The Artificial Scientist - Leveraging In-transit Machine Learning for Plasma Simulations

Sunita Chandrasekaran
David L. & Beverly J.C. Mills Career Development Chair
Associate Professor, Dept. of Computer & Information Sciences
Co-Director, AI Center of Excellence
State of Delaware, Vice Chair AI Commission

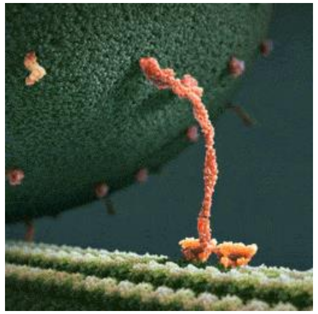
University of Delaware
schandra@udel.edu

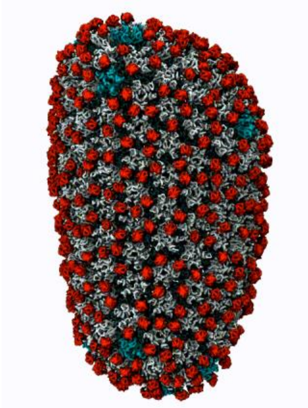
May 6th, 2025, NHR PerfLab Seminar

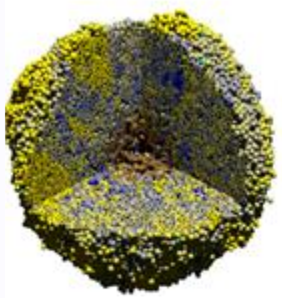
CRPL

Computational Research Programming Lab Overview

- Enabling scientific advancements by migrating real-world applications on novel and powerful hardware
- Advancing foundational computer science by building effective compiler tools
- Understanding physics simulations with ML-enhanced models
- Exploring GenAI/Large Language Models to complement manual software testing



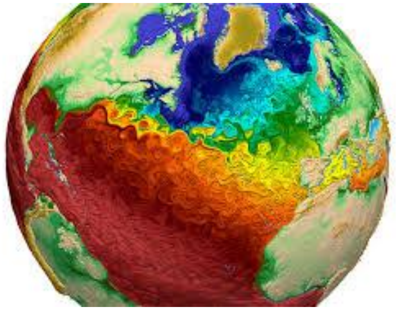




NSF U.S. DEPARTMENT OF ENERGY NVIDIA leidos OpenACC More Science, Less Programming HZDR HELMHOLTZ ZENTRUM DRESDEN ROSSENDORF CASUS CENTER FOR ADVANCED SYSTEMS UNDERSTANDING PLASMA PEPSC

A Grand Challenge: Extracting meaningful insights from complex data streams

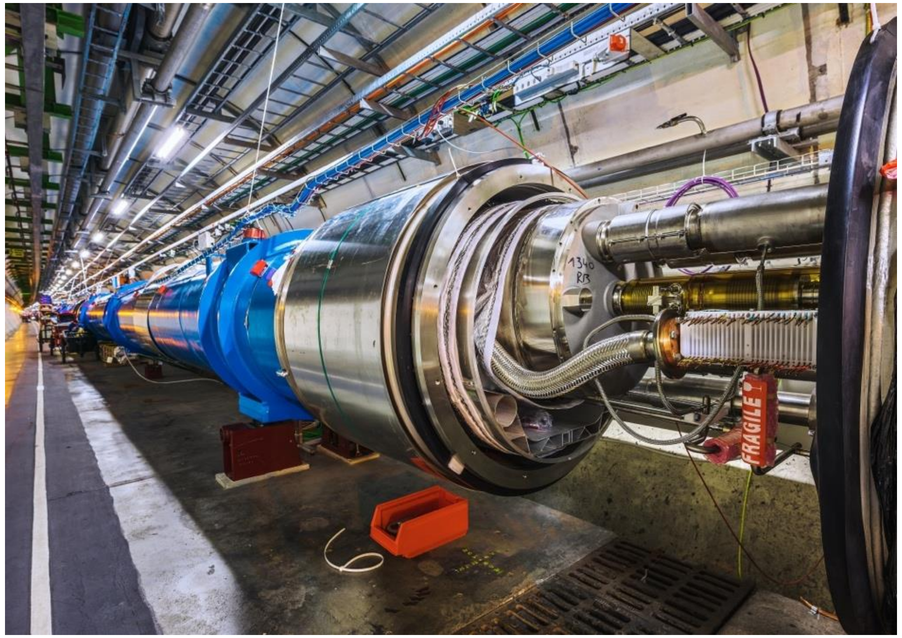
- Simulation or scientific instruments produce complex data at massive scale
- Storage of such data for offline analysis is impractical
- Physical constraints of the file system pose a massive challenge
- Data reduction will simply NOT suffice
- A complete view of full data is NEVER available
- Need solutions for online analysis of data generated at high rate and volume - to extract meaningful information



Climate & Weather Modeling

- Earth System Models (ESMs) dataset is multi-dimensional, diverse, high-resolution including structured and unstructured.
  - Models can generate TBs of data, with long-term simulations spanning centuries producing PBs of data.
- Destination Earth project – Digital Twin
- ECMWF, ICON, MPAS-A, MPAS-O

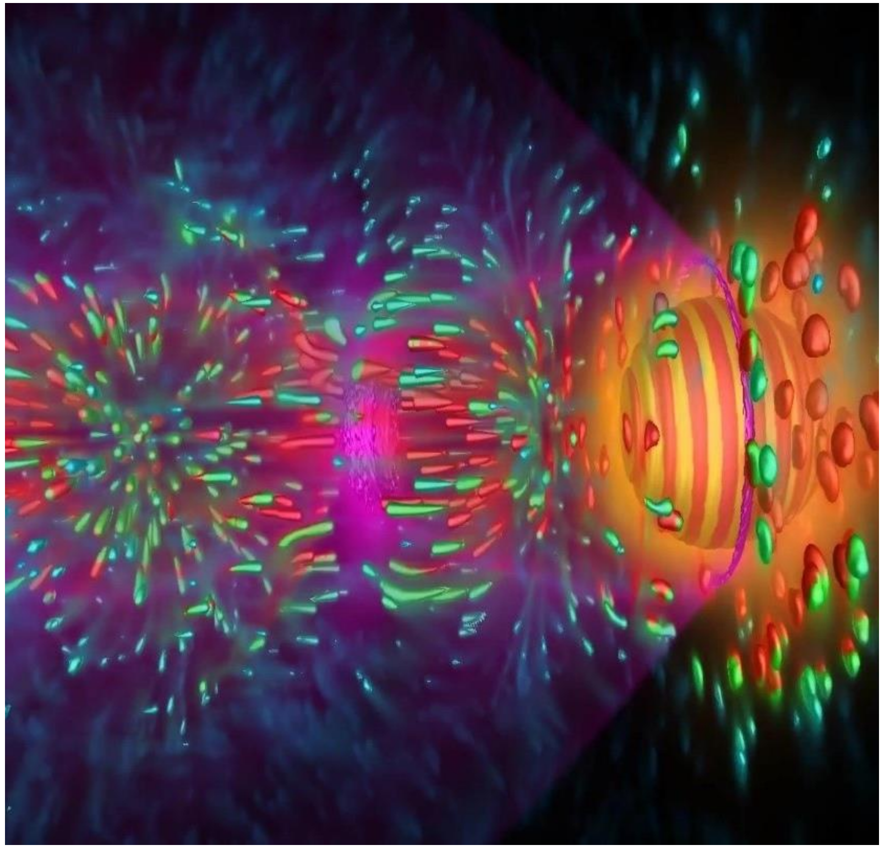
CERN's Large Hadron Collider (LHC)



- CERN's Large Hadron Collider (LHC) detectors generate over 1 billion collisions per second, with only a fraction of them being recorded and analyzed due to the sheer volume of data. This still results in TBs of data per day.
- Discovery of the Higgs boson, the LHC produced around 30 PB of data per year from collisions

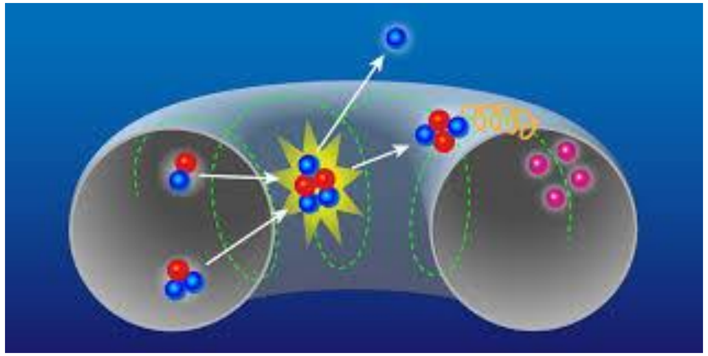
Plasma Physics – governed by the physics of charged particles

- Diverse data properties
- Microscopic particle interactions to macroscopic magnetic field
- Every simulation tracks millions to billions of particle
- Plasma Instabilities



Credit: Felix Meyer (former HZDR, now NVIDIA) Real-Time Vector Field Visualization test using HZDR Hemera Cluster with 4 NVIDIA V100.

Plasma Instabilities – Implications



Sudden violent reaction in fusion research

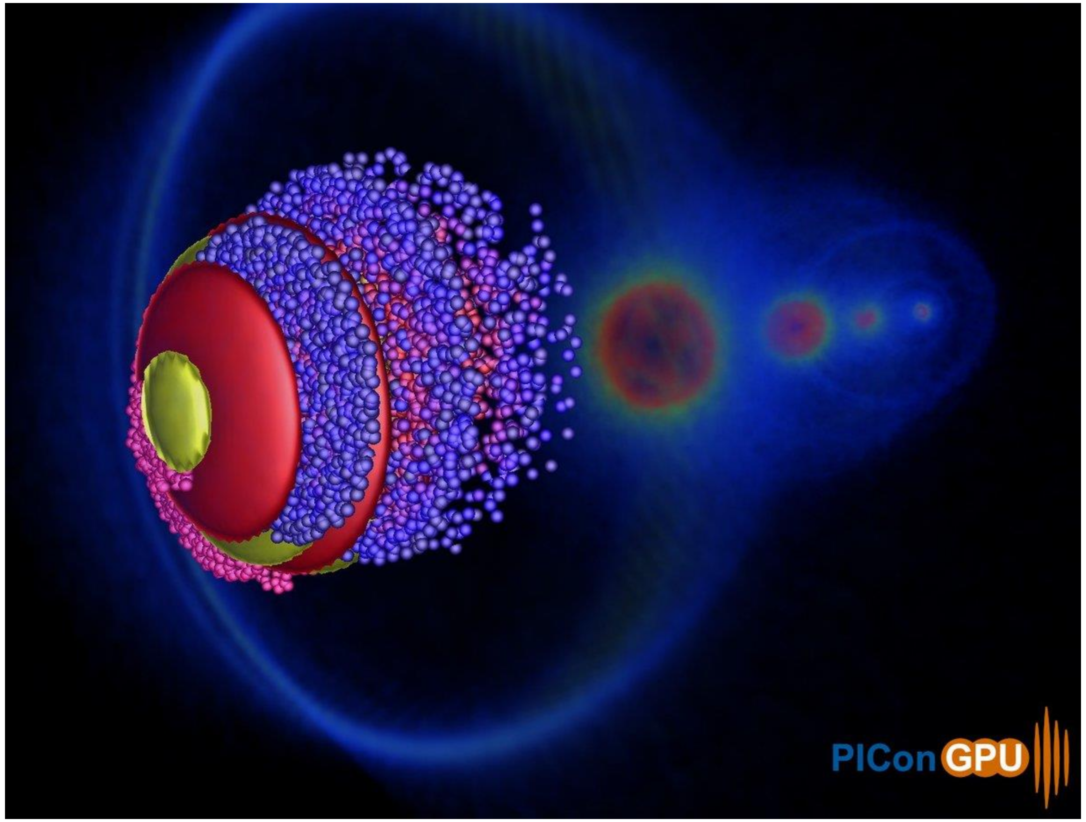
Solar flares and enormous quantities of radiation

Geomagnetic Storms and cosmic rays

Magnetic reconnection

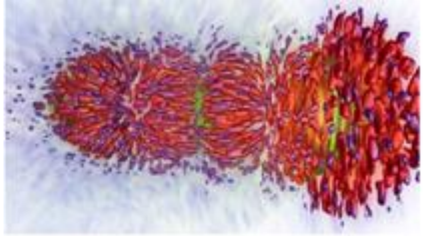
What do we want to learn?

