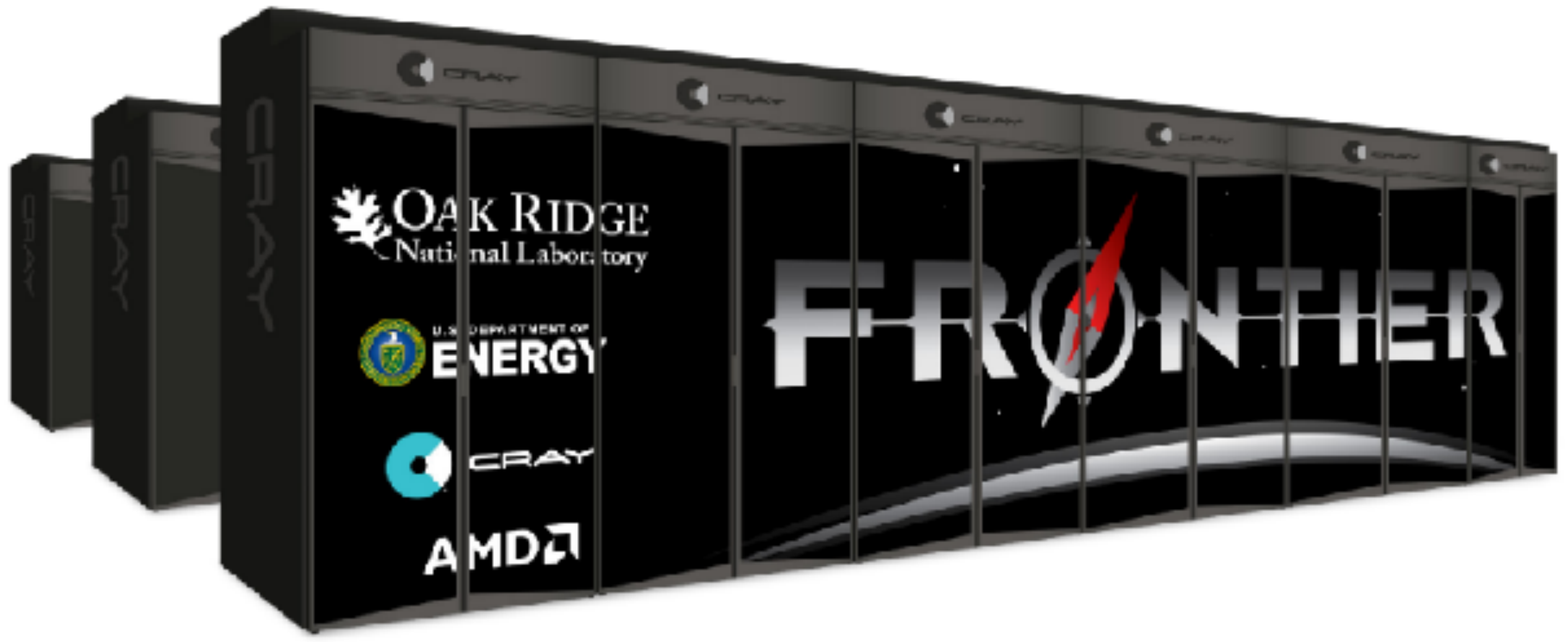
# Next-Generation High-Performance Computing: Opportunities and Challenges

Salman Habib
CPS Division
Argonne National Laboratory







## HPC system architecture "swim lanes"

- Accelerated (primarily CPU/GPU)
- Many-Core (ARM)
- Specialized

## 'Next-Generation' HPC Trends

- Complex system modeling
- Complex workflows
- Data as equal partner to compute
- AI/ML in HPC space
- Increased usage of HPC within 'science loops'

# Computing Needs for Science

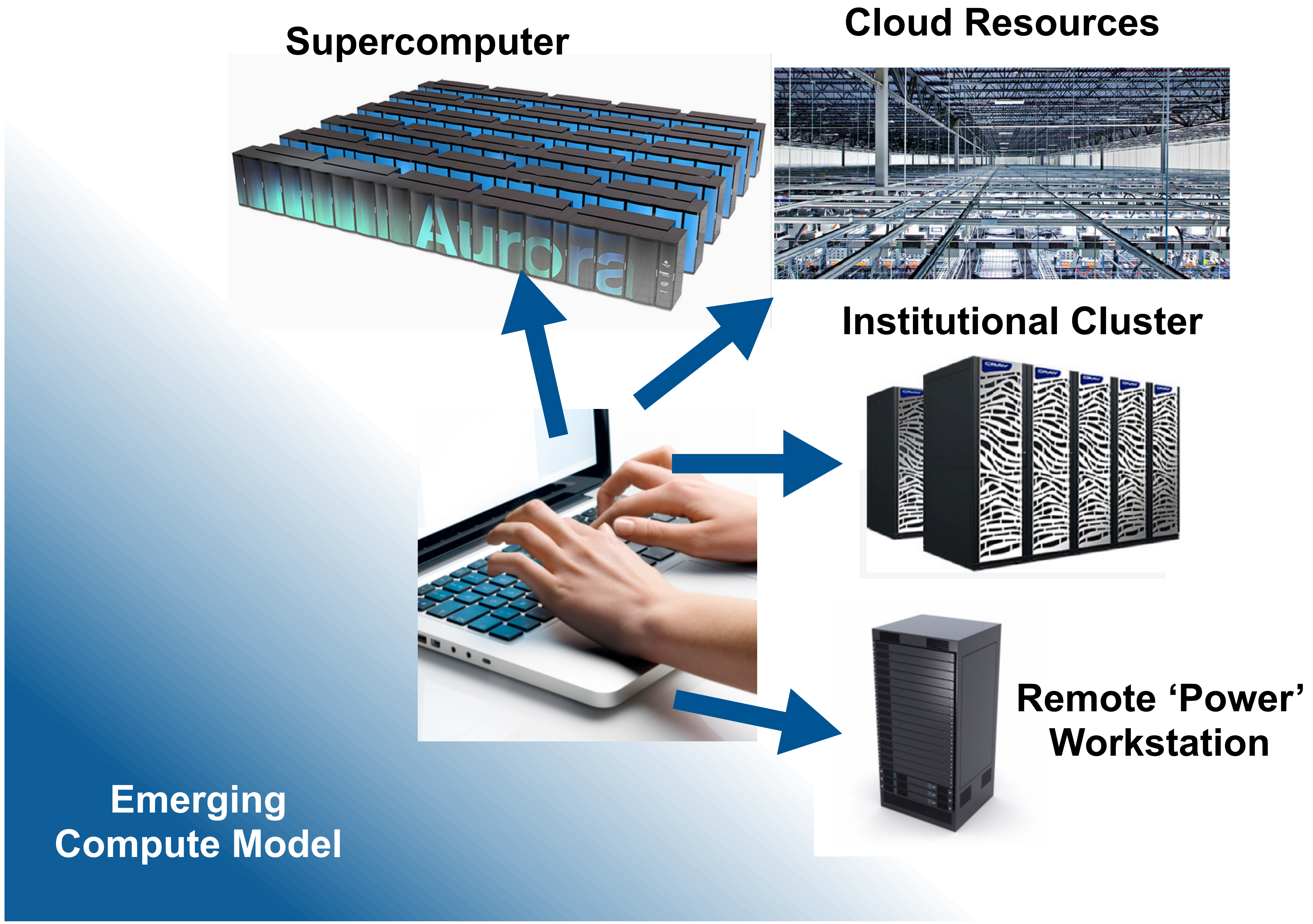
- **Many Communities Need Large-Scale Computational Resources**
  - Light Sources
  - Biology
  - Climate/Earth Sciences
  - High Energy Physics
  - Materials
  - —
- **Message:** Overall scientific computing use case is driven by large-scale data flow + volume
- Traditional HPC use cases continue to exist — well-defined “hard” problems (lattice QCD, quantum chemistry, —)
- Data-intensive applications will be ubiquitous, and will need performance, reliability, and usability
- Overall balance of compute + I/O + storage + networking will need to be thought through
- How will HPC — and the case for HPC — evolve in this complex scenario?

