Background
- Question
- Environments
- Evaluation and Results
- Conclusion and Current Work

Motivation

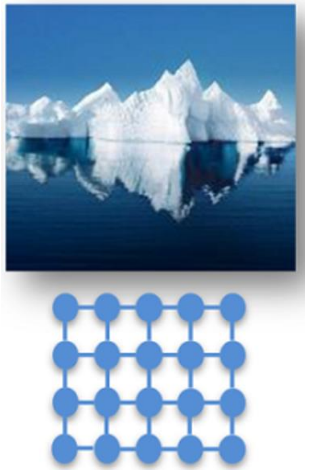
Motivation: Plasma (4^{TH} State of Matter) and Particle Energy



Solar Flares

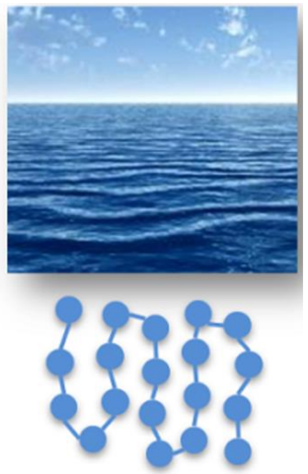
Northern Lights

Plasma TVs



Solid

Particles fixed in a grid



Liquid

Particles moving freely

Gas

Particles moving randomly and expanding

Plasma

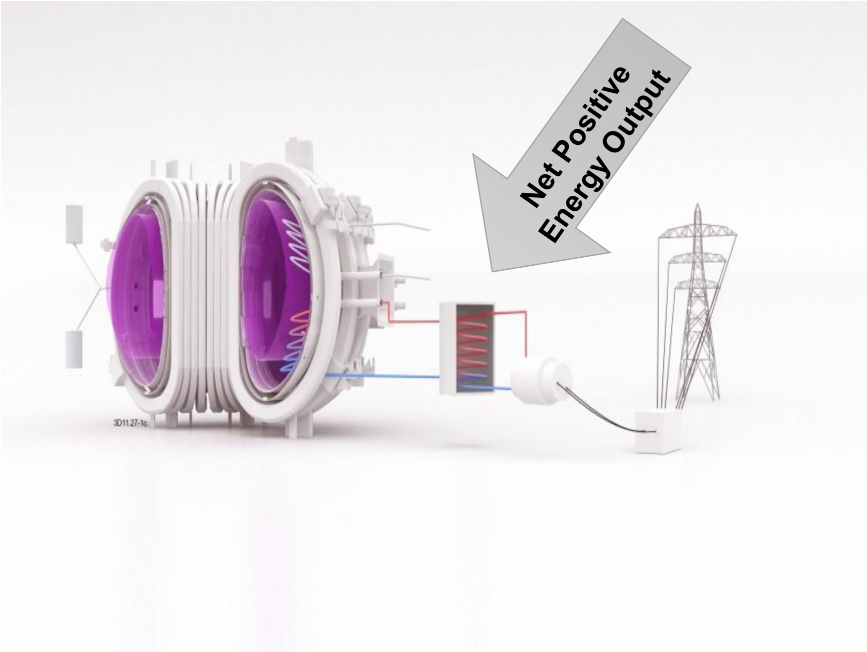
Charged particles, ions and elections, interacting

Particle Temperature → Particle Temperature → Particle Temperature → Particle Temperature

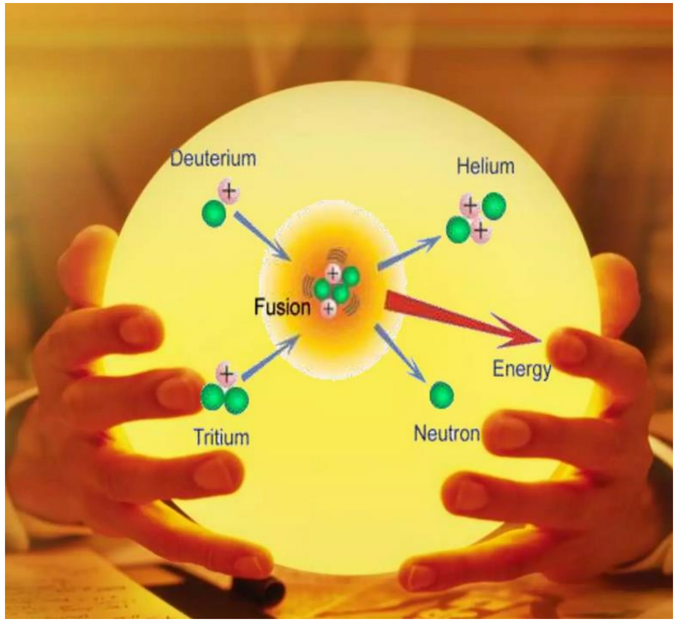
Motivation: Fusion Energy and The Grand Challenge

ITER [1]

Next Generation
Magnetic Fusion Device



Fusion Reactor



Fusion Energy

Grand Challenge:
Generate More Energy From Fusion than Invested in Starting and Maintaining it.

[1] International Thermonuclear Experimental Reactor, "What is ITER?", Accessed 30 June 2024. https://www.iter.org/proj/inafewlines

Motivation: Petascale to Exascale Supercomputers[1]

10 15

*__First__ EU __Petascale__
Supercomputer

VEGA
2021

Peak Performance: ***10.1** petaflops

Fastest EU **Pre-Exascale**
Supercomputer

LUMI
2022

Peak Performance: ****539.13** petaflops

*Petascale

**Pre-Exascale

Applications:

- Traditional Computational AI (including Deep Learning) and Big Data/HPDA
- Large-Scale Data Processing with Massive Scale Data Analytics

10 18

***__First__ EU __Exascale__
Supercomputer

JUPITER
2024/5

Booster Module: Close to **24,000** NVIDIA GH200 **GPUs**
Performance (Set To Achieve): Just over *****1 exaflops**

Application **Codes** will (**primarily**) need:

- **Efficient** Use Of Exascale **Parallelism** and **Scalability**
- **Efficient** Use Of **Complex Memory Hierarchies**
- **Efficient** Use of **I/O** for **Storage**, **Retrieval**, and **Analysis**

Applications:

- Most Demanding Simulations and Compute-Intensive AI Applications

***1 petaflop** - **1 quadrillion (10^{15}) FLOPS (or 1,000 teraflops)**

*****1 exaflop** - **1 quintillion (10^{18}) FLOPS (or 1,000 petaflops)**

[1] The European High Performance Computing Joint Undertaking (EuroHPC JU). "European Commission: Our Supercomputers", Accessed 30 June 2024. https://eurohpc-ju.europa.eu/supercomputers/our-supercomputers_en

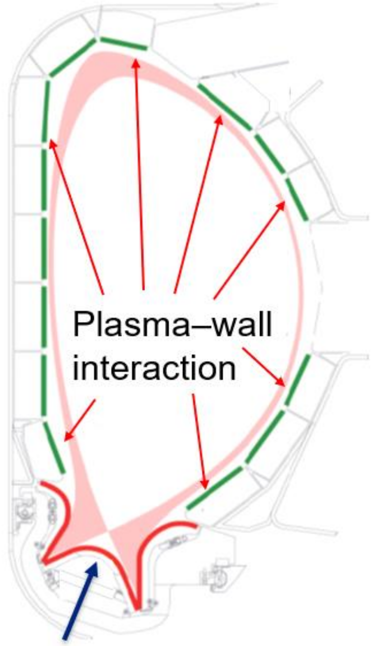
Motivation: BIT Grand Challenge and Exascale Readiness

BIT (BIT1 and BIT3) – Particle-in-Cell (PIC) Code

- Controlling the Complexities of **Plasma-Material Interfaces** for the Next-Generation Fusion Devices.

BIT Exascale Readiness

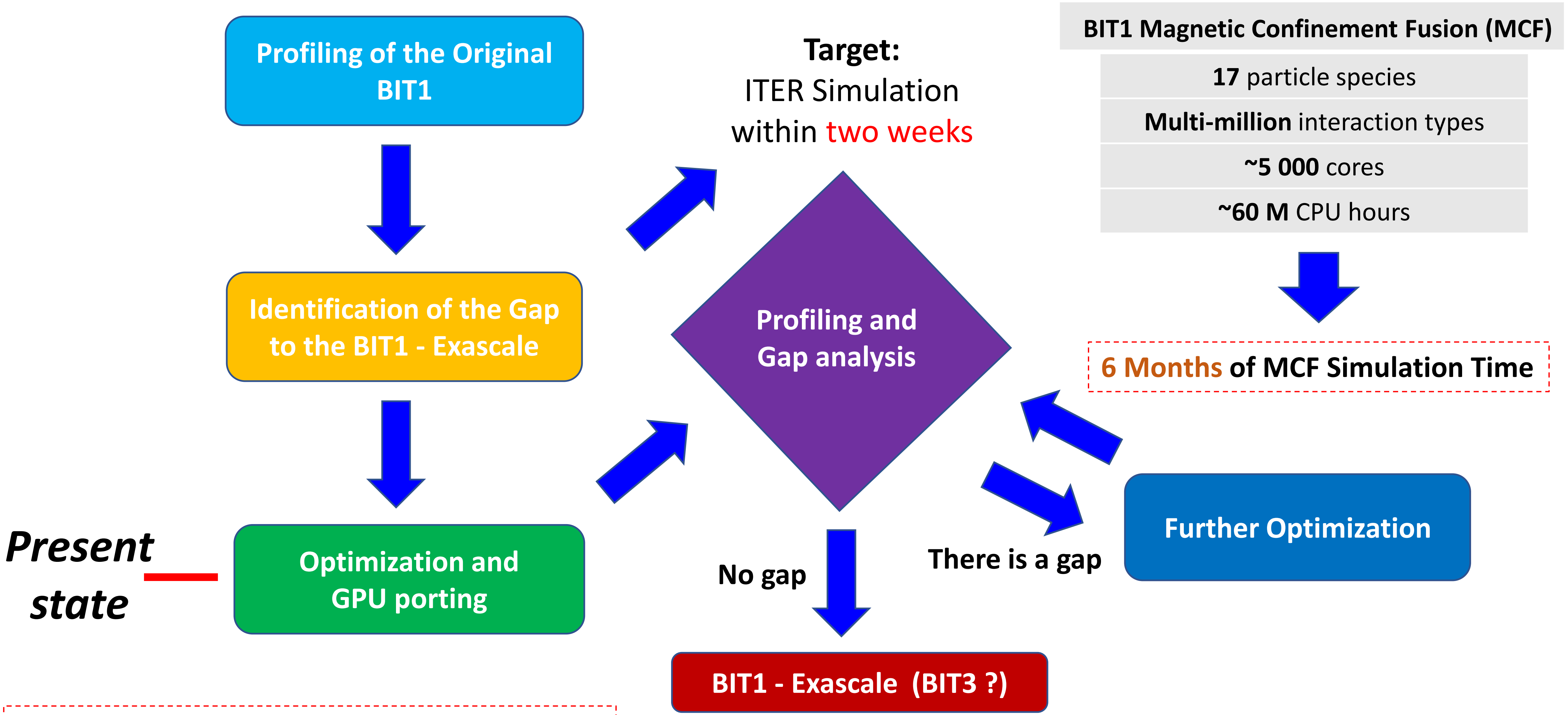
- **Investigate**, **Characterize**, and **Understand** the **Computing Performance**
  - Main Tasks:
    - ✓ **Identify** Compute-Intensive Areas
    - ✓ **Analyze** Single-Node Performance
    - ✓ **Evaluate** MPI Communication, Load Balancing, and Scaling
    - ✓ **Characterize** I/O and **Identify** Performance Enhancements



BIT1* simulates **plasma behavior** in the **tokamak divertor** (**blue arrow**), such as in the **ITER** fusion device.

Jet Discharge
Edge Part of the **Tokamak**
"High **Temperature** & **Pressurized** Plasma"

Motivation: Enabling BIT1* and Exascale Workflow

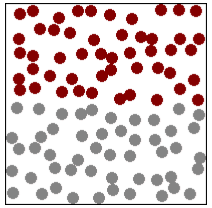


*1 exaflop - 1 quintillion (10^{18}) floating-point operations per second

Background

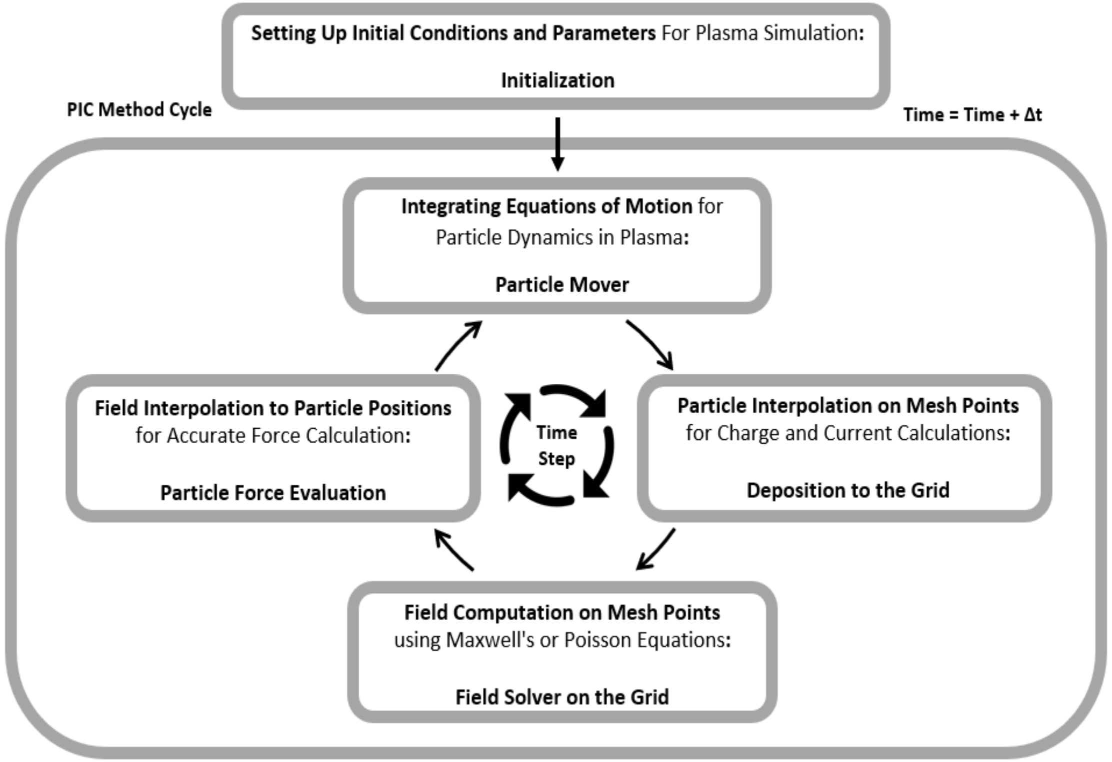
Background: Particle-in-Cell Method and HPC Architectures[1,2]

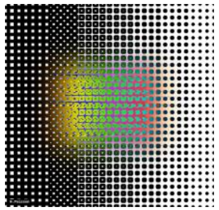
Particle-in-Cell (PIC) codes are widely used to simulate **plasma behavior**, involving an **initialization** phase and **four** main phases on HPC architectures.



(1) Particle Mover:
Solve equations of motion for particles to update their positions and velocities.

(2) Deposition to the Grid:
Distribute particle charges and currents onto a fixed mesh to calculate densities.





(3) Field Solver on the Grid:
Obtain electromagnetic / electrostatic fields by solving Maxwell's / Poisson's equations.

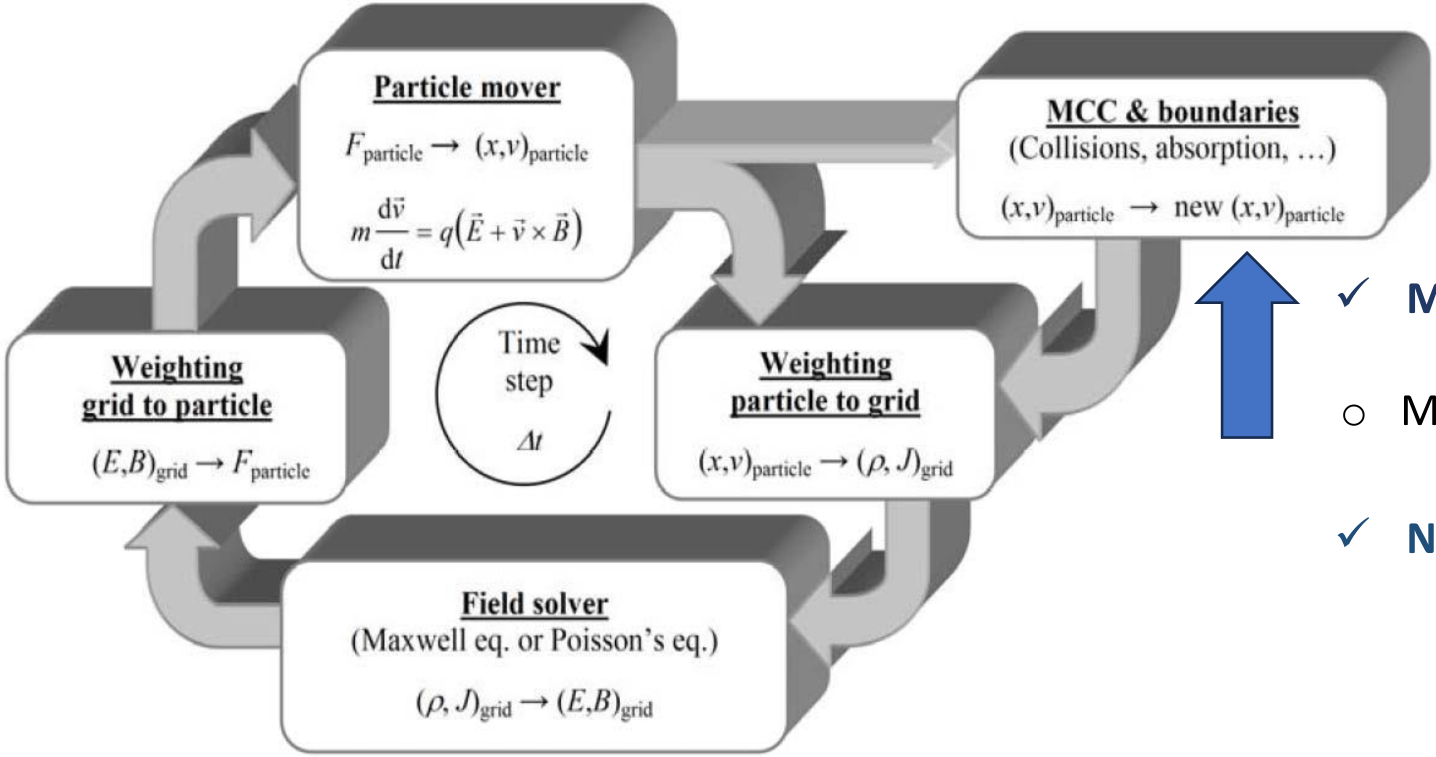
(4) Particle Force Evaluation:
Interpolate fields back to particle locations to evaluate forces.

The **PIC Method** repeats these steps for **subsequent time steps** to simulate plasma behavior over time.
Each step corresponds to a **small fraction** of the characteristic plasma **timescale** for accurate dynamics.

[1] J. J. Williams et all. (2024). "A Comprehensive Review of High-Performance Particle-in-Cell Codes for Large-Scale Plasma Simulations." Department of Computer Science, EECS, KTH. Working Paper.
[2] J. J. Williams et al. "Leveraging HPC Profiling & Tracing Tools to Understand the Performance of Particle-in-Cell Monte Carlo Simulations." Euro-Par 2023: Parallel Processing Workshops: Euro-Par 2023 International Workshops, Limassol, Cyprus, August 28–September 1, 2023, Revised Selected Papers, Part I, LNCS 14351. Springer Nature

Background: BITx, x=1,3 Code Structure

1D3V (BIT1) and 3D3V (BIT3) electrostatic PIC + Monte-Carlo Collisions (MCC)



- Originated from **XPDP1** code from Berkeley University [1]

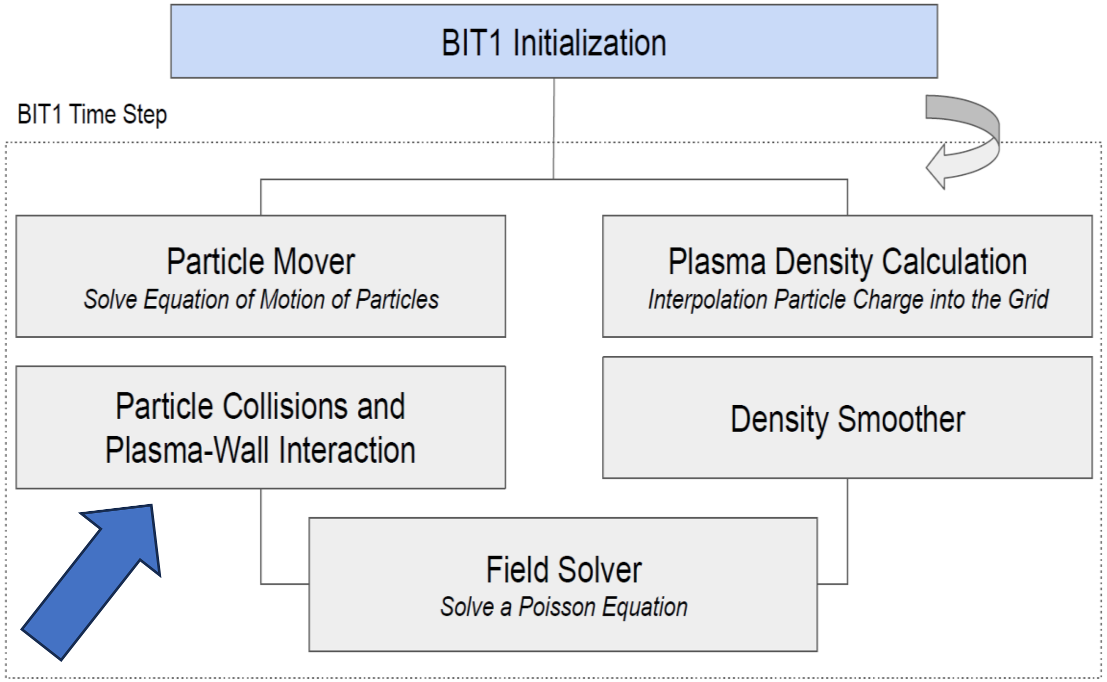
✓ **MCC Model**
- Fast, no limits on collision types [3],
- Multi-step collision processes for high density plasmas [4]

✓ **Natural Sorting with cell-based particle indexing**
- Optimal use of the cache hit,
- Easy space decomposition and
- No limitation on the size of the system[5]

✓ **BIT1 - Physics based Poisson solver**
accurate, fast and highly scalable [2]

[1] J.P. Verboncoeur, et. al., J. Comput. Phys., 104 (2), 321 (1993).
[2] David Tskhakaya, et al., 18th Euromicro Conference proceedings (2010)
*[3] David Tskhakaya, et al., Contr. Plasma Phys., **48** (2008)*
[4] David Tskhakaya, Eur. J. Phys. D., online (2023)
*[5] David Tskhakaya, et al., J. of Comp. Phys., **225** (2007)*

Background: BIT1 (PIC-MCC) Simulations and Input Parameters[1]



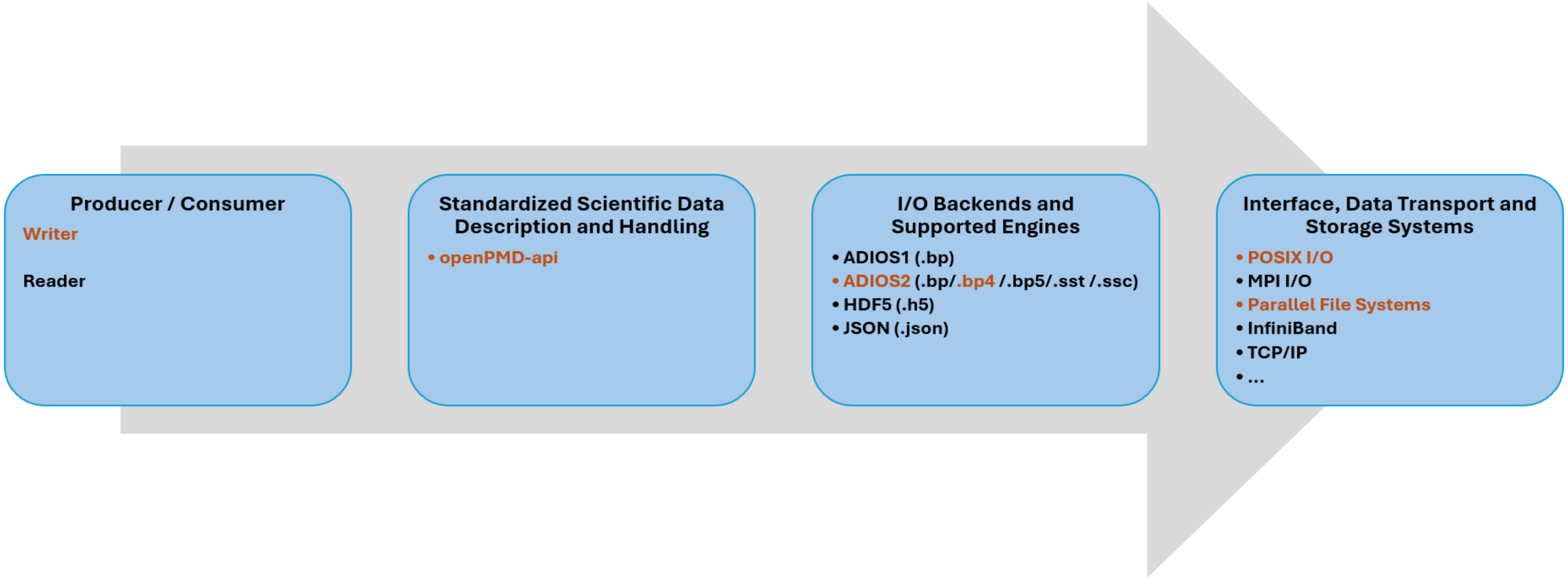
- **Particle-In-Cell code with Monte-Carlo Collisions (PIC-MCC)**
  - **1D3V**
    - x**[s][c][p]**
    - v_x **[s][c][p]**
    - v_y **[s][c][p]**
    - v_z **[s][c][p]**
    - s = **species** (electrons, ions, neutrals)
    - c = **cell index** (cell-based particle indexing)
    - p = **particle index**
  - Domain **decomposition** scheme
  - Data layout **particle information** stored in **memory**
  - MPI **communication** between processes (**non-blocking**)

- **Two Key** Plasma Diagnostic **Flags**[1]
  - **mvflag**: Activates and Enables **time-dependent** diagnostics.
  - **mvStep**: Counts the **time steps** for the interval between **time-dependent** diagnostics.