- Complete reconstruction of phase space from observations to get a detailed view of the growth and dynamics of instabilities
- Automatic detection of correlations between plasma dynamics and emitted radiation spectrum
  - Otherwise this requires extensive post-processing and analysis spanning years
- Extracting meaningful info out of complex heterogeneous data being generated from source or simulation at an exponential rate



Credit: Rene Widera, HZDR
Real-Time Vector Field Visualization test using HZDR Hemera Cluster with 4 NVIDIA V100.

The Plasma-In-Cell on GPU (PIConGPU)team



S. Chandrasekaran (PI) [2, 3], A. Debus[1], T. Kluge[1], R. Widera[1], K. Steiniger[1], M. Garten[4], M. Werner, J.Kelling[1], R. Pausch[1], B. Hernandez[7] , F. Meyer[7], V. Gutta[3], F. Mora[3], F. Pöschel[1,2], J. Young[2,5], B. Worpitz, A. Huebl[4], D. Rogers[6], G. Juckeland[1], M.Bussmann[1,2]

[1] Helmholtz-Zentrum Dresden-Rossendorf, Dresden, Germany
[2] CASUS, Center for Advanced Systems Understanding, Goerlitz, Germany
[3] University of Delaware, Newark, Delaware, USA
[4] Lawrence Berkeley National Laboratories, Berkeley, CA, USA
[5] Georgia Institute of Technology, Atlanta, GA, USA
[6] Oak Ridge National Laboratory, Knoxville, TN, USA
[7] NVIDIA

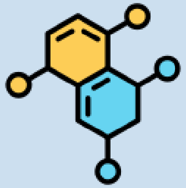
PIConGPU is funded by the Plasma-PEPSC EuroHPC Center of Excellence by the European Union through Grant Agreement No. 101093261.

Thank you very much HPE Cray/AMD Center of Excellence (COE) for your tremendous hardware/software support!

This research partially used resources of the Oak Ridge Leadership Computing Facility (OLCF) at the Oak Ridge National Laboratory, which is supported by the Office of Science of the U.S. Department of Energy under Contract No. DE-AC05-00OR22725.

*This work was partly funded by the **C**enter for **A**dvanced **S**ystems **U**nderstanding (CASUS) which is financed by the German Federal Ministry of Education and Research (BMBF) and by the Saxon Ministry for Science, Art, and Tourism (SMWK) with tax funds on the basis of the budget approved by the Saxon State Parliament.*
We would like to acknowledge the Gauss Centre for Supercomputing e.V. (www.gauss-centre.eu) for funding this project by providing computing time through the John von Neumann Institute for Computing (NIC) on the GCS Supercomputer JUWELS at Jülich Supercomputing Centre (JSC).

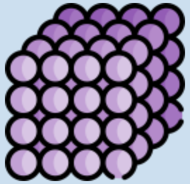
PIConGPU applicability



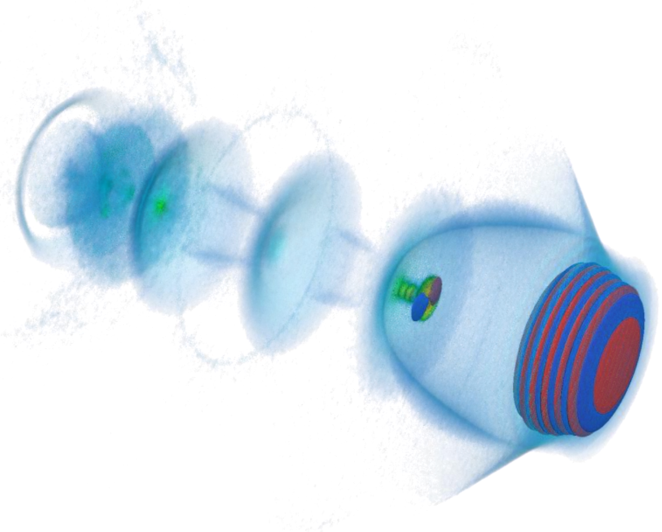
Compact table top X-Ray sources of high brightness, e.g. Free-Electron Lasers to create snapshots of ultrafast processes in material science

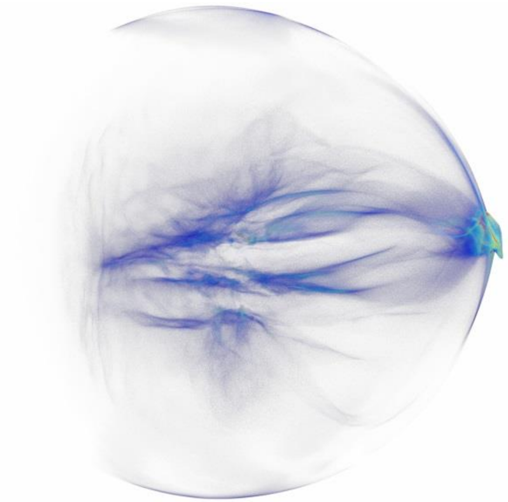
Extend plasma-based electron accelerators from multi-GeV towards TeV electron energies

Applications in radiation therapy of cancer.



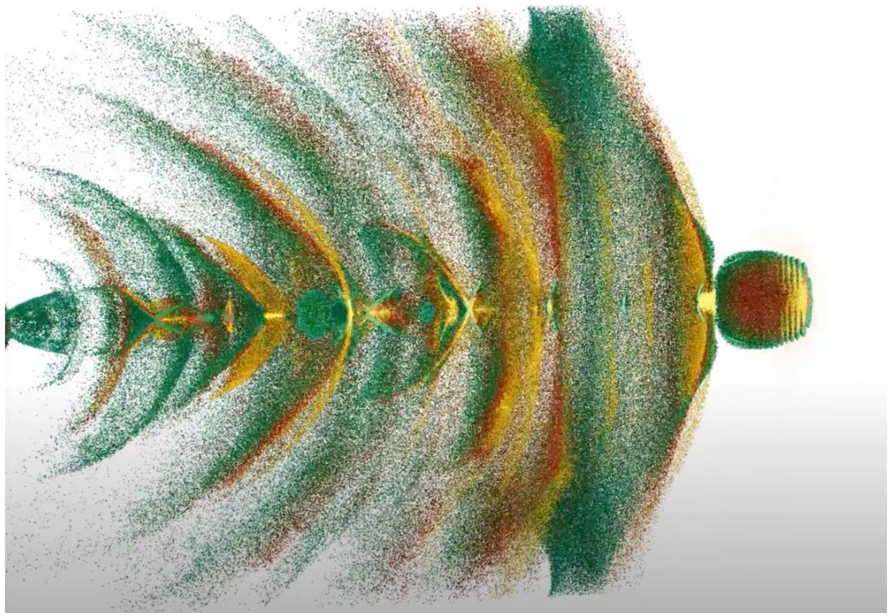
Fundamental studies of warm-dense matter and high energy density physics





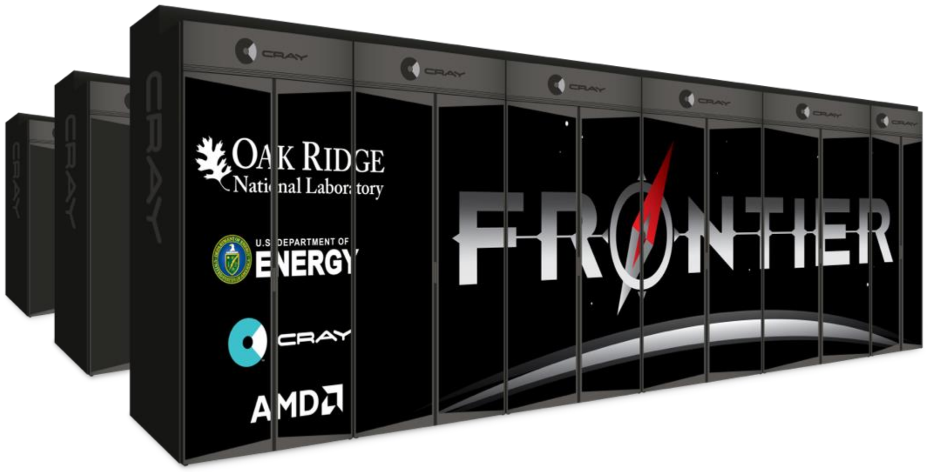
ORNL's Center for Accelerated Application Readiness (CAAR)

- To stress test Frontier's hardware & software stack



~>= 4 x

Vs Summit at ORNL



ACK: Felix Meyer (NVIDIA, former HZDR), Richard Pausch, HZDR
Still image from an uncut LWFA simulation video using OLCF Summit and 48 NVIDIA V100s using ISAAC 1.5 (in-situ library)

OLCF Frontier's AMD EPYC CPU + AMD Radeon Instinct MI250x GPU

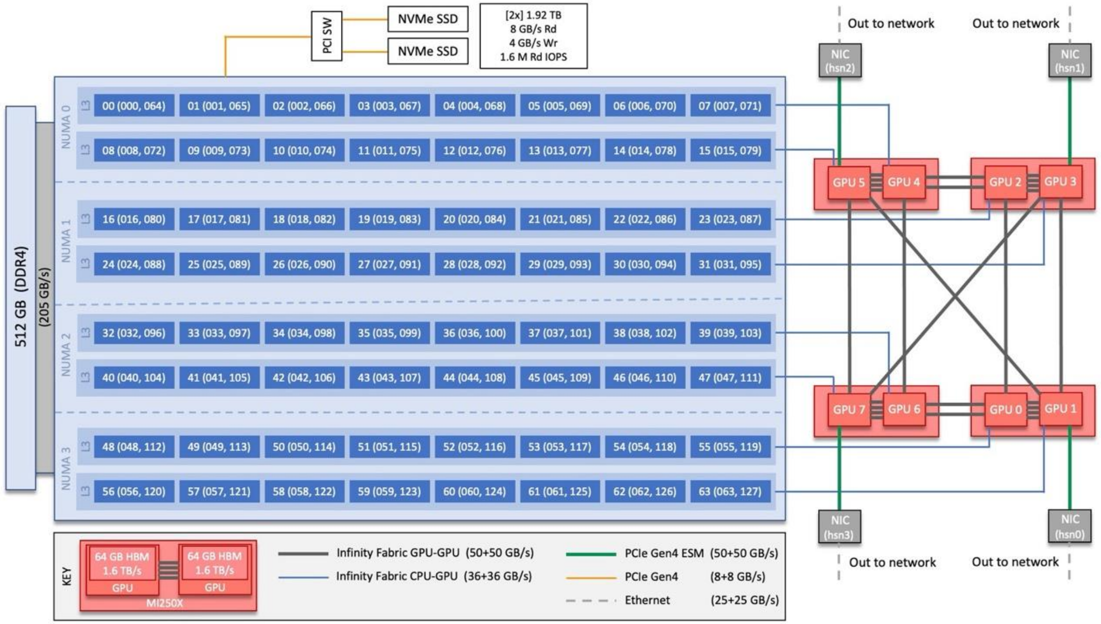
$$\text{FOM} = \frac{(\,90\%\times\text{particle updates} + 10\%\times\text{cell updates}\,)}{\text{second}}$$

PIConGPU targets….