# Different Flavors of Computing

- **High Performance Computing ('PDEs')**
  - Parallel systems with a fast network
  - Designed to run tightly coupled jobs
  - High performance parallel file system
  - Batch processing
- **Data-Intensive Computing ('Interactive Analytics')**
  - Parallel systems with balanced I/O
  - Designed for data analytics
  - System level storage model
  - Interactive processing
- **High Throughput Computing ('Events'/'Workflows')**
  - Distributed systems with 'slow' networks
  - Designed to run loosely coupled jobs
  - System level/Distributed data model
  - Batch processing

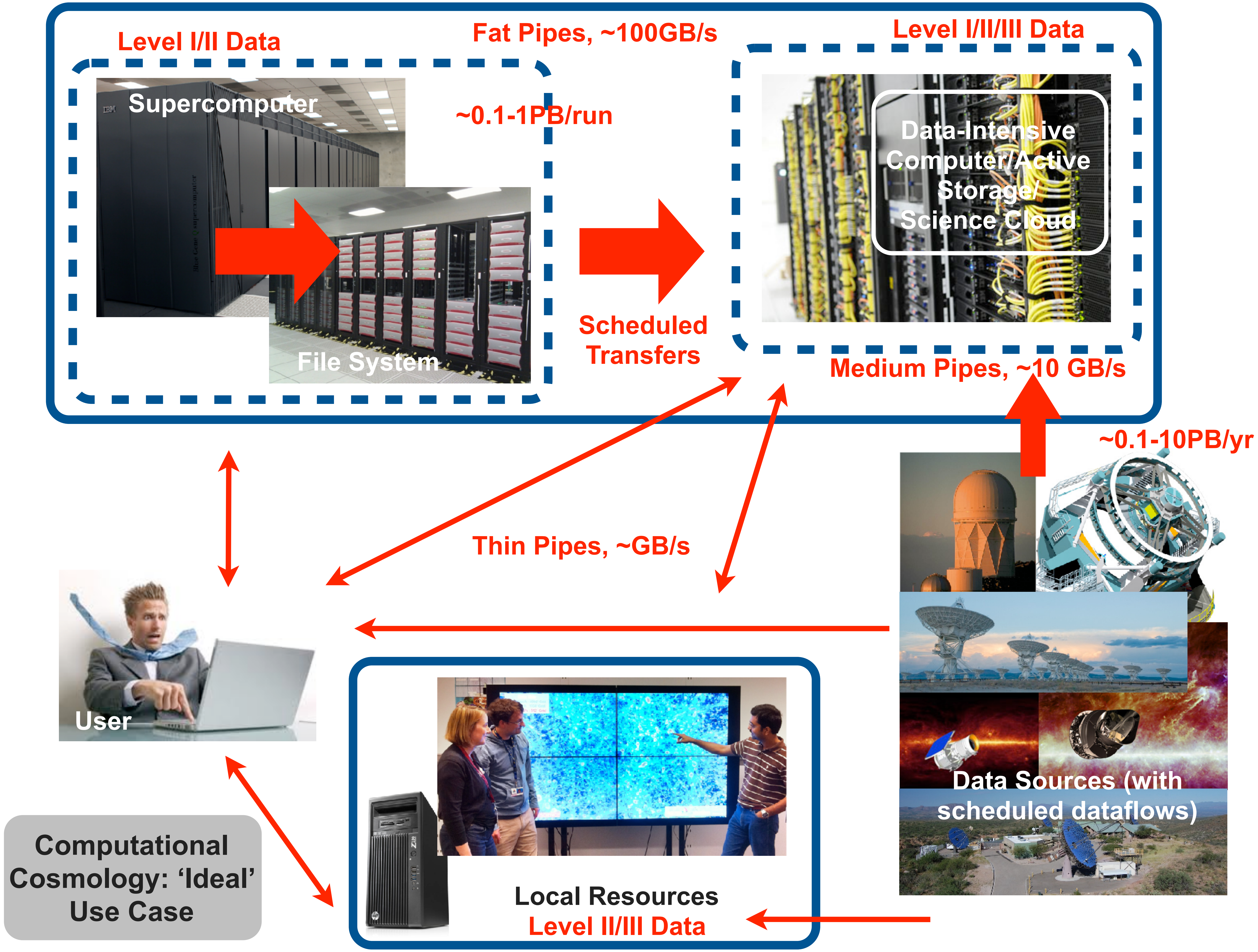
# Notional Compute Models for Computational Science

- **Computing Niches**
  - Laptop Scale
  - Research Group
  - Institutional
  - Cloud/Supercomputing
- An integrated computing model is very desirable, but changing hardware/ software boundaries and use cases have made this difficult to achieve
- Resulting sociological changes — the idea of a "Renaissance Computational Scientist" has given way to specialization and the need to support high-level programming models accessible to a large user base
- But the need remains to be fully addressed —