The **PIC-MCC Method** simulates plasma behavior over time.

[1] J. J. Williams et al. "Leveraging HPC Profiling & Tracing Tools to Understand the Performance of Particle-in-Cell Monte Carlo Simulations." Euro-Par 2023: Parallel Processing Workshops: Euro-Par 2023 International Workshops, Limassol, Cyprus, August 28–September 1, 2023, Revised Selected Papers, Part I, LNCS 14351. Springer Nature

Background: BIT1 **openPMD** Standard & **openPMD-api** Integration[1,2,3,4]



openPMD standardizes file formats for efficiently storing and exchanging data.

openPMD BP4 (Binary Pack 4) ADIOS2 backend prioritizes I/O efficiency at a large scale.

open Particle-Mesh Data

Adaptable Input/Output System 2

[1] A. Huebl et al. openPMD: A meta data standard for particle and mesh based data. https://doi.org/10.5281/zenodo.591699. (2015)
[2] A. Huebl et al. openPMD-api: C++ & Python API for Scientific I/O with openPMD. https://doi.org/10.14278/rodare.27. (2018)
[3] F. Poesche et al. Transitioning from file-based HPC workflows to streaming data pipelines with openPMD and ADIOS2. In Smoky Mountains Computational Sciences and Engineering Conference. Springer, 99–118. (2021)
[4] J. J. Williams et al. Enabling High-throughput Parallel I/O in Particle-In-Cell Monte Carlo Simulations with openPMD and Darshan I/O Monitoring. In: 2024 IEEE International Conference on Cluster Computing Workshops (CLUSTER Workshops). Kobe, Japan. IEEE (2024)

Question

Question: BIT1 (PIC-MCC) Exascale Readiness

How can we **efficiently handle** massive **plasma simulation data** on current **extreme-scale** systems?

- **We need to avoid**:
  - Data Volume and Storage
  - Data Transfer and Communication
  - Processing and Analysis
  - Computational Resource Allocation
  - Time-Consuming Insight Retrieval

- **We need to know**:
  - Expensive Call Functions?
  - Communication vs. Computation Balance?
  - Commonly used MPI calls?
  - Impact of Workload Imbalance?
  - Streamlined Analysis and Debugging Techniques?

Environments

Environments: BIT1 Original on Distinct Systems[1,2]

Greendog

Type	Workstation (w/ Admin Privileges)
Processor	Intel i7-7820X (8 cores)
Memory	32 GB DRAM
GPU	NVIDIA RTX2060 SUPER
L1 Cache	256 KiB
L2 Cache	8 MiB
L3 Cache (LLC)	11 MiB

NJ

Type	HPC System
Processor	AMD EPYC 7302P 16-Core processor (32 cores)
Memory	256 GB DRAM
GPU	2 x NVIDIA A100
L1 Cache	32 KiB
L2 Cache	512 KiB
L3 Cache (LLC)	16 MiB

Dardel

Type	HPE Cray EX Supercomputer (CPU Partition, 554 Nodes)
Processor	2 x AMD EPYC™ Zen2 2.25 GHz (64 cores)
Cores per Node	128 (2 x 64 cores)
Memory per Node	256GB DRAM
L1 Cache	32 KiB
L2 Cache	512 KiB
L3 Cache (LLC)	16.38 MiB

[1] J. J. Williams et al. "Leveraging HPC Profiling & Tracing Tools to Understand the Performance of Particle-in-Cell Monte Carlo Simulations." Euro-Par 2023: Parallel Processing Workshops: Euro-Par 2023 International Workshops, Limassol, Cyprus, August 28–September 1, 2023, Revised Selected Papers, Part I, LNCS 14351. Springer Nature
[2] J. J. Williams, et al. "Optimizing BIT1, a Particle-in-Cell Monte Carlo Code, with OpenMP/OpenACC and GPU Acceleration." 24th International Conference on Computational Science, Málaga, Spain, July 2-4, 2024, Part I, LNCS 14832. Springer Nature

Environments: BIT1 openPMD BP4 on Distinct Systems[1,2,3]

Discoverer

Type	Petascale EuroHPC Supercomputer (1128 Compute nodes)
Processor	2 x AMD EPYC™ Zen2 7H12 GHz (64 cores)
Cores per Node	128 (2 x 64 cores)
Memory per Node	256 GB DDR4 SDRAM (regular nodes) 1TB DDR4 SDRAM (fat nodes)
OS	Red Hat Enterprise Linux 8.44
Storage	4 TB NFS and 2.1 PB LFS with 4 OSTs

Vega

Type	Petascale EuroHPC Supercomputer (CPU Partition, 554 Nodes)
Processor	2 x AMD EPYC™ Zen2 7H12 GHz (64 cores)
Cores per Node	128 (2 x 64 cores)
Memory per Node	256 GB DDR4 SDRAM (80% nodes) 1TB DDR4 SDRAM (20% nodes)
OS	Red Hat Enterprise Linux 8
Storage	23 PB CephFS and 1 PB LFS with 80 OSTs

Dardel

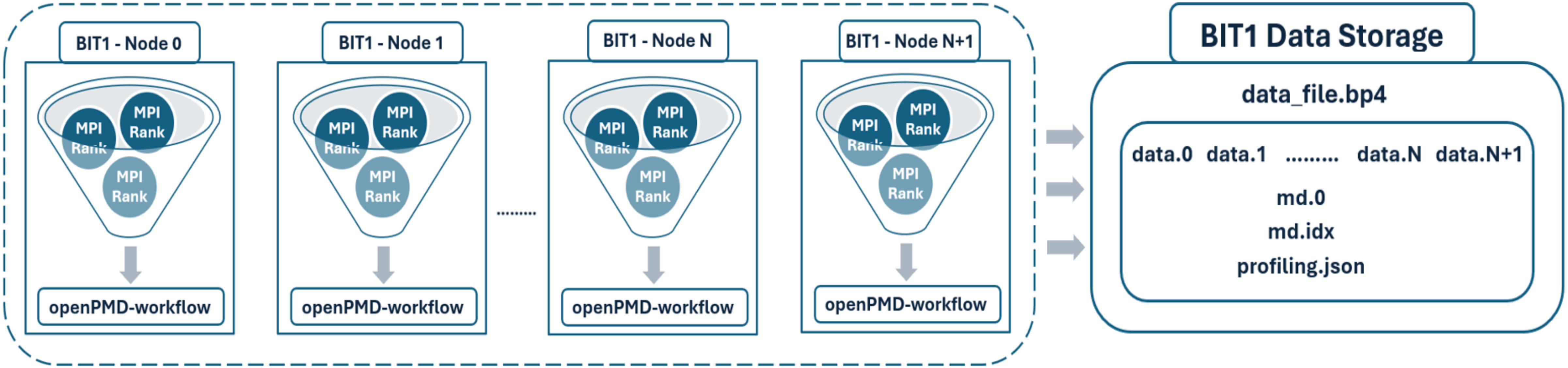
Type	HPE Cray EX Supercomputer (1270 compute nodes)
Processor	2 x AMD EPYC™ Zen2 2.25 GHz (64 cores)
Cores per Node	128 (2 x 64 cores)
Memory per Node	256GB DRAM
OS	SUSE Linux Enterprise Server 15 SP3
Storage	12 PB LFS with 48 OSTs

[1] J. J. Williams, et al. "Leveraging HPC Profiling & Tracing Tools to Understand the Performance of Particle-in-Cell Monte Carlo Simulations." Euro-Par 2023: Parallel Processing Workshops: Euro-Par 2023 International Workshops, Limassol, Cyprus, August 28–September 1, 2023, Revised Selected Papers, Part I, LNCS 14351. Springer Nature

[2] J. J. Williams, et al. "Understanding the Impact of openPMD on BIT1, a Particle-in-Cell Monte Carlo Code, through Instrumentation, Monitoring, and In-Situ Analysis." Euro-Par 2024: Parallel Processing Workshops: Euro-Par 2024 International Workshops, Madrid, Spain, August 26–30, 2024. Springer Nature.

[3] J. J. Williams, et al. Enabling High-throughput Parallel I/O in Particle-In-Cell Monte Carlo Simulations with openPMD and Darshan I/O Monitoring. In: 2024 IEEE International Conference on Cluster Computing Workshops (CLUSTER Workshops). Kobe, Japan. IEEE

Environments: BIT1 **openPMD BP4** I/O **Workflow**[1,2]



ADIOS2 Engines: BIT1 I/O **workflow** uses **ADIOS2 engines** and **output extensions** (e.g., **.bp4**) generating **directories** containing **data**, **metadata**, **index**, and optional **profiling files.**[1]

open Particle-Mesh Data

openPMD BP4 Integration: BIT1 I/O **workflow** with **openPMD** and the **ADIOS2 BP4** engine outputs **simulation results** into the directory, "**data_file.bp4**".[2]

Adaptable Input/Output System 2

[1] A. Huebl et al. openPMD-api: C++ & Python API for Scientific I/O with openPMD. https://doi.org/10.14278/rodare.27. (2018)
[2] J. J. Williams et al. Enabling High-throughput Parallel I/O in Particle-In-Cell Monte Carlo Simulations with openPMD and Darshan I/O Monitoring. In: 2024 IEEE International Conference on Cluster Computing Workshops (CLUSTER Workshops). Kobe, Japan. IEEE

Environments: Simple vs Complex BIT1 Test Cases[1,2]

*Neutral Particle Ionization Simulation

Uniform plasma interacts with Deuterium gas, causing ionization

Cells	100,000 (100K)
Species	electrons (e), ions (D^+), neutrals (D)
Collisions	elastic, excitation, ionization
Wall effects	ion recycling into neutrals
Total number of particles	30,000,000 (30M)
Fields	No electric or magnetic fields
Time steps	200,000 (200K)
Minimum (Baseline) HPC system	Dardel/Vega, 1 Node (128 MPI Processes)

Outcome: *D concentration **decreases** with time during **ionization**.

**High-Density Sheath Simulation

Plasma reaches wall, transforms to neutrals, forms charged sheath

Cells	3,000,000 (3M)
Species	electrons (e), ions (D^+), neutrals (D)
Collisions	elastic, excitation, ionization
Wall effects	ion recycling into neutrals
Total number of particles	2,200,000,000 (2.2B)
Fields	Self-consistent electric field, fixed magnetic field (3T)
Time steps	100,000 (100K)
Minimum (Baseline) HPC system	Dardel/Vega, 5 Node (640 MPI Processes)

Outcome: **e and D^+ **transform** into D, forming **front-wall** plasma **charged-sheath**

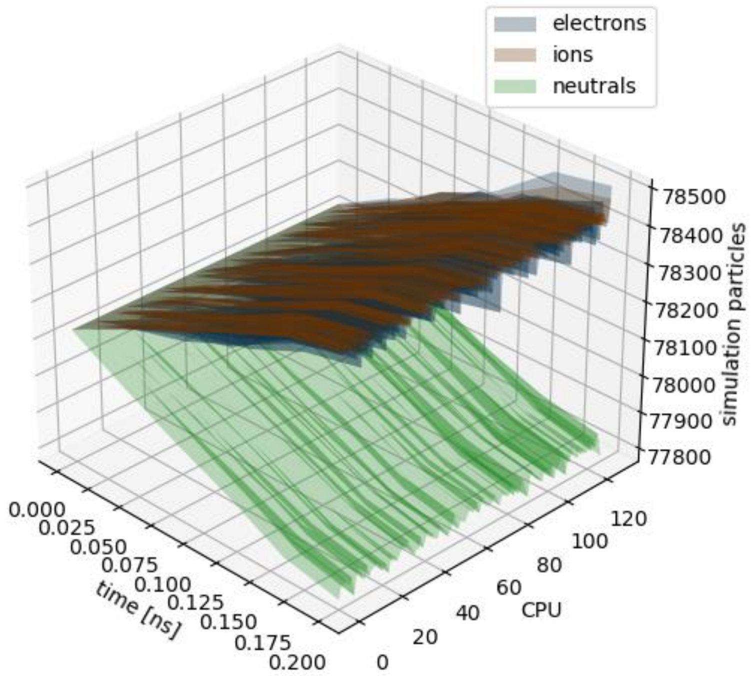
[1] J. J. Williams et al. "Leveraging HPC Profiling & Tracing Tools to Understand the Performance of Particle-in-Cell Monte Carlo Simulations." Euro-Par 2023: Parallel Processing Workshops: Euro-Par 2023 International Workshops, Limassol, Cyprus, August 28–September 1, 2023, Revised Selected Papers, Part I, LNCS 14351.

[2] J. J. Williams et al. Enabling High-throughput Parallel I/O in Particle-In-Cell Monte Carlo Simulations with openPMD and Darshan I/O Monitoring. In: 2024 IEEE International Conference on Cluster Computing Workshops (CLUSTER Workshops). Kobe, Japan. IEEE (2024)

Environments: BIT1 openPMD BP4 Test Case[1,2]

*__Neutral Particle Ionization__: Uniform **plasma** interacts with **Deuterium gas**, causing **ionization**

Cells	100,000 (100K)
Species	electrons (e), ions (D^+), neutrals (D)
Collisions	elastic, excitation, ionization
Wall effects	ion recycling into neutrals
Total number of particles	30,000,000 (30M)
Fields	No electric or magnetic fields
Time steps	200,000 (200K)
Minimum (Baseline) HPC system	Dardel/Vega, 1 Node (128 MPI Processes)



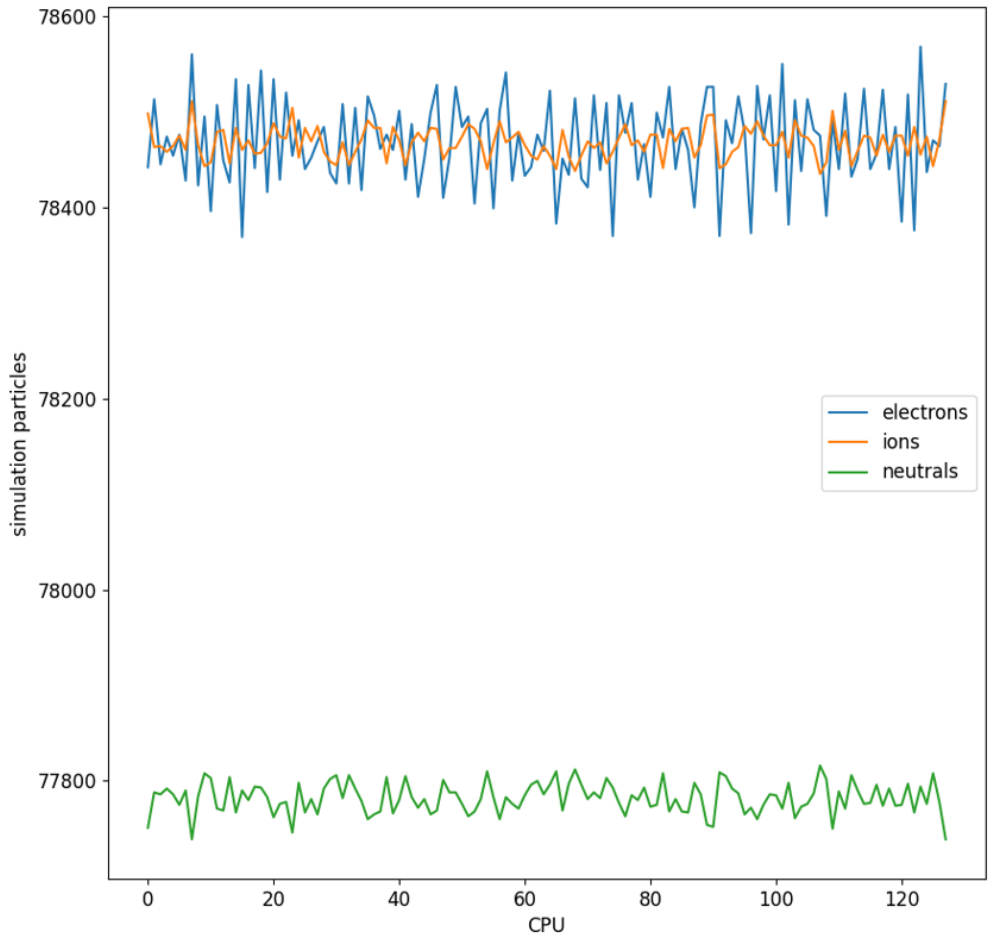
Outcome: *D concentration **decreases** with time during **ionization**.

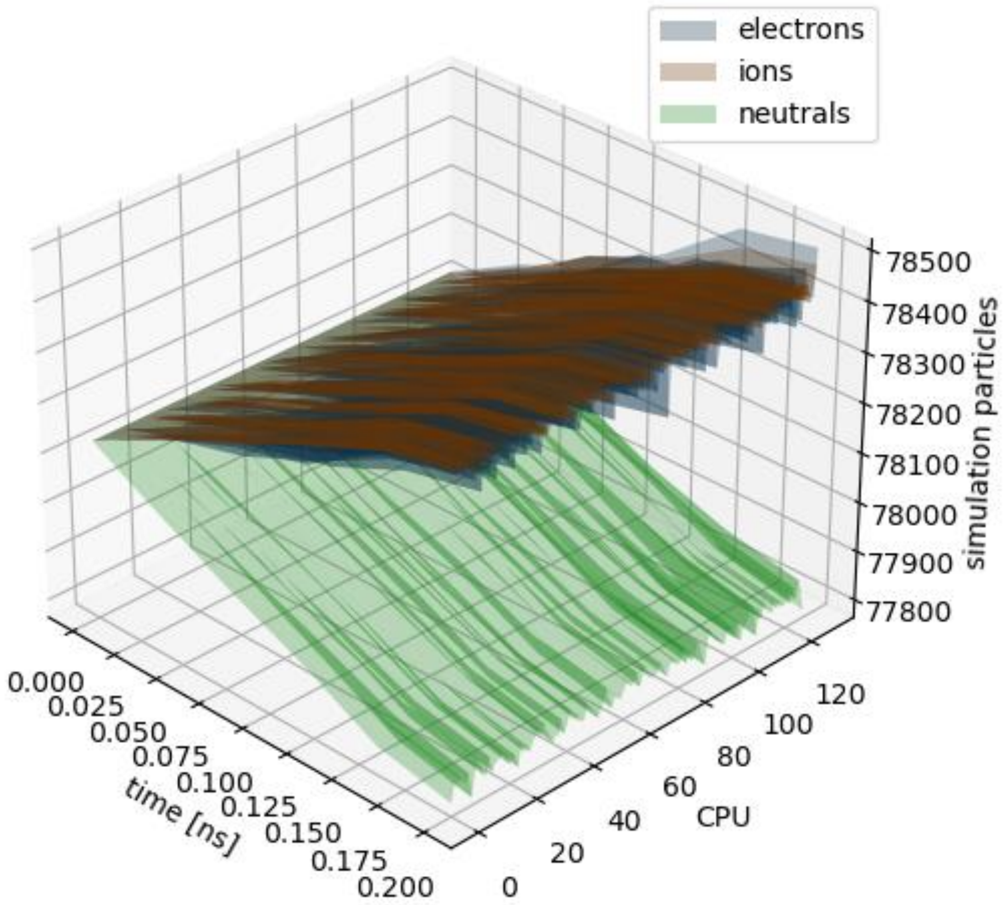
* **3D time evolution** of the number of **simulation particles** for the first **5000** timesteps: **electrons**, **ions** and **neutrals**

[1] J. J. Williams et al. "Leveraging HPC Profiling & Tracing Tools to Understand the Performance of Particle-in-Cell Monte Carlo Simulations." Euro-Par 2023: Parallel Processing Workshops: Euro-Par 2023 International Workshops, Limassol, Cyprus, August 28–September 1, 2023, Revised Selected Papers, Part I, LNCS 14351. Springer Nature
[2] J. J. Williams et al. Enabling High-throughput Parallel I/O in Particle-In-Cell Monte Carlo Simulations with openPMD and Darshan I/O Monitoring. In: 2024 IEEE International Conference on Cluster Computing Workshops (CLUSTER Workshops). Kobe, Japan. IEEE

Evaluation and Results

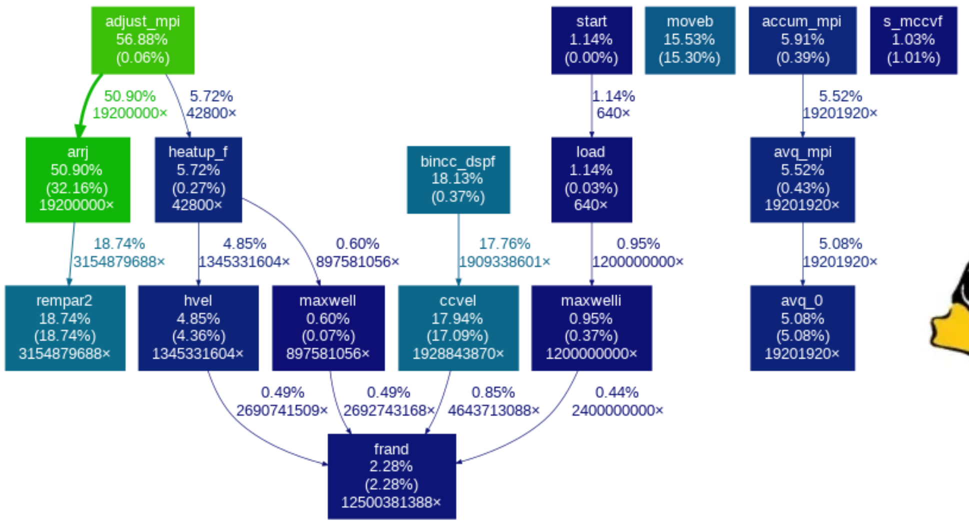
Evaluation: BIT1 Original (CPU) Simulations[1]

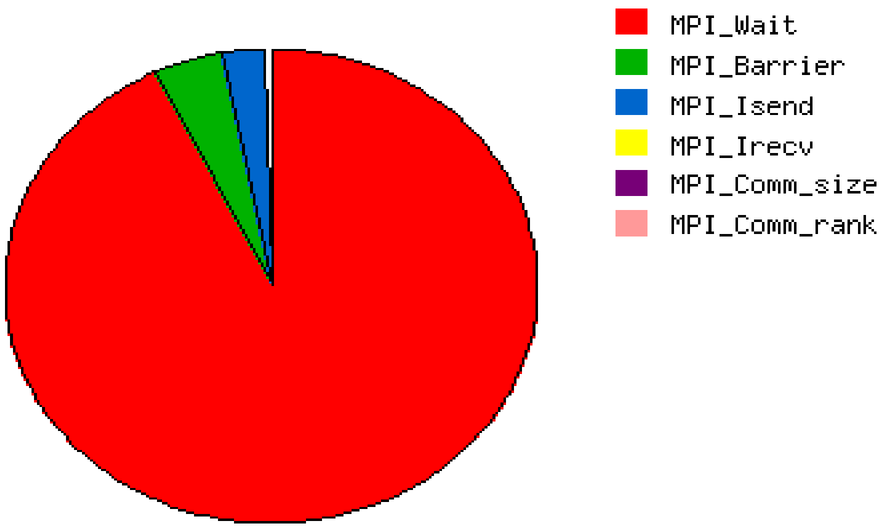




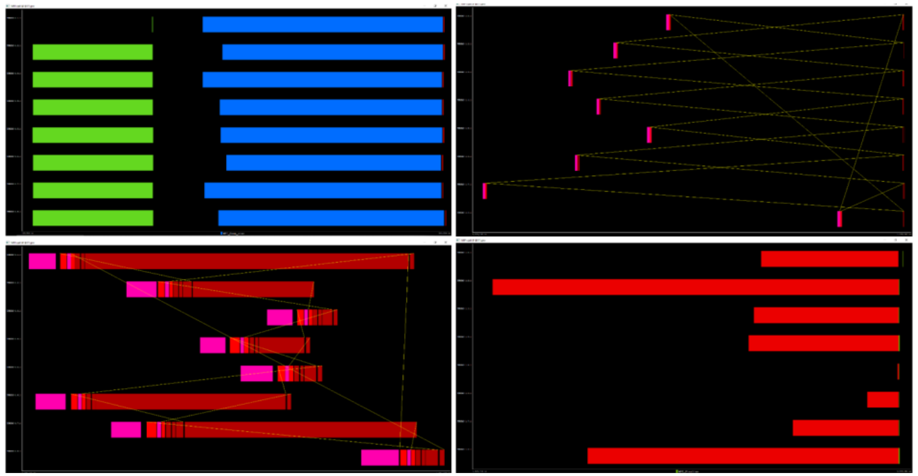
[1] J. J. Williams et al. "Leveraging HPC Profiling & Tracing Tools to Understand the Performance of Particle-in-Cell Monte Carlo Simulations." Euro-Par 2023: Parallel Processing Workshops: Euro-Par 2023 International Workshops, Limassol, Cyprus, August 28–September 1, 2023, Revised Selected Papers, Part I, LNCS 14351. Springer Nature

Evaluation: Leveraging HPC Profiling & Tracing Tools[1]

