Present and Future Computing Requirements for Computational Cosmology

Presenter:
Salman Habib
Argonne National Laboratory

Cosmic Frontier Computing Collaboration

Computational Cosmology SciDAC-3 Project

Jim Ahrens (LANL)
Scott Dodelson (FNAL)
Katrin Heitmann (ANL)
Peter Nugent (LBNL)
Anze Slosar (BNL)
Risa Wechsler (SLAC)

Ann Almgren (LBNL)
Nick Gnedin (FNAL)
Dave Higdon (LANL)
Rob Ross (ANL)
Martin White (UC Berkeley/LBNL)

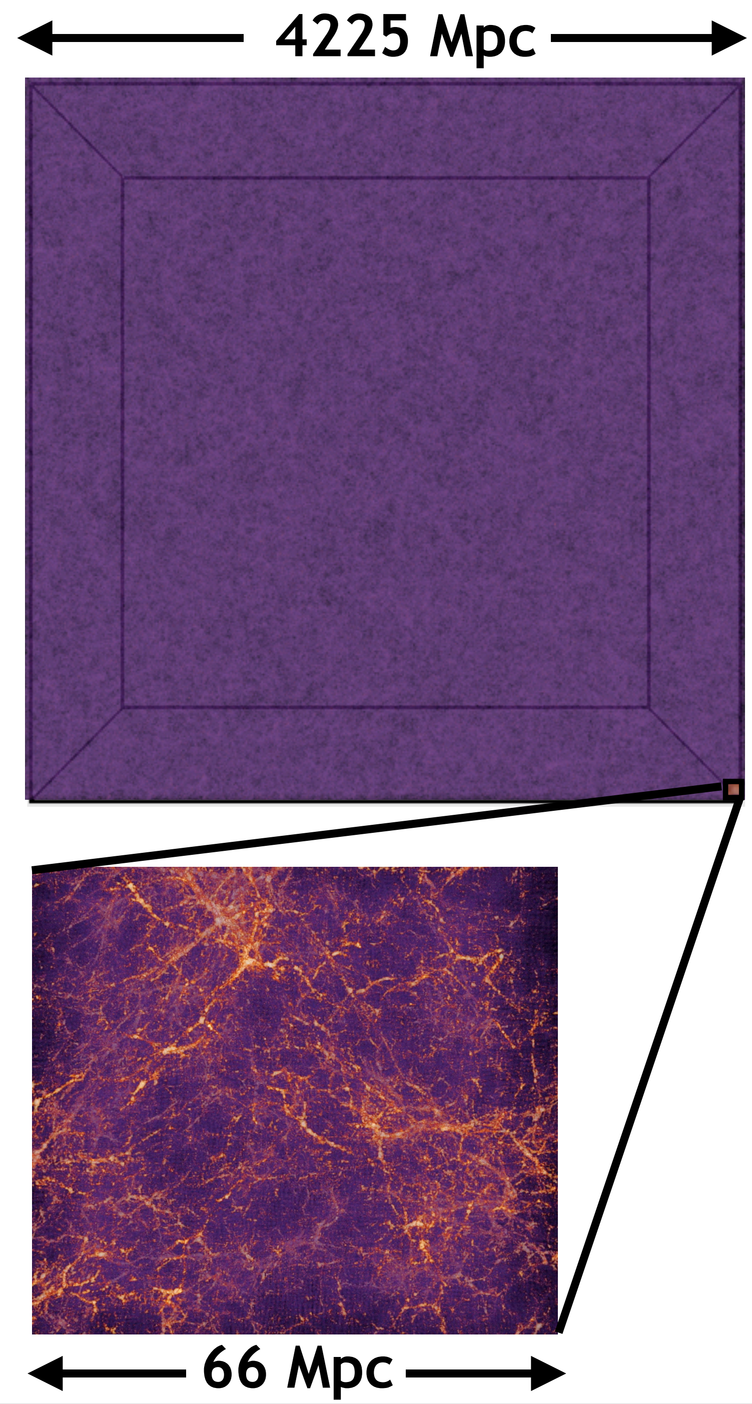
Large Scale Production Computing and Storage Requirements for High Energy Physics Research