# Target Example

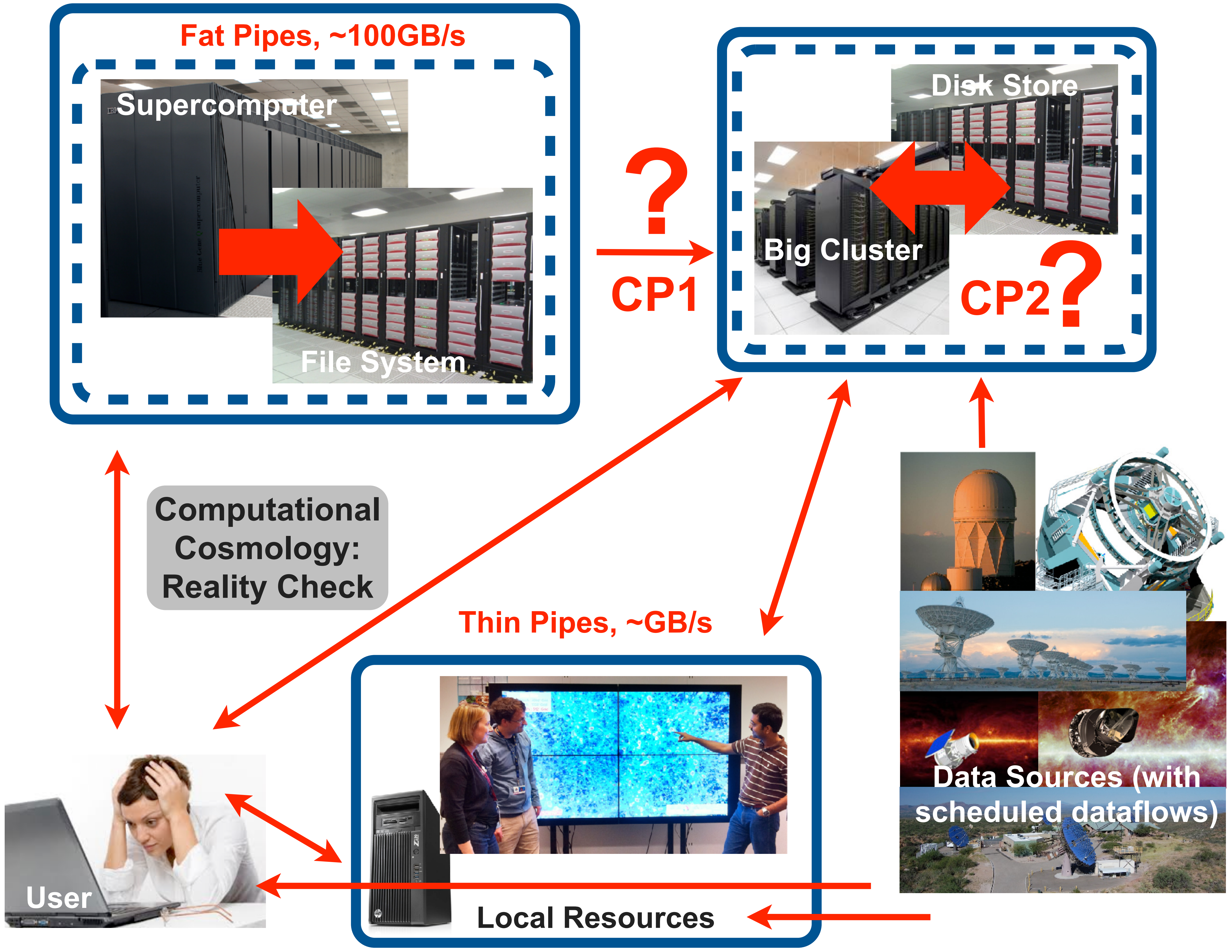
- Community use of next-generation systems is highly desirable
- Link-ups of various computational and data resources would greatly enhance scientific discovery
- Requires integration of local, intermediate, and global resources, with a (relatively) seamless mechanism for running and marshalling compute campaigns/jobs
- There is a growing realization that this is where we want to be, i.e., traditional HPC should not be isolated, but become part of multiple scientific ecosystems



# Target Example: Reality

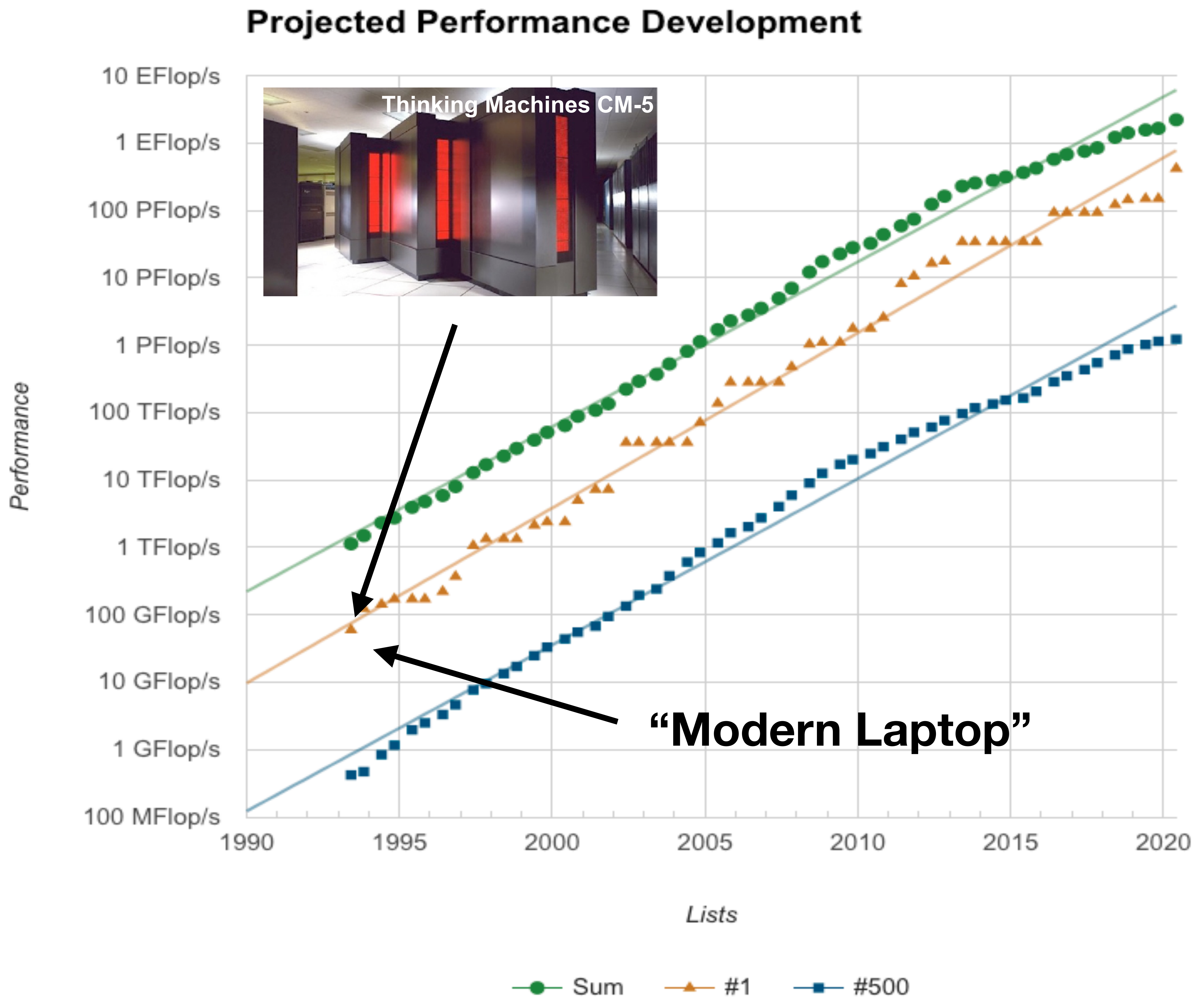
- Unfortunately, we are not there yet!
- Too many infrastructural chokepoints/ bottlenecks
- Lack of a sufficiently unified software base (largely due to moving targets, but this is getting better)