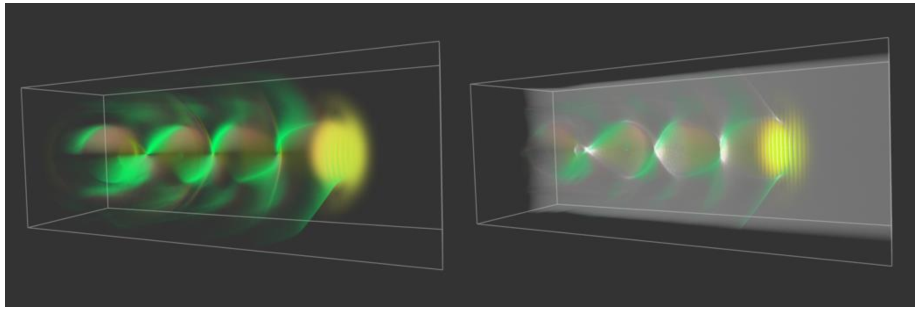
- Since ORNL’s TITAN supercomputer NVIDIA’s K20s GPUs
- ORNL’s Summit supercomputer NVIDIA’s V100 GPUs
- LBNL’s Perlmutter NVIDIA A100 GPUs
- JSC’s JEWELS Booster NVIDIA A100 GPUs
- NVIDIA’s H100 GPUs
- Julich’s H200 GPUs
- Frontier’s AMD MI250x GPUs
- Ampere computing Altra Q80 6—bit CPUs (based on Arm Neoverse N1)

ORNL’s Frontier supercomputer

- **Compute Node** 1 64-core AMD “Optimized 3rd Gen EPYC” CPU 4 AMD Instinct MI250X GPUs = 606,208 cores
- **GPU Architecture** AMD Instinct MI250X GPUs, each feature 2 Graphics Compute Dies (GCDs) for a total of 8 GCDs per node = 37,888 Instinct GPUs
- **System Interconnect** 4-port HPE Slingshot 200 Gbps (25 GB/s) NICs providing a node-injection bandwidth of 800 Gbps (100 GB/s)
- **Storage** 700 PB HDD+11 PB Flash Performance Tier, 9.4 TB/s and 10 PB Metadata Flash Lustre
- **System Size** ~9400nodes
- **Ranking** No. 1 in the Top500 as of June 2024



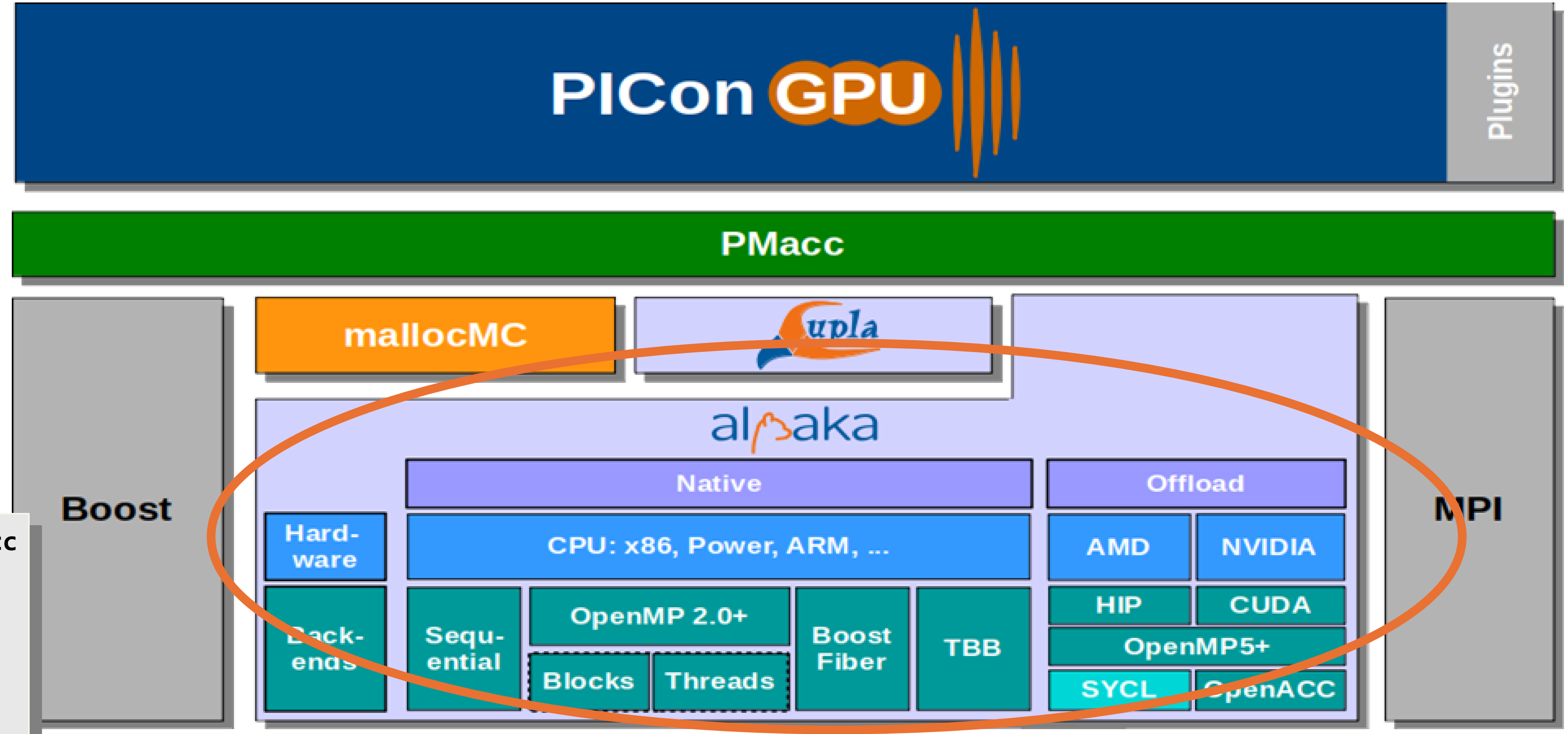
PIConGPU Exascale challenges



ACK: Benjamin Hernandez, ORNL
LWFA Simulation. Using Summit's 8 nodes (48 V100 GPUs) with ~2 billion particles using ISAAC v1.5.1 running on OLCF's cloud environment (SLATE)

- **Portability:** Run code on different compute architectures (single-source, run everywhere)
- **Performance:** Cannot lose performance while maintaining portability
- **Scalability:** Code profiling & scaling tests to ensure science cases scale to Frontier
- **Visualizations:** Create and develop tools to visualize PIConGPU on the new system
- **Exascale workflows:** Extend I/O capabilities, provide in-situ analysis, data reduction and visualization workflows

alpaka software stack

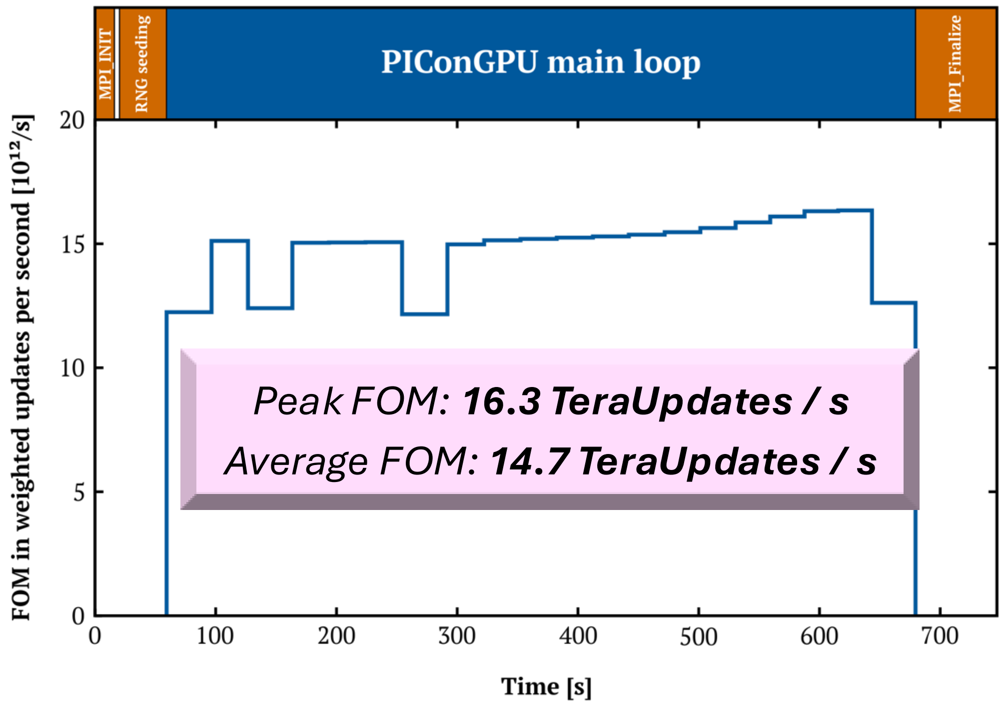



```
    template< typename T_Acc
>
    ALPAKA_FN_ACC void
operator()(
        T_Acc const & acc,
        // ...
    ) const
    {
        // ...
    }
```

Huebl, Axel, et al. (2018) Zero Overhead Modern C++ for Mapping to Any Programming Model.
Software Stack updated by René Widera (2020)

Kelling, J., Bastrakov, S., Debus, A., Kluge, T., Leinhauser, M., Pausch, R., ... & Juckeland, G. (2022, May). Challenges porting a C++ template-metaprogramming abstraction layer to directive-based offloading. In Accelerator Programming Using Directives: 8th International Workshop, WACCPD 2021, Virtual Event, November 14, 2021, Proceedings (pp. 92-111). Cham: Springer International Publishing.