A DOE Technical Program Review

November 27-28, 2012

Project Description

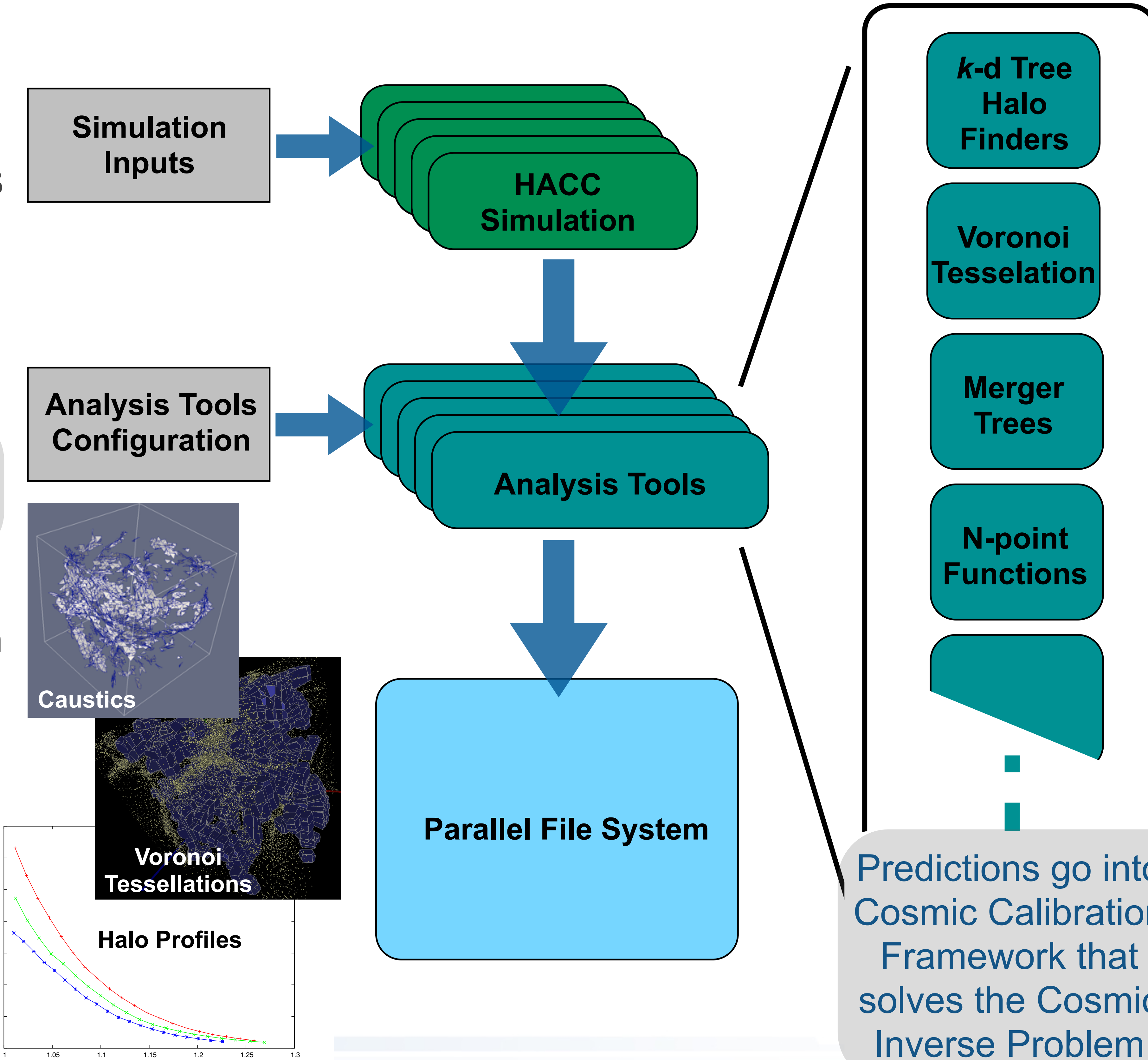
- **Project Science (will continue through and beyond 2017):**
  - **Large-scale structure simulations**
  - **Multi-physics simulations at smaller scales**
  - **Cosmic probes, mock catalogs**
  - **Fast prediction for forward models, uncertainty quantification**
  - **Support HEP Cosmic Frontier -- BOSS, DES, SPT, BigBOSS/DESpec, LSST projects**
  - **Large data analytics; in situ and post-processing**
- **Where Do We Expect to be in 2017:**
  - **Large-scale structure simulations in the 30+ trillion particle class**
  - **Sophisticated particle-based and AMR hydro simulations for galaxy formation at increased volumes**
  - **Very high degree of realism and fidelity in ‘mock skys’ for systematics control**
  - **Precision ‘emulators’ at the <1% error range**
  - **Extensive use of large-scale data analytics with 10+ PB**

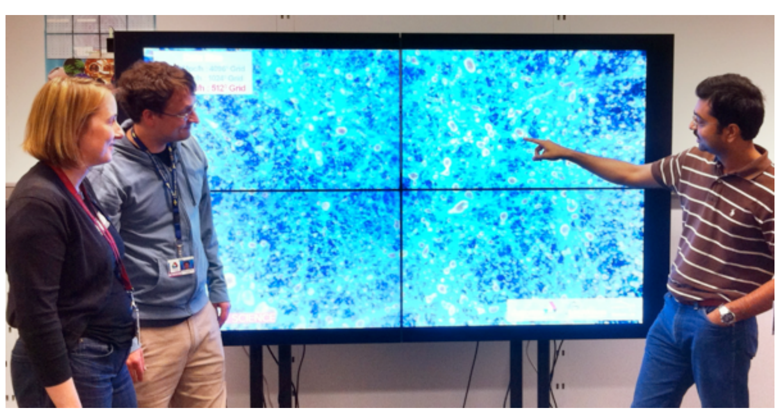


1.1 trillion particle HACC science run at z = 3 on Mira illustrating the dynamic range of a large, high-resolution, cosmological N-body simulation