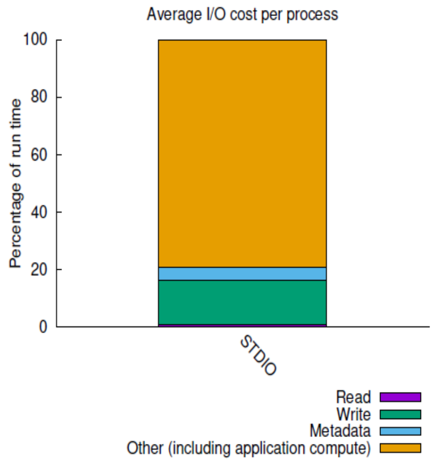


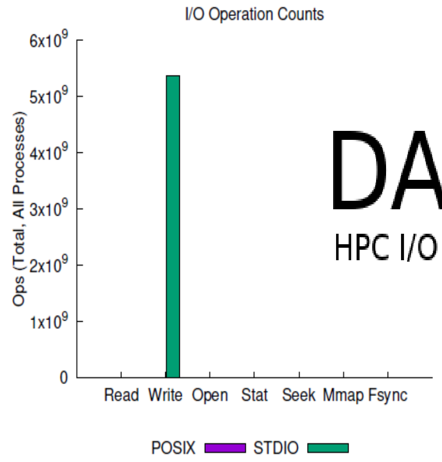


IPM



BSC





DARSHAN
HPC I/O Characterization Tool

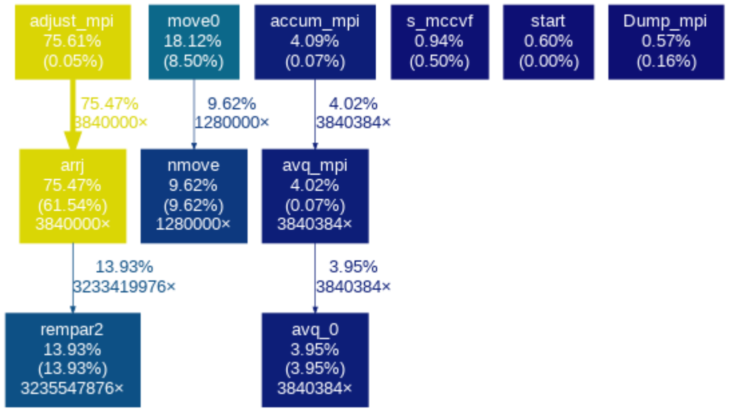
[1] J. J. Williams et al. "Leveraging HPC Profiling & Tracing Tools to Understand the Performance of Particle-in-Cell Monte Carlo Simulations." Euro-Par 2023: Parallel Processing Workshops: Euro-Par 2023 International Workshops, Limassol, Cyprus, August 28–September 1, 2023, Revised Selected Papers, Part I, LNCS 14351. Springer Nature

gprof & perf

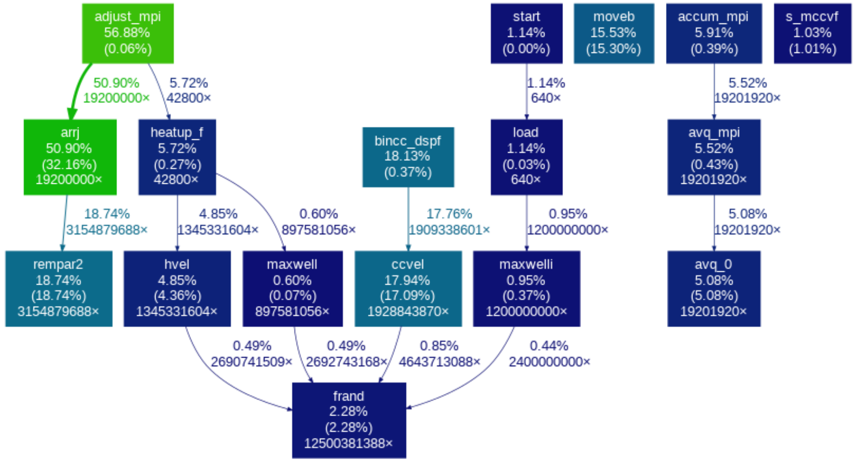
Evaluation: gprof [1]

- Open-source profiling tool
- Gathers execution time information
- Reports frequently used functions by the processor

Neutral Particle Ionization



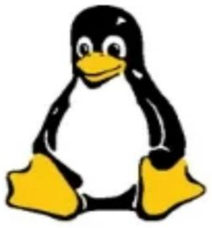
High-Density Sheath



[1] J. J. Williams et al. "Leveraging HPC Profiling & Tracing Tools to Understand the Performance of Particle-in-Cell Monte Carlo Simulations." Euro-Par 2023: Parallel Processing Workshops: Euro-Par 2023 International Workshops, Limassol, Cyprus, August 28–September 1, 2023, Revised Selected Papers, Part I, LNCS 14351. Springer Nature

Evaluation: perf [1]

- Profiler tool
- Gathers hardware performance counters and events
- Mostly used in Linux subsystems



L1 - Cachetest

Performance counter stats for './cachetest':

12 996 L1-dcache-load-misses # 5,53% of all L1-dcache accesses

234 862 L1-dcache-loads

0,001922798 seconds time elapsed

0,001948000 seconds user
0,000000000 seconds sys

LLC (L3) – Cachetest

Performance counter stats for './cachetest':

449 LLC-load-misses # 18,95% of all LL-cache accesses

2 370 LLC-loads

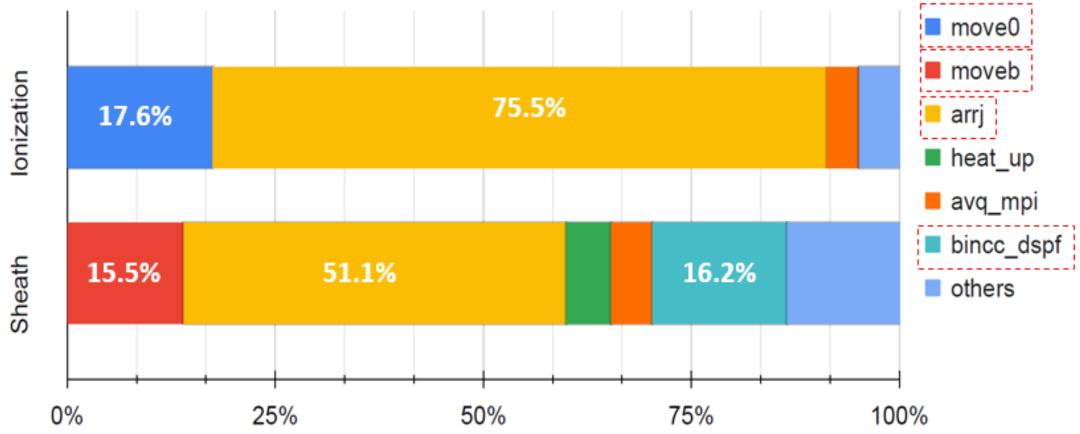
0,001688897 seconds time elapsed

0,001797000 seconds user
0,000000000 seconds sys

[1] J. J. Williams et al. "Leveraging HPC Profiling & Tracing Tools to Understand the Performance of Particle-in-Cell Monte Carlo Simulations." Euro-Par 2023: Parallel Processing Workshops: Euro-Par 2023 International Workshops, Limassol, Cyprus, August 28–September 1, 2023, Revised Selected Papers, Part I, LNCS 14351. Springer Nature

Results: Leveraging HPC Profiling and Tracing Tools[1]

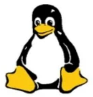
gprof: Reports Frequently Used Functions



arrj() – **Particle Arrangement** within appropriate **cells** and **ranks**.

move0() / **moveb()** – **Particle Pusher (Mover)**

bincc_dspf() – **Binary Collision Operator**



perf: Cache and Memory Usage

Baseline Size	10% Reduction Size	20% Reduction Size	cache-test
L1 Load Misses	L1 Load Misses	L1 Load Misses	L1 Load Misses
3.43%	2.51%	2.17%	5.53%
LLC Load Misses	LLC Load Misses	LLC Load Misses	LLC Load Misses
99.07%	52.25%	47.51%	18.95%

All **L1** performance **consistent** with **low** miss rates.

Baseline:

- Performance **hindered** by **99%** cache misses, causing **frequent** slow main **memory** access.

Reduction (20%):

- Smaller problem size **fits** in LLC (L3), **improving** LLC (L3) and BIT1's performance.

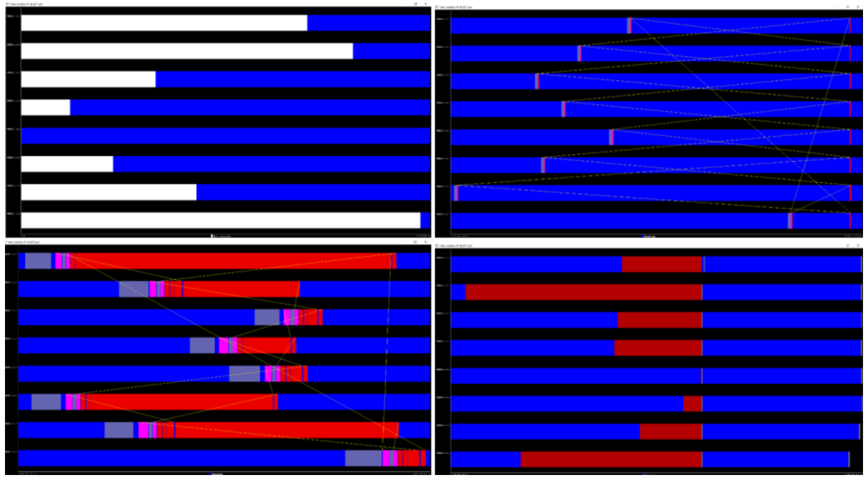
[1] J. J. Williams et al. "Leveraging HPC Profiling & Tracing Tools to Understand the Performance of Particle-in-Cell Monte Carlo Simulations." Euro-Par 2023: Parallel Processing Workshops: Euro-Par 2023 International Workshops, Limassol, Cyprus, August 28–September 1, 2023, Revised Selected Papers, Part I, LNCS 14351. Springer Nature

Extrae, Paraver & IPM

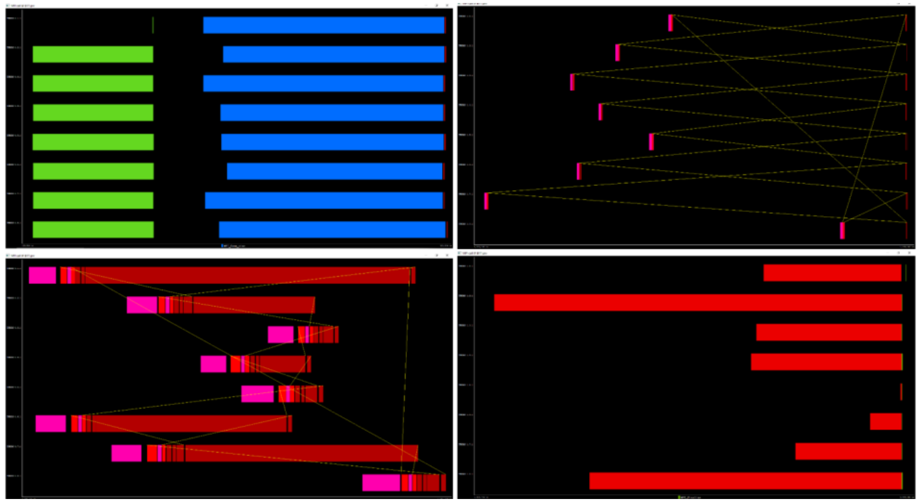
Evaluation: Extrae & Paraver [1]

- **Extrae** & **Paraver:** Parallel performance tracing and profiler tools by *BSC
- **Extrae**: Instruments code for parallel performance tracing
- **Paraver**: Post-processes Extrae output and provides visualization

Ionization (8 MPI Ranks): Runtime-Trace



Ionization (8 MPI Ranks): MPI-Call-Trace



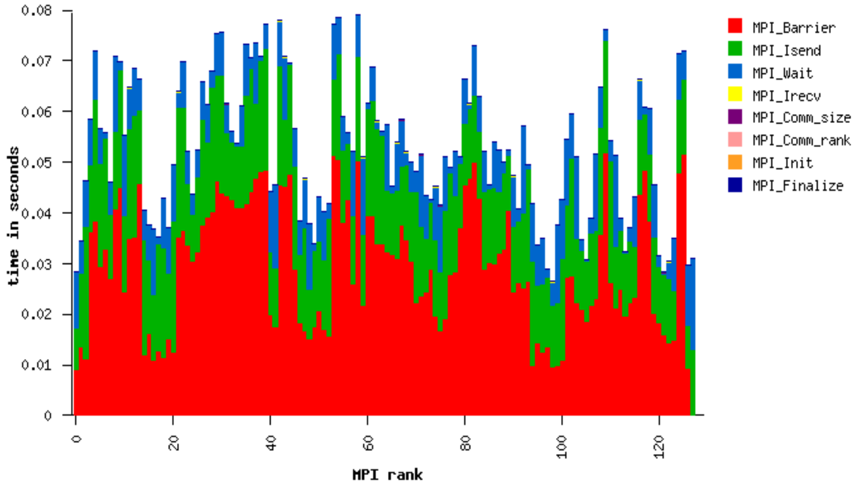
*Barcelona Supercomputing Center (BSC)

[1] J. J. Williams et al. "Leveraging HPC Profiling & Tracing Tools to Understand the Performance of Particle-in-Cell Monte Carlo Simulations." Euro-Par 2023: Parallel Processing Workshops: Euro-Par 2023 International Workshops, Limassol, Cyprus, August 28–September 1, 2023, Revised Selected Papers, Part I, LNCS 14351. Springer Nature

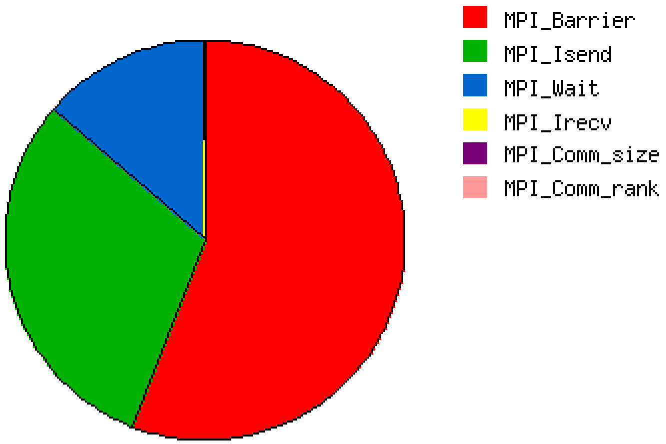
Evaluation: IPM[1]

- Portable profiling performance monitoring tool
- Offers low-overhead performance profiles for parallel programs
- Focus on communication, computation, and I/O aspects

Ionization (1 node): mpi_stack_byrank



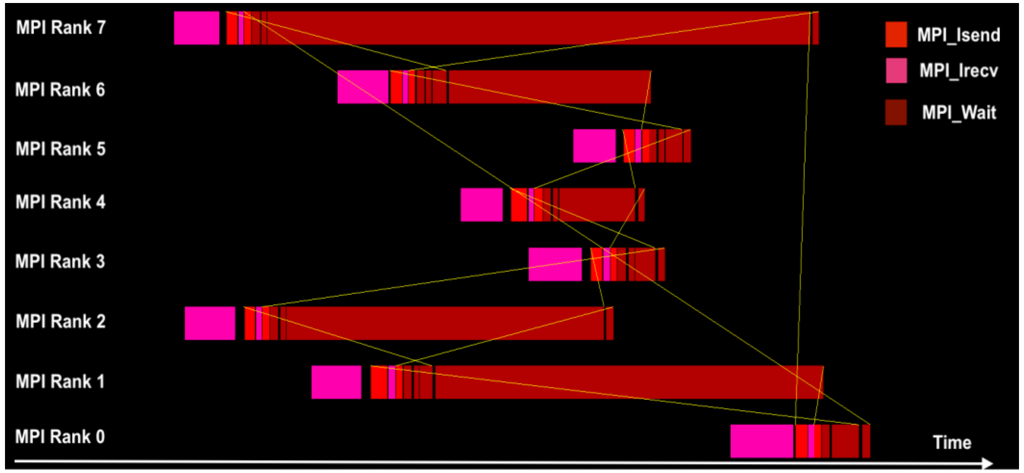
Ionization (1 node): mpi_pie (%)



[1] J. J. Williams et al. "Leveraging HPC Profiling & Tracing Tools to Understand the Performance of Particle-in-Cell Monte Carlo Simulations." Euro-Par 2023: Parallel Processing Workshops: Euro-Par 2023 International Workshops, Limassol, Cyprus, August 28–September 1, 2023, Revised Selected Papers, Part I, LNCS 14351. Springer Nature

Results: Leveraging HPC Profiling and Tracing Tools[1]

Extrae & Paraver: Communication Pattern



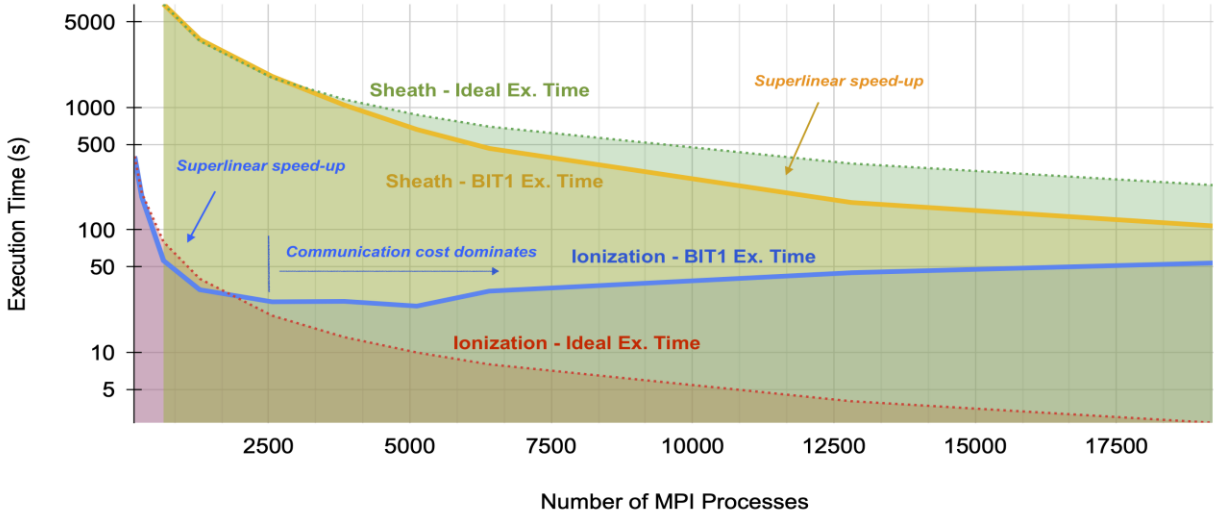
Workload imbalance:

- MPI **rank 0** is **slowest**, causing **neighbouring** MPI processes to **wait**.

Reason:

- MPI **rank 0** performs **data gathering** and **transmission** to other ranks.
- **Synchronization** is needed for the **field solver** and **plasma profile** calculator.

IPM: Parallel and MPI Performance Analysis



Superlinear speed-up observed in **both** cases.

Sheath:

- Over **2,560** MPI processes, **consistent** superlinear speed-up **detected**.
  - **Superlinear effect** due to **decreasing** problem **size** per **process**.
  - **Performance** significantly **improves** when the problem **fits** into the **L3** cache.

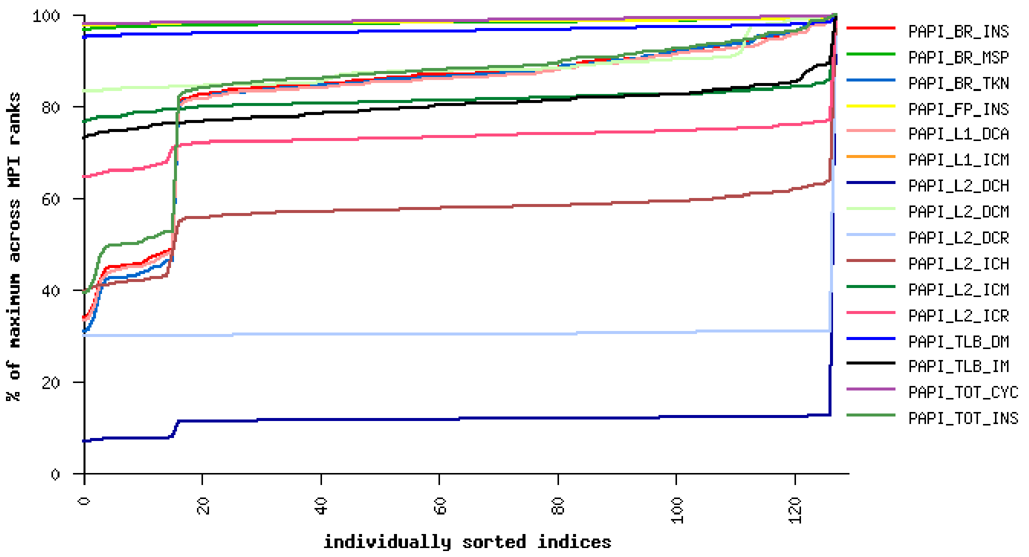
[1] J. J. Williams et al. "Leveraging HPC Profiling & Tracing Tools to Understand the Performance of Particle-in-Cell Monte Carlo Simulations." Euro-Par 2023: Parallel Processing Workshops: Euro-Par 2023 International Workshops, Limassol, Cyprus, August 28–September 1, 2023, Revised Selected Papers, Part I, LNCS 14351. Springer Nature

IPM & Darshan

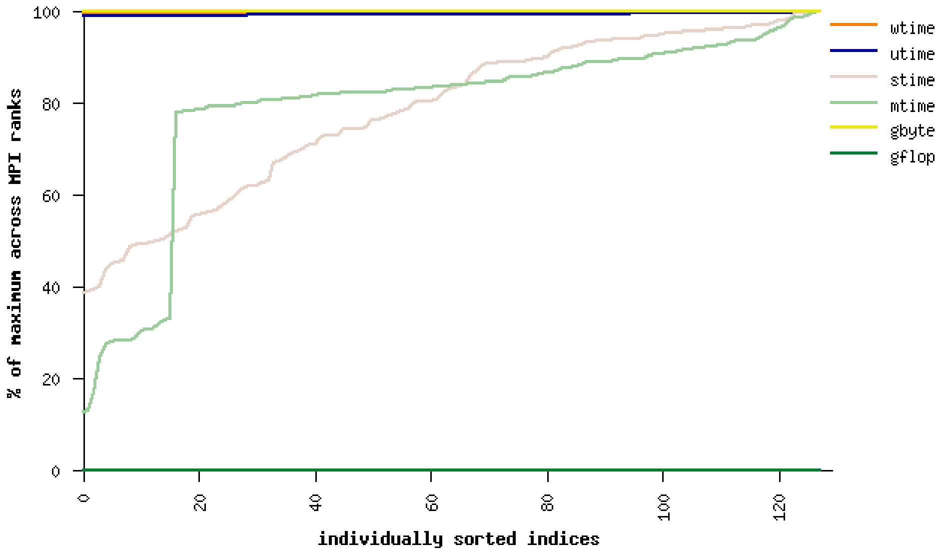
Evaluation: IPM[1]

- Offers selectable detail levels at runtime with text and web reports.
- Scalable and requires no source code modification.
- Used in HPC centers, application development, debugging, and academia.

