Cosmological Simulations on Large, Heterogeneous Supercomputers

Adrian Pope (LANL)

Future of AstroComputing, SDSC

December 17, 2010

People

- LANL: Jim Ahrens, Lee Ankeny, Suman Bhattacharya, David Daniel, Pat Fasel, Salman Habib, Katrin Heitmann, Zarija Lukic, Pat McCormick, Jon Woodring
- ORNL: Chung-Hsing Hsu
- Berkeley: Jordan Carlson, Martin White
- Virginia Tech: Paul Sathre
- IBM: Mike Perks
- Aerospace: Nehal Desai

HACC (Hardware Accelerated Cosmology Code)

- **Precise theoretical predictions to match (future) survey observations**
  - Large-scale structure, weak lensing, clusters
- **Throughput**
  - Dynamic range
    - Volume for long wavelength modes
    - Resolution for halos/galaxy locations
  - Repeat runs
    - Vary initial conditions
    - Sample parameter space, emulators for observables (Coyote Universe, Cosmic Calibration)
  - (At least) weak scaling to current and future large supercomputers (many MPI ranks, many cores)
- **On-the-fly analysis, data reduction**
  - Reduce size/number of outputs, ease file system stress
- **Flexibility**
  - Applications often do not choose supercomputer architecture (CPU,Cell, GPGPU, BG)
  - Compute intensive code takes advantage of hardware
  - Bulk of code easily portable (MPI)
- **Development/maintenance**
  - Few developers
  - Simpler code easier to develop, maintain, and port to different architectures

Collisionless Gravity

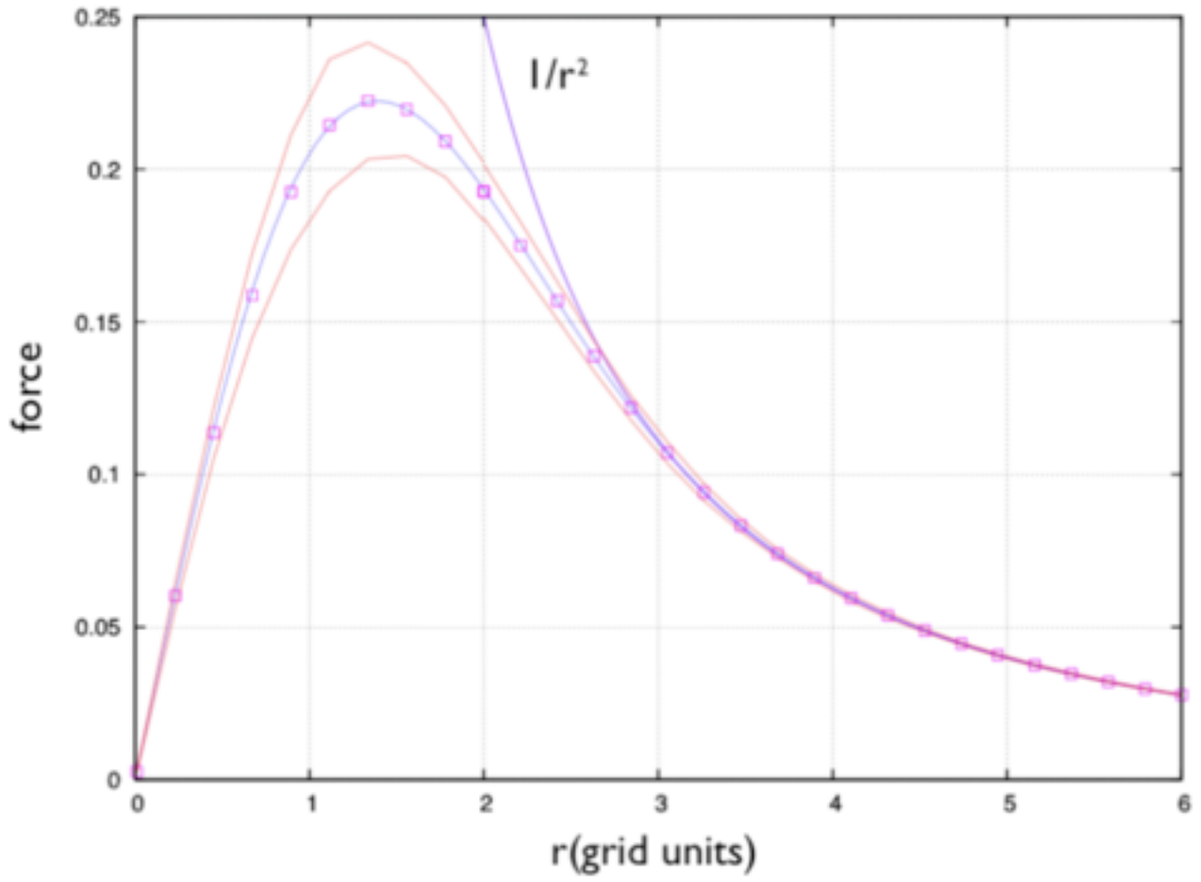
$$\frac{\partial f}{\partial t} + \dot{\mathbf{x}} \cdot \frac{\partial f}{\partial \mathbf{x}} - \nabla\phi \cdot \frac{\partial f}{\partial \mathbf{p}} = 0, \qquad \mathbf{p} = a^2 \dot{\mathbf{x}}$$