FOM baseline run on OLCF Summit (2019) TWEAC case study (Single Precision)



*Peak FOM: **16.3 TeraUpdates / s***

*Average FOM: **14.7 TeraUpdates / s***

*Peak power: **8MW***
*Sustained power: **5.8MW***

№ timesteps	1000
№ NVIDIA V100 GPUs	27600 (4600 nodes)
№ cells total	404 billion
№ cells per GPU	14.6 million
№ particles total	10.1 trillion
№ particles per GPU	365 million
№ simulation data	324 TB

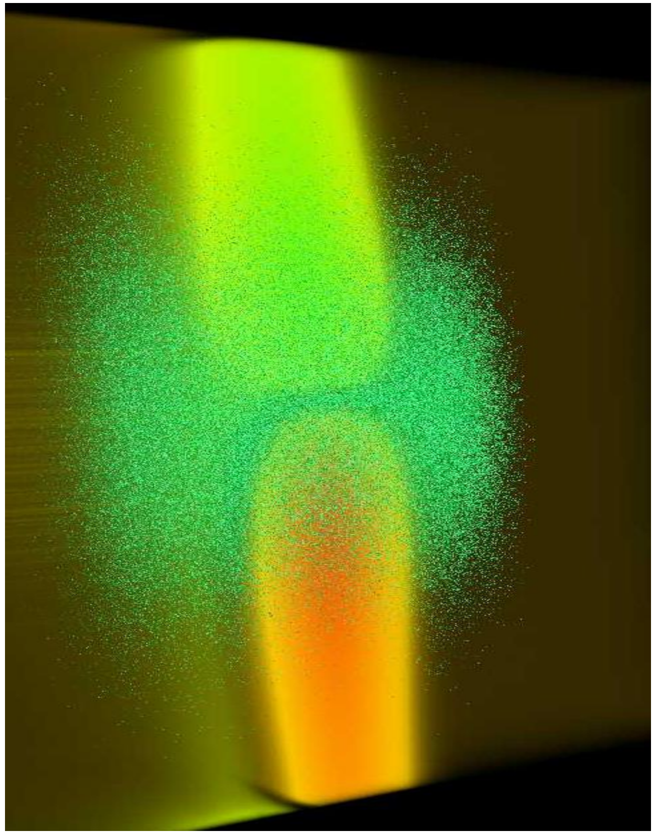
Particle Data	313.36 TB
Cell Data	14.52 TB
Particles Processed	16.2 Trillion particles/ sec
Cells Processed	656 Billion cells/sec
GPU Kernel Calls	9 Million kernels/sec

$$\text{FOM} = \frac{(\,90\%\ \text{x particle updates} + 10\%\ \text{x cell updates}\,)}{\text{second}}$$

PIConGPU

Major Improvements to PIConGPU since Summit run (2019)

- Algorithmic improvements
  - Optimized laser functor [TWTSfast]
  - New field background algorithm [SuperPusher]
  - New laser algorithm [IncidentField]– 2 years' worth work
- Performance optimizations
  - GPU-aware MPI
  - Optimized particle assignment
  - Enhance device utilization



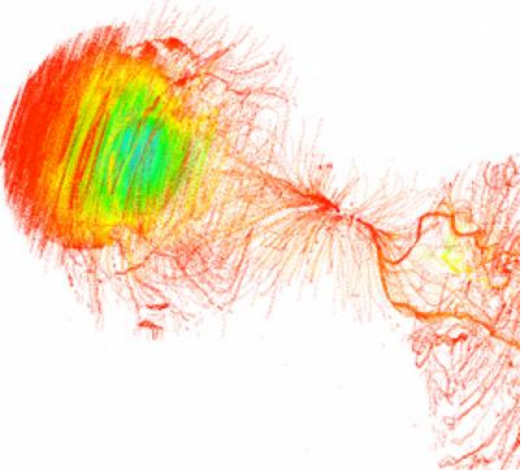
No. of commits: Autumn 2019: cupla 136, Alpaka 1057, PIConGPU 1278, mallocMC 93

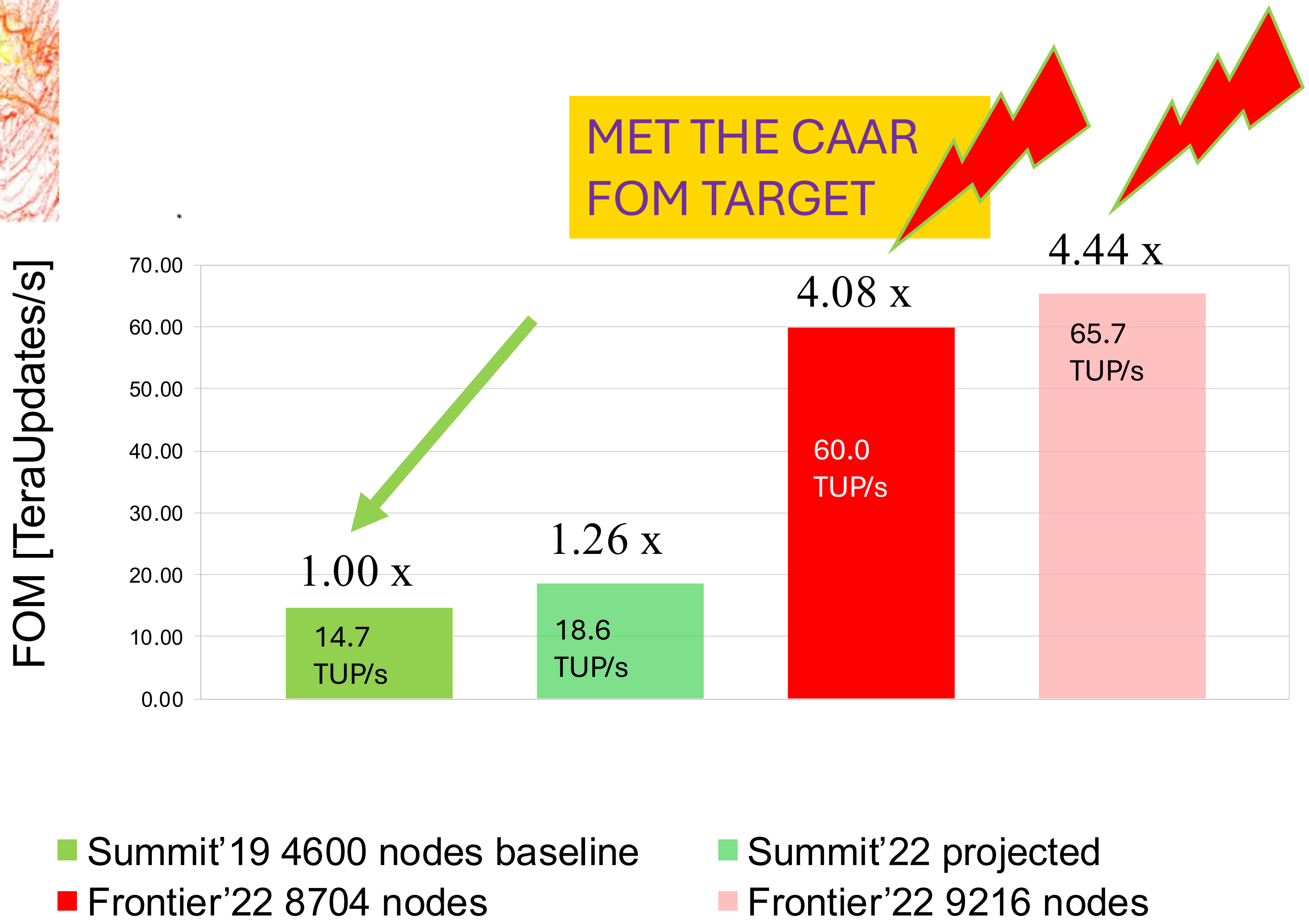
Red Queen's race – staying in the same place is falling behind

Leinhauser, M., Widera, R., Bastrakov, S., Debus, A., Bussmann, M. and Chandrasekaran, S., 2022. Metrics and design of an instruction roofline model for AMD GPUs. ACM Transactions on Parallel Computing, 9(1), pp.1-14.

Hardware is only as good as its software and tools and a close COMMUNICATION between developers & users!

Exascale FOM runs for TWEAC case study

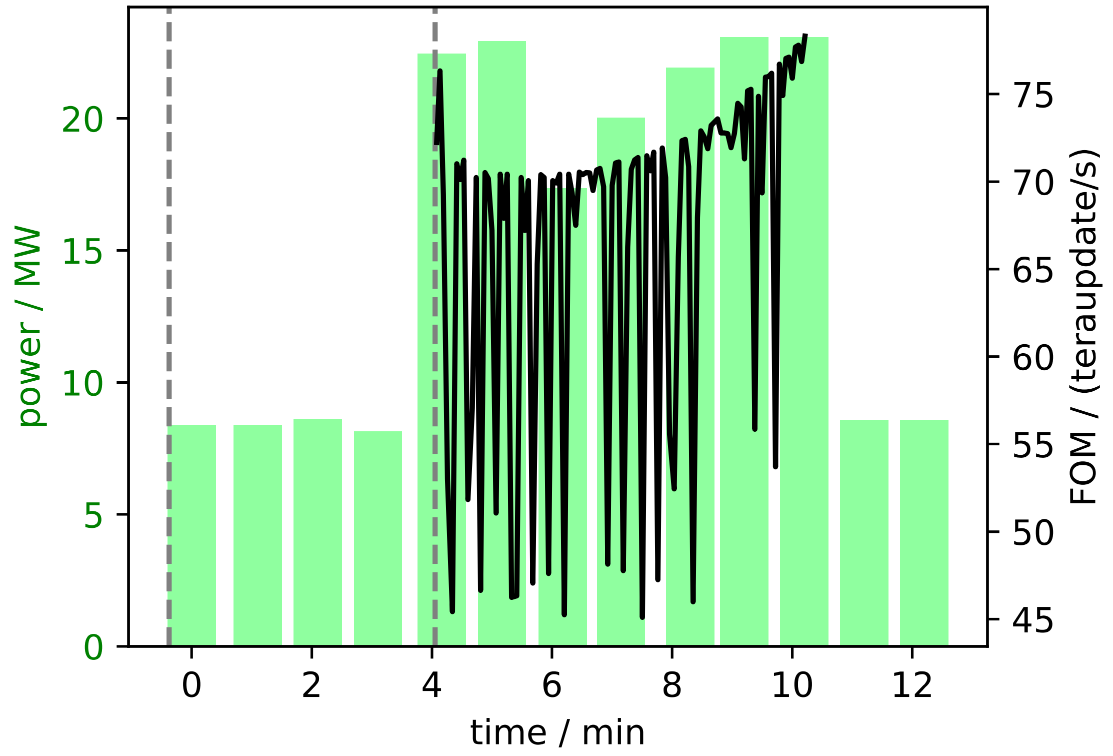




Note, these results use single precision.

