Macroheterogeneity: Enabling Hybrid HPC and AI Workflows

Samantika Sury

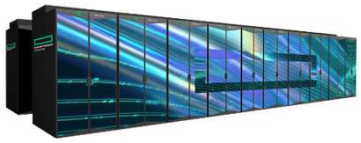
Pete Mendygral, Matt Turner, Robert Wisniewski
HPC and AI Infrastructure Solutions, HPE

8/14/2026

Agenda

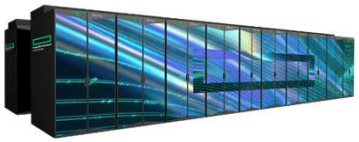
- Macroheterogeneous systems and why we need them
- Real world use-cases
- HW and SW technologies to optimize data sharing
- Early performance trends
- Summary

AI+ HPC Workflow Vision



Today

Exascale Supercomputer

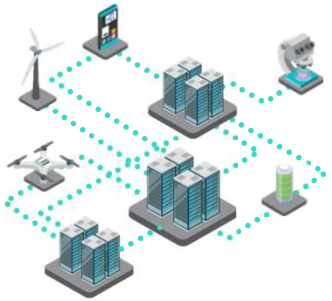


~5x Exascale

Productivity and agility for HPC and AI applications

Multi-faceted, complex workflows

For modeling, simulation, data analytics, and artificial intelligence



Diverse Scientific Systems

Integrate, automate, and optimize workflows that span multiple organizations, locations, and vendors