**Change will require multiple communities working together —**



# HPC Evolution: General Remarks

- **Top 500:** Peak perfromance has maintained a steep rate of increase, although some slowing down is evident
- **Caveat:** Overall, this is somewhat misleading, because systems have gotten ***bigger***, not ***faster***
- Ability to make use of concurrency is essential; modern systems are complex (memory hierarchy, node complexity, network imbalance, —), hard to get computational efficiency
- Laptop to supercomputer is roughly 6 orders of magnitude in performance, 5 in RAM, 4 in storage
- Key problem — “software wall”, laptops are easy to use and versatile, providing access to diverse and powerful software suites, while supercomputers are not (relatively speaking)



# What are Supercomputers Good For?

- Quantitative predictions for complex systems
- Discovery of new physical mechanisms
- Ability to carry out 'virtual' experiments (system scale simulations)
- Solving large-scale inverse problems
- Ability to carry out very large individual runs as well as simulation campaigns
- New communities, new directions?
- Supercomputing + Large-Scale Data Analytics?
- Scalable AI/ML Applications

OLCF

## OLCF Science



ALCF

SCIENCE HIGHLIGHTS



NERSC

## Table of Contents

# However, —

- **Dealing with supercomputers is painful!**
  - HPC programming is tedious (MPI, OpenMP, CUDA, OpenCL, —)
  - Batch processing ruins interactivity
  - File systems corrupt/eat your data
  - Software suite for HPC work is very limited
  - Analyzing large datasets is frustrating
  - HPC experts are not user-friendly
  - Machine crashes are common
  - Ability to ‘roll your own’ is limited

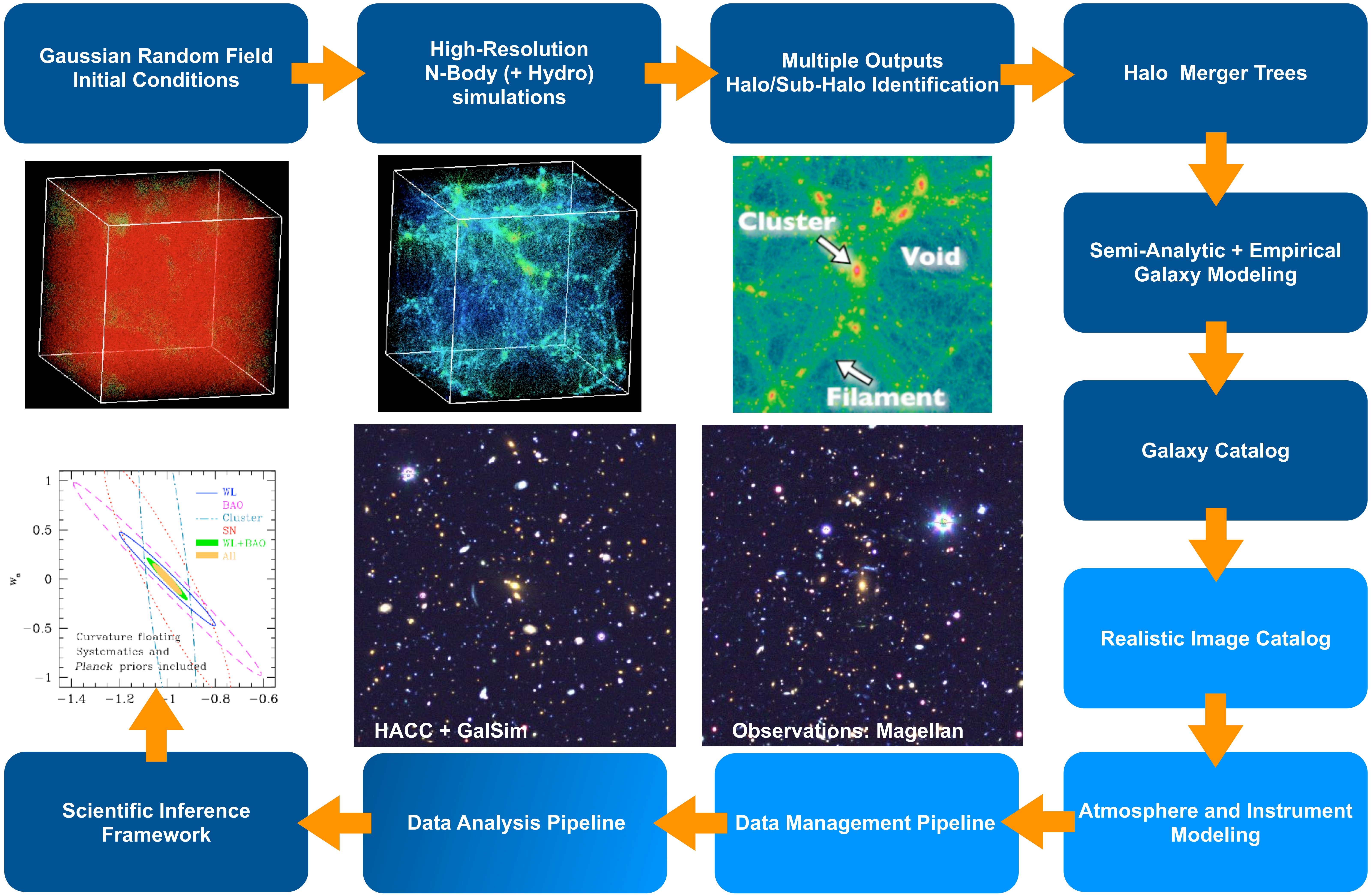

Running Jobs | Queued Jobs | Reservations

**Total Queued Jobs**: 172

| Job Id | Project | Score | Walltime | Queued Time | Queue | Nodes |
|---|---|---|---|---|---|---|
| 307941 | SkySurvey | 8351.7 | 1d 00:00:00 | 5d 01:10:03 | prod-capability | 32768 |
| 307942 | SkySurvey | 8350.5 | 1d 00:00:00 | 5d 01:09:42 | prod-capability | 32768 |
| 309793 | NucStructReact_2 | 7069.0 | 01:00:00 | 1d 19:13:34 | prod-capability | 32768 |
| 309794 | NucStructReact_2 | 7065.1 | 01:00:00 | 1d 19:12:28 | prod-capability | 32768 |
| 309795 | NucStructReact_2 | 7056.8 | 01:00:00 | 1d 19:10:04 | prod-capability | 32768 |
| 309271 | LatticeQCD_2 | 6121.1 | 03:00:00 | 3d 03:40:34 | prod-capability | 12288 |
| 309314 | LatticeQCD_2 | 5036.1 | 04:50:00 | 2d 22:51:59 | prod-capability | 12288 |
| 309315 | LatticeQCD_2 | 5034.8 | 03:00:00 | 2d 22:51:38 | prod-capability | 12288 |
| 309316 | LatticeQCD_2 | 5034.0 | 04:50:00 | 2d 22:51:24 | prod-capability | 12288 |
| 309317 | LatticeQCD_2 | 5033.0 | 03:00:00 | 2d 22:51:08 | prod-capability | 12288 |
| 309318 | LatticeQCD_2 | 5032.6 | 04:50:00 | 2d 22:51:01 | prod-capability | 12288 |