8.5x for double precision

FOM run on Frontier (2023) TWEAC case study (Single Precision)



№ timesteps	1000
№ AMD GCDs	73,728 (9216 nodes)
№ cells total	1.1 Trillion
№ cells per GCD	14.6 Million
№ particles total	27 trillion
№ particles per GCD	365 million

Particle Data	760.7 TB	41% more
Cell Data	35.3 TB	41% more
Particles Processed	72 Trillion/sec	22.5% more
Cells Processed	2.6 Trillion/sec	25% more
GPU Kernel Calls	24 Million kernels/sec	37% more

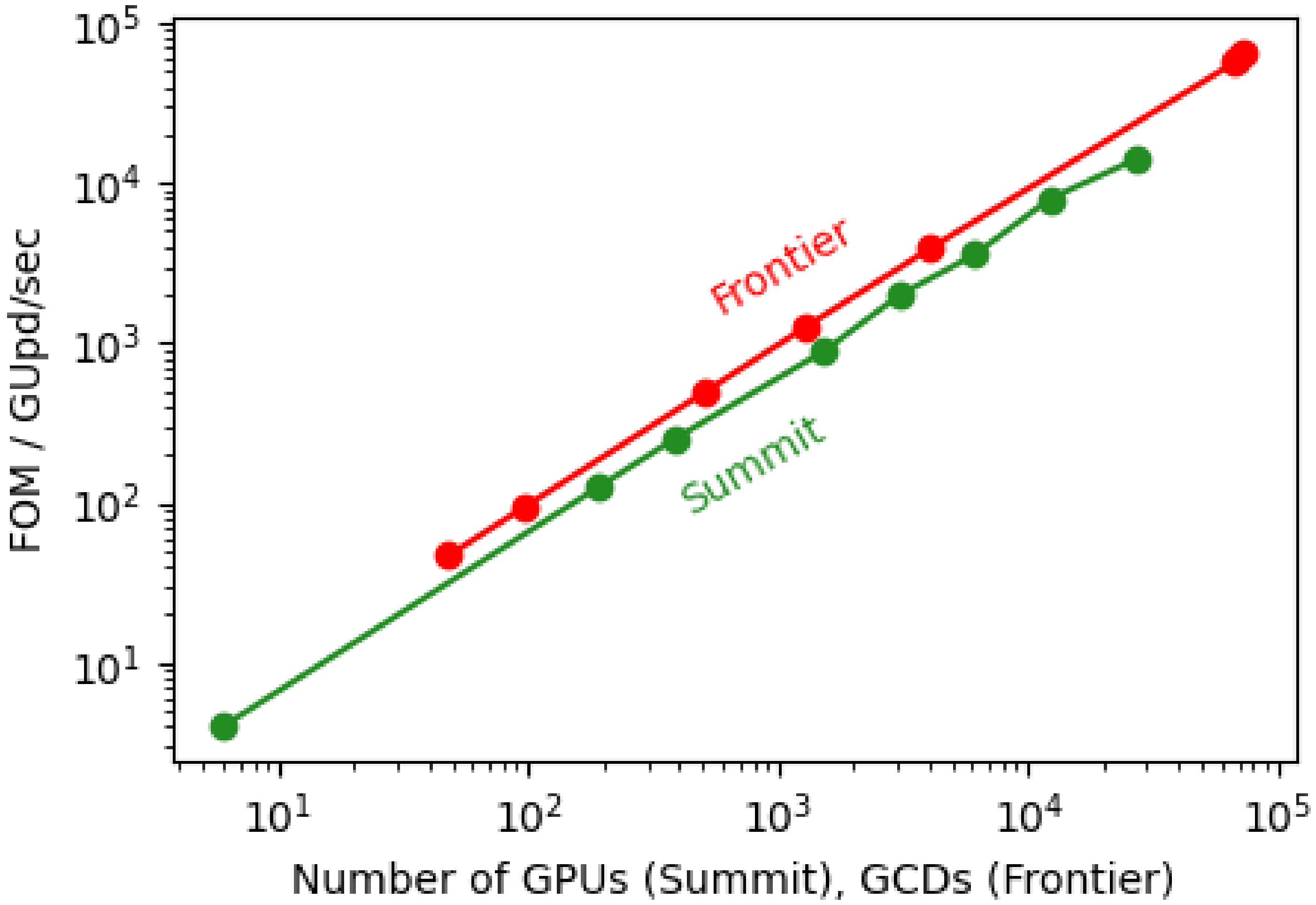
PIConGPU on 98% of Frontier
Peak Power: **23 MW**
Average Power: **21.5 MW**

*Peak FOM: **78.3 TeraUpdates / s***
*Average FOM: **65.7 TeraUpdates / s***

$$\text{FOM} = \frac{(\,90\%\times\text{particle updates} + 10\%\times\text{cell updates}\,)}{\text{second}}$$

Weak Scaling FOM case on Frontier

- № Iterations: 1000
- Runtime: ~10 mins
  ~ 0.37 secs per iteration
- FOM Science case
- Scaling:
  - 6 nodes → 9,216 nodes
  - 48 GCDs → 73,728 GCDs
  - 24 GPUs → 36,864 GPUs
  - 96-98% GPU utilization

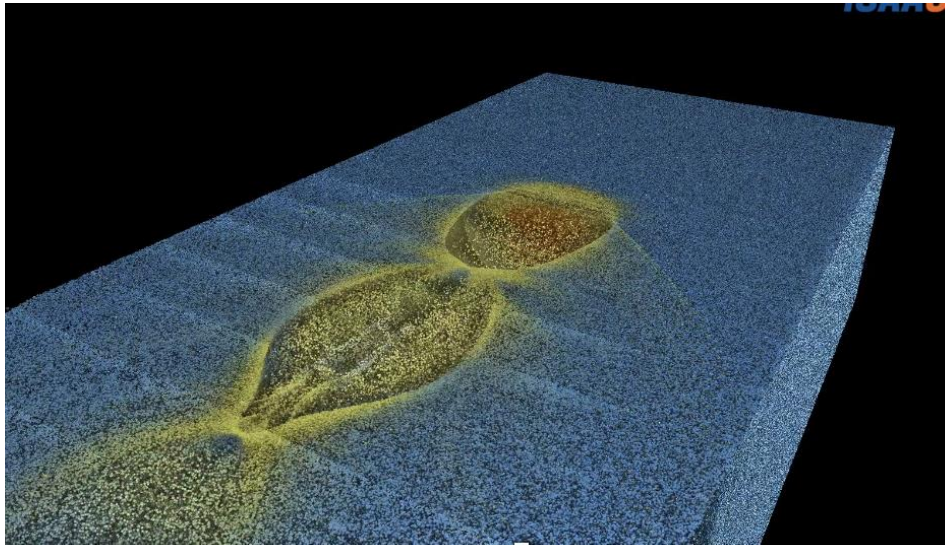


There is more to do....

- With Frontier we can ONLY perform a few long-running simulations where we will hopefully observe acceleration of electrons to highest 100GeV scale energies, BUT the parameter space of laser electron acceleration is HUGE; we are still just able to catch a TINY bit of it.

- Need to explore LARGER parameter range, need a **sophisticated WORKFLOW** to further advance science

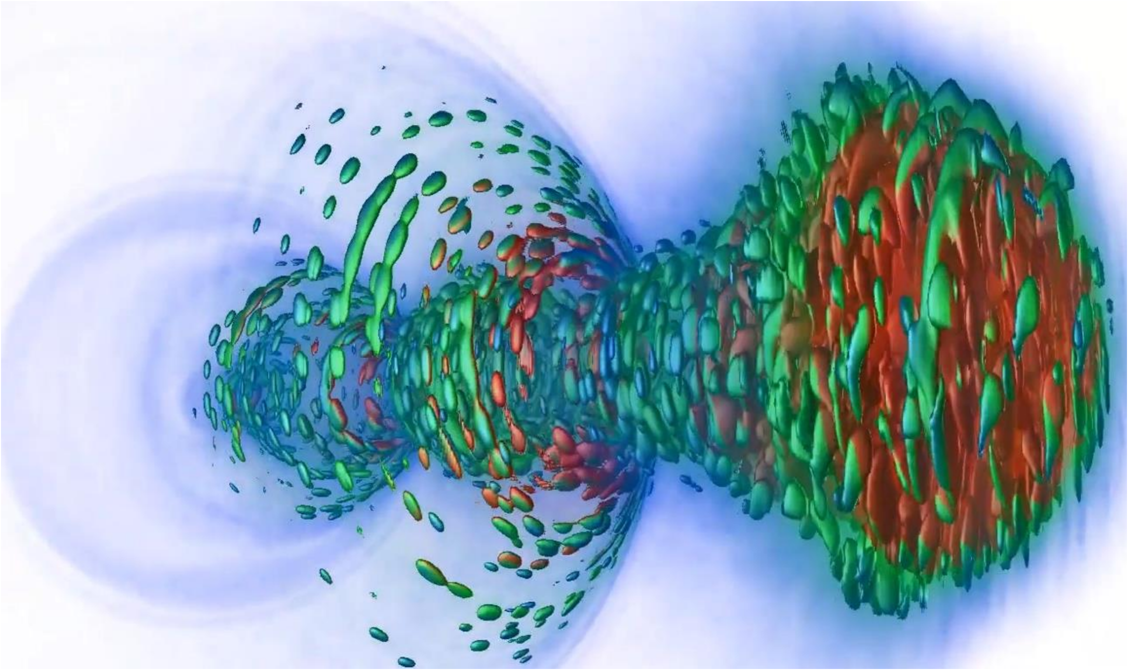
- Need to close gap between simulation scenarios on supercomputers and the experimental setup in the labs



ACK: Vincent Gerber, HZDR, Germany
Using In-Situ viz library for Animation of Accelerated Computations (ISAAC)

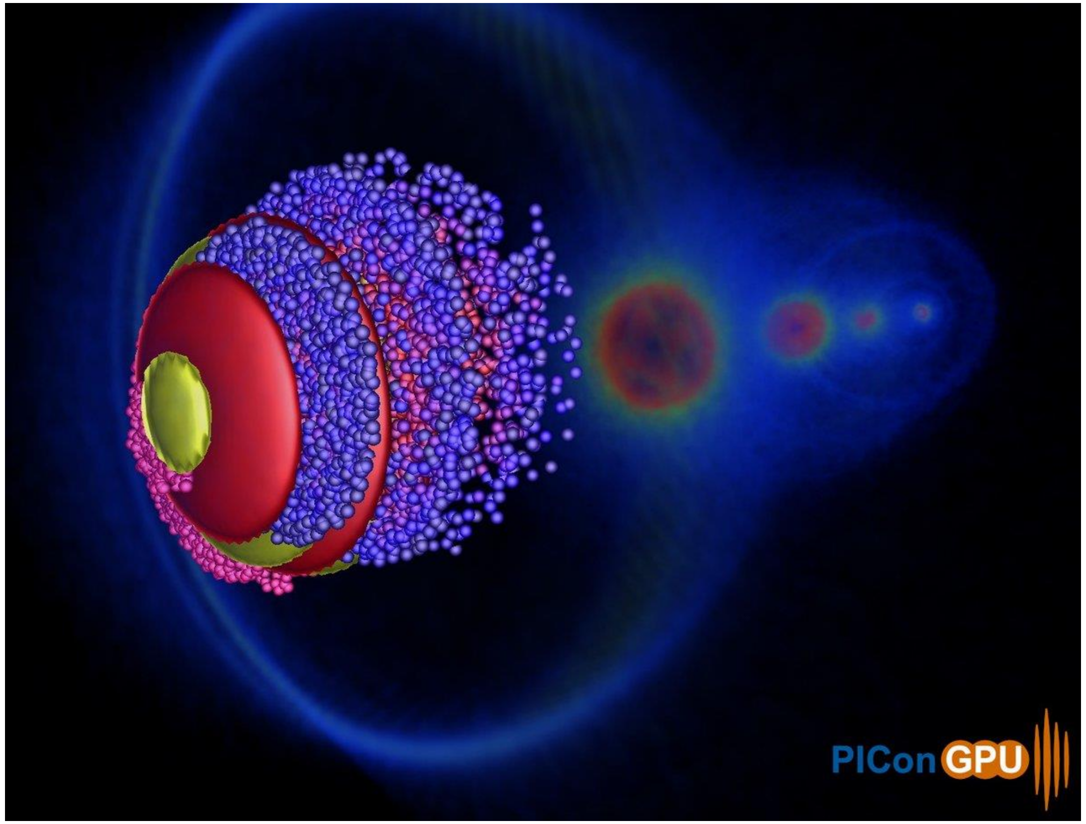
PIConGPU data volume

- Forward calculation incurs a heavy computational cost prohibitively expensive for a wide range of simulation parameters.
- Simulations of plasma behavior, involve solving complex nonlinear equations for trillions of particles creating **TBs** of data
- Due to the scale of the data, we cannot save all the raw data to disk
- We need a different solution!