HACC Example: *Fast* In Situ Analysis

- **Data Reduction:** A trillion particle simulation with 100 analysis steps has a storage requirement of ~4 PB -- in situ analysis reduces it to ~200 TB
- **I/O Chokepoints:** Large data analyses difficult because I/O time > analysis time, plus scheduling overhead
- **Fast Algorithms:** Analysis time is only a fraction of a full simulation timestep
- **Ease of Workflow:** Large analyses difficult to manage in post-processing

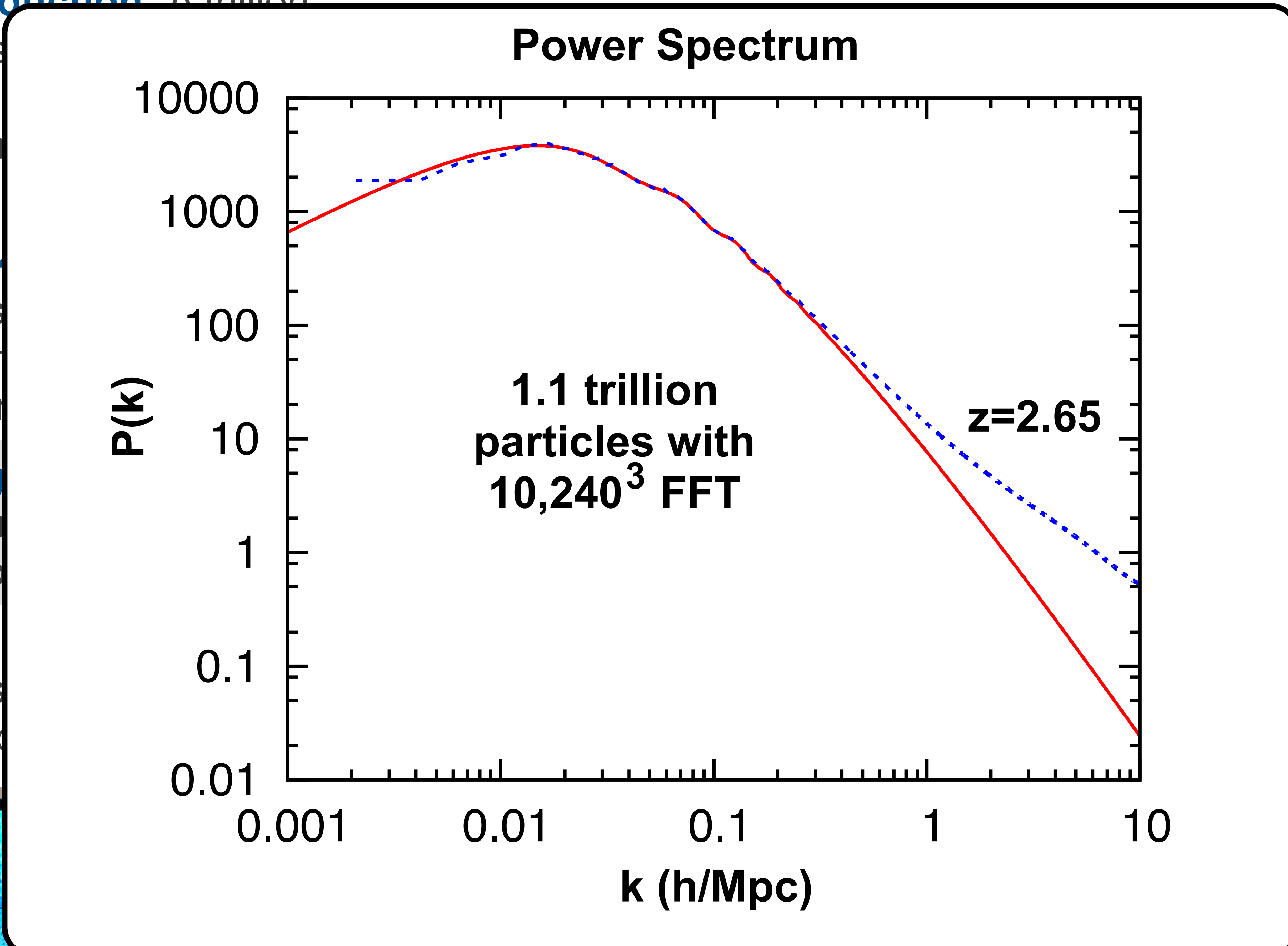


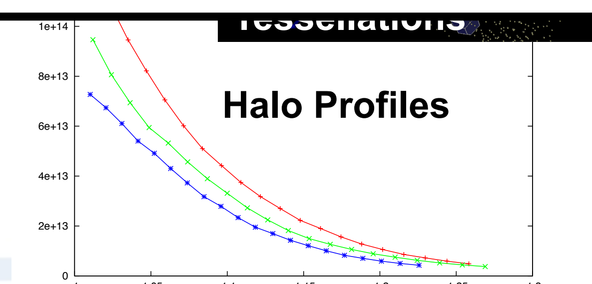


Predictions go into Cosmic Calibration Framework that solves the Cosmic Inverse Problem

HACC Example: *Fast* In Situ Analysis

- **Data Reduction:** A trillion particle s analysis requirem analysis
- **I/O Chok** analyses time > ar schedulir
- **Fast Alg** time is or simulatio
- **Ease of** analyses post-proc