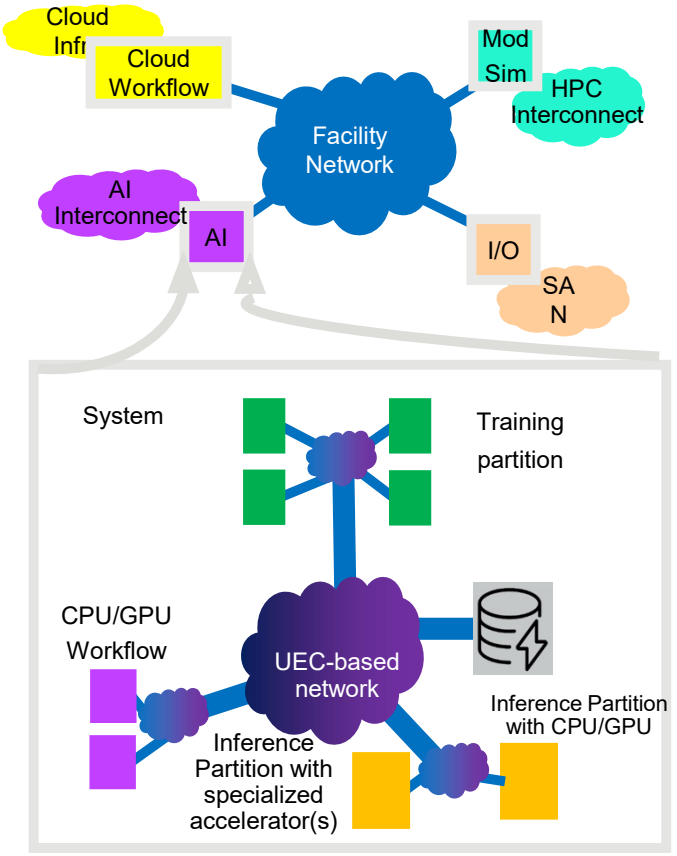
World's fastest **Supercomputer**

World's fastest **Workflows**

Macroheterogeneity: A System of Systems/Partitions

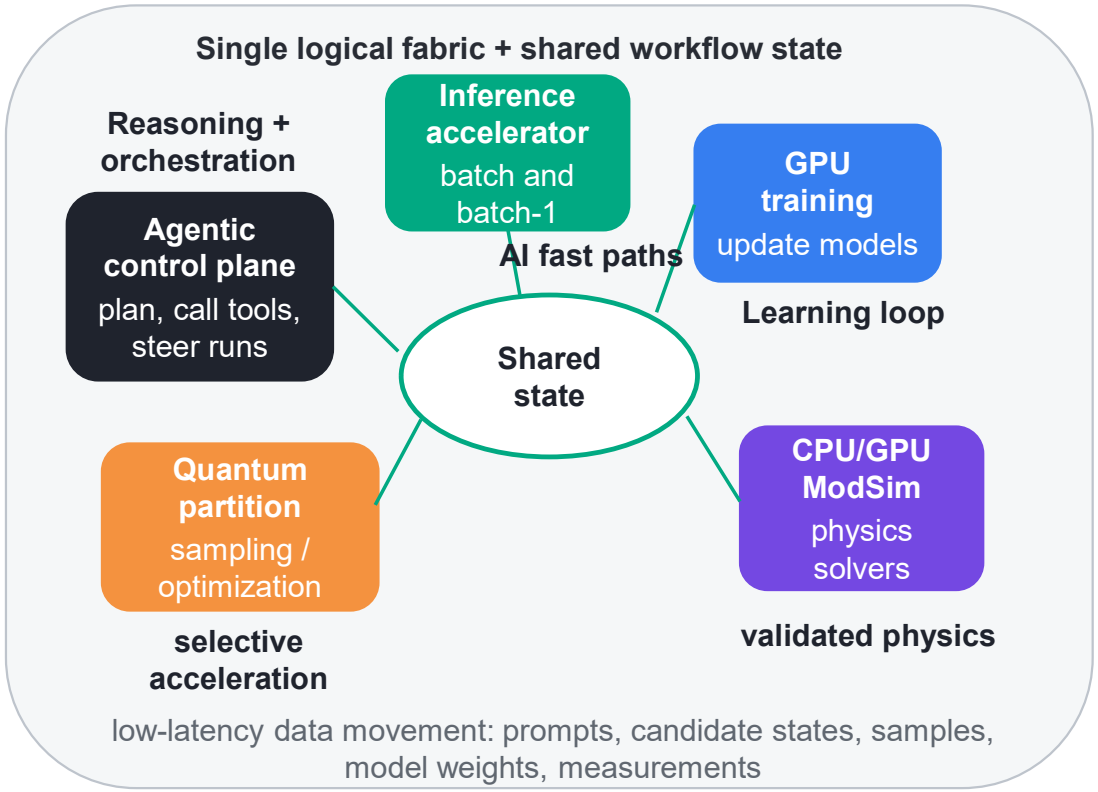
Motivation

- Optimized system architecture for AI-enabled HPC
- Integrated and broad spectrum of workflows and new use-cases
  - Batch and batch-1 inferencing from ModSim
  - Reasoning models require CPUs
  - AI steering of ensembles
  - Co-scientist with closed loop models
- Diverse and highly optimized hardware availability
  - Accelerators with 10-50X GPU perf
  - Rackscale architectures provide very high token throughput
- Path to introduce new technologies at scale
- Each phase lands on a domain-efficient substrate, while the fabric keeps the closed loop interactive



Macroheterogeneity for Agentic and Quantum-based ModSim

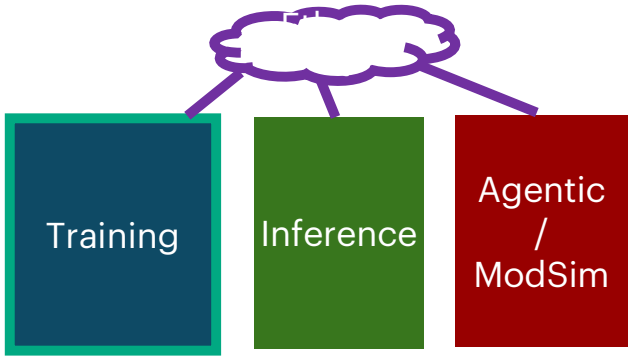
Agentic control, AI inference/training, classical ModSim and Quantum acceleration mapped to the most efficient partition over one logical fabric