Ionization (1 node): load_hpm_all (HPM counters)



Ionization (1 node): load_multi (memory, flops, timings)



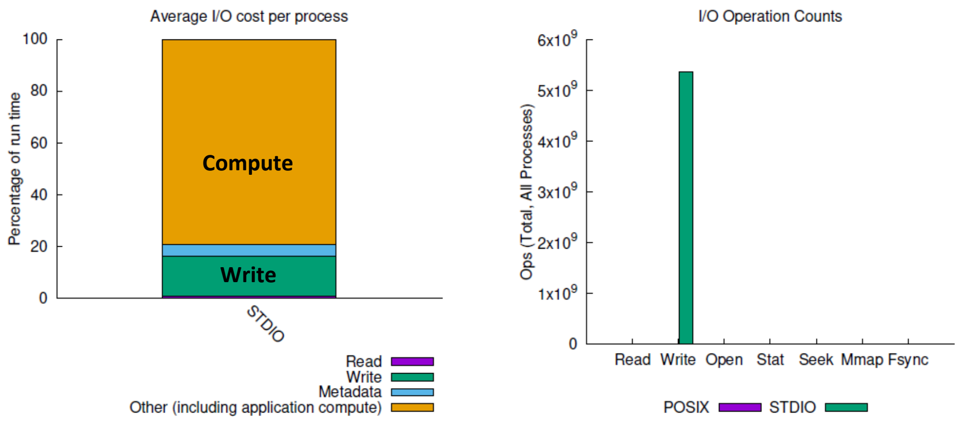
[1] J. J. Williams et al. "Leveraging HPC Profiling & Tracing Tools to Understand the Performance of Particle-in-Cell Monte Carlo Simulations." Euro-Par 2023: Parallel Processing Workshops: Euro-Par 2023 International Workshops, Limassol, Cyprus, August 28–September 1, 2023, Revised Selected Papers, Part I, LNCS 14351. Springer Nature

Evaluation: Darshan[1]

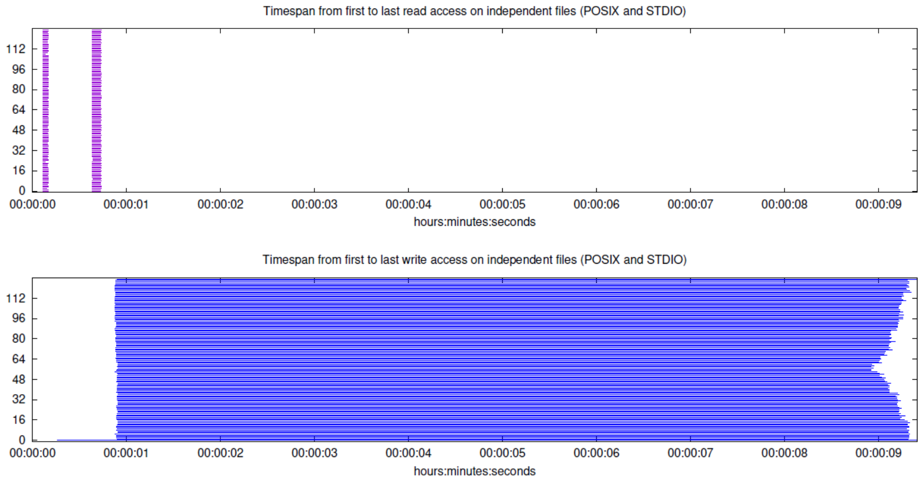
- Performance monitoring tool
- Designed for analyzing serial and parallel I/O workloads
- Used for investigating and tuning I/O behavior

DARSHAN
HPC I/O Characterization Tool

Ionization I/O performance estimate (1 Node)



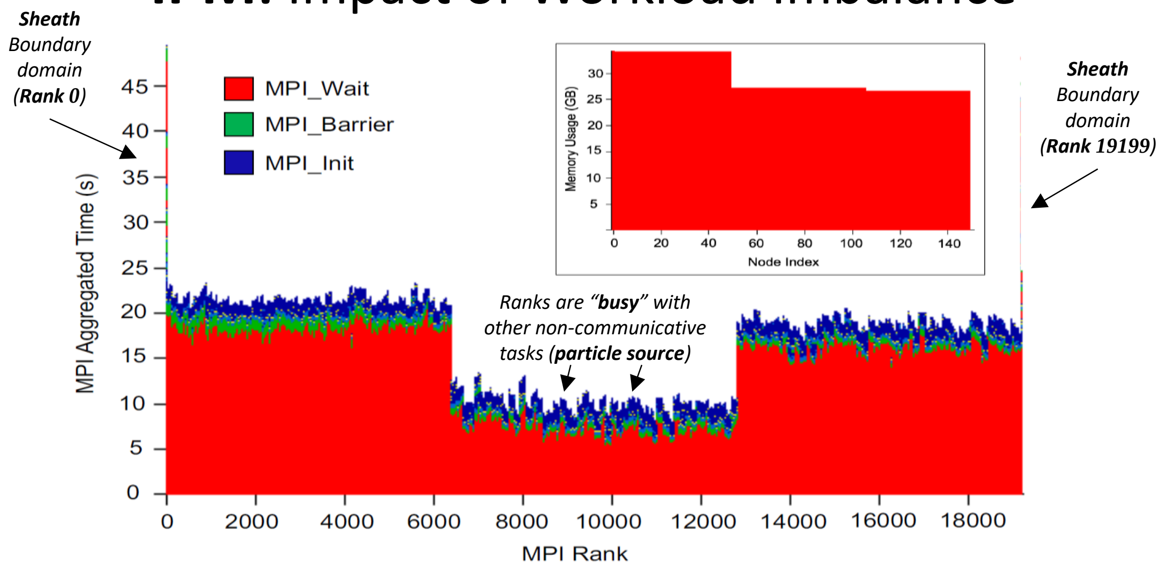
Ionization timespan for read and writes (1 Node)



[1] J. J. Williams et al. "Leveraging HPC Profiling & Tracing Tools to Understand the Performance of Particle-in-Cell Monte Carlo Simulations." Euro-Par 2023: Parallel Processing Workshops: Euro-Par 2023 International Workshops, Limassol, Cyprus, August 28–September 1, 2023, Revised Selected Papers, Part I, LNCS 14351. Springer Nature

Results: Leveraging HPC Profiling and Tracing Tools[1]

IPM: Impact of Workload Imbalance



Workload imbalance:

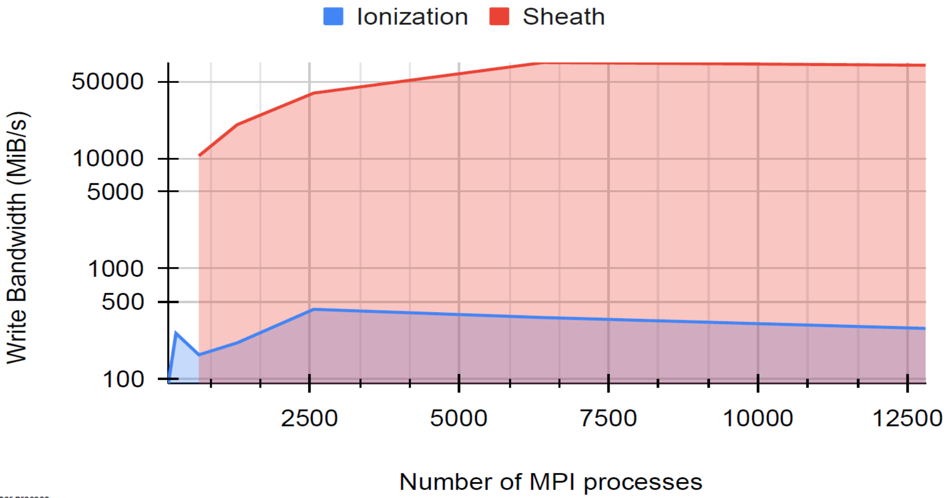
- Boundary MPI ranks (0 and 19199) **2x longer** MPI times due to **MPI_Wait synchronization**.

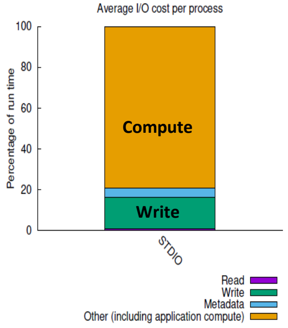
Node Memory Usage:

- Varying **memory use** (~ 23% difference)
  - **Largest** is ~34 GB and **Smallest** is ~26 GB.

IPM

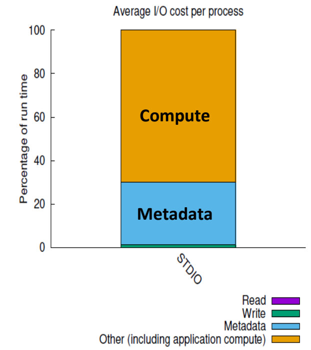
Darshan: Evaluate I/O Performance





Both Cases:

Write bandwidth **increases** and then **saturates.**



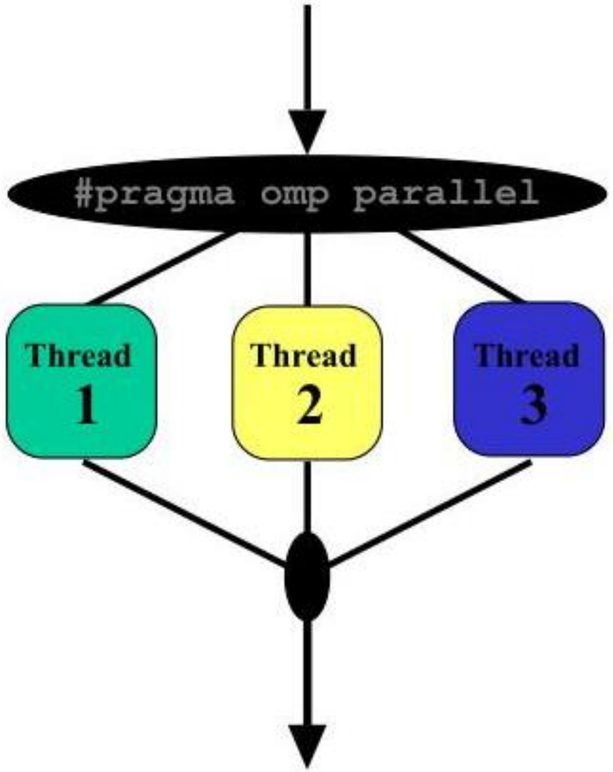
Peak I/O write bandwidth **linked** to problem **size**.

Post-peak, performance **drops** from **metadata** writing cost.

DARSHAN
HPC I/O Characterization Tool

[1] J. J. Williams et al. "Leveraging HPC Profiling & Tracing Tools to Understand the Performance of Particle-in-Cell Monte Carlo Simulations." Euro-Par 2023: Parallel Processing Workshops: Euro-Par 2023 International Workshops, Limassol, Cyprus, August 28–September 1, 2023, Revised Selected Papers, Part I, LNCS 14351. Springer Nature

Evaluation: Empowering (CPU/GPU) BIT1 Original Simulations[1]




```
main()
 {
   <serial code>
   #pragma acc kernels
   {
     <parallel code>
   }
 }
```

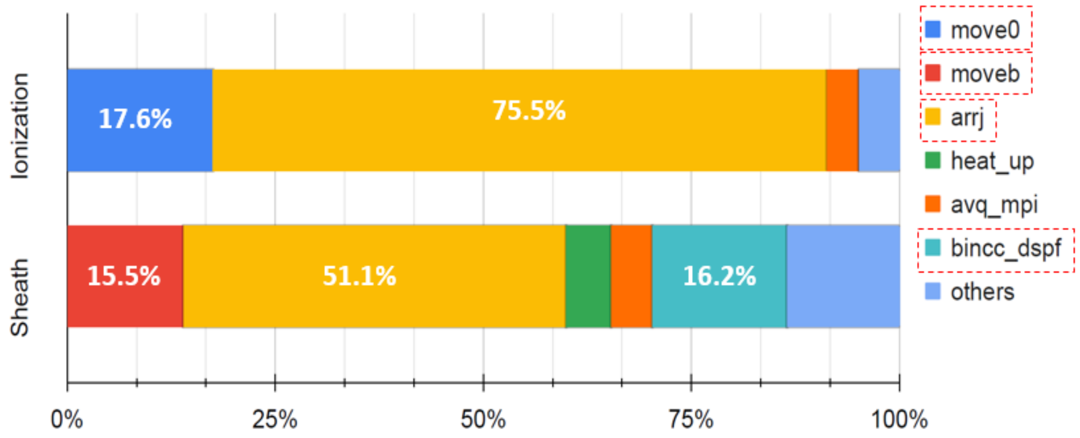
OpenMP

OpenACC

[1] J. J. Williams, et al. "Optimizing BIT1, a Particle-in-Cell Monte Carlo Code, with OpenMP/OpenACC and GPU Acceleration." 24th International Conference on Computational Science, Málaga, Spain, July 2-4, 2024, Part I, LNCS 14832. Springer Nature.

Evaluation: HPC Analysis [1] and Empowering BIT1 Simulations[2]

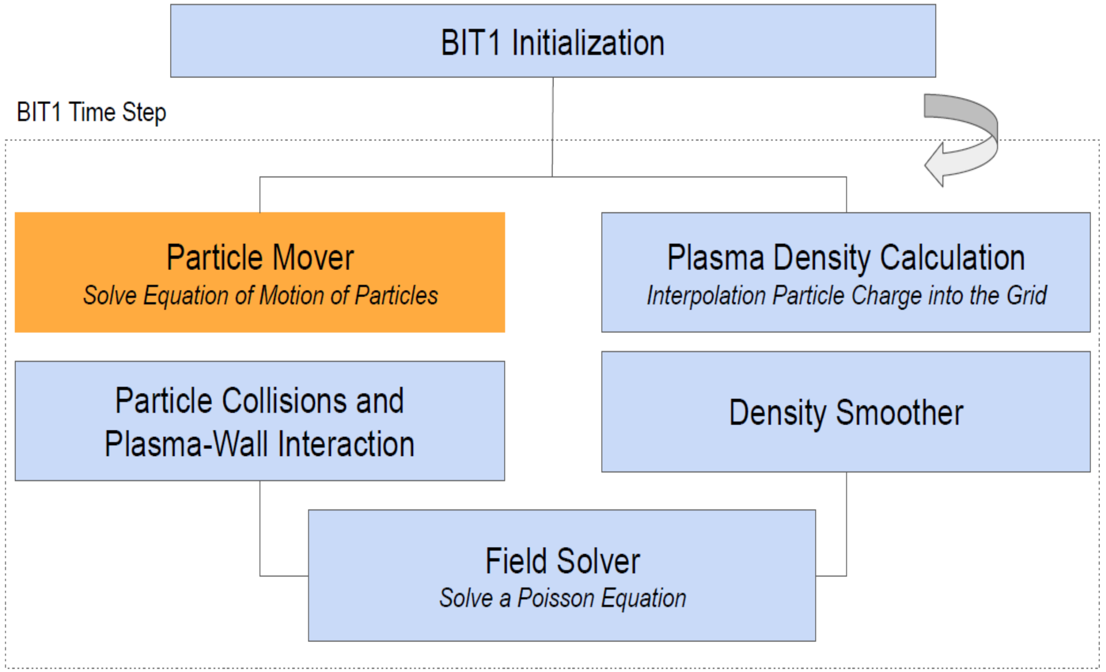
gprof: Reports Frequently Used Functions [1]



arrj() – **Particle Arrangement** within appropriate **cells** and **ranks**.

move0() / **moveb()** – **Particle Pusher (Mover)**

bincc_dspf() – **Binary Collision Operator**



- **Confinements** [1]
  - **Workload Imbalance (during particle movement)**
  - Relies solely on MPI (lacks hybrid parallel computing)
  - CPU-bound (lacks GPU utilization and support)

[1] J. J. Williams et al. "Leveraging HPC Profiling & Tracing Tools to Understand the Performance of Particle-in-Cell Monte Carlo Simulations." Euro-Par 2023: Parallel Processing Workshops: Euro-Par 2023 International Workshops, Limassol, Cyprus, August 28–September 1, 2023, Revised Selected Papers, Part I, LNCS 14351. Springer Nature
[2] J. J. Williams, et al. "Optimizing BIT1, a Particle-in-Cell Monte Carlo Code, with OpenMP/OpenACC and GPU Acceleration." 24th International Conference on Computational Science, Málaga, Spain, July 2-4, 2024, Part I, LNCS 14832. Springer Nature.

Evaluation: Empowering BIT1 Simulations

OpenMP (Open Multi-Processing) and **OpenACC** (Open Accelerators) Directives[1]

OpenMP Tasks

```
#pragma omp parallel shared(chsp, sn2d, dinj, nstep, np, x, yp, yx, vy) \
    private(isp, i, j) firstprivate(nsp, nc)
{
#pragma omp single
 {
 for (isp = 0; isp < nsp; isp++) {
    ...
#pragma omp taskloop grainsize(500) nogroup
    for (j = 0; j < nc; j++) {
#pragma omp simd
      for (i = 0; i < np[isp][j]; i++)
        x[isp][j][i] += nstep[isp] * vx[isp][j][i];
        }
        ...
#pragma omp taskloop grainsize(500) nogroup
     for (j = 0; j < nc; j++)
#pragma omp simd
      for (i = 0; i < np[isp][j]; i++)
        x[isp][j][i] += nstep[isp] * vx[isp][j][i];
    }
  }
}
```

OpenACC Multicore

```
#pragma acc parallel loop present(chsp[:lenA],
    sn2d[:lenA], dinj[:lenA], nstep[:lenA],
    np[:lenA][:lenB], x[:lenA][:lenB][:lenC],
    yp[:lenA][:lenB][:lenC],
    vx[:lenA][:lenB][:lenC],
    vy[:lenA][:lenB][:lenC])
{
    for (isp = 0; isp < nsp; isp++) {
    ...
#pragma acc loop gang vector
    for (j = 0; j < nc; j++) {
#pragma acc loop vector
    for (i = 0; i < np[isp][j]; i++)
      x[isp][j][i] += nstep[isp] * vx[isp][j][i];
        }
        ...
#pragma acc loop gang vector
    for (j = 0; j < nc; j++) {
#pragma acc loop vector
     for (i = 0; i < np[isp][j]; i++)
       x[isp][j][i] += nstep[isp] * vx[isp][j][i];
    }
  }
}
```

OpenMP Target

```
#pragma omp target enter data map(to: chsp[:lenA],
    sn2d[:lenA], dinj[:lenA], nstep[:lenA],
    np[:lenA][:lenB], x[:lenA][:lenB][:lenC],
    yp[:lenA][:lenB][:lenC],
    vx[:lenA][:lenB][:lenC],
    vy[:lenA][:lenB][:lenC])
{
    for (isp = 0; isp < nsp; isp++) {
    ...
#pragma omp target teams distribute parallel for thread_limit(256) num_teams(391)
    for (j = 0; j < nc; j++) {
#pragma omp simd
    for (i = 0; i < np[isp][j]; i++)
      x[isp][j][i] += nstep[isp] * vx[isp][j][i];
        }
        ...
#pragma omp target teams distribute parallel for thread_limit(256) num_teams(391)
    for (j = 0; j < nc; j++) {
#pragma omp simd
     for (i = 0; i < np[isp][j]; i++)
       x[isp][j][i] += nstep[isp] * vx[isp][j][i];
    }
#pragma omp target exit data map(from: x[:lenA][:lenB][:lenC]...)
  }
}
```

OpenACC Parallel

```
#pragma acc enter data copyin(chsp[:lenA],
    sn2d[:lenA], dinj[:lenA], nstep[:lenA],
    np[:lenA][:lenB], x[:lenA][:lenB][:lenC],
    yp[:lenA][:lenB][:lenC],
    vx[:lenA][:lenB][:lenC],
    vy[:lenA][:lenB][:lenC])
{
    for (isp = 0; isp < nsp; isp++) {
    ...
#pragma acc parallel loop gang worker vector vector_length(128) \
    present(np[:lenA][:lenB], nstep[:lenA],
    x[:lenA][:lenB][:lenC],
    vx[:lenA][:lenB][:lenC])
    firstprivate(nc,isp,nsp) private(i)
    for (j = 0; j < nc; j++) {
#pragma acc loop
    for (i = 0; i < np[isp][j]; i++)
      x[isp][j][i] += nstep[isp] * vx[isp][j][i];
        }
        ...
#pragma acc parallel loop gang worker vector vector_length(128) \
    present(np[:lenA][:lenB], nstep[:lenA],
    x[:lenA][:lenB][:lenC],
    vx[:lenA][:lenB][:lenC])
    firstprivate(nc,isp,nsp) private(i)
    for (j = 0; j < nc; j++) {
#pragma acc loop
     for (i = 0; i < np[isp][j]; i++)
       x[isp][j][i] += nstep[isp] * vx[isp][j][i];
    }
#pragma acc exit data copyout(x[:lenA][:lenB][:lenC]...)
  }
}
```

Multicore **CPUs**

OpenMP Tasks**:**

- **Fine-grained** parallelism with **tasks** for **dynamic workload** distribution.

OpenACC Multicore**:**

- Offers **straightforward method** for **parallelizing** code on multicore **CPUs**.

GPU Acceleration

OpenMP Target**:**

- **Extension** of the **OpenMP** standard for **offloading computation** on **GPUs**.

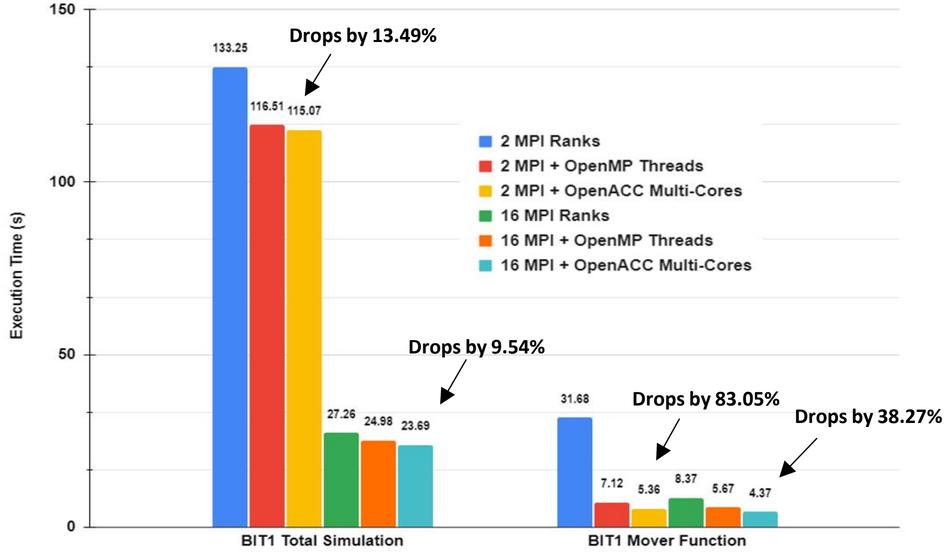
OpenACC Parallel**:**

- Offers a **high-level directive-based** approach for **parallelizing code** on **GPUs**.

[1] J. J. Williams, et al. "Optimizing BIT1, a Particle-in-Cell Monte Carlo Code, with OpenMP/OpenACC and GPU Acceleration." 24th International Conference on Computational Science, Málaga, Spain, July 2-4, 2024, Part I, LNCS 14832. Springer Nature.

Results: Hybrid MPI and OpenMP/OpenACC BIT1 on CPUs [1]

Improving **Total** Simulation and **Mover** Function



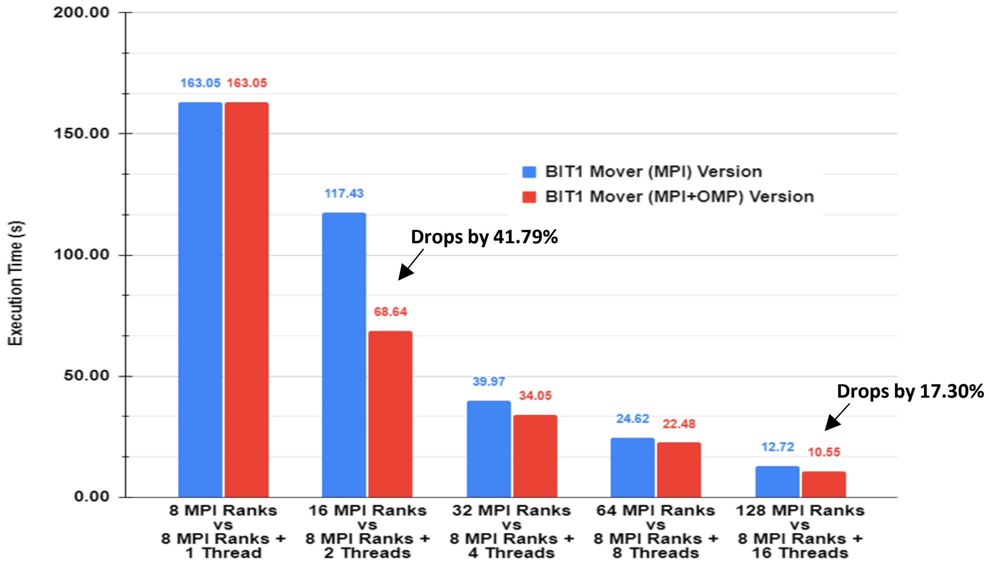
For **2 Ranks** (Baseline), **Hybrid** MPI+OpenACC **decreases**:

- **Total** simulation time by **13.49%** and **Mover** function time by **83.05%.**

For **16 Ranks** (Baseline), **Hybrid** MPI+OpenACC **decreases**:

- **Total** simulation time by **9.54%** and **Mover** function time by **38.27%.**

Enhancing **Communication** Performance and **Scalability**



For **16 Ranks** vs **8 Ranks + 2 Threads**:

- **Execution** time **decreased** by **41.79%**.
  - **Adding** 2 threads to 8 Ranks **significantly boosts** BIT1 performance.

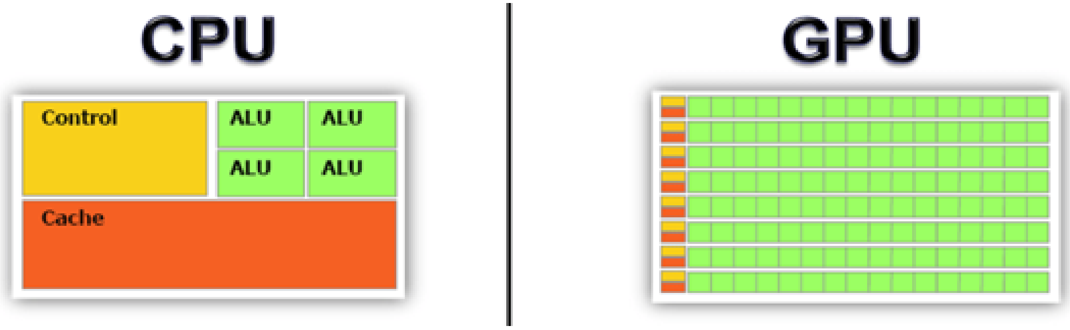
For **128 Ranks** vs **8 Ranks + 16 Threads**:

- **Execution** time **decreased** by **17.30%**.
  - **Adding** 16 threads to 8 Ranks **demonstrates scalable** BIT1 performance.

[1] J. J. Williams, et al. "Optimizing BIT1, a Particle-in-Cell Monte Carlo Code, with OpenMP/OpenACC and GPU Acceleration." 24th International Conference on Computational Science, Málaga, Spain, July 2-4, 2024, Part I, LNCS 14832. Springer Nature.