***k*-d Tree Halo Finders**

Voronoi Tesselation

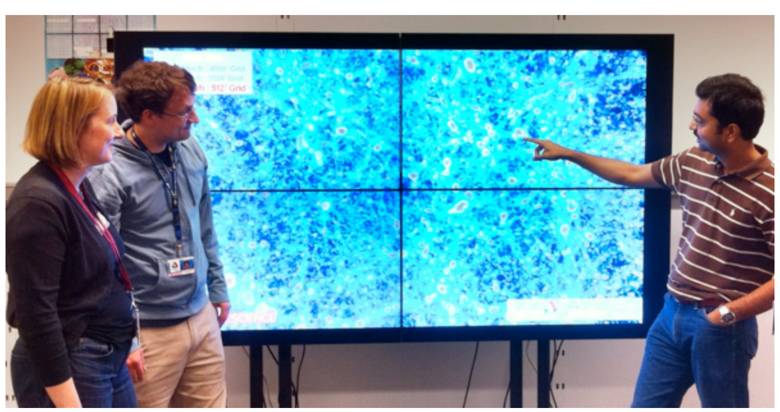
Merger Trees

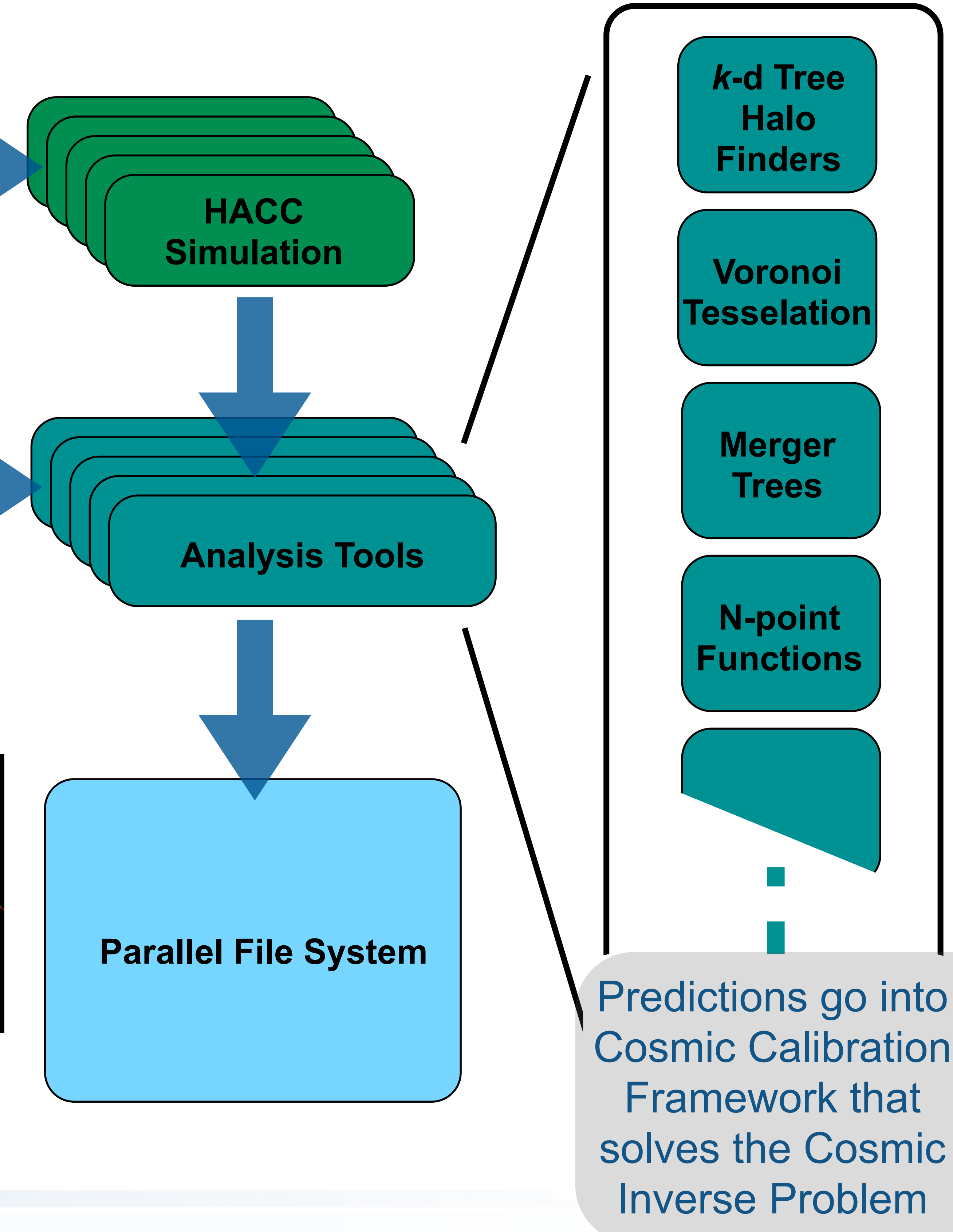
N-point Functions

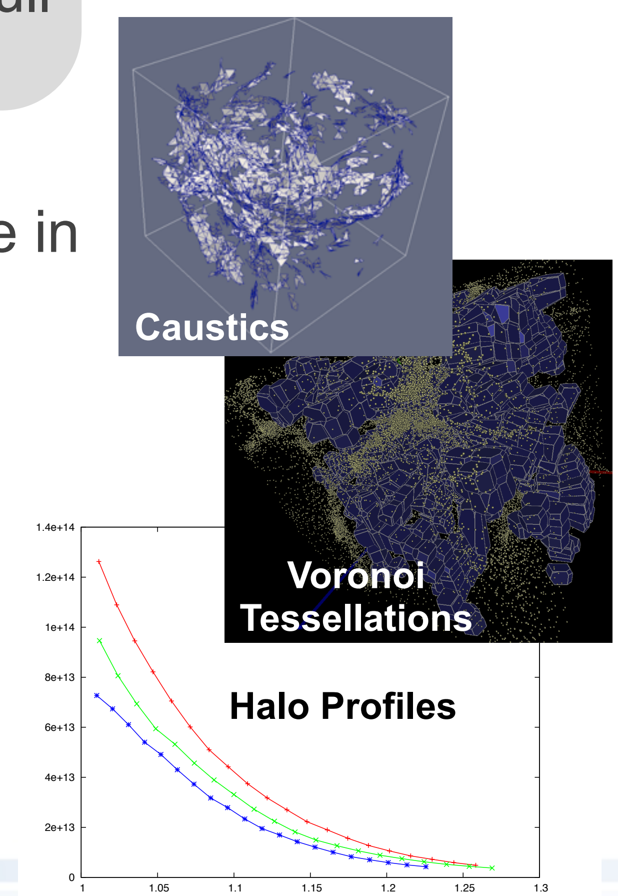
Predictions go into Cosmic Calibration Framework that solves the Cosmic Inverse Problem

HACC Example: *Fast* In Situ Analysis

- **Data Reduction:** A trillion particle simulation with 100 analysis steps has a storage requirement of ~4 PB -- in situ analysis reduces it to ~200 TB
- **I/O Chokepoints:** Large data analyses difficult because I/O time > analysis time, plus scheduling overhead
- **Fast Algorithms:** Analysis time is only a fraction of a full simulation timestep
- **Ease of Workflow:** Large analyses difficult to manage in post-processing