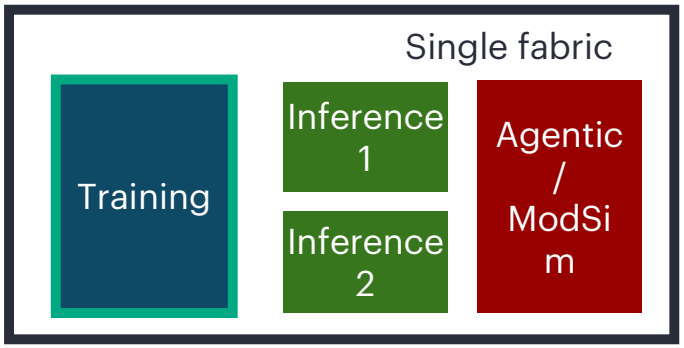
Partition mapping

Phase	Best partition	Role
Agentic reasoning	CPU + inference accelerator	keeps workflow state, chooses next experiment
Classical ModSim	CPU/GPU partition	high-fidelity physics and validation
Surrogate inference	AI accelerator partition	fast scoring, steering, batch-1 decisions
Training / adaptation	GPU training partition	fine-tune surrogate or policy models
Quantum kernel	QPU / quantum emulator	optimization, sampling, small hard subproblems

Characteristics



Loosely coupled Macroheterogeneity



Tightly coupled Macroheterogeneity

Dimension	LC macroheterogeneity	TC macroheterogeneity
Execution & Workload Mapping	Workloads mostly run end-to-end on a single compute type (CPU-only, GPU-only, inference-only)	Workloads decomposed and mapped phase-by-phase to the most efficient compute engine
Data Movement & Memory Locality	Cross-silo data movement via network or storage tiers	Locality-aware execution with tighter memory coupling
System-Level Energy Accounting	Power optimized per device, not per workflow	Power optimized across compute, memory, and interconnect
Scheduling & Control Model	Coarse-grained scheduling with limited phase awareness	Phase-aware scheduling with dependency and readiness awareness
Bottleneck & Scaling Behavior	Performance limited by slowest silo or hand-off boundary	Bottlenecks may be mitigated by redistributing work across compute types

Example Use Case: Climate Modeling

- Climate models and other Earth System Models are large-scale simulations that couple multiple different component models together into a single simulation
  - DoE's Energy Exascale Earth System Model (E3SM) couples an atmosphere model with land, river transport, ocean, and sea ice models
  - Each of these component models implement different physics routines that may perform better on different hardware (e.g. fog models)
  - In nature, **all these processes are very tightly coupled** → weather patterns impact the formation and flow of ice sheets and when modeling, data must be moved between the components at a high frequency to obtain more accurate forecasts
- Macroheterogeneity provides a unique opportunity for accurate and fast simulations
  - Less computationally intense components (e.g., ocean, sea ice) can run on the CPU partition
  - AI surrogate models can run on GPU partitions
  - Natural convection processes can be run on a wafer-scale partition if needed

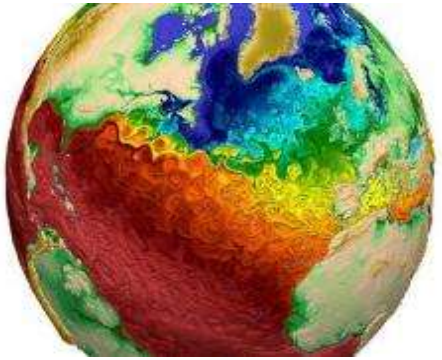


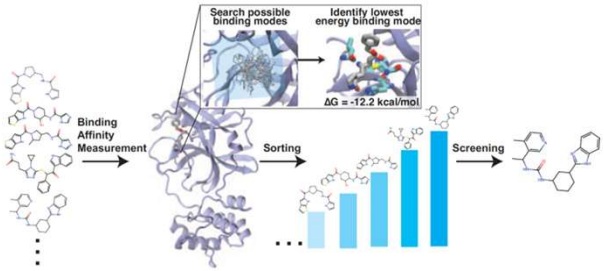
Image Source: https://www.anl.gov/article/updated-exascale-system-for-earth-simulations

Example Use Case: NekRS-ML (Physics-Informed CFD + ML)