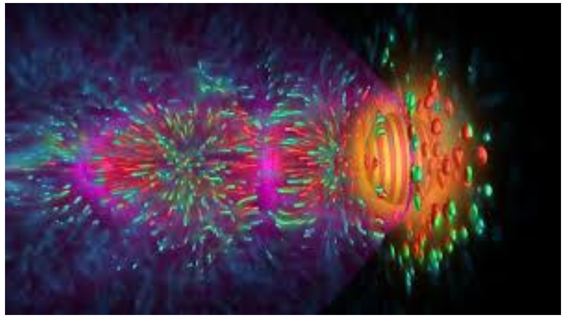


What do we want to learn?

- Extracting meaningful info out of complex heterogeneous data being generated from source or simulation at an exponential rate
- Need a complete reconstruction of phase space from observations to get a detailed view of the growth and dynamics of instabilities
- Need for automatic detection of correlations between plasma dynamics and emitted radiation spectrum
  - Otherwise this requires extensive post-processing and analysis spanning years

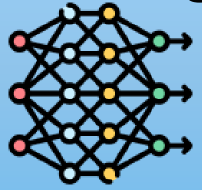


Extracting knowledge from large scale simulations is a challenge!!



Particle data streaming

Learning



Learning Outcomes

Jeffrey Kelling.....Michael Bussmann, Sunita Chandrasekaran, “The Artificial Scientist - Leveraging In-transit Machine Learning for Plasma Simulations” Accepted to 39th IEEE International Parallel & Distributed Processing Symposium (IPDPS) 2025, Best Paper Nomination

Case Study: Kelvin-Helmholtz instability (KHI) in PIConGPU

- Well known shear surface instability observed in fluids and plasma
- When 2 layers exhibit different velocity/density
- Viz using ISAAC

Richard Pausch, HZDR, Germany; Uses 4 V100 GPUs

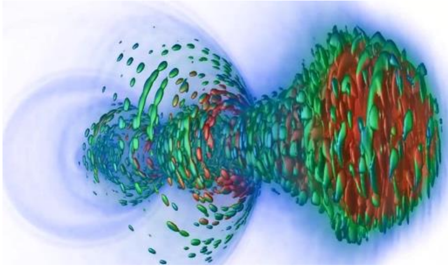
In plasmas, the KHI is driven by a self-amplifying cycle of small density or velocity fluctuations that lead to a growing magnetic field at the shear surface, which further amplifies the initial fluctuations as depicted

openPMD and ADIOS2

- openPMD
  - Data standard for **p**article **m**esh **d**ata
- ADIOS2
  - I/O library handling the data transfer
  - SST (Staging Transport Engine) is a backend of ADIOS2 that supports in-memory data streaming.
    - Multiple transport methods - TCP, libfabric, MPI_Open_port()
- PIConGPU sends openPMD-formatted particle data directly from memory to another process in memory using ADIOS2 with the SST engine.
  - This **bypasses the slow file system** entirely.

Poeschel, Franz, et al. "Transitioning from file-based HPC workflows to streaming data pipelines with openPMD and ADIOS2." *Smoky Mountains Computational Sciences and Engineering Conference*. Cham: Springer International Publishing, 2021.

Huebl, A., Lehe, R., Vay, J. L., Grote, D. P., Sbalzarini, I., Kuschel, S., ... & Bussmann, M. (2015). openPMD: A meta data standard for particle and mesh based data. *URL https://doi. org/10.5281/zenodo, 591699*.

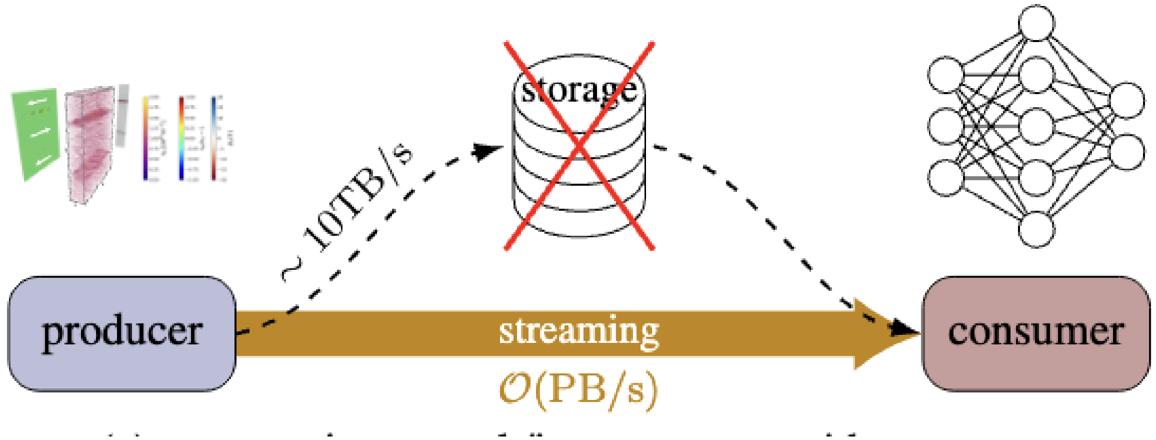


Challenges - System Constraints

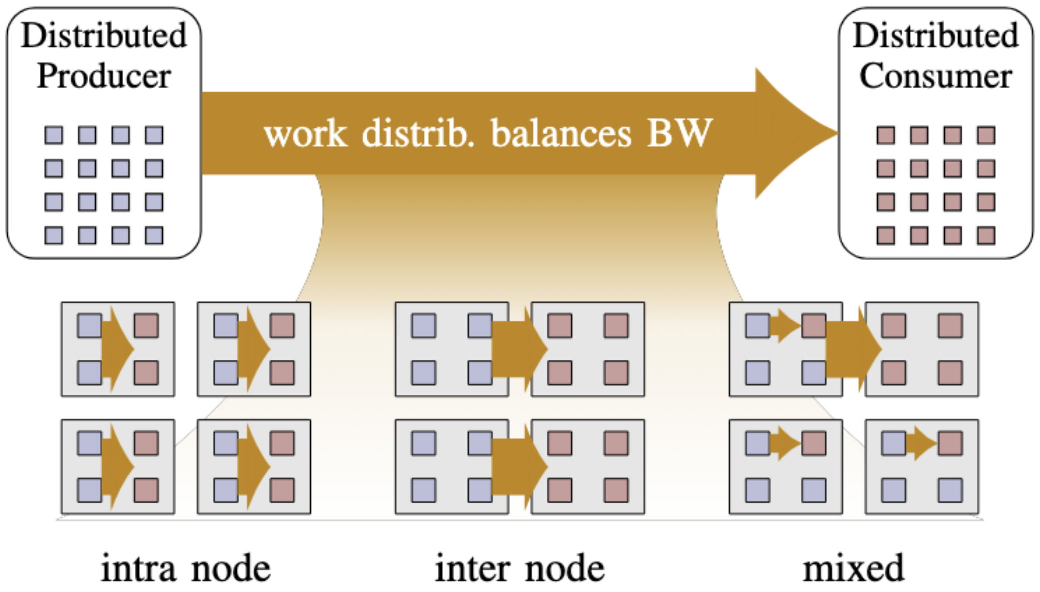
- PIConGPU simulation produces 5.86GB of particle data per node per time step
  - Simulations of plasma behavior, involve solving complex nonlinear equations for trillions of particles creating **TBs** of data
- Application scaling to just 25% of Frontier system would produce 1 PB of data every time step
  - BUT Frontier's Orion file system can write at a max of 10 TB/s
- 1 PB per time step, 1000 time steps = 1 EB of total data
  - Even if each time step only takes 0.1 to 1 second, 1 to 10 PB/s of data is being produced
- That means in 100 to 1,000 seconds, the total data volume would reach 10 EB.
- **Implication:** Entire file system can be exhausted in 100 to 1000 seconds

We need to circumvent the file system!!!!!!

Three in-transit workflow approaches on Frontier



Streaming without going through storage unlocks more bandwidth



Reducing simulation data close to the producer lowers bandwidth requirements

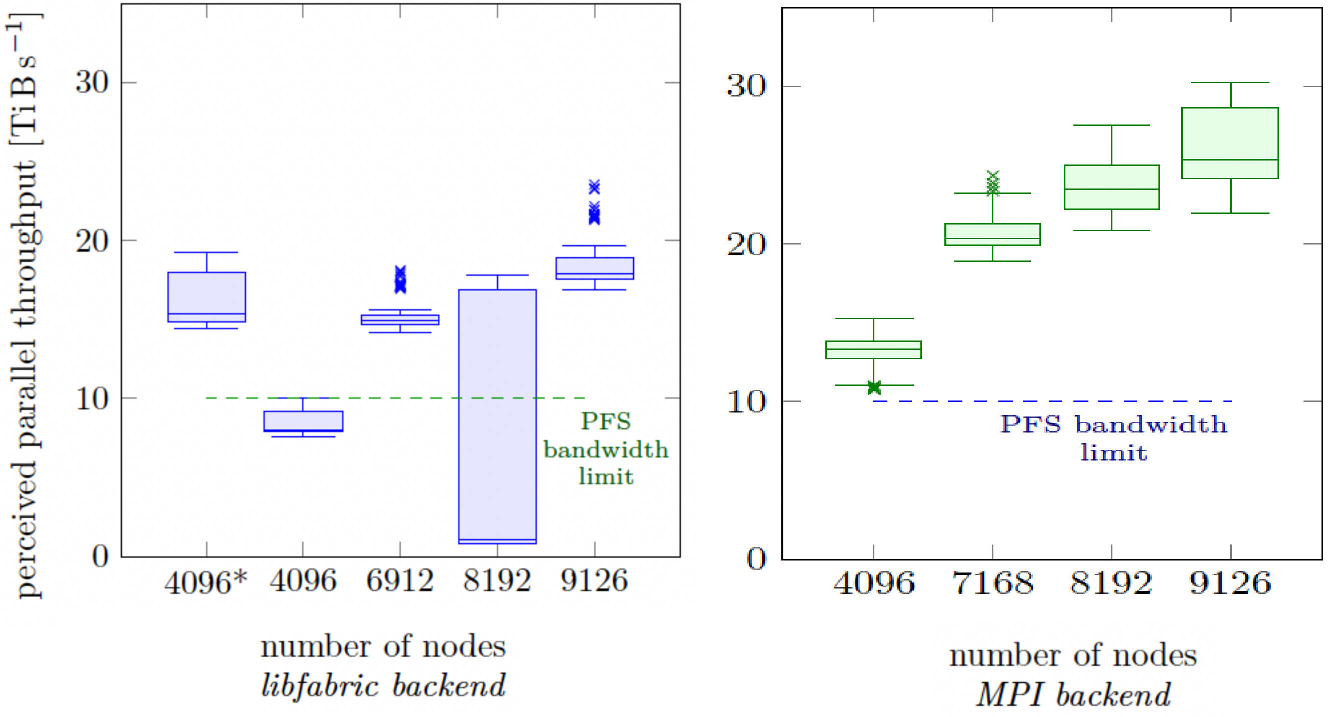
Distributed producer and consumer, system topology presents communication paths with vastly different bandwidths which must be reconciled with the loosely-coupled application's communication requirements.