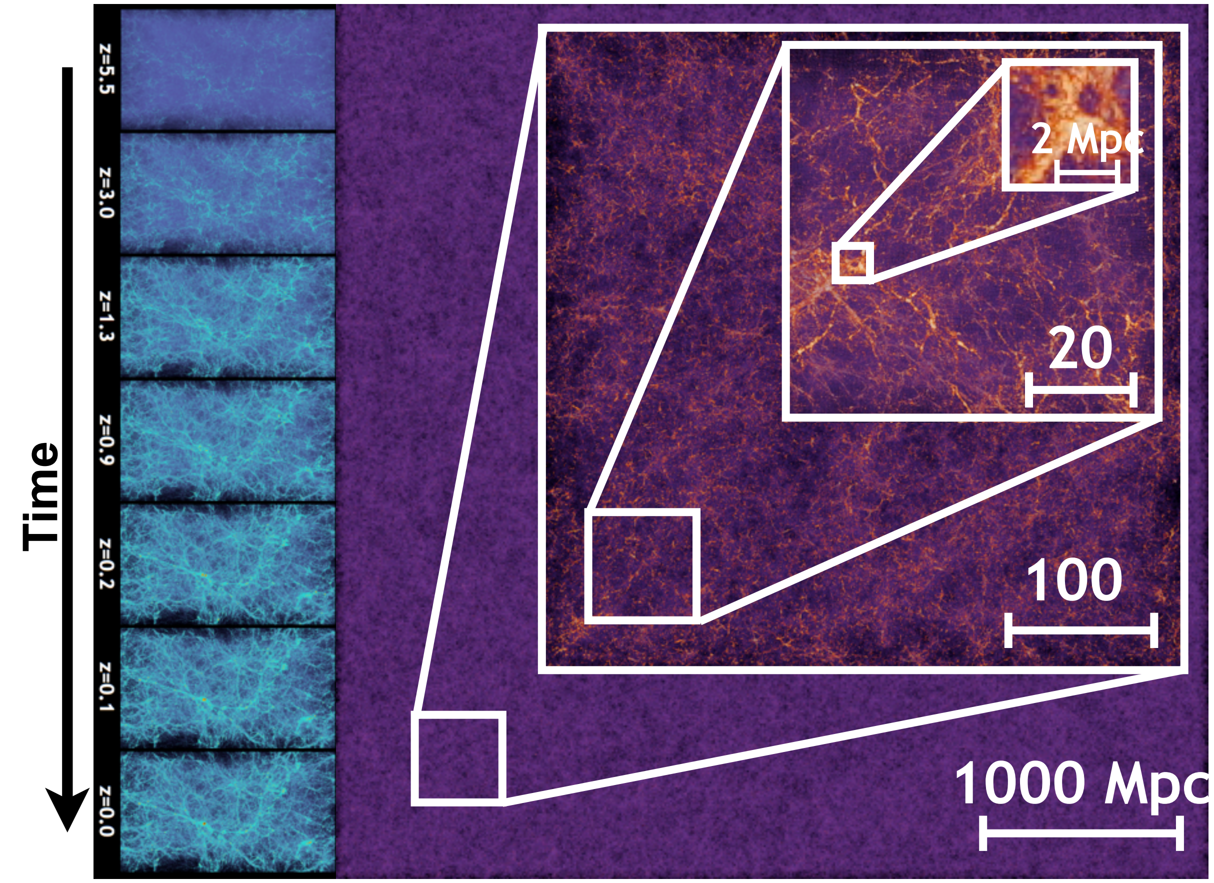






Computational Cosmology: Not One Problem, But Many!

- **Structure Formation:**
  - Initial conditions -- non-Gaussian, multi-scale, baryons, GR effects in large boxes, etc.
  - Gravity-only simulations (N-body) require dynamic ranges of > million-to-one
  - Gasdynamics simulations at smaller scales
- **Physics:**
  - Gravity dominates at scales greater than ~Mpc
  - At small scales: galaxy distribution modeling, sub-grid models, full hydro too difficult
- **Algorithms:**
  - N-Body: Melds of direct particle-particle, tree, and particle-mesh algorithms (including AMR)
  - Hydro: AMR, SPH, and variations
- **Computational Challenges/Scaling Limitations:**
  - Complex data structures
  - Large memory/core requirements
  - Inherent limitations on Flops/Bytes ratio
  - Analytics (in situ or post-processing)



Gravitational Jeans Instablity

- **Changes in Approach:**
  - Code algorithmic structure (e.g., HACC for N-body simulations)
  - Revamped data structures
  - New algorithms with higher Flops/Bytes ratio
  - However, not everything will change (long duration software cycles, especially for community codes)

Primary Codes

- **N-Body:**
  - Solvers: Poisson solver for the PM component; direct particle-particle, tree, and multipole methods for the short-range solvers
  - Gadget -- TreePM, public domain, primarily written by Volker Springel, currently scales to 1000's of MPI ranks, available in MPI and MPI/OpenMP versions (can be run to hundreds of billions of particles)
  - HACC (Hardware/Hybrid Accelerated Cosmology Codes) Framework -- PPTreePM, primary development team at Argonne, supports a number of programming models (MPI, MPI/OpenMP, MPI/OpenCL, MPI/Cell_SDK, --), arbitrarily scalable (~exascale design point), has been run on > 1.5M cores/MPI ranks at 14 PFlops with 3.6 trillion particles; typical runs in the tens of billions to multi-trillions of particles (memory bound)
- **AMR Hydro:**
  - Solvers: Poisson solvers on an AMR mesh (relaxation/Multigrid); Euler solvers on AMR meshes, various rad. transfer algorithms, local methods for feedback, cooling, star formation, etc.
  - ART -- cell-structured (refinement trees) AMR code for gravity + gasdynamics + feedback + --, primary development at Fermilab and UChicago, currently scales to ~10K cores (work is underway to improve this); can run up to billions of particles with 2000^3 AMR meshes
  - Gadget -- Adds SPH hydro to TreePM solver
  - Nyx -- new block-structured (nested hierarchy of rectangular grids) AMR code for gravity + gasdynamics + rad. transfer + feedback + --, based on BoxLib framework, primary development at LBNL, supports MPI and MPI/OpenMP; weak scaling verified to 200K processors, currently capability extends to running up to tens of billions of particles with 4000^3 AMR meshes