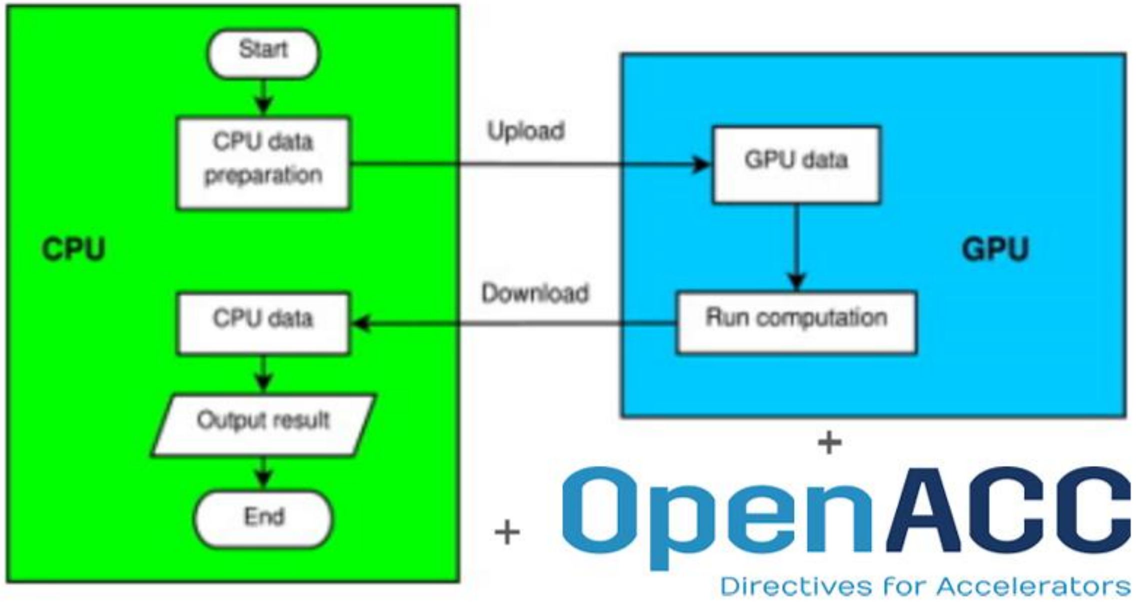
Evaluation: Accelerating BIT1 with GPUs[1]

CPU (Host) vs **GPU (Device)**



Have **powerful** Arithmetic Logic Units (ALUs)	Have **energy efficient** and **resourceful** ALUs
Have **large** caches	Have very **small** cache
Have a **sophisticated** control structure	Have a **basic simple** control structure
No **processor threads to tolerate** latencies	Have **massive number of threads** to tolerate latencies

A Simple **GPU Accelerated** Procedure

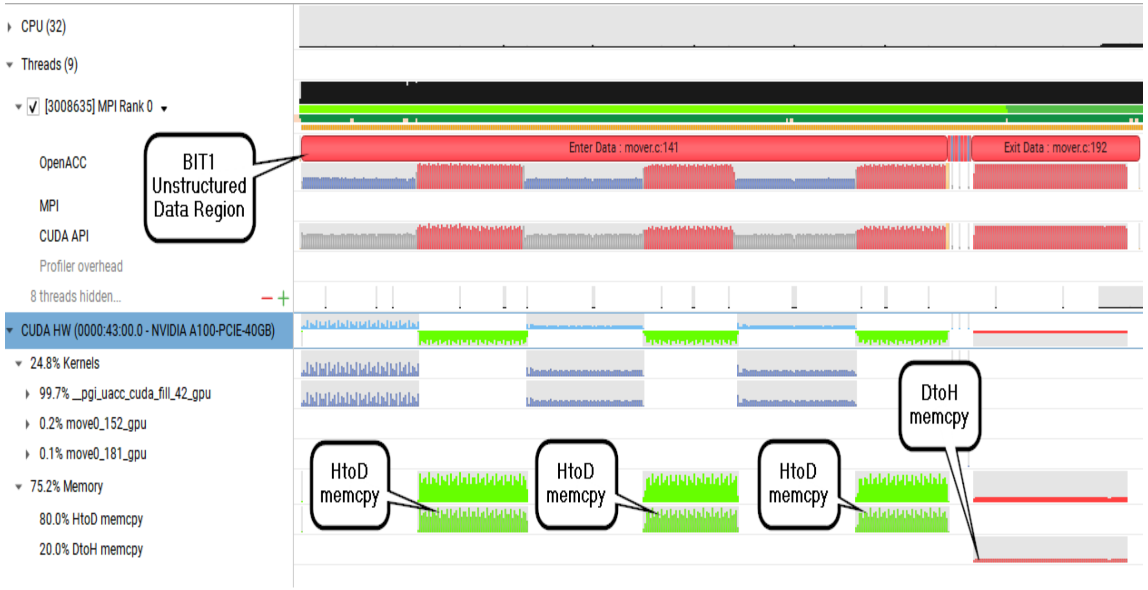


CPU + OpenACC + **GPU**[Co-processor] = **GPU** Acceleration

[1] J. J. Williams, et al. "Optimizing BIT1, a Particle-in-Cell Monte Carlo Code, with OpenMP/OpenACC and GPU Acceleration." 24th International Conference on Computational Science, Málaga, Spain, July 2-4, 2024, Part I, LNCS 14832. Springer Nature.

Results: Accelerating BIT1 with GPUs[1]

NVIDIA **Nsight** Systems OpenACC **Explicit**



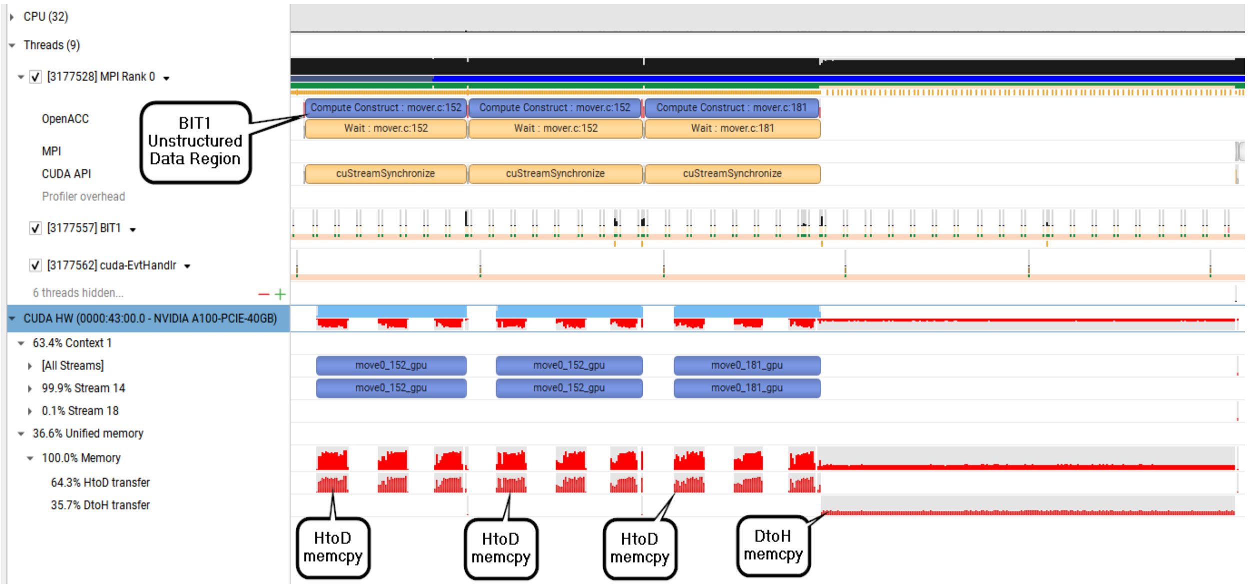
What?

- Explicitly (**manually**) **control** data movement from **CPU** to **GPU**.

Why?

- Provides **fine-grained control** for **optimizing** specific data transfer **patterns**.

NVIDIA **Nsight** Systems OpenACC **Unified Memory**



What?

- Runtime system **automatically handles** data movement from **CPU** to **GPU**.

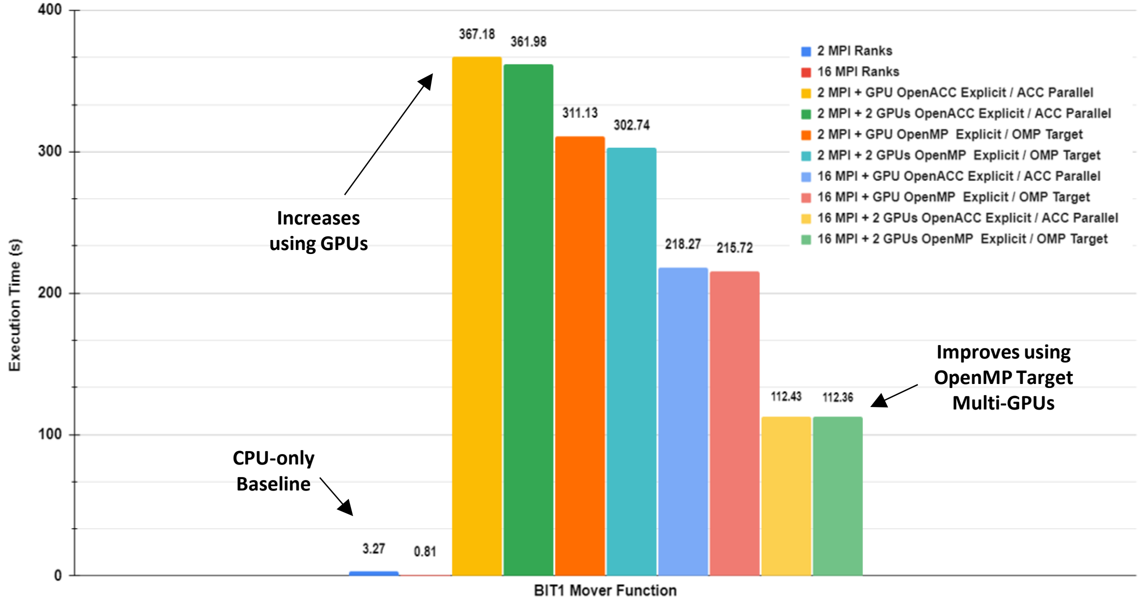
Why?

- Minimizes **manual intervention** with **limited fine-grained control** for **simpler** data transfer.

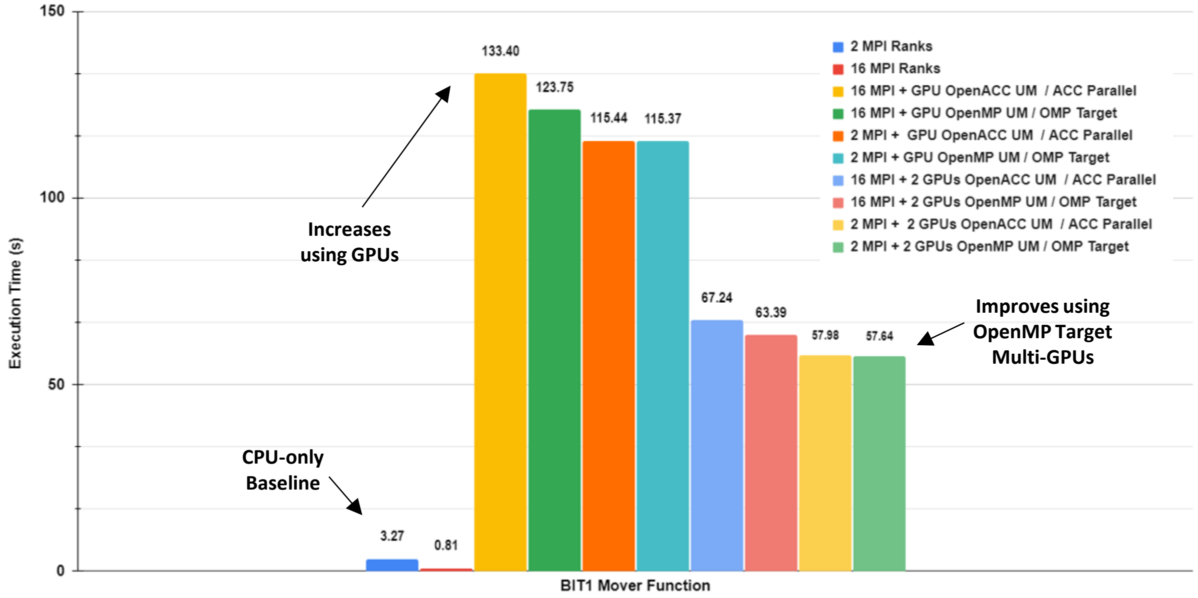
[1] J. J. Williams, et al. "Optimizing BIT1, a Particle-in-Cell Monte Carlo Code, with OpenMP/OpenACC and GPU Acceleration." 24th International Conference on Computational Science, Málaga, Spain, July 2-4, 2024, Part I, LNCS 14832. Springer Nature.

Results: Accelerating BIT1 with GPUs[1]

OpenMP/OpenACC **Explicit**



OpenMP/OpenACC **Unified Memory**



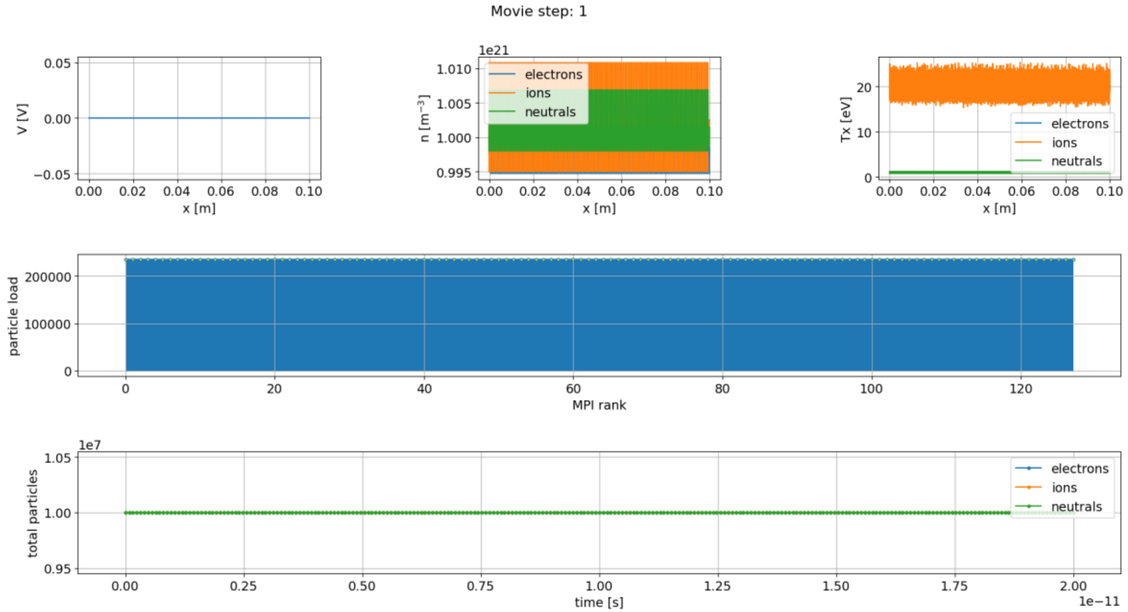
- GPUs (**OpenMP** or **OpenACC**) usage **increases** execution times compared to **2 MPI** and **16 MPI** Ranks **CPU-only** Baseline.
- **OpenMP Target** with **2 GPUs** shows **notable** execution time **decrease**, indicating **improved** performance with **Multi-GPUs**.

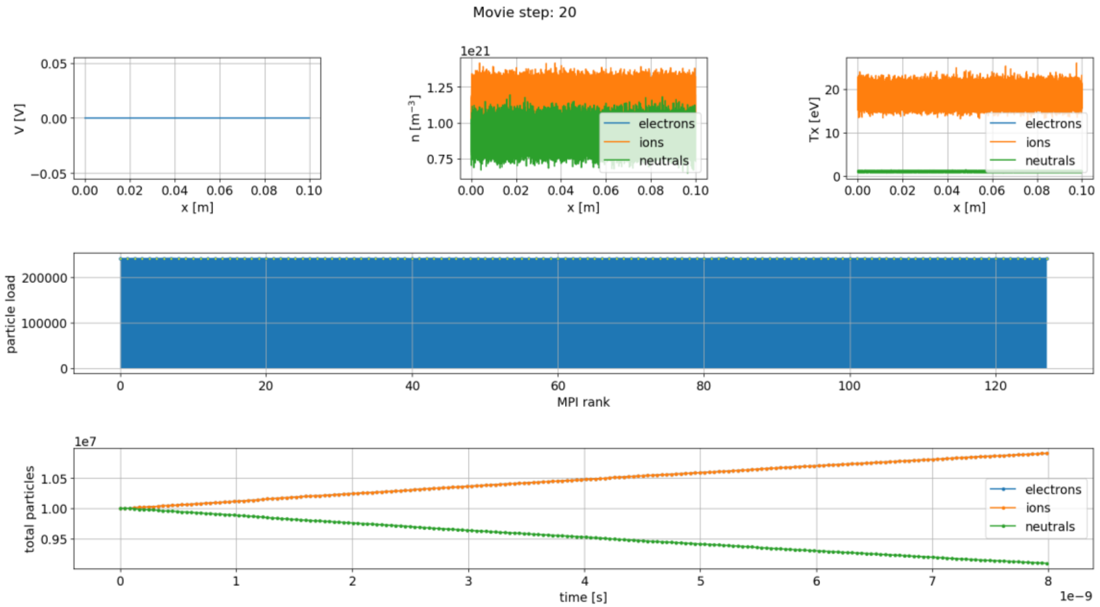
Data Transfer Problem **Discovered** [1]
- We **should not** copy/upload very **large amounts** of **data** from **CPU** to the **GPU** every **time step**.

Explore New **GPU Porting** Possible **Solution** [2]
- Use **CUDA streams** and **particle batch** processing with **OpenMP** Target **Multi-GPUs** per **node**.

[1] J. J. Williams, et al. "Optimizing BIT1, a Particle-in-Cell Monte Carlo Code, with OpenMP/OpenACC and GPU Acceleration." 24th International Conference on Computational Science, Málaga, Spain, July 2-4, 2024, Part I, LNCS 14832. Springer Nature.
[2] Steven WD Chien et al. "sputniPIC: an implicit particle-in-cell code for multi-GPU systems." 2020 IEEE 32nd International Symposium on Computer Architecture and High Performance Computing (SBAC-PAD). IEEE, 2020

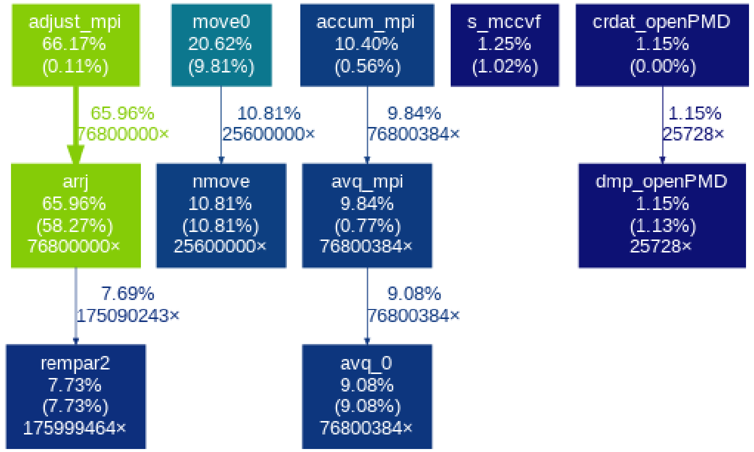
Evaluation: BIT1 openPMD BP4 (CPU) Simulations[1]

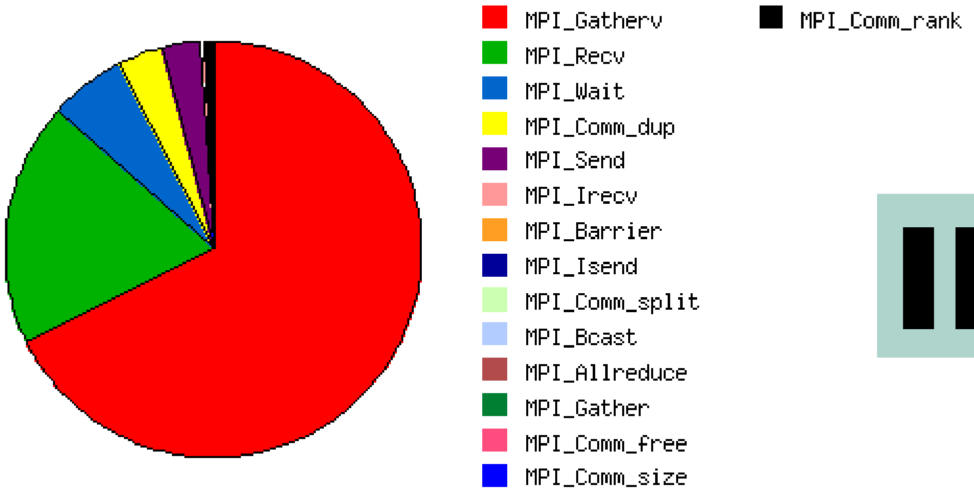


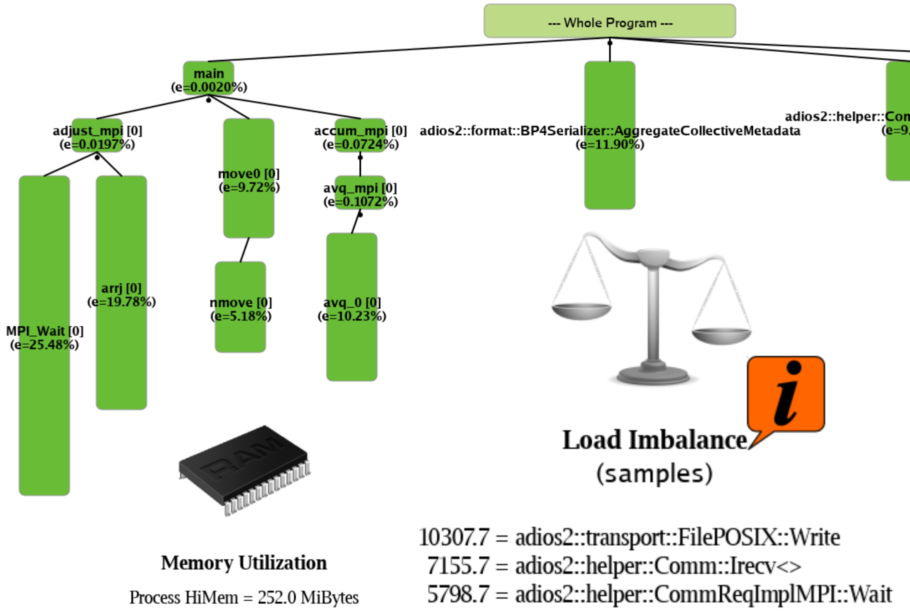


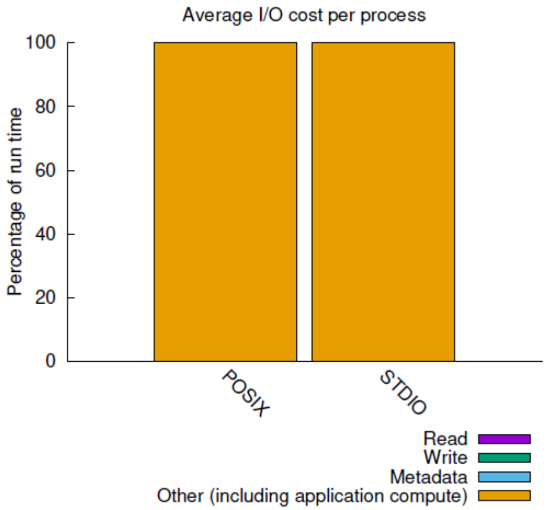
[1] J. J. Williams et al. "Understanding the Impact of openPMD on BIT1, a Particle-in-Cell Monte Carlo Code, through Instrumentation, Monitoring, and In-Situ Analysis." Euro-Par 2024: Parallel Processing Workshops: Euro-Par 2024 International Workshops, Madrid, Spain, August 26–30, 2024. Springer Nature.

Evaluation: HPC Instrumenting, Profiling & Monitoring Tools[1]

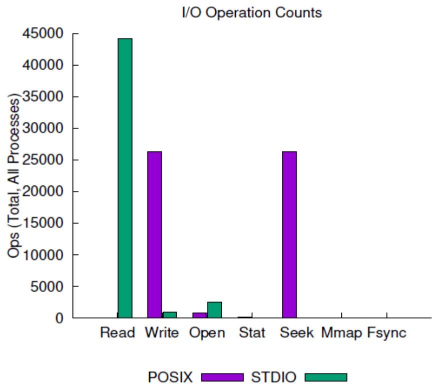








IPM



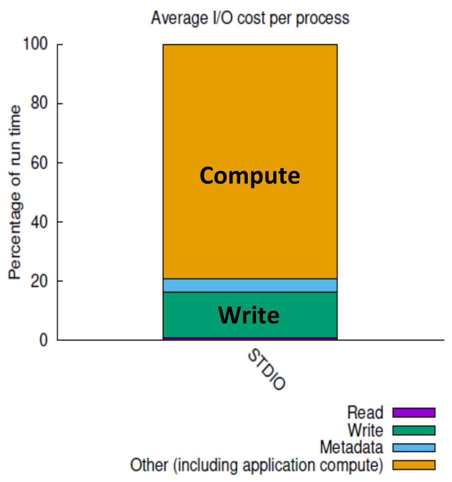
DARSHAN
HPC I/O Characterization Tool

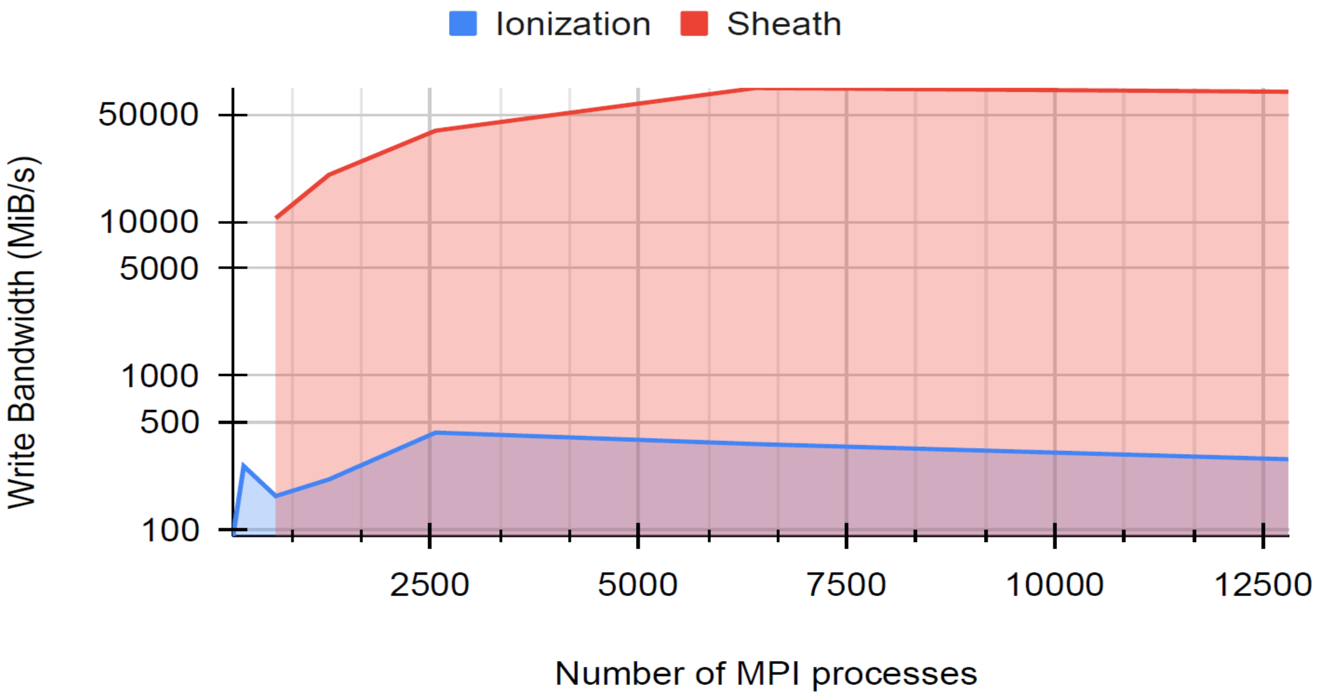
[1] J. J. Williams et al. "Understanding the Impact of openPMD on BIT1, a Particle-in-Cell Monte Carlo Code, through Instrumentation, Monitoring, and In-Situ Analysis." Euro-Par 2024: Parallel Processing Workshops: Euro-Par 2024 International Workshops, Madrid, Spain, August 26–30, 2024. Springer Nature.