$$\nabla^2 \phi = 4\pi G a^2 (\rho(\mathbf{x}, t) - \rho_b(t)), \quad \rho(\mathbf{x}, t) = a^{-3} m \int d^3\mathbf{p} f(\mathbf{x}, \dot{\mathbf{x}}, t)$$

- **Evolution of over-density perturbations in smooth, expanding background (Vlasov-Poisson)**
- **Gravity has infinite extent and causes instabilities on all scales**
- **N-Body**
  - Tracer particles for phase-space distribution
  - Self-consistent force
  - Symplectic integrator

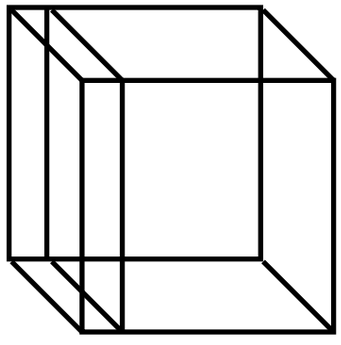
Force

- **Long-range (PM = particle-mesh)**
  - Deposit particles on grid (CIC)
  - Distributed memory FFT (Poisson)
  - Pros: fast, good error control
  - Cons: uses memory
- **Short-range**
  - Inter-particle force calculation
  - Several short steps per long step
  - Limited spatial extent
    - Local n^2 comparisons
  - Several choices for implementations
    - Direct particle-particle (P^3M: Cell, OpenCL)
    - Tree solver (TreePM: CPU in development)
- **Spectral smoothing at handover**
  - More flexible than real-space stencils (eg. TSC)

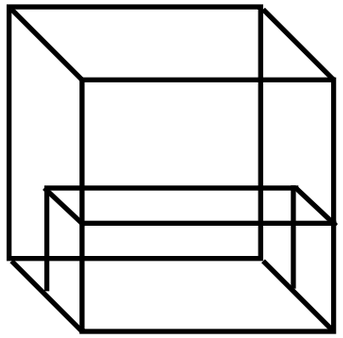


FFT Decomposition

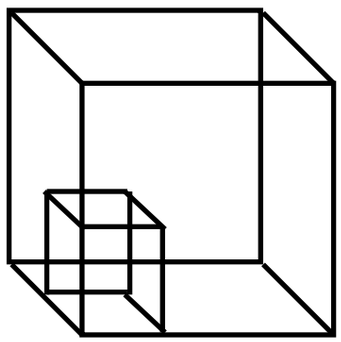
- **Compute:Communication::Volume:Area**
- **Independent of particle decomposition**
  - Buffers to re-arrange
- **Roadrunner 1D tests**
  - (Weak) scaling up to 9000^3, up to 6000 MPI ranks
  - Probably about as far as 1D will go (thin slabs)
- **Analysis: 2D should work for likely exascale systems**
  - 2D FFT is under testing
- **Not as critical to calculate on accelerated hardware**
  - Network bandwidth limited
  - Still relatively fast and accurate force calculation