# Full Circle — Is (Next-Generation) HPC Your Poison?

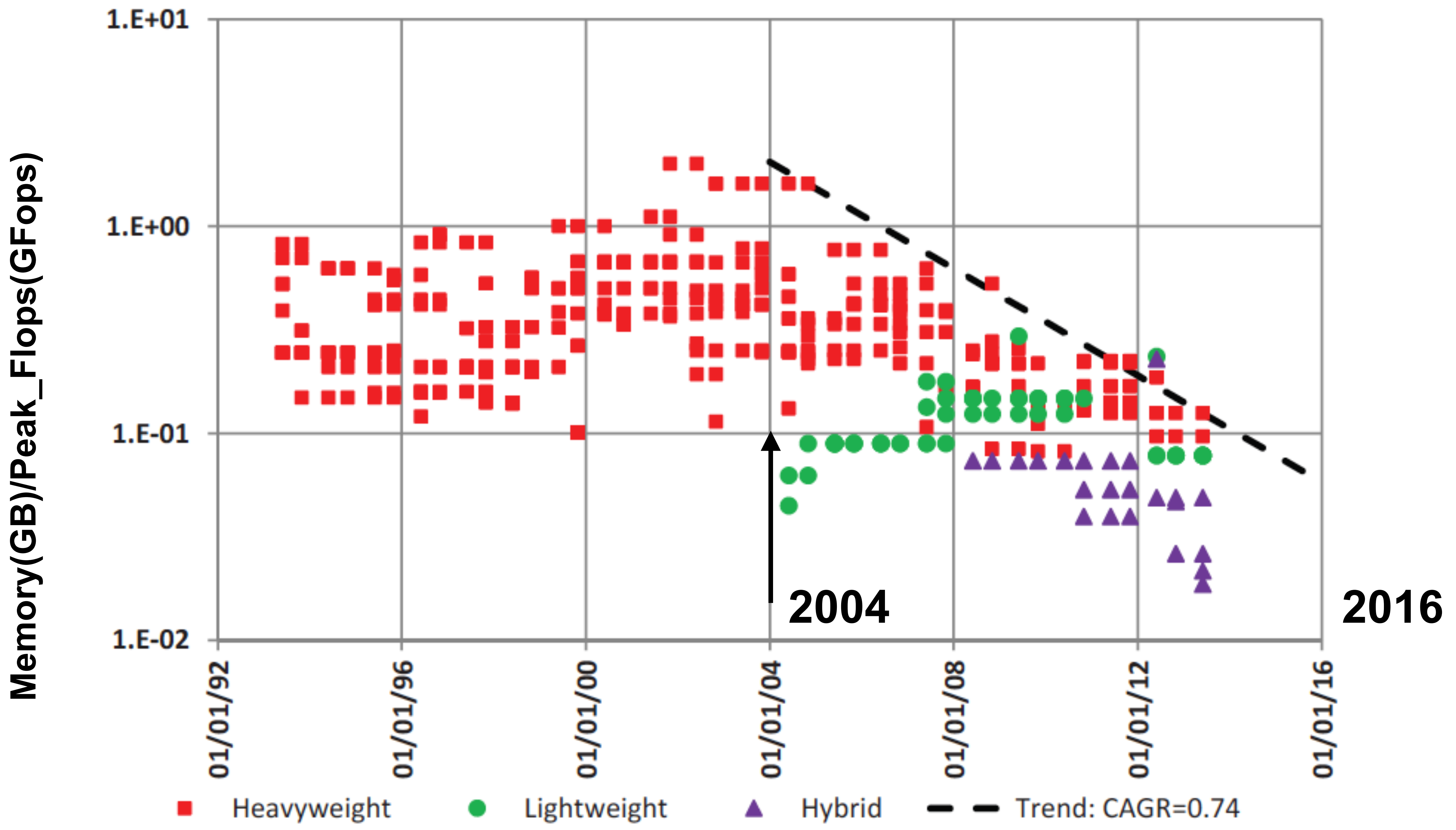
- Quantitative predictions for complex, nonlinear systems
- Discover/expose physical mechanisms underlying complex phenomena
- System-scale simulations ("impossible experiments")
- **Nature of Computational Tasks**
  - Very large simulations necessary, but not just a matter of running a few large simulations
  - High throughput essential (short wall clock times)
  - Optimal design of simulation campaigns (parameter scans)
  - Large-scale data-intensive applications




**Example of a complex End-to-End Task (Cosmology)**

# Hardware Evolution I

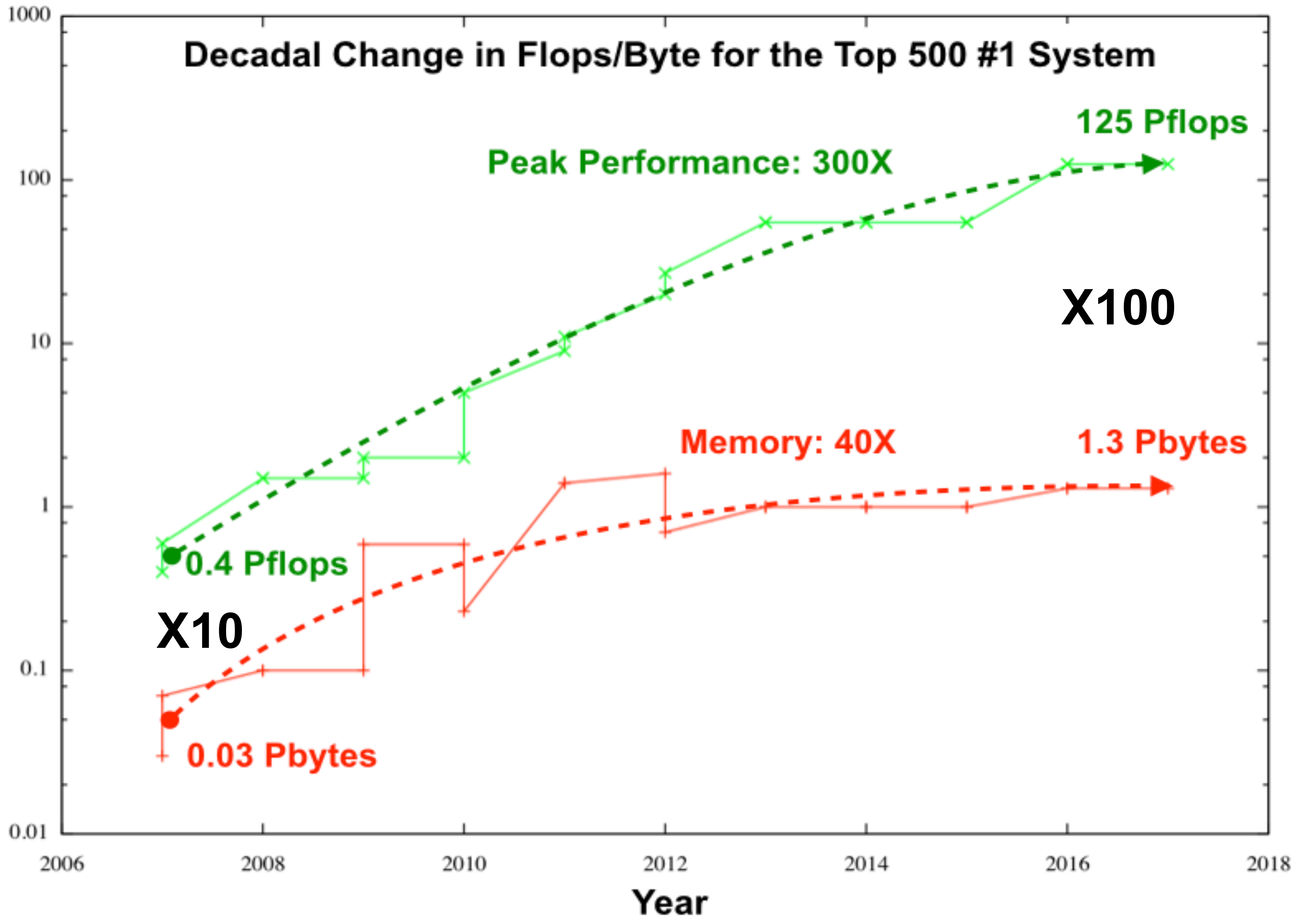
- **Power is the main constraint**
  - 50X performance gain by 2021/2022
  - ~50MW per large system
  - power/socket roughly const.
- **Only way out: more cores**
  - Several “mix/match” design choices
- **Micro-architecture gains sacrificed**
  - **Accelerate** specific tasks
  - Restrict memory access structure (SIMD/SIMT)
- **Machine balance sacrificed**
  - Memory/Flops; comm BW/Flops — all go in the wrong direction
  - Upper-level code **must be rewritten**
  - Low-level code **must be refactored**