- NekRS-ML is a fork of the ALCF-managed NekRS CFD solver, augmented to provide examples and capabilities for AI-enabled CFD research on HPC systems.
    - Physics-Based Mod/Sim:
      High-fidelity simulations to resolve turbulent flow physics and generate trusted reference data → CPU partition due to memory bandwidth sensitivity, and control-flow–intensive solvers
    - Digital Surrogate Training:
      ML models are trained in-situ using NekRS simulation data to learn mesh-based representations of flow dynamics, turbulence closures, or reduced-order operators.
      Dist-GNN: Scalable, distributed graph neural network for time-dependent and time-independent modeling on domain-decomposed CFD meshes → GPU partition/Data-flow partition may be best suited for training
    - Surrogate Inference (Replacing Mod/Sim):
      Replace repeated high-cost NekRS simulations for parameter sweeps, design exploration, and rapid prediction → GPU partition enables fast, scalable inference on large meshes
- Transitioning from full-physics simulations to ML-based digital surrogates enables scalable digital twins and efficient AI-driven CFD on leadership-class HPC systems

Example Use Case: MLDocking Drug Discovery Screening

- Couples AI inference, physics-based docking simulation, and model fine-tuning in an iterative loop to solve the drug-discovery bottleneck and search large chemical spaces for promising compounds
- Classical simulation using full Aurora system takes **1 day per target protein**
  - Inference: SST transformer screens the large ZINC22 SMILES corpus and predicts binding affinity across GPUs. SST can screen O(10^{10}) compounds in about **23 seconds per target protein**. AI accelerators with very high low-precision throughput are good candidates.
  - Mod/Sim: The simulation step takes the top candidate compounds identified by AI inference and runs OpenEye docking simulations to generate more trustworthy docking scores and are used for fine-tuning the SST model. Control-flow and memory intensive may be well suited to high-performance CPUs.
  - Training: TensorFlow fine-tunes the SST surrogate using simulated labels; updated weights feed the next screen. Training updates the AI surrogate using the latest physics-based simulation results→ GPUs are good targets for these.



Vasan et al., IPDPS, 2024

Network Optimizations for Low-latency Data Sharing

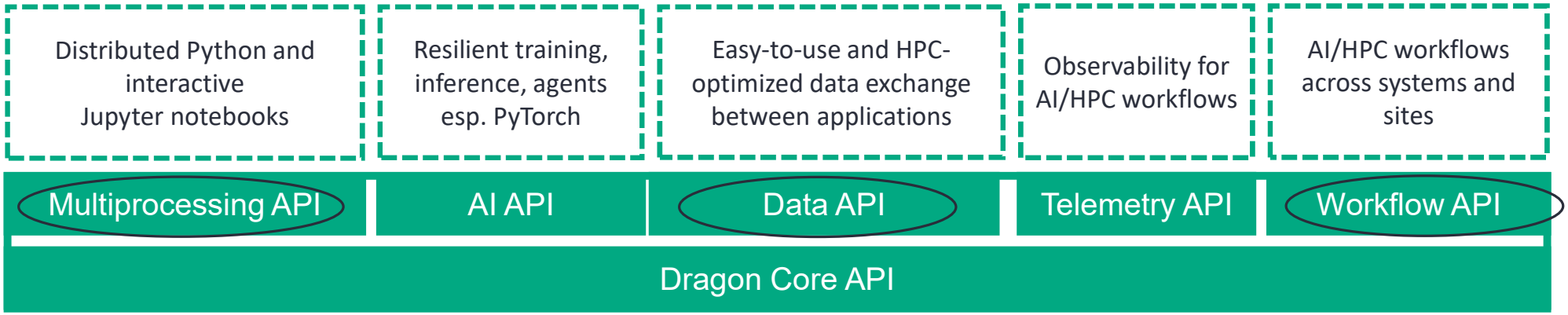
- HPE Slingshot (high-performance and low-latency UEC network) can support multiple fabrics where each partition has its own fabric and fabrics are managed independently (scale-up in one for e.g.)