1D slab



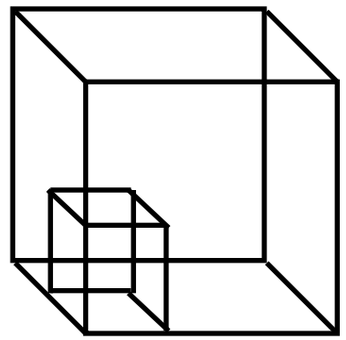
2D pencil

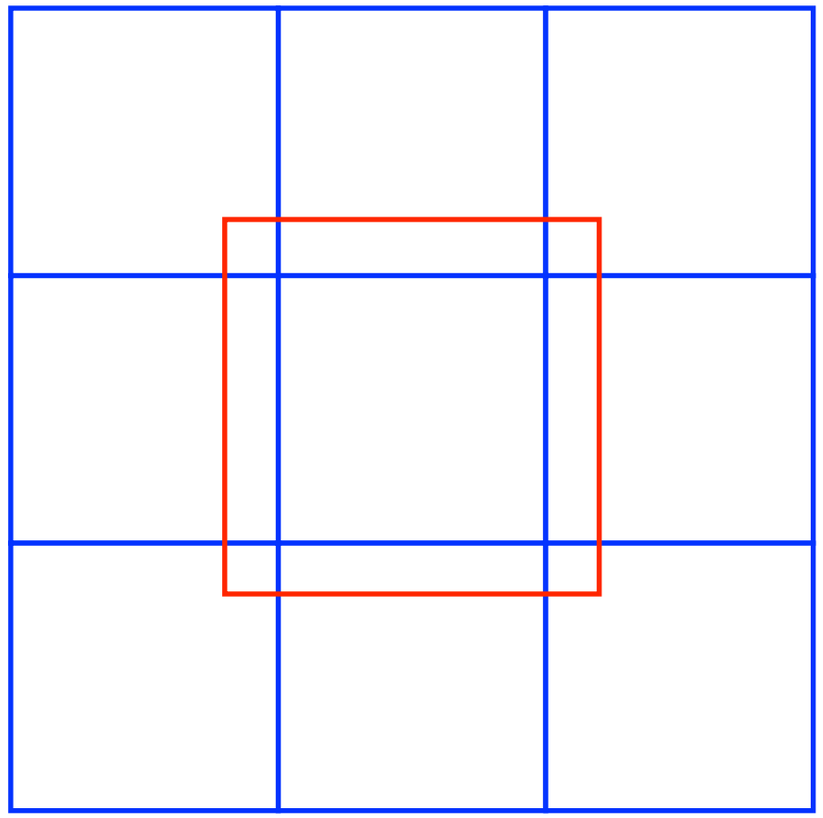


3D

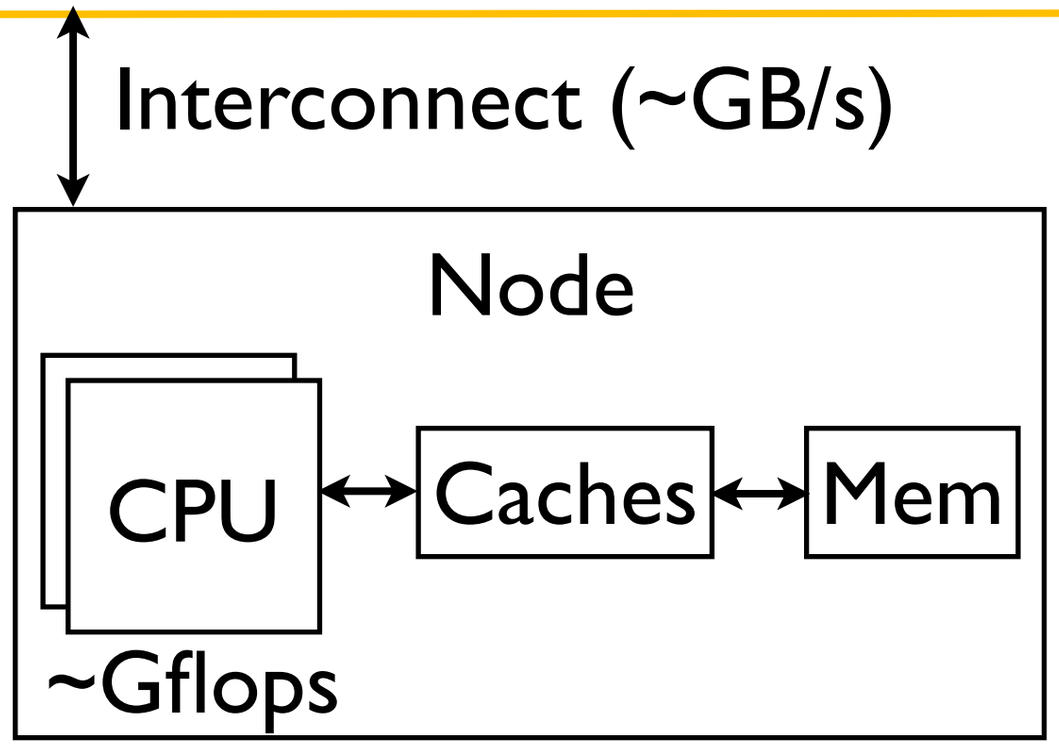
Particle Overloading

- **3D spatial decomposition (max volume:area)**
  - Large-scale homogeneity = load balancing
- **Cache nearby particles from neighbors**
- **Update cached particles like others**
  - Move in/out of sub-volumes
  - Skip short-range update at very edge to avoid anisotropy
- **Can refresh cache (error diffusion)**
  - Not every (long) time step
- **Network communication**
  - Mostly via FFT
  - Occasional neighbor communication
  - None during short-range force calculation
- **Serial code development for short-range force**