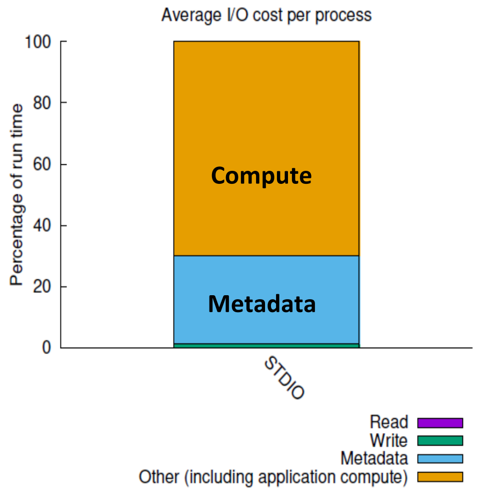
Darshan

Evaluation: BIT1 Original I/O Performance Monitoring[1]

We use **Darshan** to **evaluate** the I/O performance of **BIT1** in terms of **write bandwidth**.







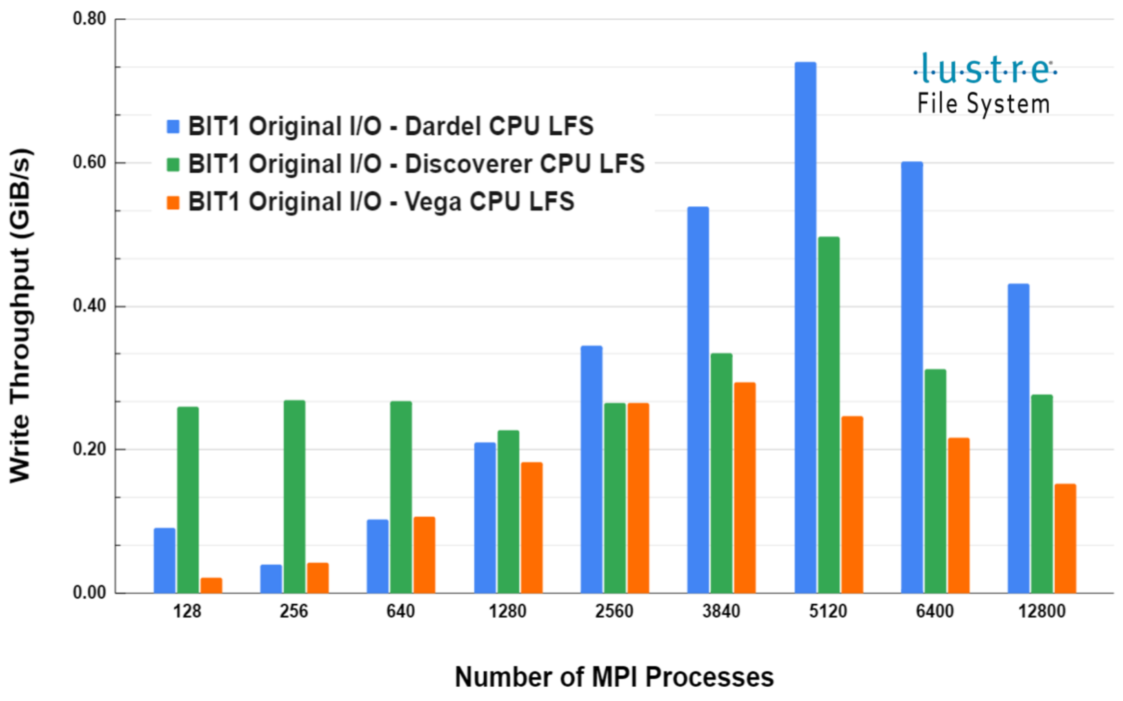
- **Both cases**: Write bandwidth **increases** and then **saturates.**
- Peak I/O write bandwidth linked to problem size.
- Post-peak, performance **drops** from **metadata** writing cost.

DARSHAN
HPC I/O Characterization Tool

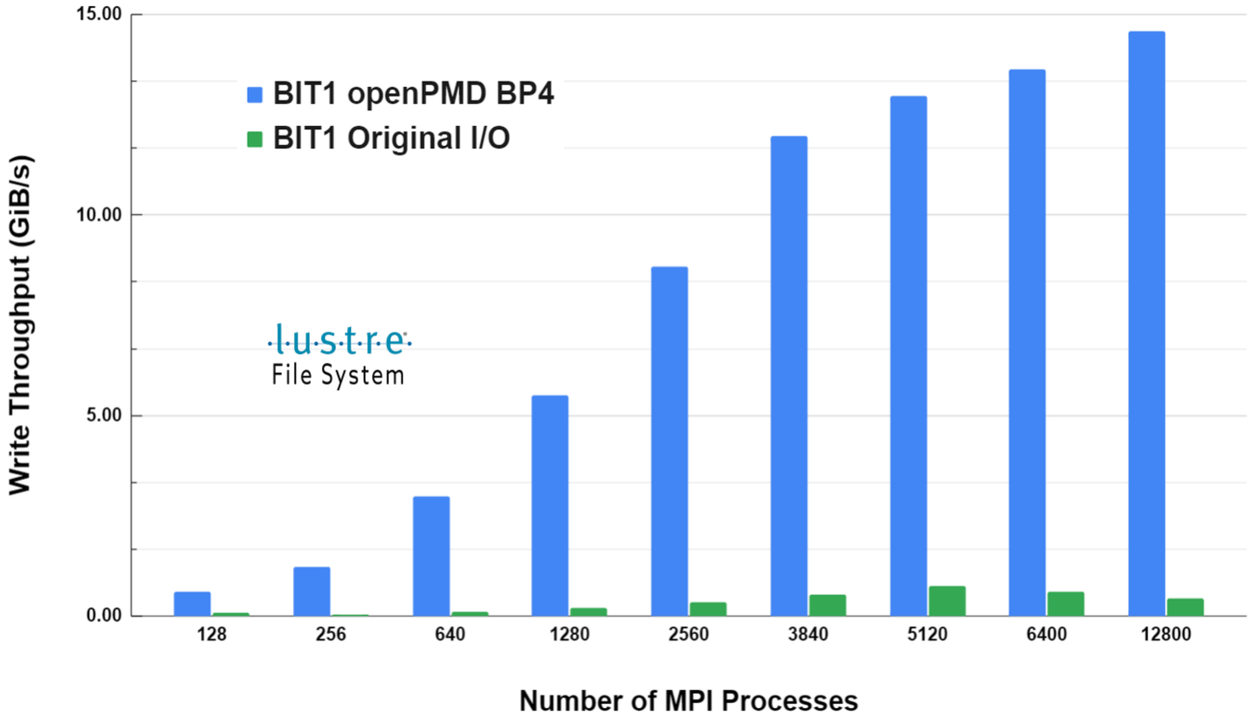
[1] J. J. Williams et al. "Leveraging HPC Profiling & Tracing Tools to Understand the Performance of Particle-in-Cell Monte Carlo Simulations." Euro-Par 2023: Parallel Processing Workshops: Euro-Par 2023 International Workshops, Limassol, Cyprus, August 28–September 1, 2023, Revised Selected Papers, Part I, LNCS 14351. Springer Nature

Results: Enabling BIT1 and Parallel I/O Implementation[1]

Original **File I/O** Write **Throughput**



Parallel I/O Write **Throughput** on **Dardel** CPU



Dardel CPU LFS **outperforms** Discoverer and Vega, **achieving** the **highest overall** write **throughput**.

BIT1 Original File I/O: Achieves **peak** throughput of **0.74 GiB/s** at **40 nodes.**

BIT1 openPMD BP4: Maintains **stable** performance**, suitable** for **tasks** requiring **high-throughput** and **efficiency**.

open Particle-Mesh Data

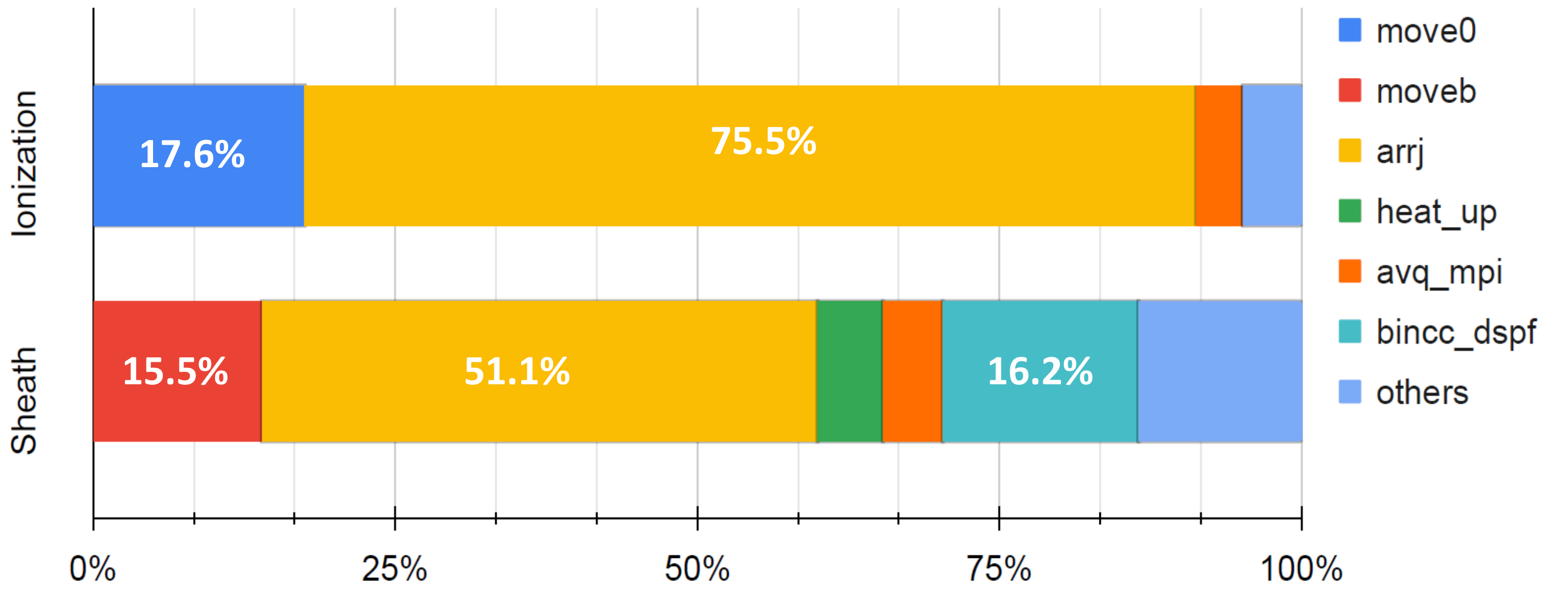
Adaptable Input/Output System 2

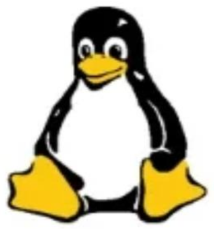
[1] J. J. Williams et al. Enabling High-throughput Parallel I/O in Particle-In-Cell Monte Carlo Simulations with openPMD and Darshan I/O Monitoring. In: 2024 IEEE International Conference on Cluster Computing Workshops (CLUSTER Workshops). Kobe, Japan. IEEE (2024)

gprof

Evaluation: BIT1 Simulation Execution and Runtime Profiling[1]

We use **gprof** to understand where further **BIT1 optimizations** should be carried out to **maximize** the performance gain.





arrj() – **Particle Arrangement** within appropriate **cells** and **ranks**.

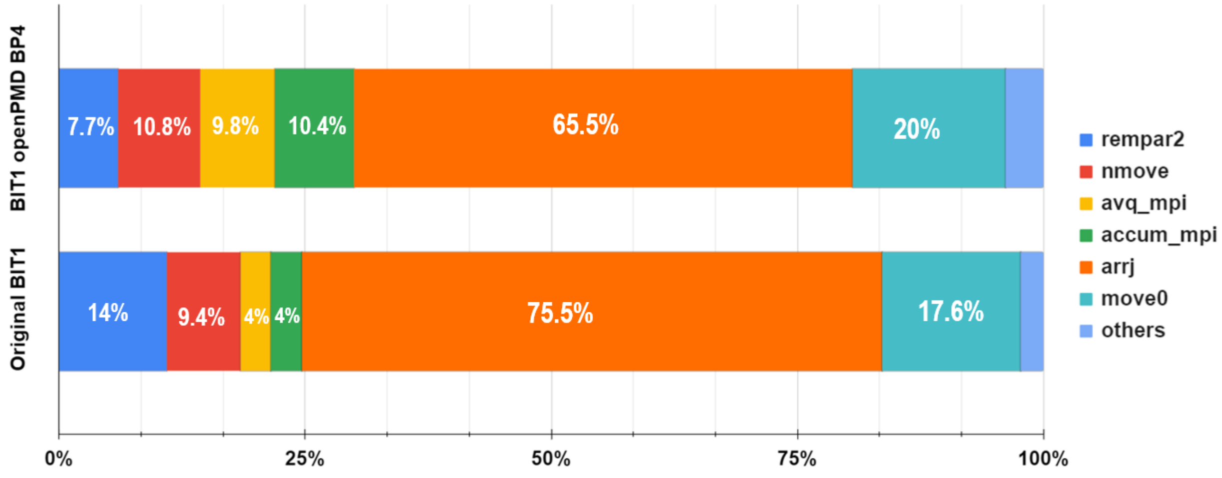
move() / moveb() - **Particle Pusher**

bincc_dspf() – **Binary Collision Operator**

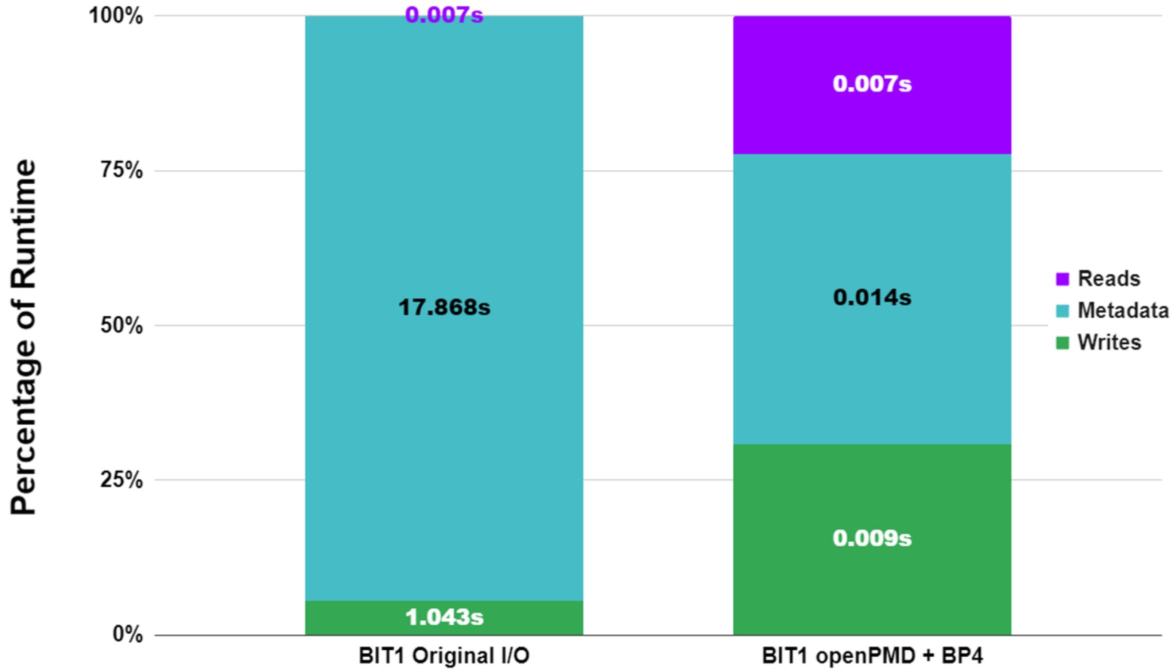
[1] J. J. Williams et al. "Leveraging HPC Profiling & Tracing Tools to Understand the Performance of Particle-in-Cell Monte Carlo Simulations." Euro-Par 2023: Parallel Processing Workshops: Euro-Par 2023 International Workshops, Limassol, Cyprus, August 28–September 1, 2023, Revised Selected Papers, Part I, LNCS 14351. Springer Nature

Results: BIT1 openPMD BP4 Execution and Runtime Profiling[1,2]

gprof: Reports Frequently Used Functions



Parallel I/O Write **Throughput** on Dardel CPU



In the **original BIT1**, the "**arrj**" function uses **75.5%** of the time, **dropping** to **65.5%** with **openPMD BP4**.

Integrating **openPMD** with **ADIOS2 BP4** significantly reduces metadata overhead from **17.868 seconds** to **0.014 seconds** per process (**99.92% reduction**).

Write **operation** time also **decreases** from **1.043 seconds** to **0.009 seconds** per process (**99.14% reduction**).

open Particle-Mesh Data

Adaptable Input/Output System 2

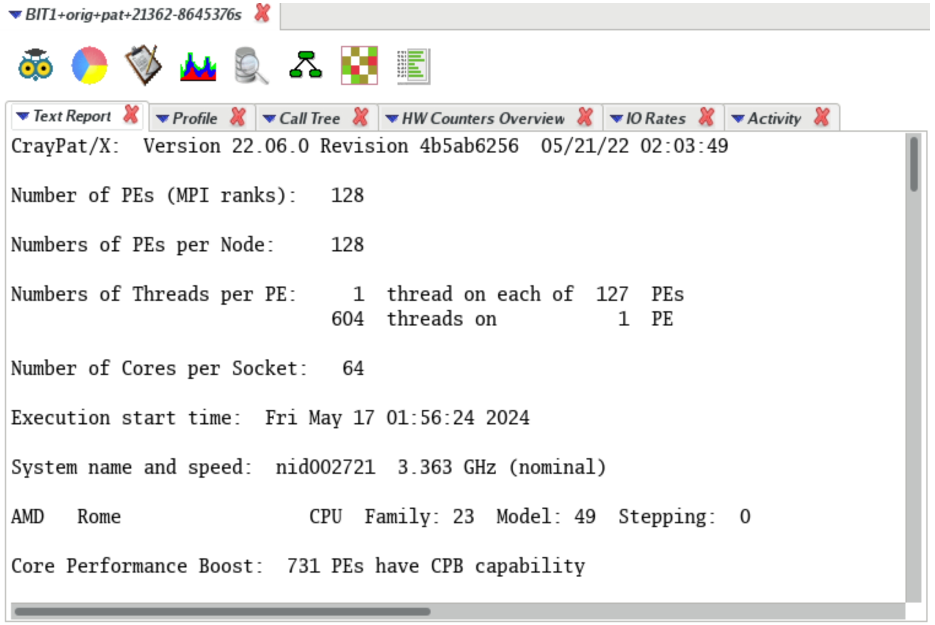
[1] J. J. Williams et al. Enabling High-throughput Parallel I/O in Particle-In-Cell Monte Carlo Simulations with openPMD and Darshan I/O Monitoring. In: 2024 IEEE International Conference on Cluster Computing Workshops (CLUSTER Workshops). Kobe, Japan. IEEE (2024)

[2] J. J. Williams et al. "Understanding the Impact of openPMD on BIT1, a Particle-in-Cell Monte Carlo Code, through Instrumentation, Monitoring, and In-Situ Analysis." Euro-Par 2024: Parallel Processing Workshops: Euro-Par 2024 International Workshops, Madrid, Spain, August 26–30, 2024. Springer Nature.

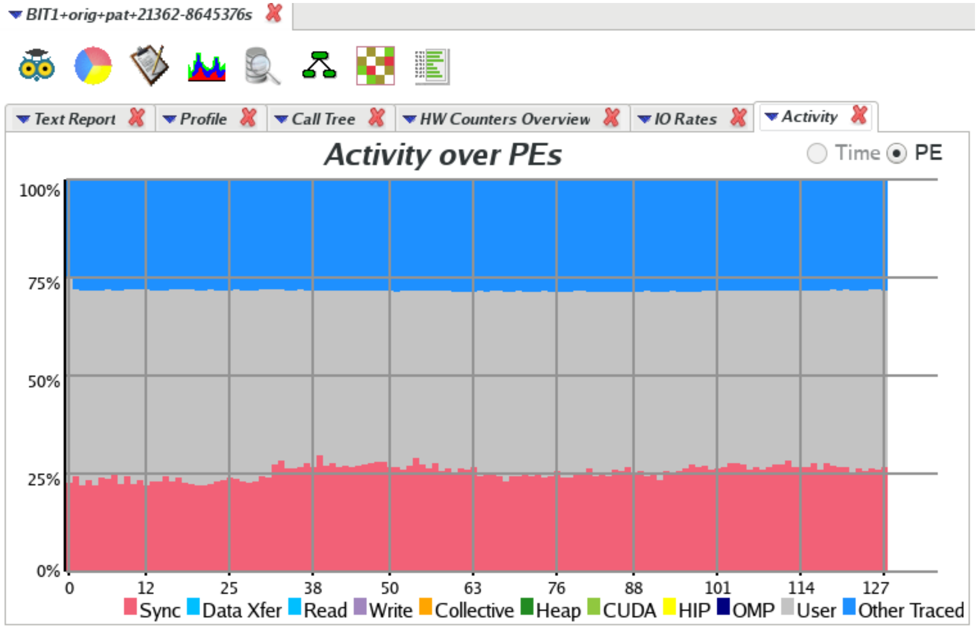
CrayPat & Apprentice2

Evaluation: *CrayPat & Apprentice2 [1]

- **CrayPat & Apprentice2:** parallel application performance tool offered by Cray
- **CrayPat**: Provides detailed information about overall application performance
- **Apprentice2**: GUI tool used to visualize performance data instrumented by CrayPat