- Partition fabrics connected with Link Aggregation Groups (LAGs)
  - Bundle multiple physical links into a single logical link to connect partitions and Links can be connected to different switches in each fabric
  - Adaptive routing and congestion management support routing to any link that connects to the destination fabric

- Communication model within and across partitions
  - Bridge from MPI apps to Pytorch apps
  - MPI inside Mod/Sim, CCL inside a scale-up inference partition for GPUs/acc
  - RDMA-based puts and gets

Runtime optimizations for low-latency data sharing

DragonHPC allows orchestration and in-memory data sharing between workflows

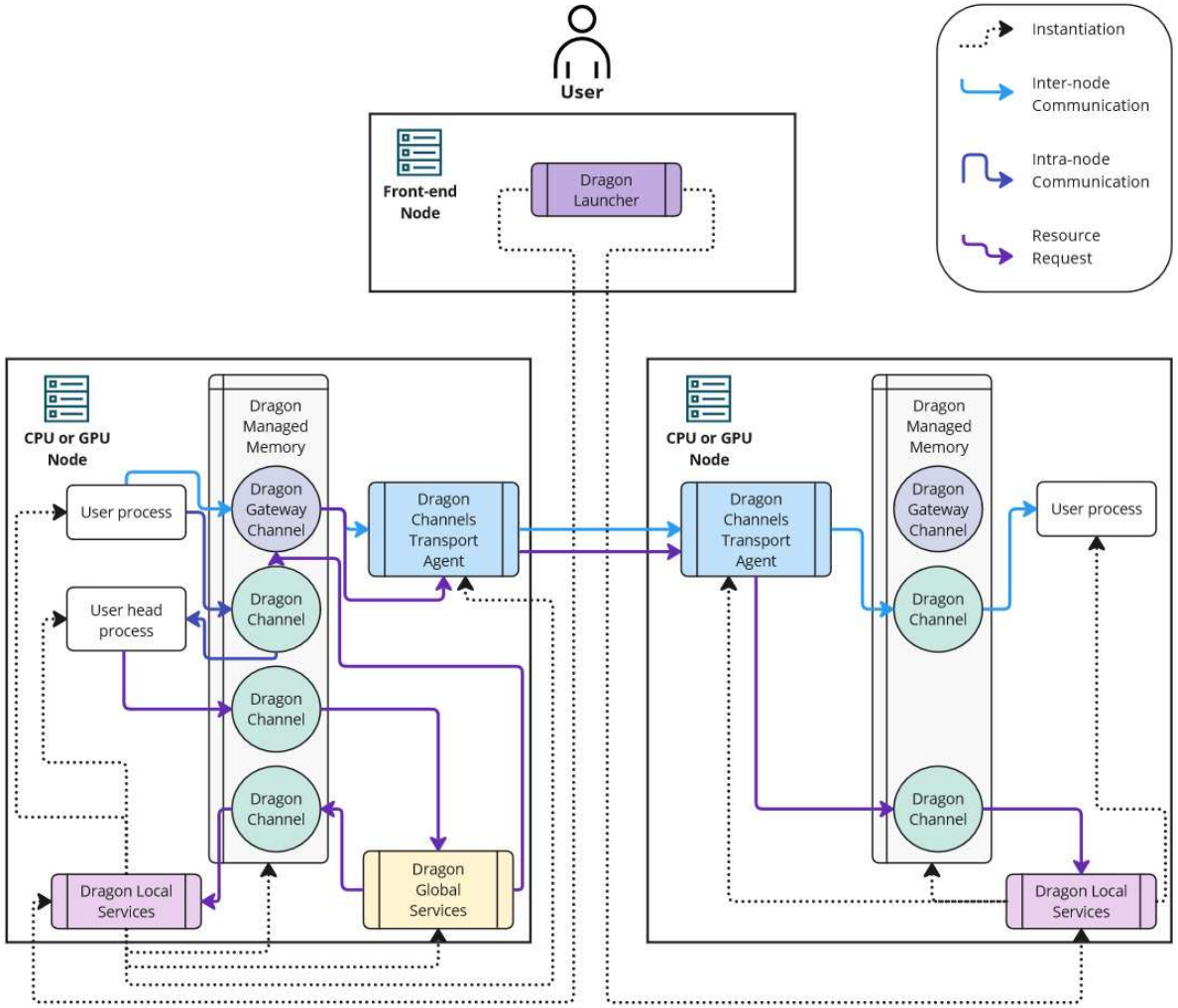


- Distributed Dictionary (Ddict) to provide in-memory key-store value store accessible
  - Pass training data from an MPI simulation directly to PyTorch — no filesystem needed →Ideal for coupling producers (simulations) with consumers (AI training)