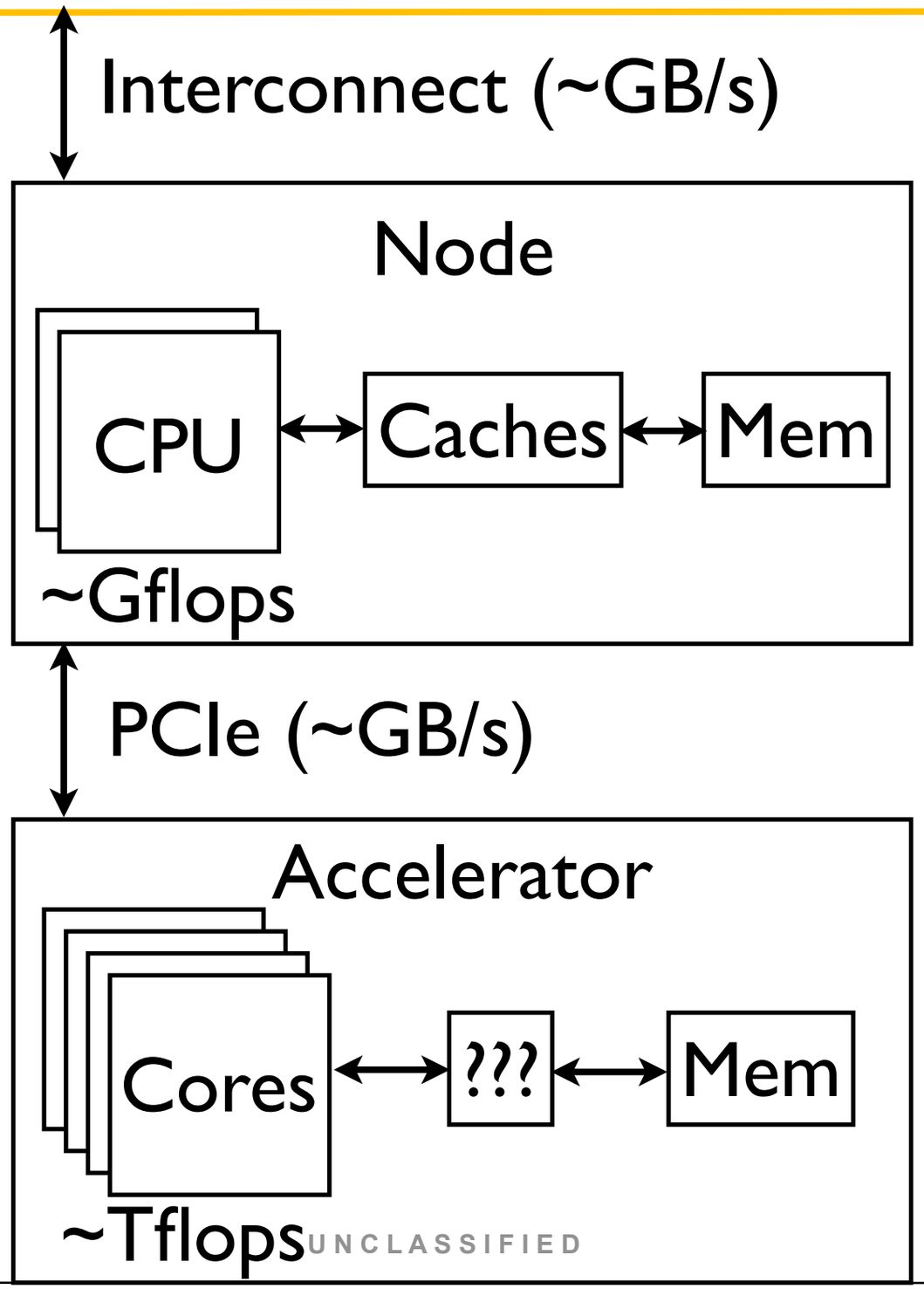




Architecture



Architecture

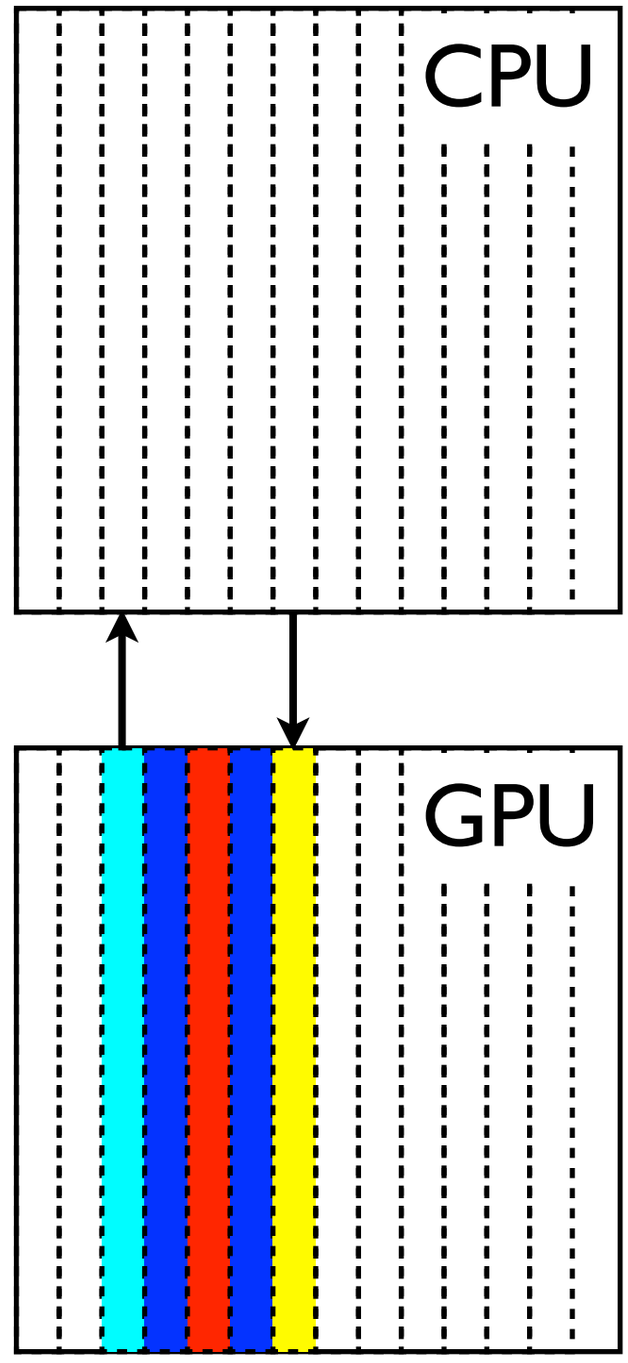


Modular Code

- **Decomposition and communication is independent of hardware (MPI)**
- **Particles class**
  - Particle/grid deposit for long-range force (CIC, CIC^{-1})
  - Particle position update (easy)
  - Short-range force, velocity update (bottleneck)
    - Use methods/datastructures to suite hardware
  - Fixed set of public methods

Accelerators

- **P³M**
  - Simpler code development
  - Exact calculation can be a reference for approximate methods
  - Chaining mesh: sort particles into buckets, ~force-length
- **Organize into independent work units**
  - Cell: concurrent scheduling by hand
  - OpenCL: data-parallel kernel execution
- **Memory hierarchy, coherence**
  - Asynchronous transfers
    - Overlap movement and computation (no “if”s)
    - No competing writes to memory
  - Cell
    - Balanced memory between CPU and Cell
    - Particles in Cell main memory, grid info over PCI
  - OpenCL
    - Possibly (probably?) unbalanced memory
    - Stream slabs through GPU memory