Wallclock time: 770.491570s



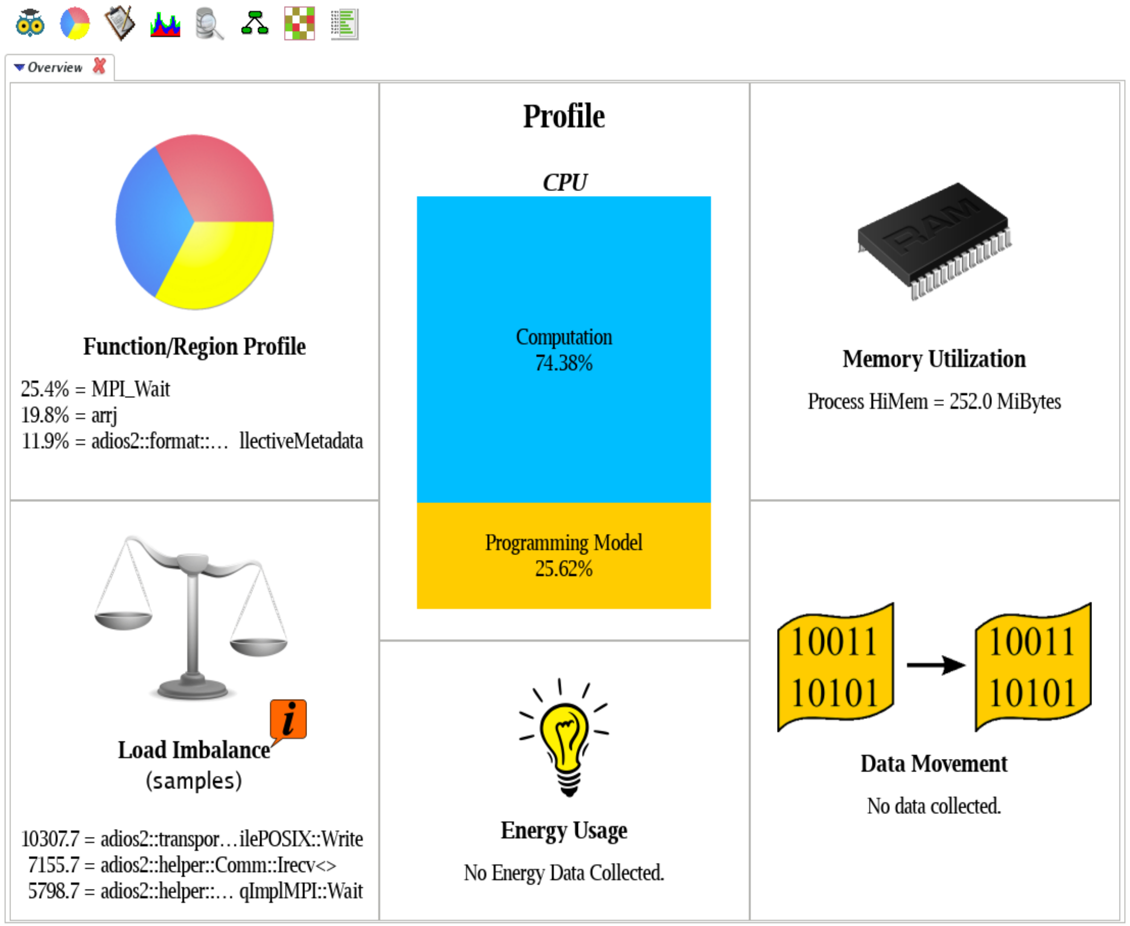
Wallclock time: 770.491570s

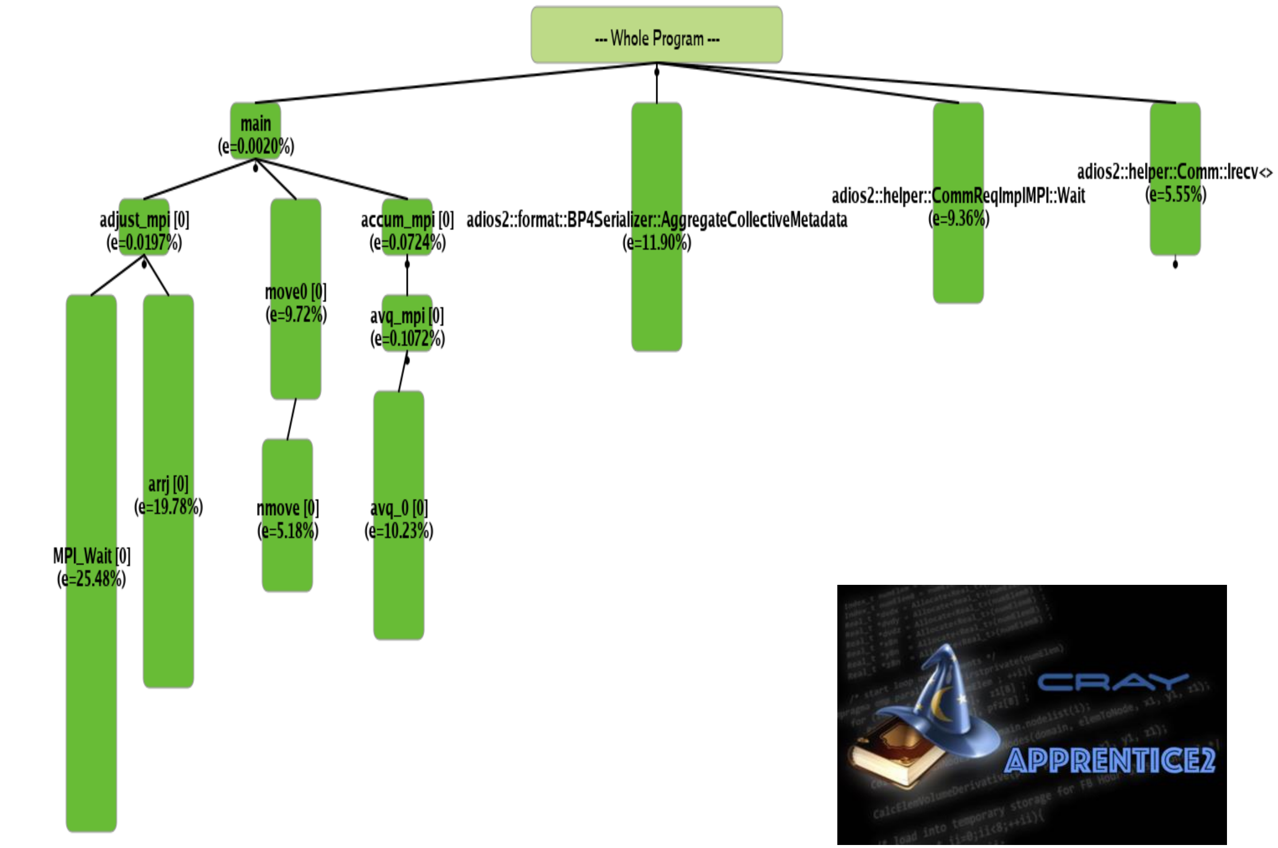
*Cray Performance Measurement and Analysis Toolset (CrayPat)

[1] J. J. Williams et al. "Understanding the Impact of openPMD on BIT1, a Particle-in-Cell Monte Carlo Code, through Instrumentation, Monitoring, and In-Situ Analysis.“ Euro-Par 2024: Parallel Processing Workshops: Euro-Par 2024 International Workshops, Madrid, Spain, August 26–30, 2024. Springer Nature.

Evaluation: BIT1 openPMD BP4 Instrumentation and Profiling[1]

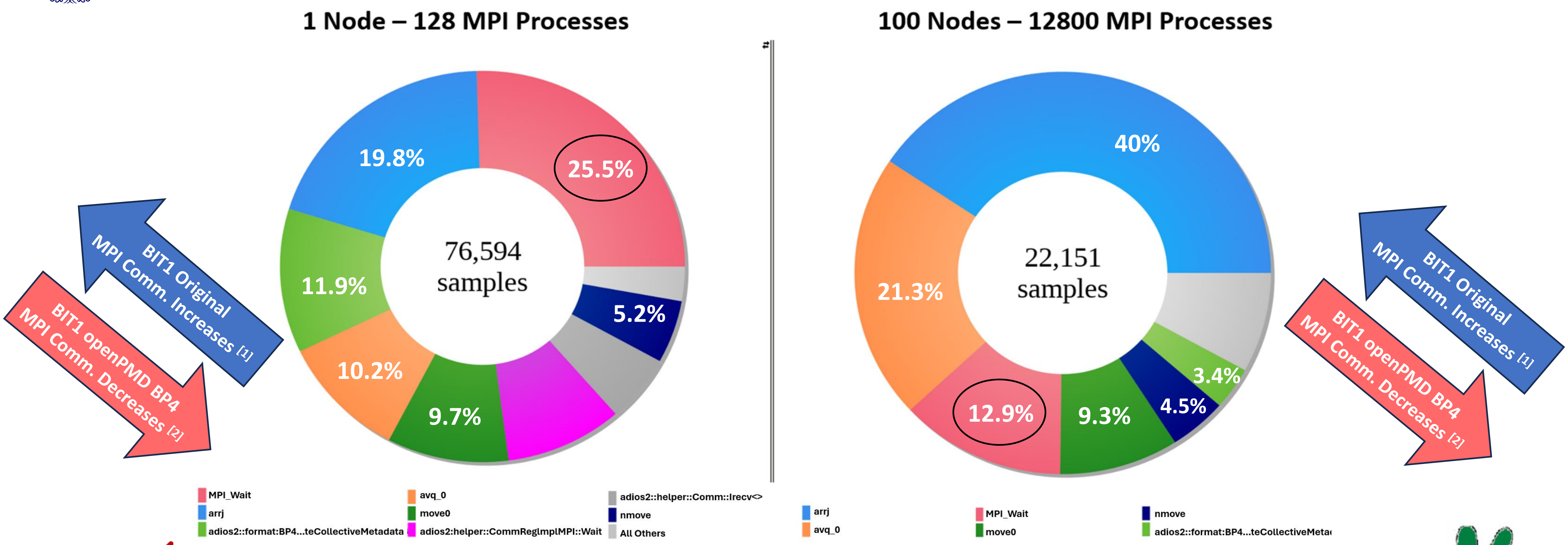
We use **CrayPAT** to instrument the **code** and **Apprentice2** for interactive **performance analysis** and **visualization**.





[1] J. J. Williams et al. "Understanding the Impact of openPMD on BIT1, a Particle-in-Cell Monte Carlo Code, through Instrumentation, Monitoring, and In-Situ Analysis." Euro-Par 2024: Parallel Processing Workshops: Euro-Par 2024 International Workshops, Madrid, Spain, August 26–30, 2024. Springer Nature.

Results: BIT1 openPMD BP4 Instrumentation and Profiling[2]



open Particle-Mesh Data

The **reduction** in MPI **communication** is due to using **openPMD** with the **ADIOS2 BP4** backend, which **optimizes** MPI **communication** and **improves performance** compared to the **original BIT1**.

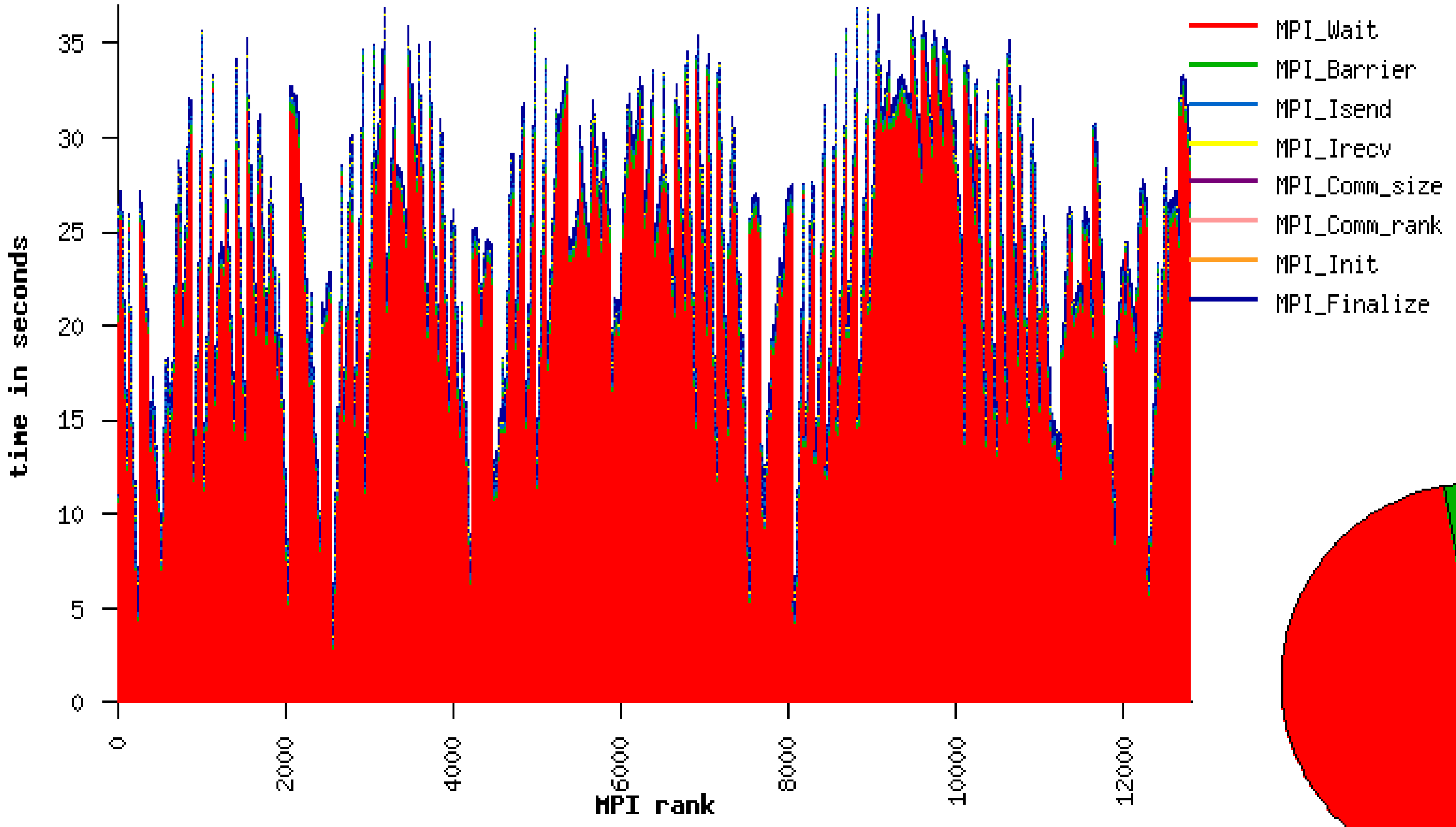
Adaptable Input/Output System 2

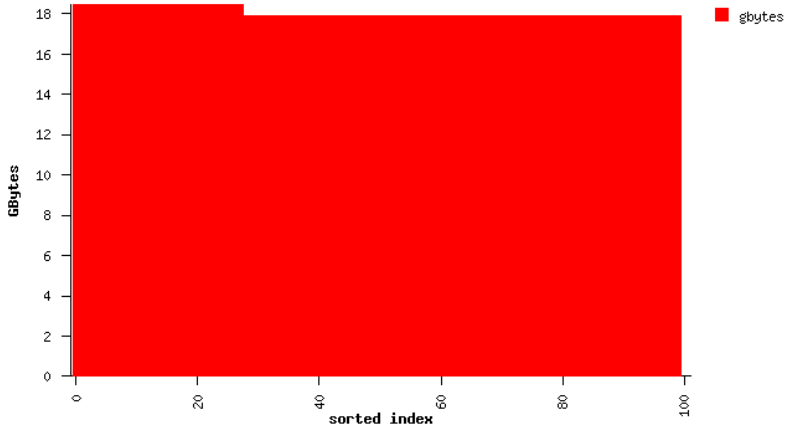
[1] J. J. Williams et al. "Understanding the Impact of openPMD on BIT1, a Particle-in-Cell Monte Carlo Code, through Instrumentation, Monitoring, and In-Situ Analysis." Euro-Par 2024: Parallel Processing Workshops: Euro-Par 2024 International Workshops, Madrid, Spain, August 26–30, 2024. Springer Nature.

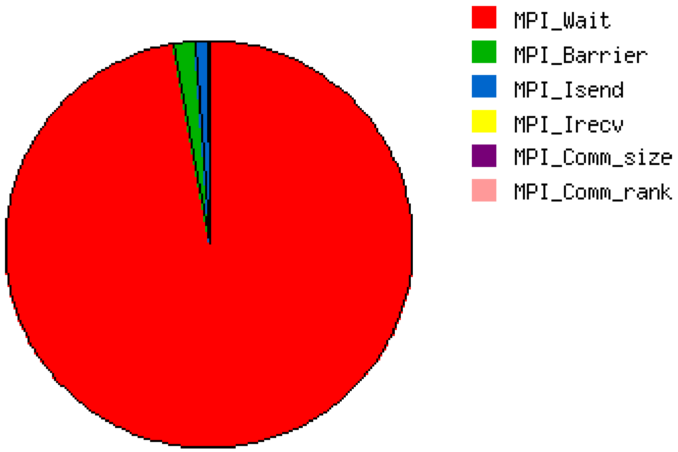
IPM

Evaluation: BIT1 Integrated Performance Monitoring and Profiling[1]

We use **IPM** to **understand** the parallel and **MPI** performance of BIT1.



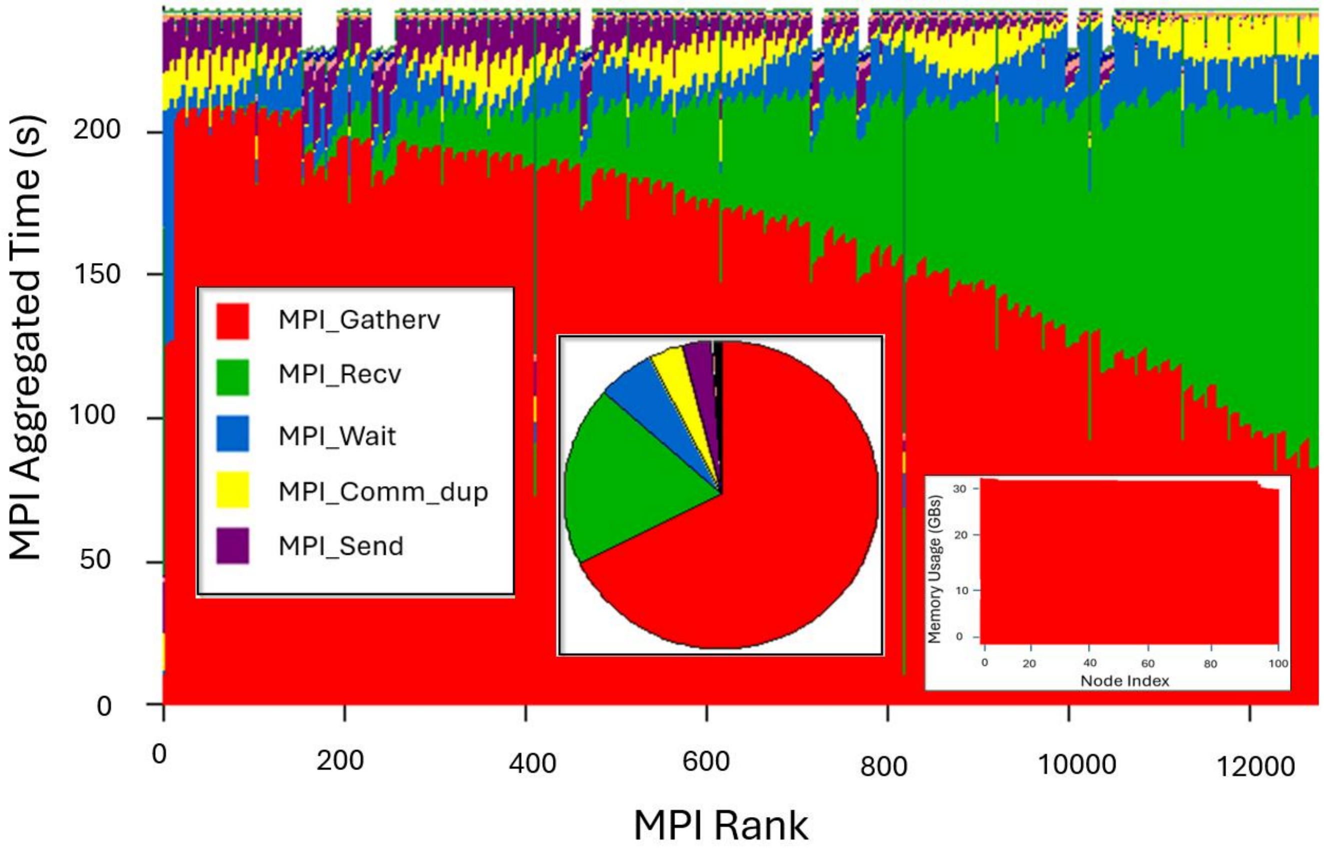




IPM

[1] J. J. Williams et al. "Leveraging HPC Profiling & Tracing Tools to Understand the Performance of Particle-in-Cell Monte Carlo Simulations." Euro-Par 2023: Parallel Processing Workshops: Euro-Par 2023 International Workshops, Limassol, Cyprus, August 28–September 1, 2023, Revised Selected Papers, Part I, LNCS 14351. Springer Nature

Results: BIT1 Integrated Performance Monitoring and Profiling[1]



open Particle-Mesh Data

"MPI_Gatherv" (67.65%) dominates communication time, suggesting a need to optimize **data gathering** processes.

"MPI_Recv" (19.04%) and **"MPI_Wait" (5.76%)** times suggest **inefficiencies** in **message handling** and **synchronization**.

Memory usage per node is **balanced**, ranging from **29 GB** to **33 GB**.

Adaptable Input/Output System 2

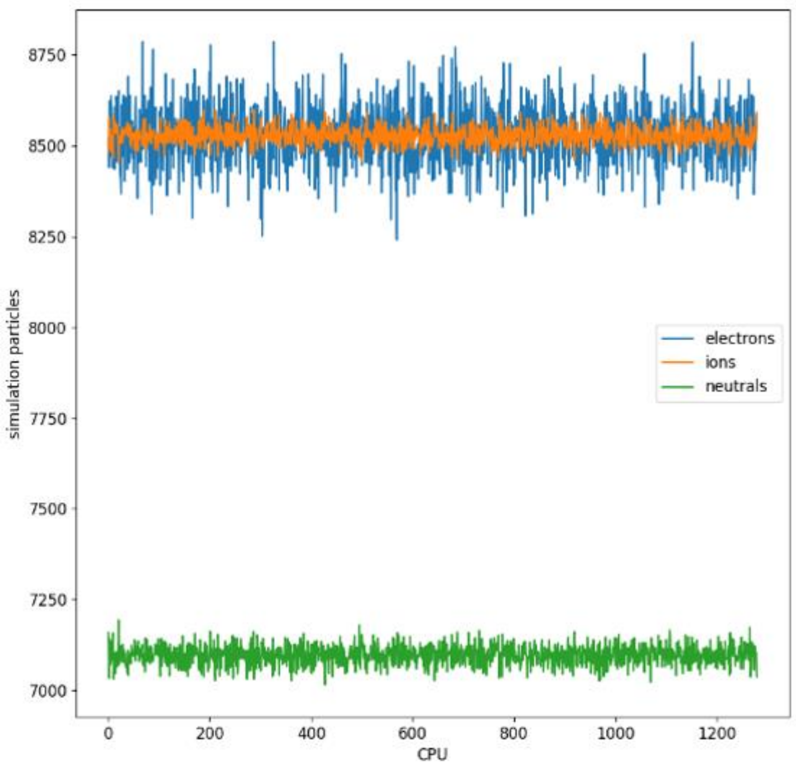
[1] J. J. Williams et al. "Understanding the Impact of openPMD on BIT1, a Particle-in-Cell Monte Carlo Code, through Instrumentation, Monitoring, and In-Situ Analysis." Euro-Par 2024: Parallel Processing Workshops: Euro-Par 2024 International Workshops, Madrid, Spain, August 26–30, 2024. Springer Nature.

Real-Time Checkpoint (In-Situ) Analysis

Evaluation: BIT1 Original Data Analysis[1]

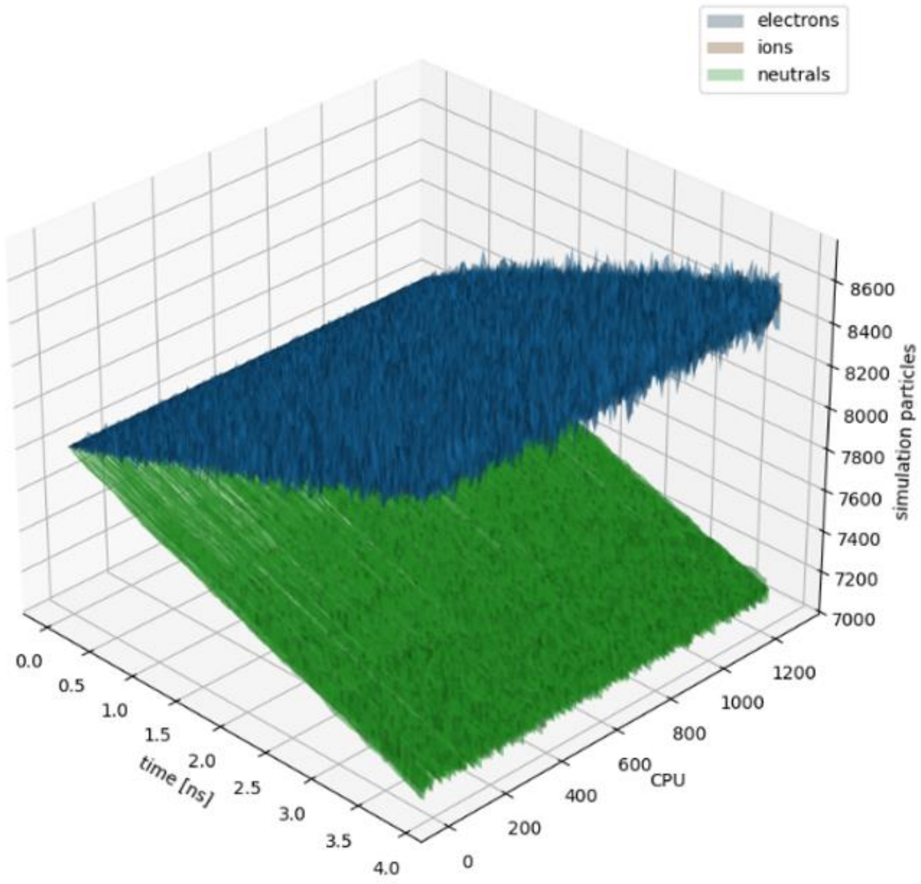
BIT1: 2D Time Evolution

X: CPU | Y: Simulation Particles



BIT1: 3D Time Evolution

X: CPU | Y: Simulation Particles | Z: Time [ns]



[1] J. J. Williams et al. "Leveraging HPC Profiling & Tracing Tools to Understand the Performance of Particle-in-Cell Monte Carlo Simulations." Euro-Par 2023: Parallel Processing Workshops: Euro-Par 2023 International Workshops, Limassol, Cyprus, August 28–September 1, 2023, Revised Selected Papers, Part I, LNCS 14351. Springer Nature

Evaluation: BIT1 openPMD BP4 Real-Time Checkpoint (In-Situ) Analysis[1]

BIT1 Remote User Login

```
python3 /scratch/j/BIT1/Visualize/in-situ-vis.py test_load_balance.inp.bp4
```

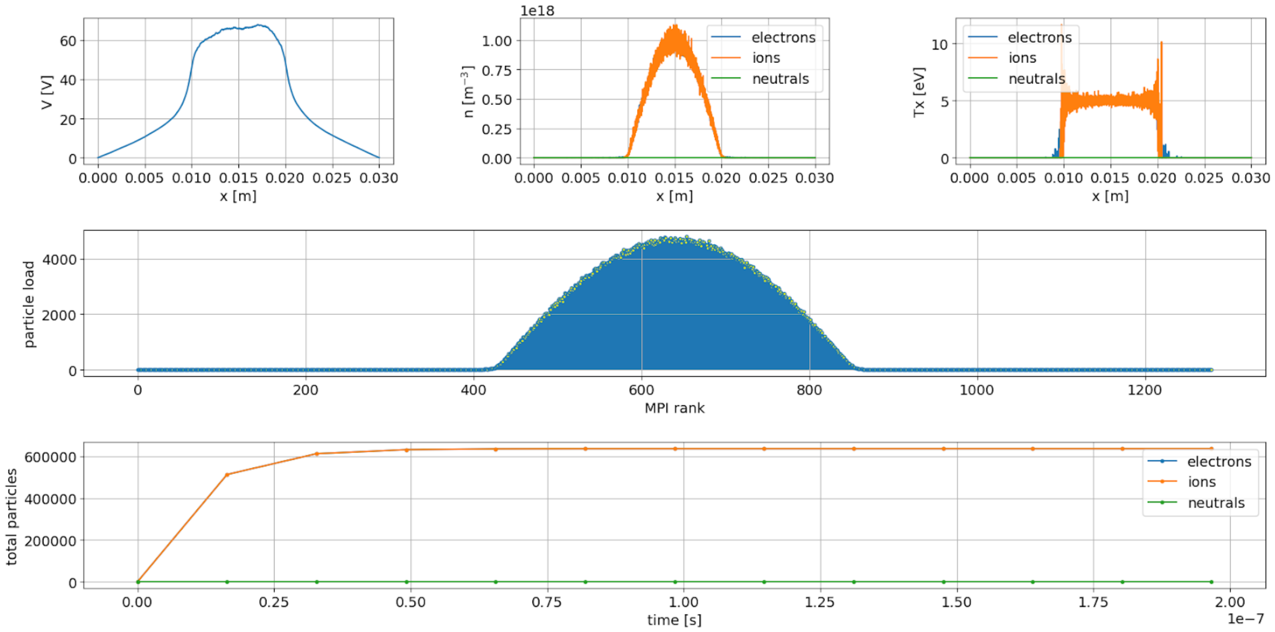
Login Node <-> Interactive Compute Node

```
In-situ visualisation available at: test_load_balance.inp.bp4
Succesfully read the input file.
BIT1 - Bounded Electrostatic 1 Dimensional PIC-MCC Code (Berkeley-Innsbruck-Tbilisi)

Author: D.Tskhakaya, IPP.CR

Code is based on XPDP1 code from University of California - Berkeley

BIT1: Creating series.
Movie: Created iteration 1.
tstep= 1000
Movie: Created iteration 2.
tstep= 2000
Movie: Created iteration 3.
tstep= 3000
Movie: Created iteration 4.
tstep= 4000
Movie: Created iteration 5.
tstep= 5000
Movie: Created iteration 6.
tstep= 6000
Movie: Created iteration 7.
tstep= 7000
Movie: Created iteration 8.
tstep= 8000
Movie: Created iteration 9.
tstep= 9000
Movie: Created iteration 10.
tstep= 10000
BIT1: Close series.
Final tstep= 10000
```



Login Node <-> Interactive Compute Node

BIT1 openPMD BP4 Real-Time Checkpoint (In-Situ) Analysis

[1] J. J. Williams et al. "Understanding the Impact of openPMD on BIT1, a Particle-in-Cell Monte Carlo Code, through Instrumentation, Monitoring, and In-Situ Analysis." Euro-Par 2024: Parallel Processing Workshops: Euro-Par 2024 International Workshops, Madrid, Spain, August 26–30, 2024. Springer Nature.