- **End of weak scaling**
  - Problem size is stalling (limits on total RAM due to cost)
  - Need to add more physics in a way that can use out of balance systems
  - Current approaches face many obstacles

# Hardware Evolution II

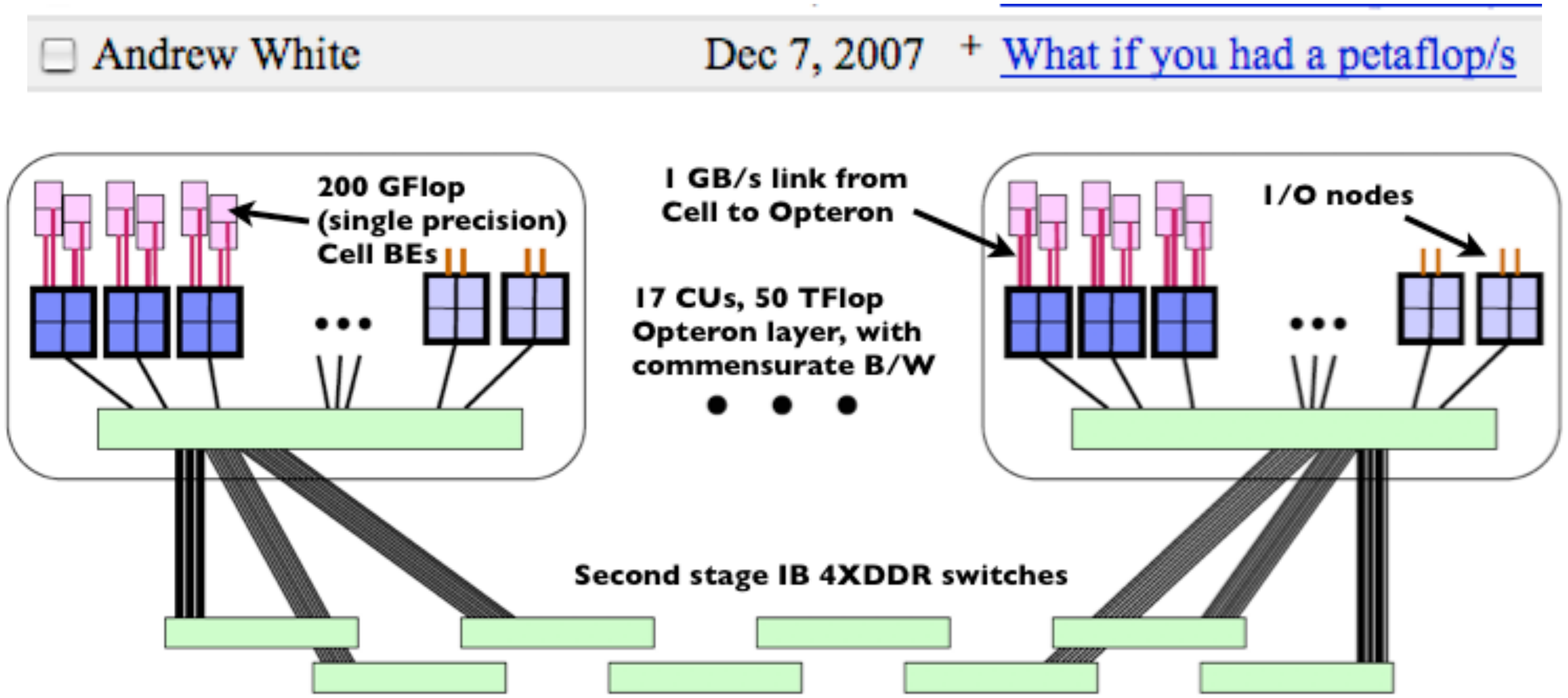
- **Evolution of HPC Systems**
  - Optimized for raw Flops
  - Poor Memory to Flops ratio
  - Poor Comm/IO to Flops ratio
  - Insufficient storage
  - Multiple technology 'swim lanes'
  - Rapid node architecture evolution in **nontrivial** directions
  - **Major lag** in software development
- **Mitigation Strategies**
  - Rethink computer architecture and design for science use cases
  - Storage caches with direct connectivity to compute nodes
  - Faster/fatter data pipes to compute platforms
  - Software strategies for portability




**Example of current supercomputer evolution: driven by a number of imperatives — economic and technological — leading to specialized nodal architectures (end of the 'Pile of PCs' model)**