- Process Group and Policy for fine-grained control over process lifecycle, placement, and GPU affinity across distributed nodes
  - Policy allows pinning ProcessGroup to diverse hardware

DragonHPC (https://dragonhpc.org/portal/index.html)
Rhapsody is a heterogeneous workflow system/middleware that can expose Dragon capabilities in an easy-to-use manner (https://github.com/radical-cybertools/rhapsody)

Using Dragon Channels to Communicate between Partitions



System Performance for MH Systems

Delivered scientific outcomes (FOM) per unit power (Science/Watt)

- HPC: time-to-solution, validated simulation throughput,
- AI training: tokens/sec × convergence efficiency
- AI inference: validated tokens/sec at target latency
- For hybrid workflows: completed workflows/day (sim+ml surrogate) combining above

Achieved performance is bounded by the lowest active architectural ridge

- Compute ridge: arithmetic throughput (precision-dependent)
- Memory ridge: bandwidth-limited phases
- Communication ridge: latency and synchronization overheads

Emerging applications can be modeled as a weighted mixture of execution phases.

- Decompose application into compute, memory, communication, and control phases
- Weight performance and power by time fraction and add application SW overheads and utilization caps
- Add cross partition latency based on amount of data and network latency

Modeling MH Architectures and Workflows

Partition Diversity

- GPUs, AI Inference accelerators, CPUs, future architectures
- Peak utilization is bounded by the ridges and software overheads
  - E.g. NVL72 partition has lower communication ridge than NVL4 GPUs connected with scale-out network
  - Validated with empirical utilization

Systems diversity

- Tightly coupled homogeneous systems
- Tightly coupled macroheterogeneous systems
- Loosely coupled /Siloed: Homogeneous or diverse but with high data-sharing penalties

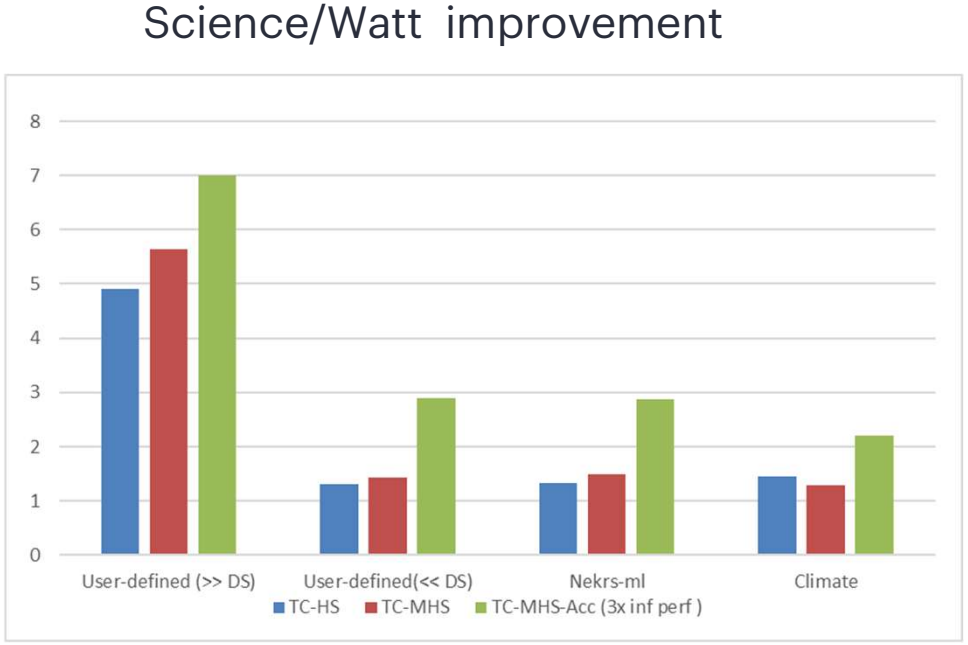
Workflow diversity

- CFD, Climate model, User-defined mix
- Arithmetic intensity, fraction, data sharing within and between partitions (% and latency)