Evaluation: BIT1 openPMD BP4 Real-Time Checkpoint (In-Situ) Analysis[1]

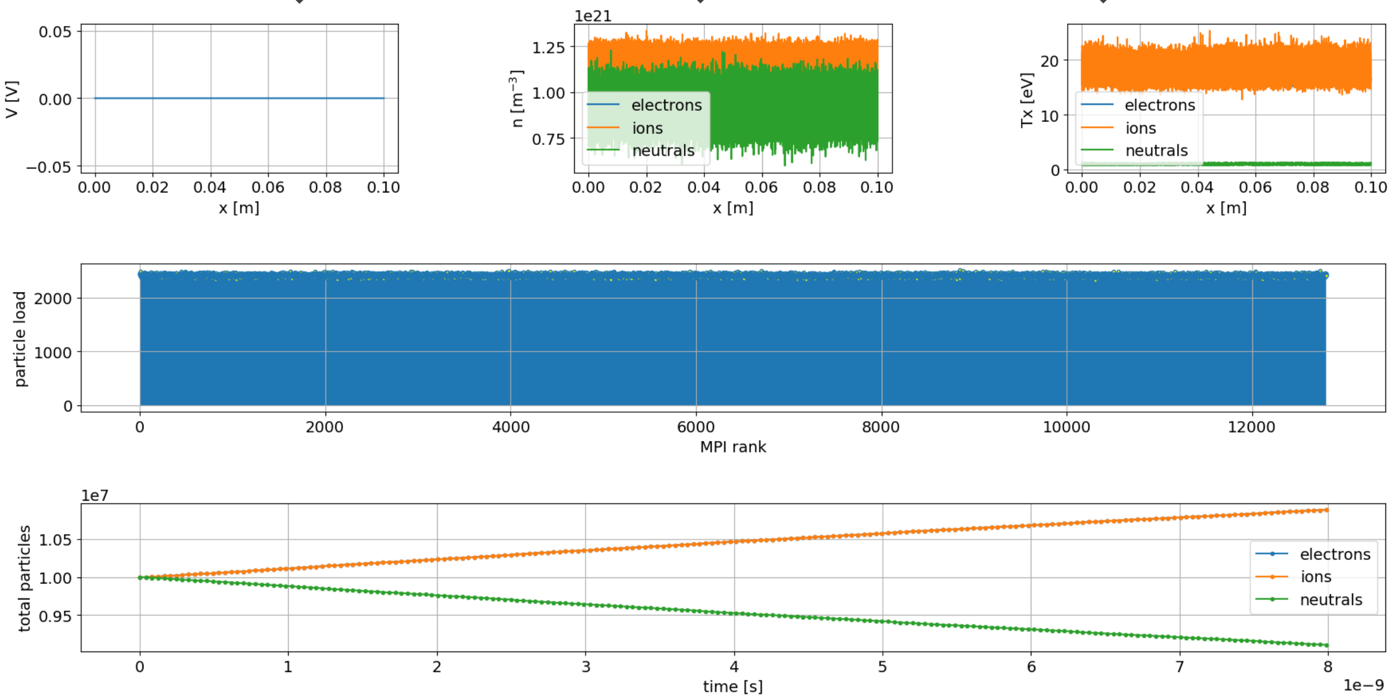
```
python3 /scratch/j/BIT1/Visualize/in-situ-vis.py DCSM_0_e21_c8.inp.bp4
```

```
In-situ visualisation available at: DCSM_0_e21_c8.inp.bp4
Succesfully read the input file.
BIT1 - Bounded Electrostatic 1 Dimensional PIC-MCC Code (Berkeley-Innsbruck-Tbilisi)

Author: D.Tskhakaya, IPP.CR

Code is based on XPDP1 code from University of California - Berkeley

BIT1: Creating series.
Movie: Created iteration 1.
tstep= 10000
Movie: Created iteration 2.
tstep= 20000
Movie: Created iteration 3.
tstep= 30000
Movie: Created iteration 4.
tstep= 40000
Movie: Created iteration 5.
tstep= 50000
Movie: Created iteration 6.
tstep= 60000
Movie: Created iteration 7.
tstep= 70000
Movie: Created iteration 8.
tstep= 80000
Movie: Created iteration 9.
tstep= 90000
Movie: Created iteration 10.
tstep= 100000
Movie: Created iteration 11.
tstep= 110000
Movie: Created iteration 12.
tstep= 120000
Movie: Created iteration 13.
tstep= 130000
Movie: Created iteration 14.
tstep= 140000
Movie: Created iteration 15.
tstep= 150000
Movie: Created iteration 16.
tstep= 160000
Movie: Created iteration 17.
tstep= 170000
Movie: Created iteration 18.
tstep= 180000
Movie: Created iteration 19.
tstep= 190000
Movie: Created iteration 20.
tstep= 200000
BIT1: Close series.
Final tstep= 200000
```



First (1st) Row:

- Plots **electric potential**, **species densities**, and **species temperatures** to **observe** plasma sheath, **monitor** particle transport**,** and **monitor** heat transport.

Second (2nd) Row:

- Displays **particle load** per **MPI rank** to **check workload distribution**; if needed, a restart with **load balancing** can be initialised.

Third (3rd) Row:

- Shows **time evolution of particle counts** to determine if the **simulation** has **reached steady-state**, indicating **balanced sources** and **sinks** with **stable species** numbers.

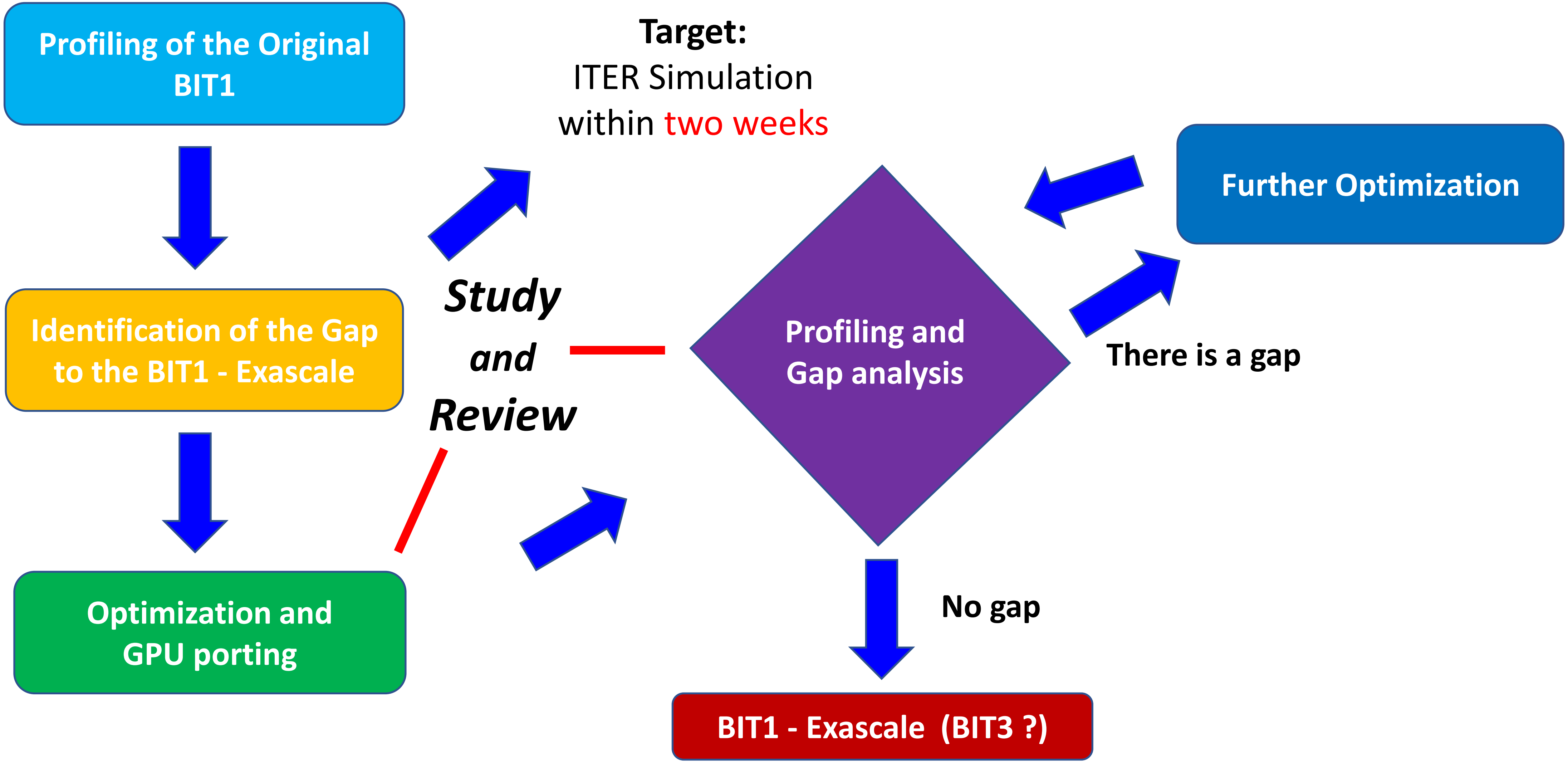
[1] J. J. Williams et al. "Understanding the Impact of openPMD on BIT1, a Particle-in-Cell Monte Carlo Code, through Instrumentation, Monitoring, and In-Situ Analysis." Euro-Par 2024: Parallel Processing Workshops: Euro-Par 2024 International Workshops, Madrid, Spain, August 26–30, 2024. Springer Nature.

Conclusion and Current Work

Conclusion

- **Performance Insights Revealed**:
  - Profiling tools **highlight efficiency improvements** in **data management** and **processing** compared to the **original BIT1 setup**.
- **Data Format Standardization Improved:**
  - openPMD enhances **accuracy** and e**fficiency** in **plasma-material interaction** studies, **advancing** the **development** of **fusion device** designs.
- **Key Enhancements Achieved:**
  - ADIOS2 BP4 **backend** integration **improves** write throughput **bottlenecks.**
  - **Enhanced** visualization and in-situ analysis have **reduced post-processing times** and **provided real-time insights** without **system runtime** interruptions.

Current Work: Enabling BIT1* and Exascale Workflow

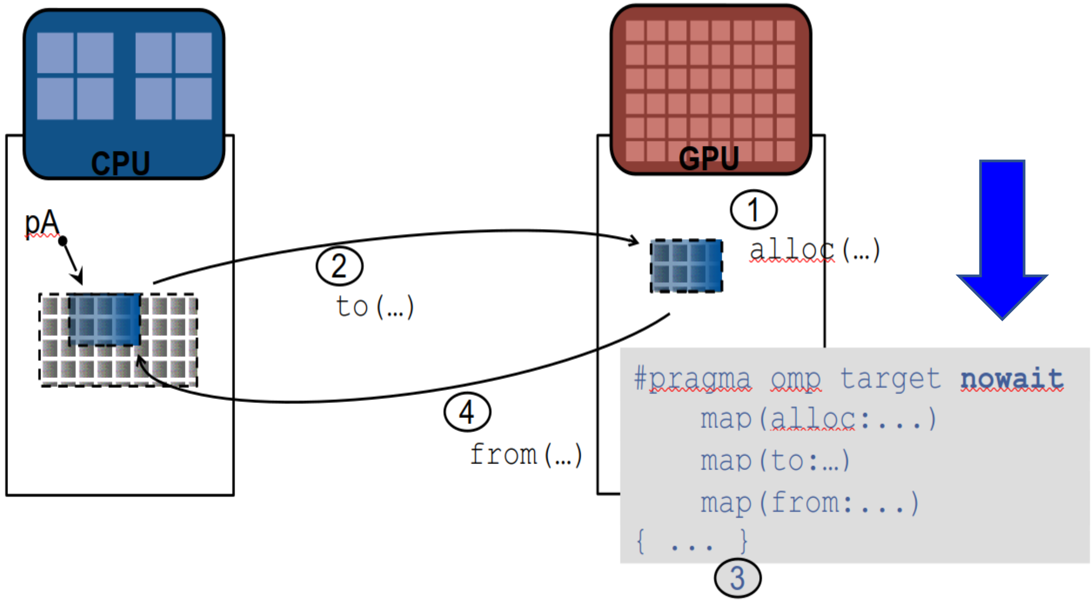


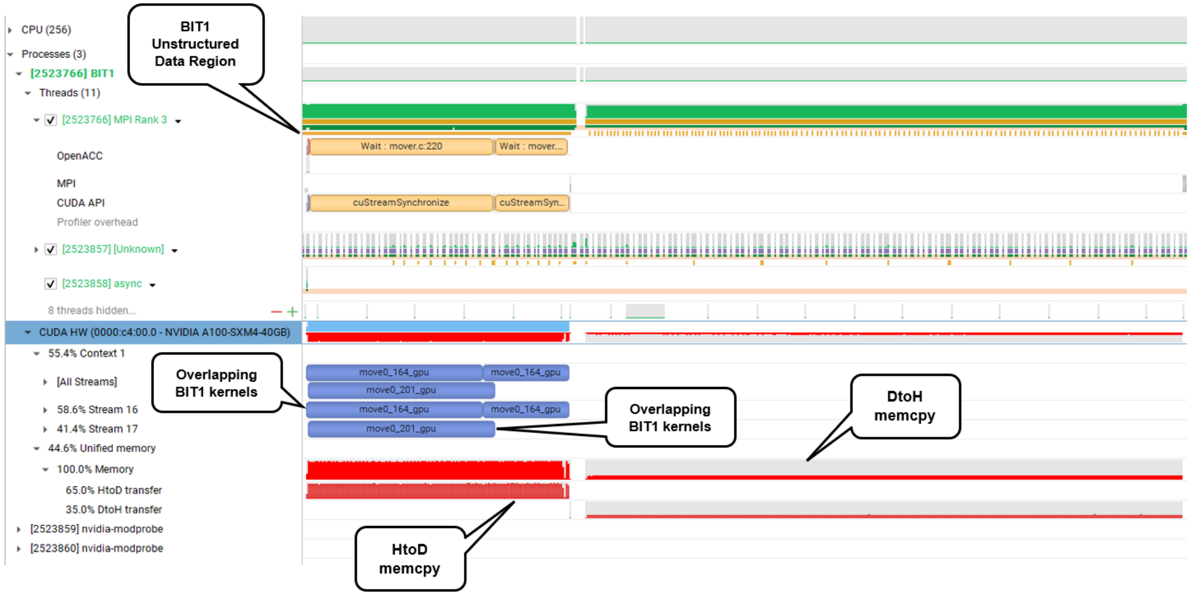
*1 exaflop - 1 quintillion (10^{18}) floating-point operations per second

Current Work

- **Explore and Introduce Hybrid Approaches**:
  - **Maximize Performance:** Combine MPI with Thread/Multicore Utilization and openPMD.
  - **Scalable Optimization:** Balance Hybrid Strategy & Reduce Execution Time.

- **Optimizing Data Transfer with Asynchronous Execution[1]:**
  - **Asynchronous Kernel Offloading (OpenACC and OpenMP):** Improve **data transfer speed** and **efficiency** in the **HPC** system (e.g., **memory**, **storage**, **network, etc.**).



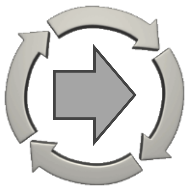


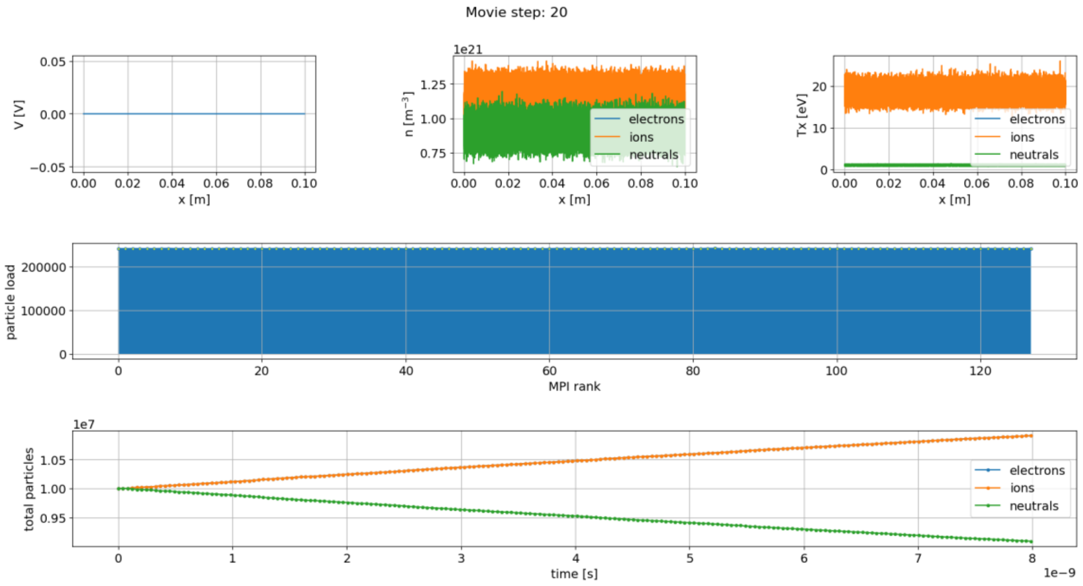
[1] Performance Optimisation and Productivity, A Centre of Excellence in HPC, Accessed 30 August 2024. https://pop-coe.eu/blog/26th-pop-webinar-asynchronous-gpu-programming-in-openmp

Current Work

- **Extend openPMD Supported Features:**
  - Work on **checkpoint restart** and **particle load balancing** with **openPMD** to extend support for **exascale computing** and improve **plasma-material interface** control and **system** resilience.

- **Optimize MPI Communication** and **Improve Disk Storage:**
  - Focus on **reducing** and **improving** MPI **communication overhead** on **large runs**.
  - Explore integrating **high-performance streaming** with the **Sustainable Staging Transport (SST)** backend, extending **real-time checkpoint analysis** capabilities in **memory** rather than **file I/O**.

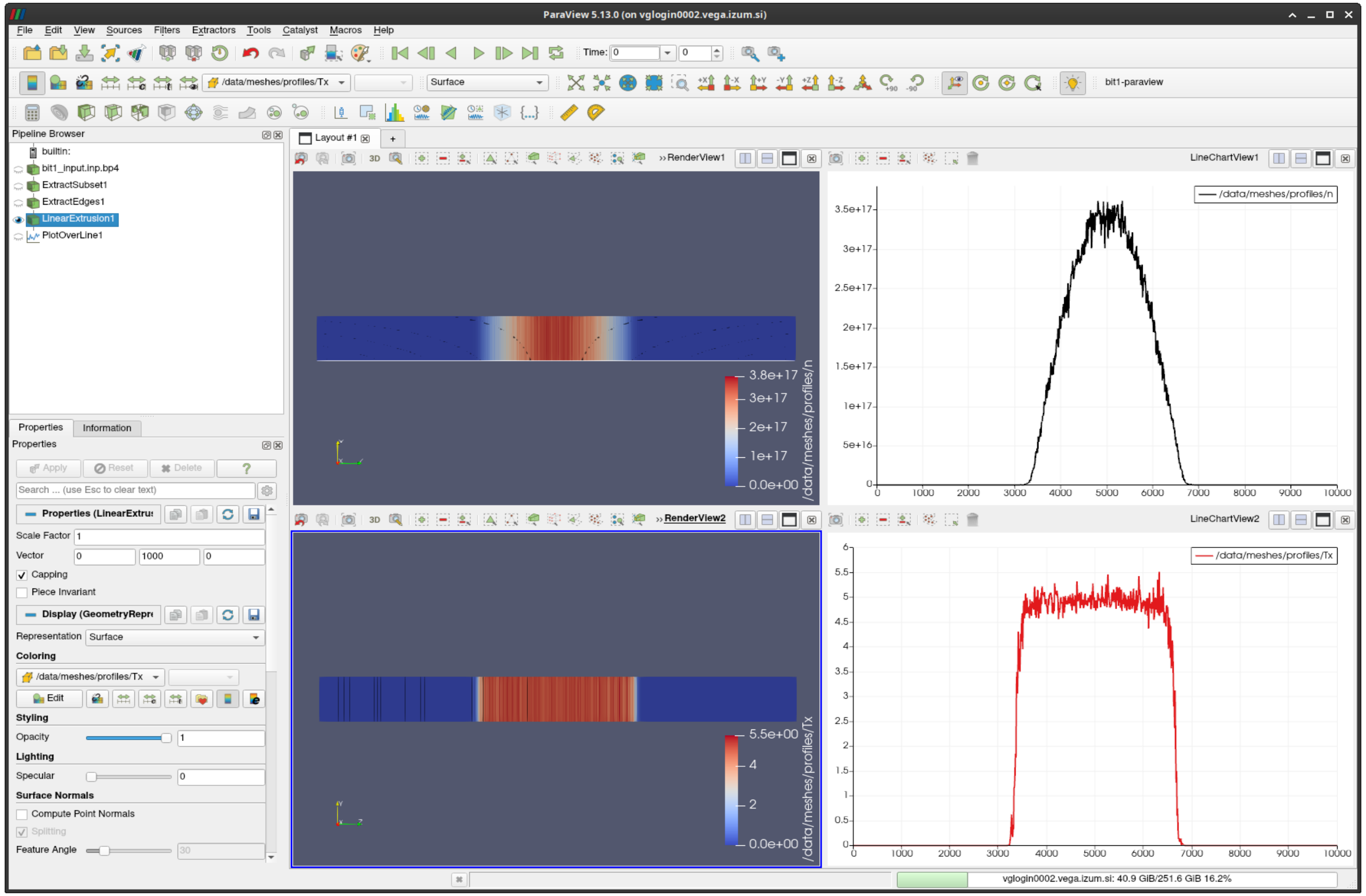




BIT1 Remote User Steaming

Current Work

- **Introduce / Enable Enhanced Features with the SST backend:**
  - Explore **in-situ visualization** using **ParaView Catalyst 2** and **ADIOS2** to support **real-time** debugging.



Thanks!

This project has received funding from the European High-Performance Computing Joint Undertaking (JU) under grant agreement No 101093261. The JU receives support from the European Union's Horizon 2020 research and innovation programme and Sweden, Germany, France, Spain, Finland, the Czech Republic, Slovenia, and Greece.