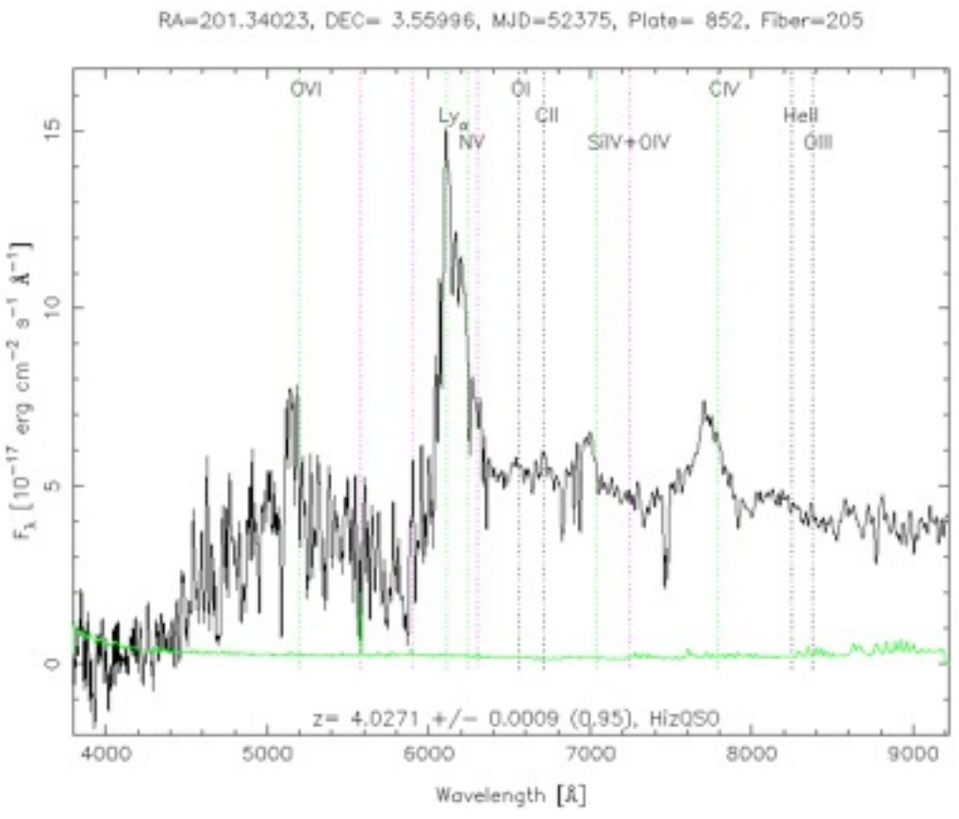
PM Science

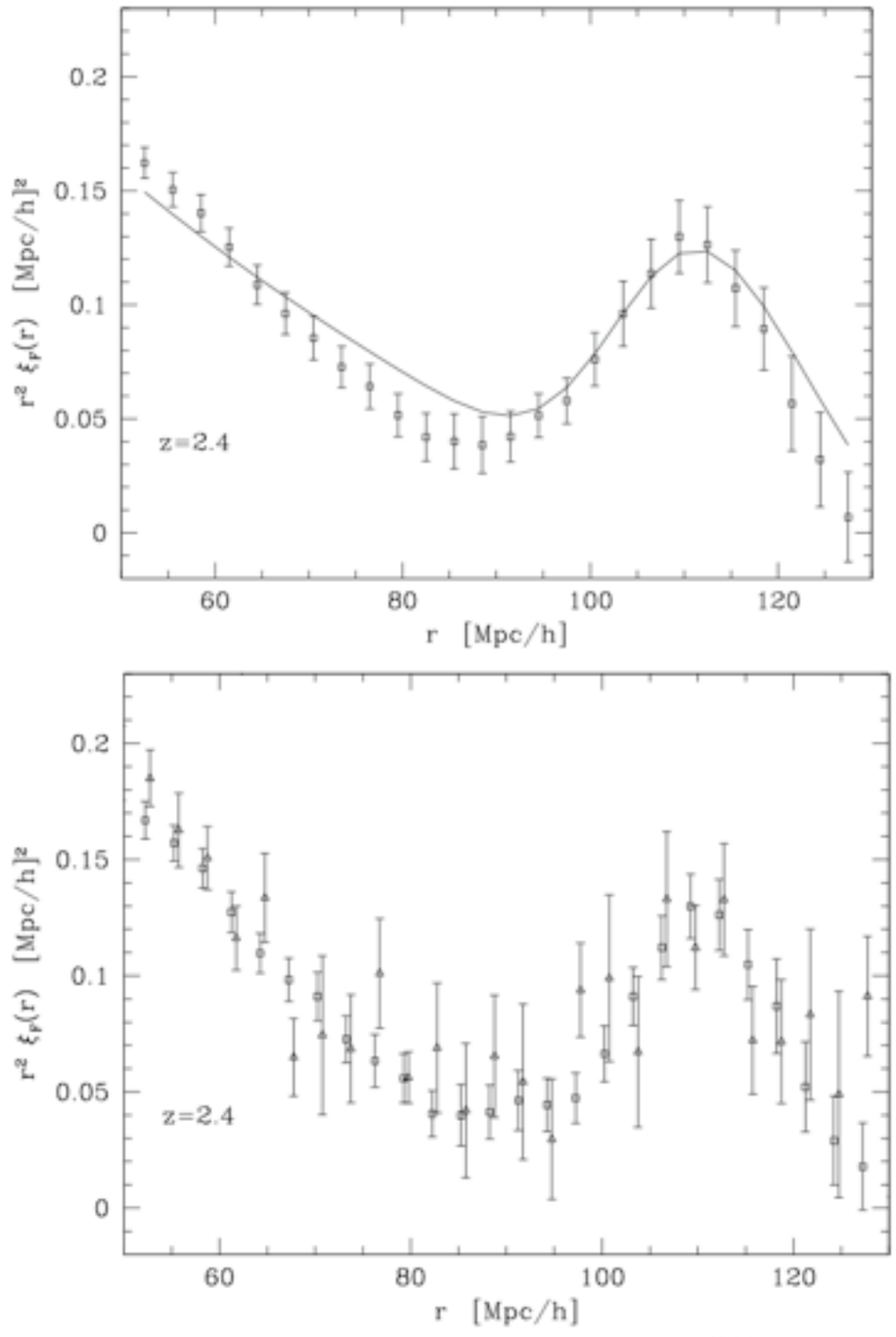
First Roadrunner Universe (RRU) Science Runs