Data streaming on 9126 Frontier nodes

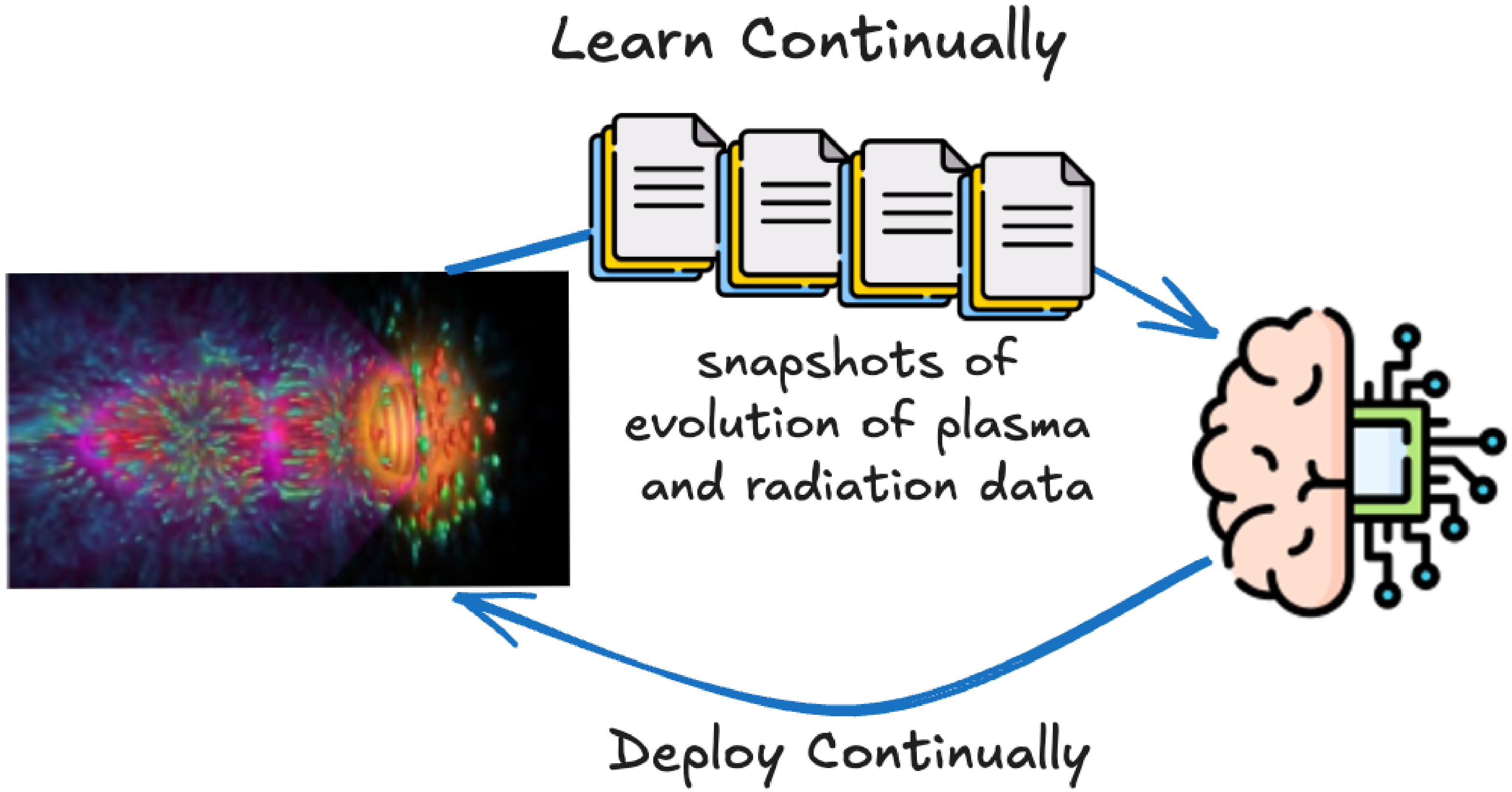
- The amount of particle data produced by PIConGPU KHI is 5.86 GB per compute node and time step.

- Libfabric per-node achieved per-node throughput of 3.5 ~ 4.7 GB/s but 1.9 - 2.6 GB/s at 9126 compute nodes (a total of 16.5 - 23.0 TB/s).

- Conversely, the MPI data plane yields a per-node throughput from 2.6 - 3.7 GB/s at 4096 compute nodes (a total of 10.5 - 14.9 TB/s) to 2.4 - 3.3 GB/s at 9126 nodes (a total of 21.4 - 29.5 TB/s

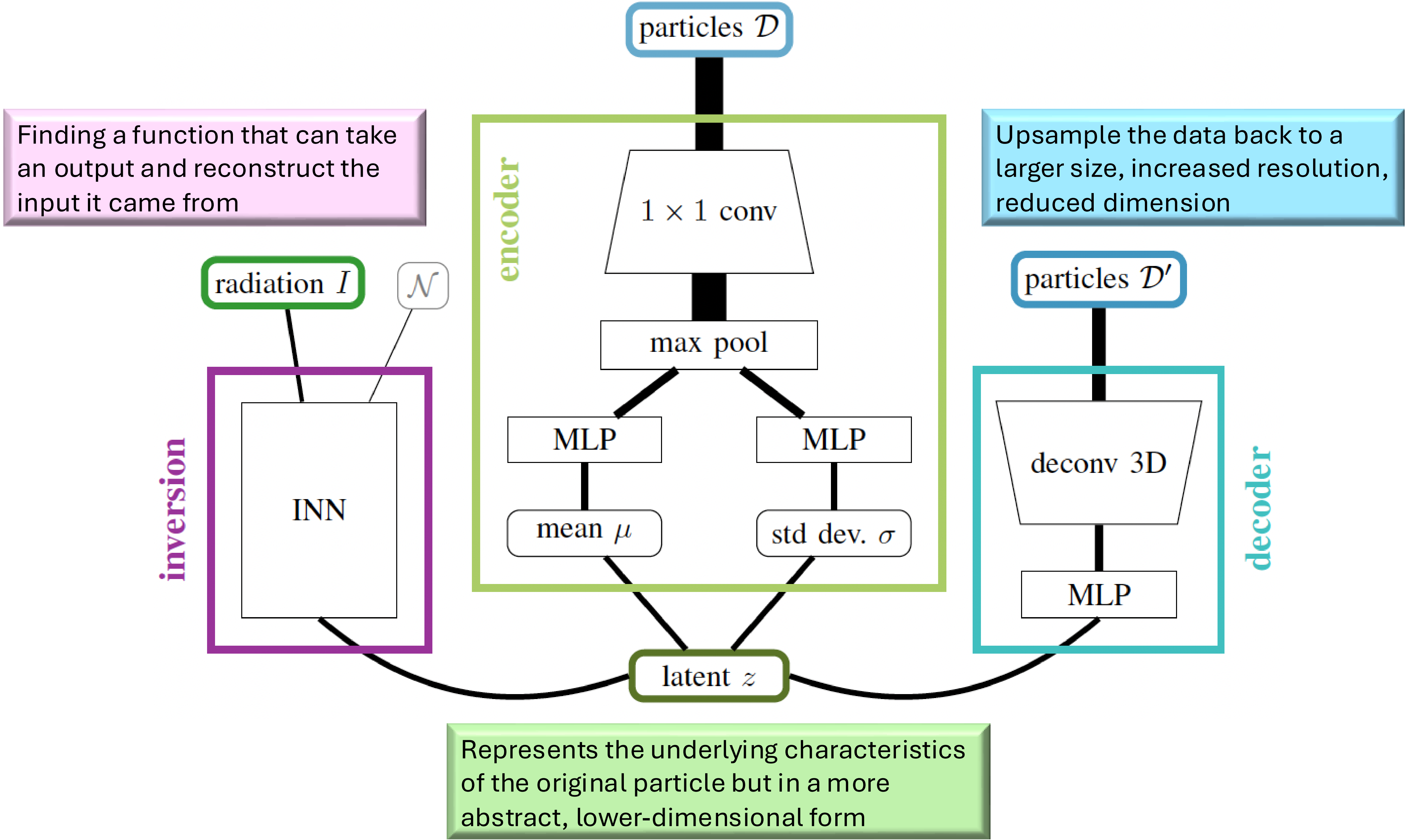


In-transit continual learning



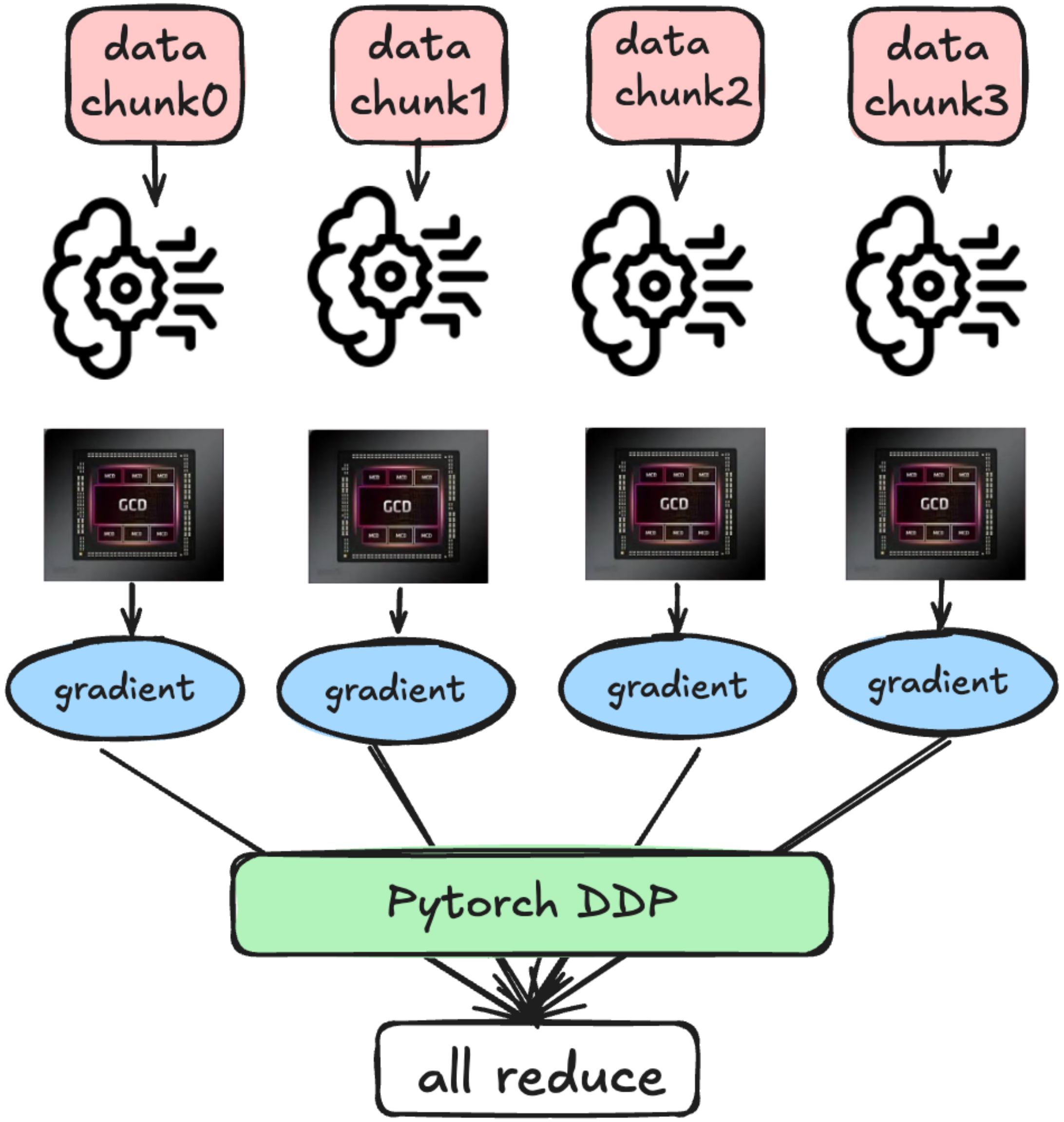
- Model is continuously trained online from subsequent snapshots of the evolution of plasma and radiation data without storing every data point to disk
- Continuous learning circumvents lack of adequate disk capacity and bandwidth by enabling data to reside and distribute in-memory via network interconnects.

