

Scalability in HPC/AI

Focus of this session

How far can AI/HPC methods scale when the limiting factors are memory, heterogeneity, interconnect, and numerical precision?

Chair: Mohamed WAHIB (RIKEN)

Co-chairs: Emmanuel Jeannot (INRIA)

Arthur Lorenzon (UFRGS)

Philippe Navaux (UFRGS)

Brian Spears (LLNL)

Why scalability is now harder

- AI/HPC workflows scale across compute, memory tiers, storage, and communication domains at the same time.
- Peak flops keep increasing, but usable performance is often set by memory capacity, bandwidth, network behavior, and software complexity.

Working definition

Scalability means sustaining useful performance and scientific fidelity as problem size, model size, and system size increase.

Session scope

Four bottlenecks: memory limits, heterogeneous architectures, slow interconnects, and low precision.

Scalability axis 1 — memory limits

- Model sizes and simulation data products grow faster than on-package memory capacity.
- HBM delivers bandwidth, but capacity remains constrained. This pushes applications into tiered memory, recomputation, compression, or data staging.
- The result is a shift from compute-bound scaling to memory-aware scaling.

Challenge

How do we scale when the fast memory per device is the real resource being exhausted?

What this affects

Training, inference, surrogate models, in situ analysis, graph workloads, and coupled simulation–AI pipelines.

Typical responses

Partitioning, offload, memory pooling, checkpoint trade-offs, sparsity, quantization, and topology-aware placement.

Scalability axis 2 — heterogeneous architectures

- Exascale systems are built from CPUs, GPUs, multiple memory spaces, and increasingly specialized kernels and runtimes.
- The performance question becomes where to place data, where to execute each phase, and how to keep the full workflow balanced.
- Heterogeneity improves capability, but complicates portability, tuning, scheduling, and reproducibility.

Discussion point

Does heterogeneity still help when the software effort needed to exploit it grows faster than the delivered efficiency?

Core challenge

We want performance portability, but current AI/HPC stacks still depend heavily on platform-specific kernels, communication libraries, and memory behavior.

Scalability axis 3 — slow interconnects

- At scale, communication, synchronization, and data movement often dominate the marginal cost of adding more resources.
- This appears in distributed training, stencil and particle codes, multi-physics coupling, and in situ pipelines.
- The system may have exascale compute capability, yet the application scales only if communication is reduced, hidden, or reshaped.

Core issue

More devices do not guarantee more useful work when the fabric becomes the bottleneck.

Typical symptoms

Low strong-scaling efficiency, idle accelerators, noisy iteration times, and sensitivity to placement and routing.

Needed response

Communication-avoiding algorithms, overlap, locality-aware decomposition, better collectives, and workflow co-design.

Scalability axis 4 — low precision

- Low-precision arithmetic is now the main point to AI scalability because it reduces memory traffic, storage footprint, and time-to-solution.
- But lower precision changes the numerical behavior. For HPC and AI-for-science workloads, accuracy, stability, and uncertainty matter.
- When lower precision is scientifically acceptable?

Opportunity

More throughput and larger effective memory capacity.

Risk

Loss of convergence, unstable training, silent numerical drift, or degraded physical fidelity.

Research need

Mixed-precision methods with explicit error control, validation, and domain-aware acceptance criteria.

What ties the four axes together

- Memory, heterogeneity, interconnect, and precision are not separate problems. They interact.
- Example: low precision can relieve memory pressure and communication volume, but may require extra iterations or stronger validation.
- Example: heterogeneity can improve performance, but also amplifies data-placement and communication problems.

Implication for the group

Scalability should be discussed as a system problem across algorithms, runtime, architecture, and scientific correctness.

Questions to start the discussion

- Which of these four factors is the principal limiter of scalability in today's AI/HPC workloads, and under what workload or system conditions does that ordering change?
- What should this community optimize for first at scale: time-to-solution, energy efficiency, model quality, scientific fidelity, or overall facility throughput?
- In which parts of the AI/HPC stack is low precision already mature enough to be treated as standard practice, and where does it still introduce unacceptable risk?
- How far can heterogeneous execution be pushed before programmability, portability, and software complexity outweigh the performance benefit?
- Which communication and synchronization patterns are now the clearest barrier to scalable AI/HPC workflows on exascale-class systems?
- What common programming or runtime abstractions are needed so that memory management, heterogeneity, and workflow composition stop being handled as isolated problems?