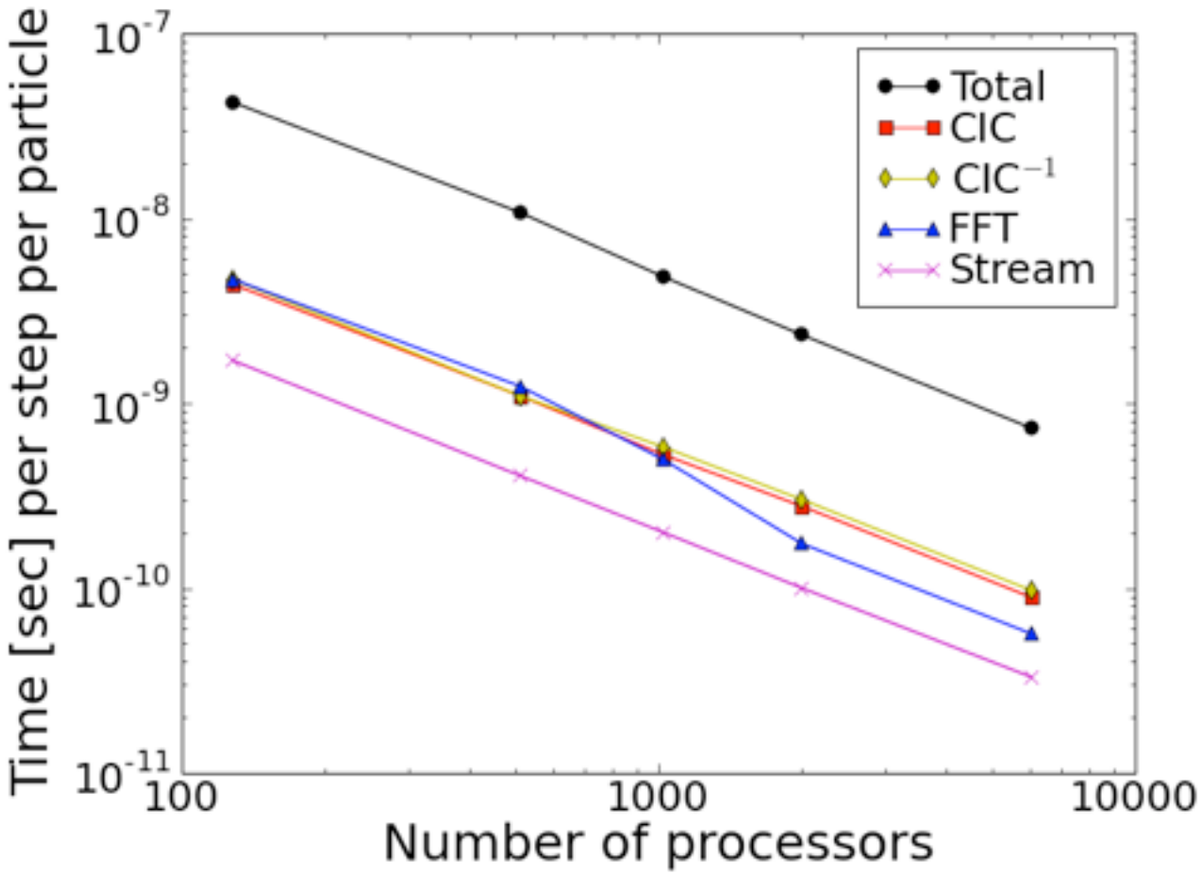
- **Roadrunner (LANL)**
  - 3060 nodes
    - 2x dual core Opterons, 10% flops
    - 4x Cell, 90% flops (8 vector processors per Cell)
  - 1 petaflops double precision, 2 petaflops single precision
- **Simulation parameters**
  - 750 Mpc/h side length
  - 64 billion particles (resolve IGM Jeans mass)
  - ~100 kpc/h (resolve IGM Jeans length)
  - 9 realizations (single cosmology)
  - 1000 nodes (1/3)
  - ~Day per simulation
- **Analysis**
  - Density along “skewers” calculated on-the-fly
  - Cross-correlations along nearby lines-of-sight in post processing

Ly-α BAO studies

- **BOSS: Cross-correlation along pairs of QSO**
- **DM only simulation, some gas physics in post processing**
- **Can test noise/error scenarios**
- **White et al. 2010 (ApJ, arXiv:0911.5341)**