ML model architecture



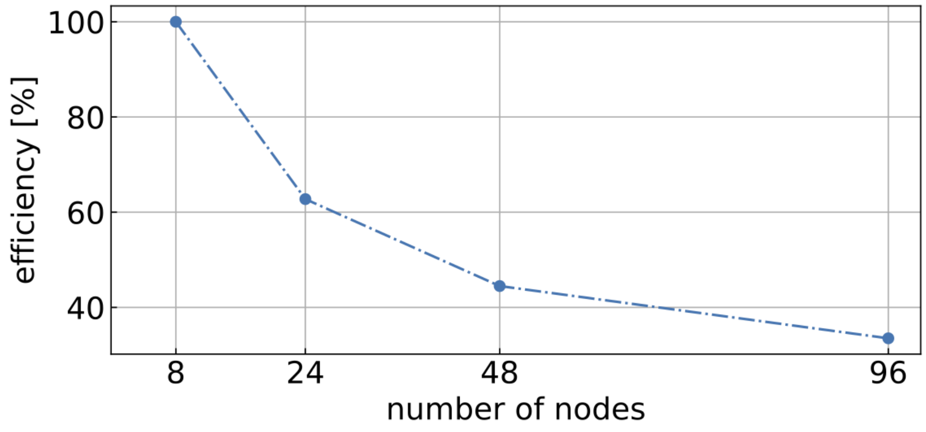
Data parallelism and scaling on Frontier

- ML model fits into 1 GCD
- Each copy of the model receives different chunks of data to train on
- Asynchronously train the model with that data
- Scaling depends on the optimization of all-to-all communication in PyTorch DDP
  - using N/RCCL backend – hits a wall after 100 Frontier nodes = 400 AMD GPUs
- Need to explore libfrabic backend for N/RCCL or PyTorch DDP's MPI backend



Weak scaling and observations

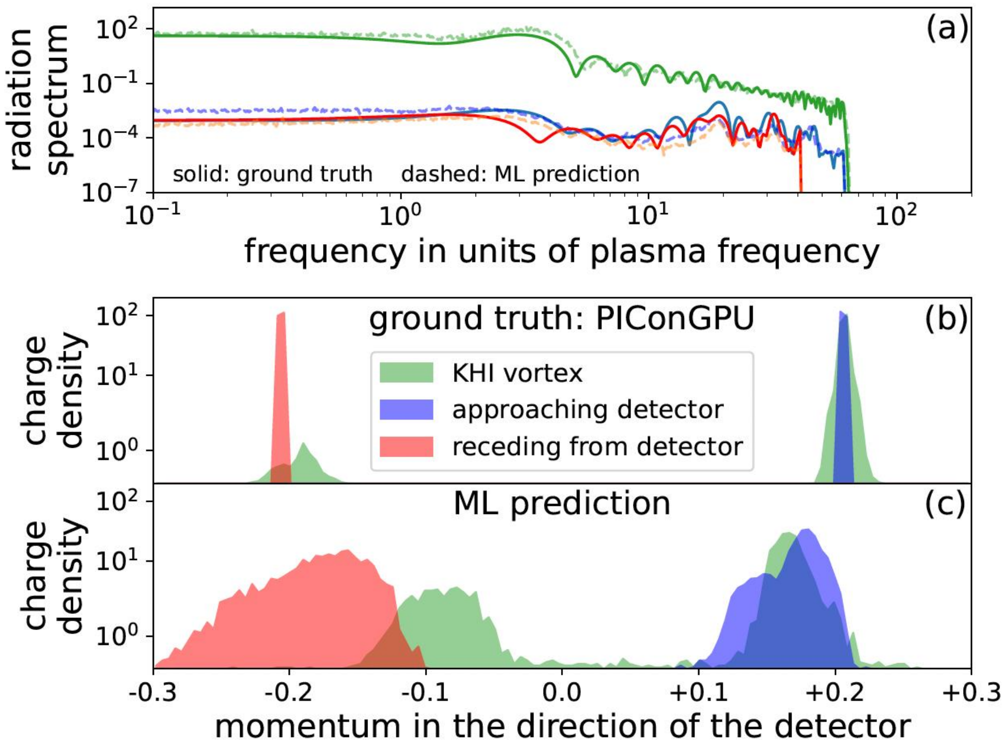
- Increasing time spent in PyTorch DDP for larger runs
- Low Efficiency - Inevitable all-to-all communication between PyTorch ranks taking place to average gradients during each backward pass – deficit of 30%
- Low Efficiency - Lack of availability of PyTorch distributed primitive for matrix dot product to evaluate INN
- In-transit training at very large batch size
  - Hyperparameter needs to happen at scale; doesn't transfer well from small scale experiments
- Loss functions to compare point cloud – CD vs EMD
- Comprehensive studies between batch sizes, block learning rate and weights need to be studied at scale with streamed simulation

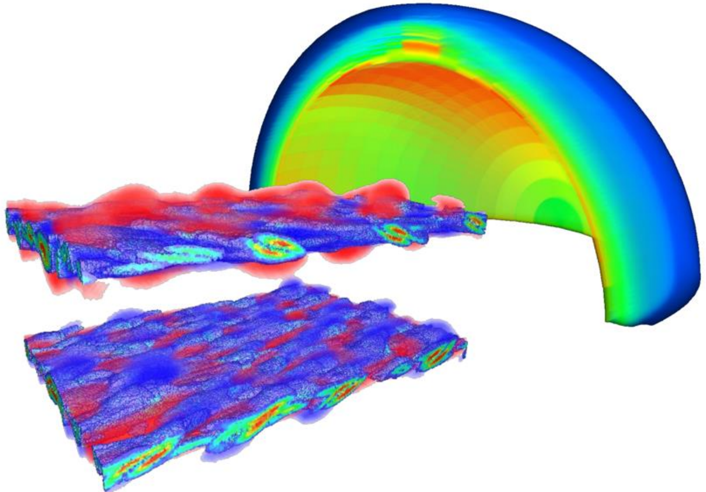


- In-transit training from 8 to 96 nodes (32 to 384 GCDs)
- Reaches around 35% at 96 nodes

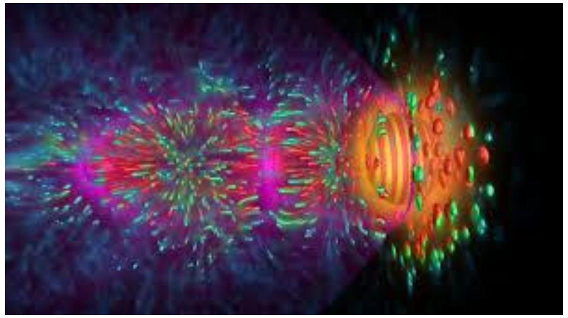
Take aways

- The model clearly learned to partition the latent space into regions for different flow directions and vortex regions
    - Both the parts of the ML system — the encoder (which compresses the data) and the inversion network (which tries to reconstruct or interpret it) — figured out how to handle those zones correctly.
- Achieved partial reconstruction of the plasma distribution, the ML prediction still clearly identifies the instability regions
- A very promising early result for using ML during a simulation (called in-transit learning) — especially in complex, changing situations (non-steady-state processes, like plasma evolving over time).



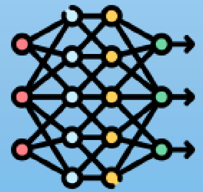


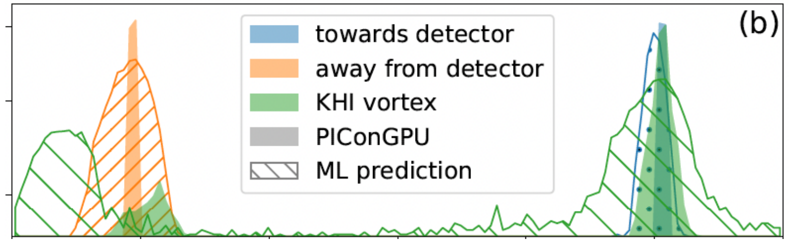
Extracting knowledge from large scale plasma simulations



Particle data streaming

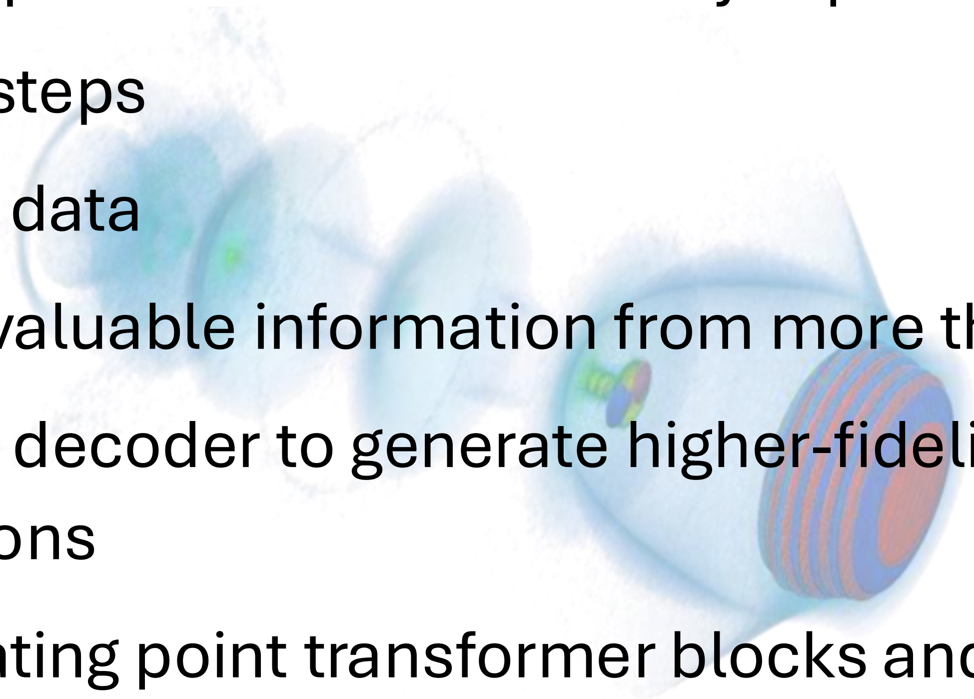
Machine Learning





So much more to do….

- Build surrogate models of simulations of different configurations – classically these experiments would be very expensive
- Collect many time steps
- Intelligently reduce data
- Learning in-transit valuable information from more than just one simulation
- More sophisticated decoder to generate higher-fidelity depictions of particle configurations
- Encoders incorporating point transformer blocks and a deeper network around the bottleneck to better extract latent information from point-features



Need for WORKFLOWs to tackle complex, data-intensive real-world applications!