Current HPC Usage

- **Facilities and Machines:**
  - Facilities -- ALCF, NERSC, OLCF, NSF Centers
  - Architectures -- IBM BG/P and BG/Q, Cray XT, IBM iDataPlex, Cray XK7, Cray XE6, various Linux clusters, CPU/GPU clusters
- **Usage:**
  - Total computational hours per year (2012, est.) -- ~75M core-hours (50M@ALCF, 20M@NERSC, 5M at other places, will use more as new systems come on line)
  - Number of cores in a typical production run -- thousands to hundreds of thousands
  - Wall clock for a single production run -- ~days to two weeks
  - Minimum memory required per core -- depends on the code, 1 GB to 4 GB
  - Data read and written per run -- ranges from ~1 TB to 100's of TB
  - Size of checkpoint files -- ranges from tens of GB to ~100 TB
  - Amount of data moved in/out of NERSC -- 10's to 100's of TB
  - On-line file storage requirement -- 300 TB currently (on data-intensive computing project at NERSC, have ~50 TB in place, more coming!)
  - Off-line archival storage requirement -- in general, not thrilled with HPSS (example bottleneck 1)
  - Example bottleneck 2: Would like faster I/O to the Global File System from Hopper

HPC Requirements for 2017(!)

- **Compute "hours" needed:**
  - In 2013, our "project" has somewhere in the realm of 200M core-hours, historically science needs are inexhaustible -- especially since in our case we have to chase down a large number of science issues -- no doubt there will be surprises
  - Extrapolating from the 2012/2013 experience and knowing that the present installed supercomputing base at major centers will be more or less stable (within a factor of a few) until the next major jump (~2015/16?), by 2017, we could be looking at multi-billion to 10-billion core-hours (in "Hopper" units)
- **Usage patterns for 2017:**
  - Parallel concurrency changes are only to be expected; next-generation architectures such as the Intel Xeon Phi are already one order of magnitude shy of memory provisioning at fixed performance compared even to a BG/Q, so average concurrency for memory-bound codes (typical of cosmology) will likely increase by factors in the 10's to 100 (or people will just give up)
  - Run times in terms of wall clock will never be longer than ~month; beyond this, doing science becomes hopeless; number of runs per year will go up as simulations become ever more important to extracting science from observations, perhaps a factor of 10 or more than current practice
  - Data read/written will likely depend on (i) simulation sizes, (ii) extent of in situ analysis and compression, (iii) external factors such as I/O bandwidth, size/stability of filesystems -- is likely to go up by a factor of 10-100
  - Minimum memory requirements -- globally, one expects something in the low 10's of PB, which would, at 100 million-way concurrency, translate to ~100 MB/"core"; this is fine for HACC but not so much for AMR codes, may need (equivalent of) ~GB/"core"
  - On-line file storage requirement -- 10's of PB, but in "active" storage (off-line, hopeless)