# Architecture Evolution: Software Challenges

**Roadrunner:** **Prototype for modern accelerated architectures (2008)**

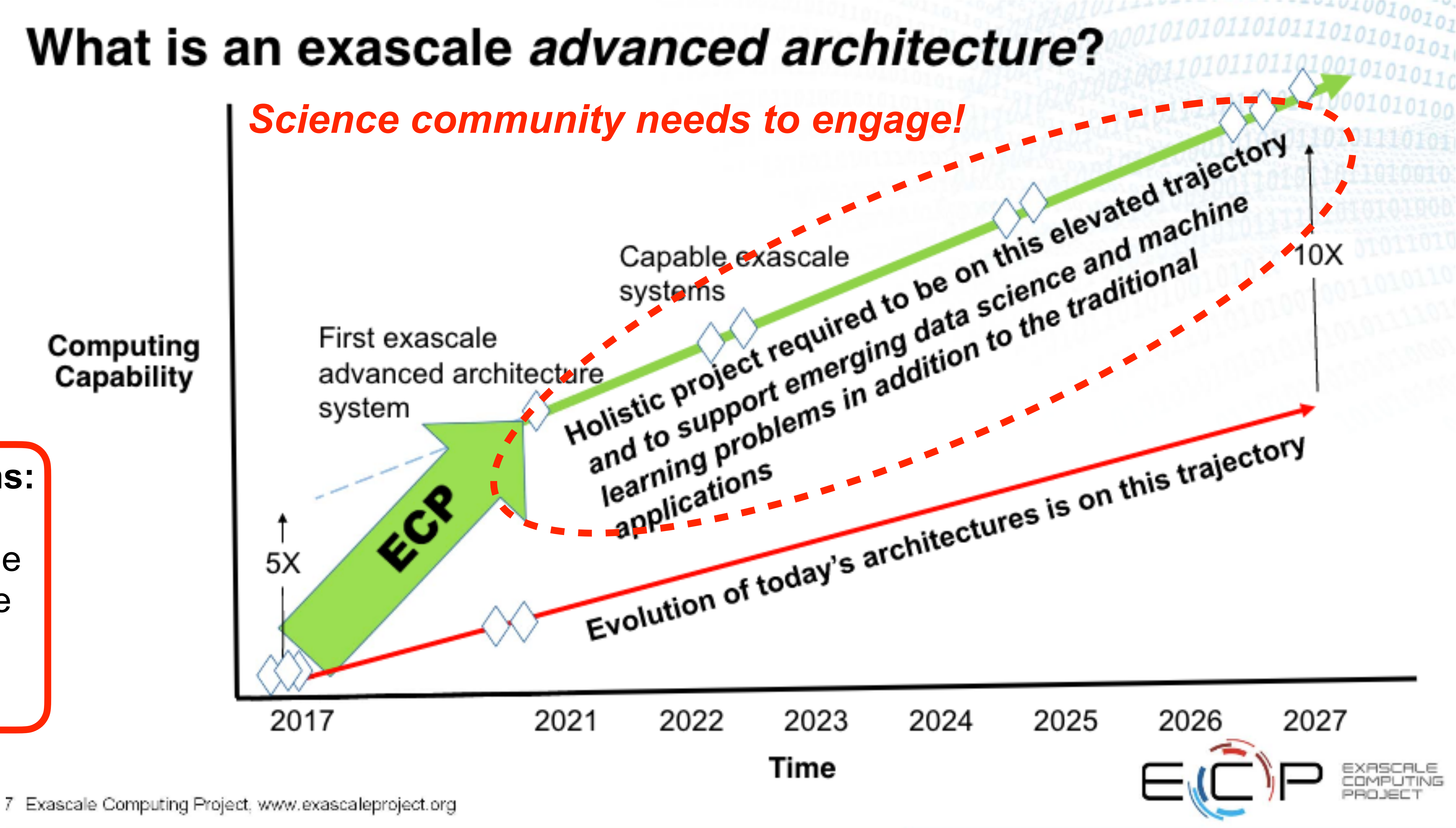



## Architectural Features/Problems of Next-Generation Systems (~2018+)

- Complexity at the node level (heterogeneity, accelerators, —)
- Simpler cores, multi-level memory hierarchy (limited DRAM/core)
- NVRAM (performance, resilience, I/O buffering)
- Skewed compute/communication balance (‘weak’ networks — PCIE, IB, Ethernet)
- Programming models? (major issue)
- I/O? File systems? Storage? (always problematic)
- Code for “Cold” vs. “Hot” systems (trend towards **computationally hot** machines)

# Exascale Computing Project

Major DOE SC and NNSA joint project to arrive at a scientifically usable architecture for exascale computing in the early 2020's — *largest science project within DOE*




**Exascale Systems:**
2021/22 Argonne
2021/22 Oak Ridge
2022/23 Livermore
2023/24 LBNL
2024 LANL

# Status Summary: Writing Code

- **HPC ‘Dreams/Myths’**
  - The magic compiler/programming model/language/ —
  - Co-Design (historically too perturbative, can this change?)
- **Dealing with Today’s Reality (Defensive Code Design)**
  - Code teams must understand all levels of the system architecture, but not be enslaved by it (software cycles are long)!
  - Software vs. hardware cycles are getting inverted in many cases, need to rethink how to write code
  - Must have a good idea of the ‘boundary conditions’ (what may be available, what is doable, etc.)
  - ‘Code Ports’ is ultimately a false notion, need an architecture-aware code design philosophy and a helpful programming environment (rich/interactive “decoration/pragmas?”)
  - Portability across architecture choices must be addressed (programming models, algorithmic choices, heterogeneity, trade-offs, etc.), “hard” vs. “soft” portability
  - Special-purpose (domain-specific) hardware, some small(er) teams are looking at such options, hard to imagine this as a community solution
  - Need to start thinking out of the box — domain scientists and computer scientists and engineers must work together