E3SM characterization

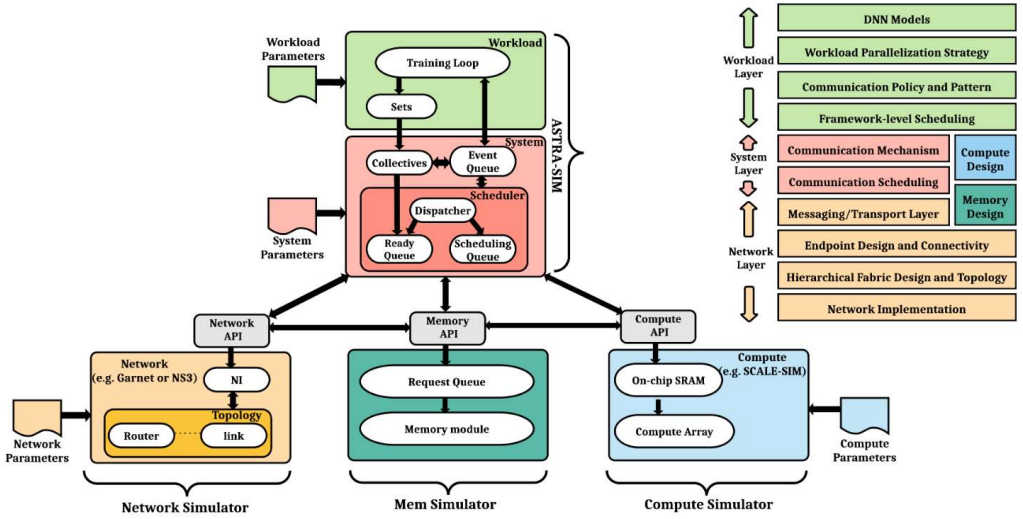
Workload	Model intensity	Share	Cross-partition? (1=yes)	Cross-partition data
Atmosphere Emulator (Ai2)	100.00	20%	1	20%
Ocean	0.20	70%	1	15%
Sea Ice	0.30	10%	1	10%

Early indicators of system efficiency



- Goal is to study AI-coupled HPC workloads and emerging workflows with high and low data sharing
- Model currently assumes concurrent execution and measures aggregate science throughput, not workflow latency
- Even small workload share can influence science/Watt
- Higher the diversity in acc and communication perf more the impact
- Match % HW to workflow phases is not trivial
- Communication optimized partitions help inference phases
- Being enhanced to capture time-to-solution

Simulation Framework for Macroheterogeneous Systems



- Need a scalable, modular, and open source system-level simulation framework that can integrate diverse computing partitions
- Data sharing impacts in particular needs better architectural models
- AstraSim is a modular and extensible framework that can incorporate high-level as well as detailed models for compute partitions
- SST (https://sst-simulator.org/sst-docs/) detailed models can be incorporated into AstraSim
- Chakra workload generator can generate AI traces being expanded to support HPC traces as well as workflows

Profiling Tools

Existing tools collect performance information for current workloads
Need workflow profiling tools to gather better information about data sharing patterns
Performance info → provide analysis of how different code regions perform on specific hardware
Use the performance data to **recommend** which partition to run workload/phase on

Observation	E.g. Recommendation
High FLOP intensity	Run on GPU or wafer-scale partition
High Integer / branch-heavy	Run on CPU partition
High mem-BW and low compute intensity	Run on dataflow accelerator partition
High cache hit/miss ratio	Run on dataflow or processing-in-memory partitions
High sparsity / irregular memory access	Run on CPU or FPGA partitions
Optimization problem	Run on Quantum partitions

Summary

Potential to significantly improve real workflow performance if hardware diversity becomes mainstream

Data sharing between partitions is a big differentiator for MH systems

Software and operational complexities still exist → communication and partitioning between partitions

Need more hybrid workflows to test across different domains

Need community-driven effort across modeling and workflows

Thank you!

Samantika Sury
Samantika.sury@hpe.com