Strategies for New Architectures

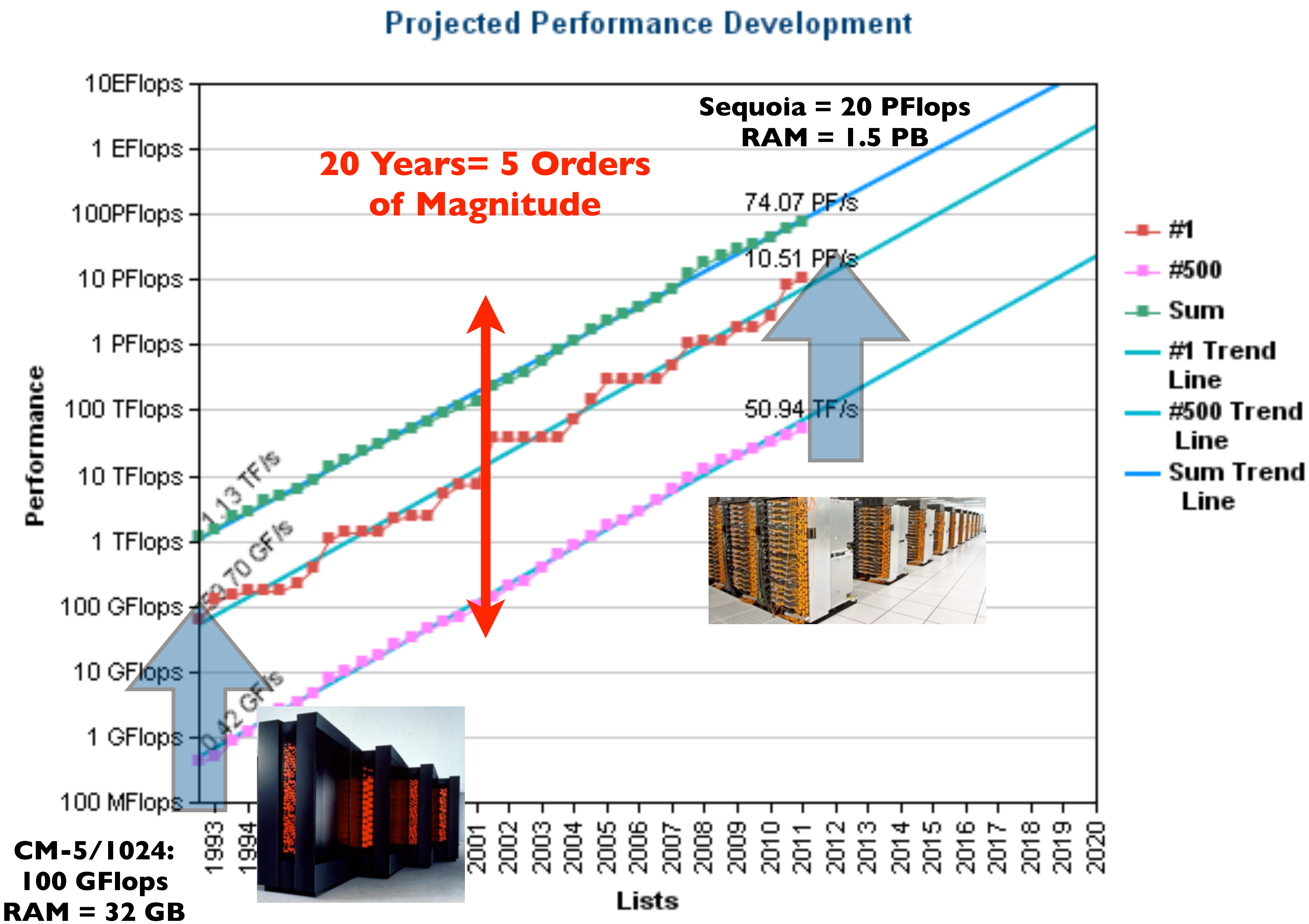
- **N-Body:**
  - Ready now: HACC runs on CPU/GPU, SoC, CPU/MIC already, based on a multi-algorithmic design targeted to future architectures (see arXiv:1211.4864 [cs.CE]), we don't expect major changes to the framework to get ready for 2017
- **AMR Codes:**
  - For AMR codes this is likely to be a research project which will not be done by 2017
  - Send computation-heavy pieces to accelerator (atomic physics, radiative transfer)
  - For many-core systems, the small memory/core and cost of nonlocal memory access within and across nodes will be a major problem -- needs to be addressed
  - Possible intermediate solution to get something to run and be more portable -- use of directives (pragmas) as in OpenACC, good place for NERSC to provide expertise, collect early science project teams
  - DSLs unlikely to be available by 2017 in production form
- **Data Analytics:**
  - A major challenge will be to rewrite a large set of analysis routines for new architectures, especially if there is an ecology of different architectures
  - We are highly competent-people-limited! Help in this area would be very useful --

Summary I

- **New Science:**
  - First, not so much new science as much as planned science -- improvements in NERSC services will be required for supporting the science of ongoing and next-generation cosmological surveys (BOSS, DES, BigBOSS/DESpec, LSST)
  - Second, our project has already planned future work based on expected improvements in computational capabilities at NERSC and other centers
  - Key science results relate to dark energy, dark matter properties, early Universe physics, neutrino mass constraints, and perhaps, new surprises --
- **Recommendations:**
  - Hard to tell what will be the “conservative option” in 2017 for NERSC to follow, there may not be one! (as there wasn’t in the mid-90’s)
  - Most important -- work to to keep the NERSC userbase informed of coming changes well in advance; perhaps availability of early prototype systems, help with code “ports”
  - Data analytics and supercomputing are likely to be intertwined by 2017, NERSC may want to be one of the leaders in this
  - Maintain continuous interaction with applications teams, many of us know what we are doing ;-)

Summary II

- **Expanded Resources:**
  - A factor of 30 is significant, but far from outrageous (in terms of a 3D problem size it's only a factor of ~3); some science cases will be unaffected by this increase, just as they have not been affected by many orders of magnitude in the past --
  - Some science cases will no doubt be affected -- we will be!



- **How?**
  - Need to expand significantly in the "supercomputing meets big data" direction
  - A "data analysis cloud" or equivalent, with dynamically allocatable resources would be a very useful complement to the supercomputer
  - Help codes with complex data structures and low Flops/ Bytes figure out strategies for next generation architectures
  - Key issue will be dealing with a number of programming models unless one clear winner emerges (highly unlikey?) -- help userbase with this