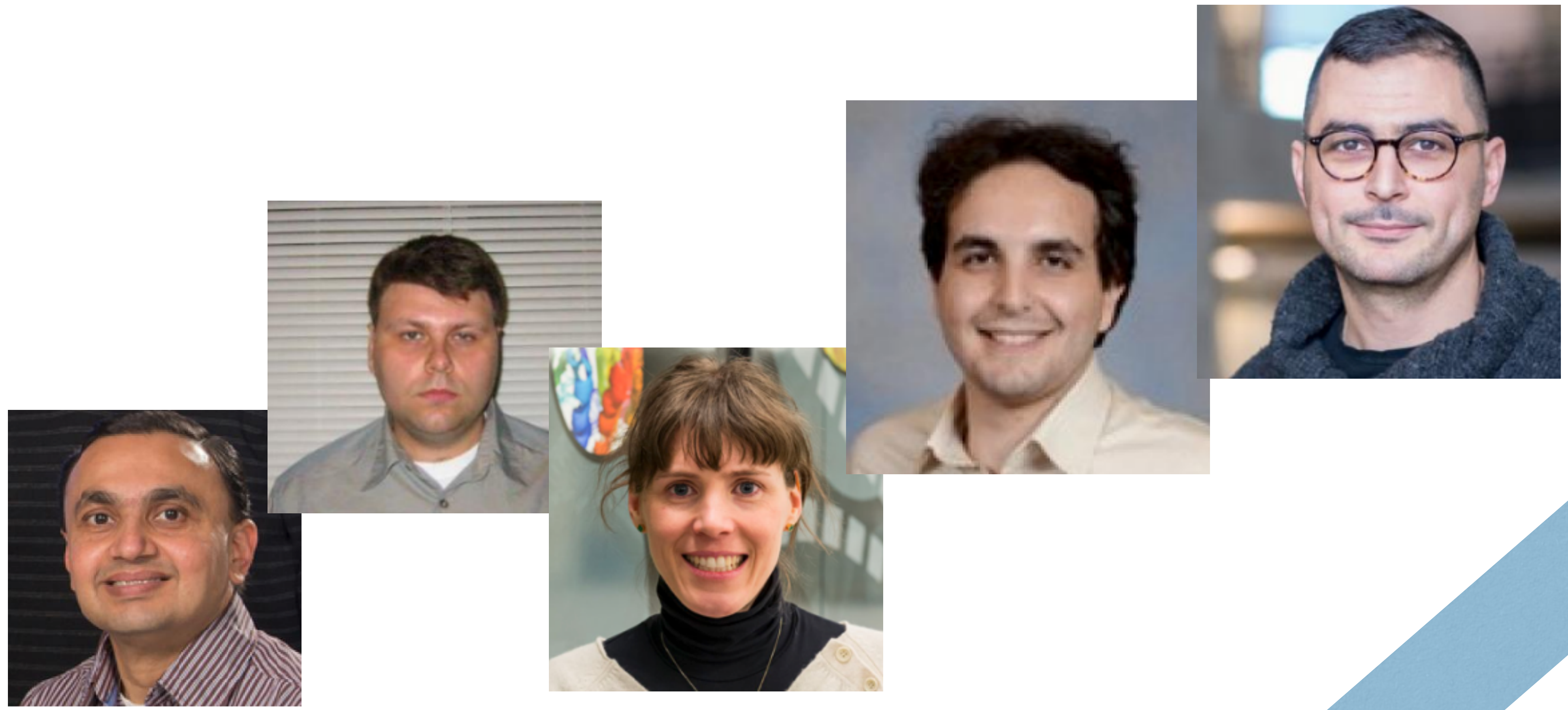
# CPS Division at Argonne

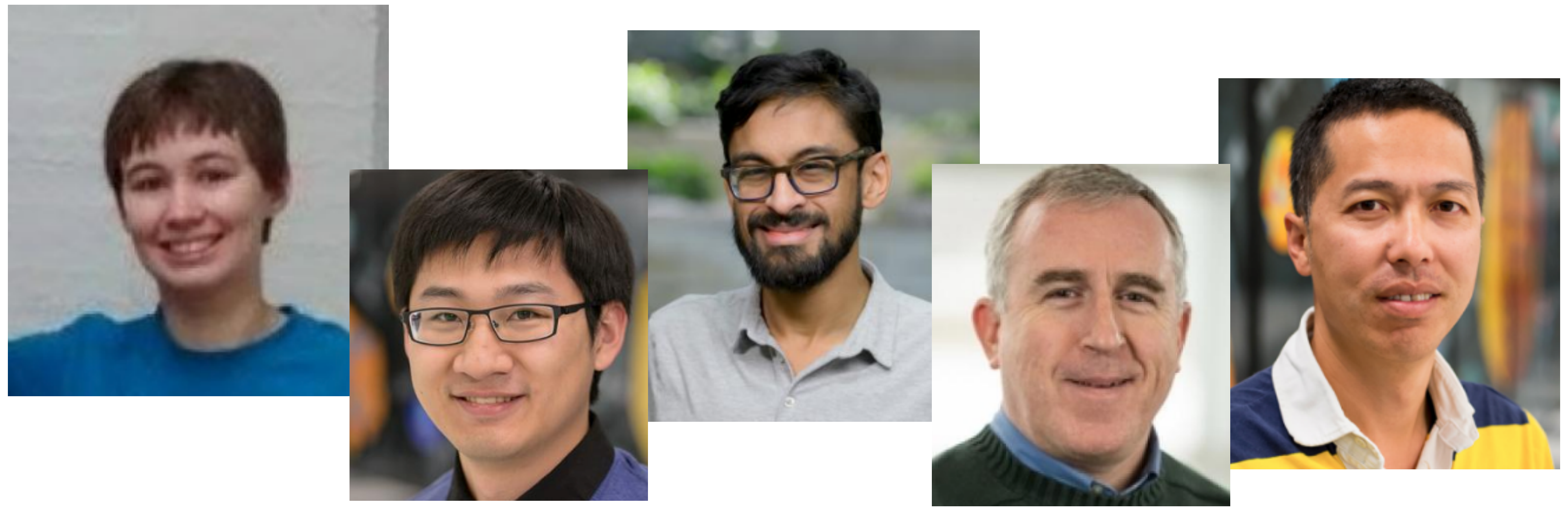
- **Scientific Focus Areas**
  - Computational chemistry and materials science
  - Computational cosmology and astrophysics
  - Computational fluid dynamics
  - Computational plasma physics
  - Computational quantum field theory
  - Particle transport
  - Quantum information science

**QMCPACK LAMMPS**
**WEST NWChem NAMD**
**QBOX GAMESS VALENCE**
**HACC NEURON Nek5000**
**PHASTA AVBP Converge**
**XGC USQCD SWFFT**
**—**

**40+ computational scientists, postdocs, and students —**

- **Computational Capabilities**
  - Advanced computing
  - Boltzmann/Monte Carlo transport
  - Data-intensive applications + AI/ML
  - Electronic structure methods
  - Parallel algorithms
  - Particle methods: MD/N-body/SPH/PIC
  - QIS/“Beyond Moore” computing
  - Quantum Monte Carlo
  - Software engineering





# CPS Division at Argonne

**Exascale science breakthrough opportunities**

Contact:
Salman Habib — habib@anl.gov
Tim Williams — zippy@anl.gov

- APS analysis/simulation
- Beam physics
- CFD/combustion
- Climate/weather modeling
- Computational astrophysics
- Computational biology
- Computational chemistry
- Computational cosmology
- Condensed matter simulations
- Device simulations
- Grid optimization
- HEP analysis/simulation
- Lattice/nonperturbative QCD
- Materials modeling
- NP analysis/simulation
- Nuclear structure
- Numerical algorithms/methods
- Plasma simulations
- Quantum simulations
- Reactor simulations
- Uncertainty quantification
- Windmills/windfarms
- —-

Argonne NATIONAL LABORATORY

# CPS: 5/10-Year Outlook I

- **5 Years: Exascale Breakthrough Opportunities (ALCF)**
  - Next-gen systems expected on the 2021/22 timescale, how can CPS contribute to **breakthroughs with exascale computing**?

- **General Issues in HPC: Application Community “Inertia”**
  - Slow evolution of scientific interests at the individual level
  - Research community inertia (shear mass)
  - Software inertia (software timecycles >> hardware timecycles)
  - Worsening **application diversity** because of the difficulty of using next-gen HPC systems; directly impacts the probability of obtaining breakthrough results

- **5 Years: Exascale “Convergence” Era Focal Points**
  - Use of **accelerated** HPC systems
  - Melding **computational science and ML** to develop new methods for solving large-scale problems
  - **Data-intensive** computing and efficient implementation of “workflow-like” applications

# CPS: 5/10-Year Outlook II

- **CPS Strategic Planning 'Pressure Points'**
  - **Disruptive computing paradigm and hardware changes** expected (FPGAs/ASICs, low-power, special-purpose systems, neuromorphic, quantum, —); must build a **future-ready** computational community
  - **Uncertain and multiple software futures** (what follows C++/MPI + X? how to support multiple computing paradigms?)
  - **New methods** for computational science in areas such as quantum/hybrid computing, AI/HPC hybrid applications, integrated data/HPC apps, wide-area data/computing integration
  - **Early** CPS activities and workforce planning should already be looking at the 10-year horizon to set appropriate priorities

**Looking for partners —**