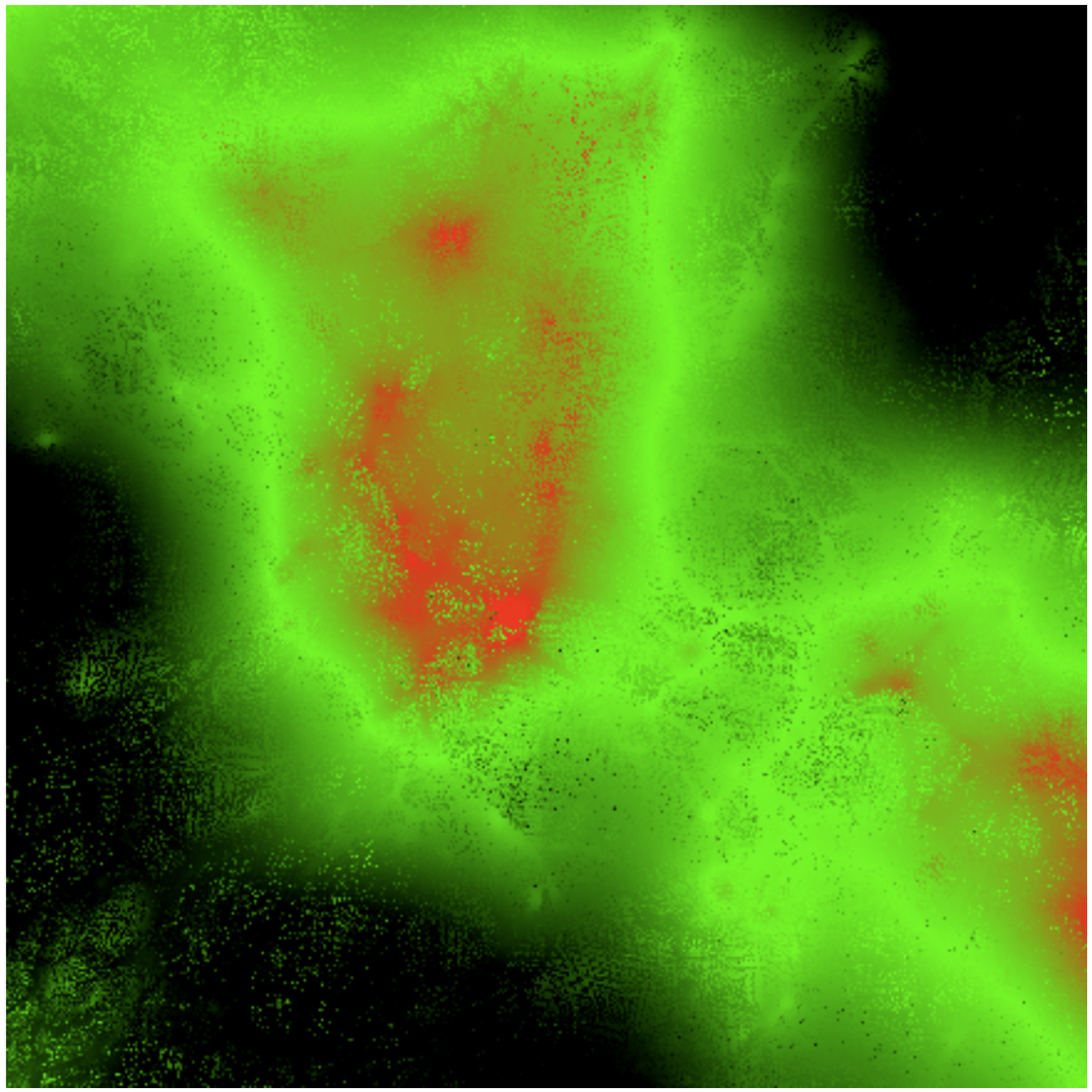



```
heitmann@rr-fe4:2PPN_6048_NEW - Shell - Konsole <2>
Session  Edit  View  Bookmarks  Settings  Help

Initializer will use 6048 processors.
6048^3 grid
Decomposing into slabs..............done

sigma_8 = 0.800000, target was 0.800000
redshift: 211.000000; growth factor = 0.006321; derivative = 9.755618


Min and max value of density in k space: -306.01 394.242
Average value of density in k space: 4.91044e-07
[heitmann@rr-fe4 2PPN_6048_NEW]$ showq

active jobs------------------------
JOBID              USERNAME      STATE PROCS   REMAINING            STARTTIME

17231              heitmann    Running 12096     7:02:48  Thu Sep 24 15:27:59

1 active job          12096 of 12112 processors in use by local jobs (99.87%)
                      3021 of 3028 nodes active      (99.77%)

eligible jobs----------------------
JOBID              USERNAME      STATE PROCS     WCLIMIT            QUEUETIME


0 eligible jobs

blocked jobs-----------------------
JOBID              USERNAME      STATE PROCS     WCLIMIT            QUEUETIME


0 blocked jobs

Total job:  1

[heitmann@rr-fe4 2PPN_6048_NEW]$ showq

active jobs------------------------
JOBID              USERNAME      STATE PROCS   REMAINING            STARTTIME

17231              heitmann    Running 12096     7:01:07  Thu Sep 24 15:27:59

1 active job          12096 of 12112 processors in use by local jobs (99.87%)
                      3021 of 3028 nodes active      (99.77%)

eligible jobs----------------------
JOBID              USERNAME      STATE PROCS     WCLIMIT            QUEUETIME


0 eligible jobs

blocked jobs-----------------------
JOBID              USERNAME      STATE PROCS     WCLIMIT            QUEUETIME


0 blocked jobs

Total job:  1

[heitmann@rr-fe4 2PPN_6048_NEW]$
Shell
```

P³M Commissioning

Cell

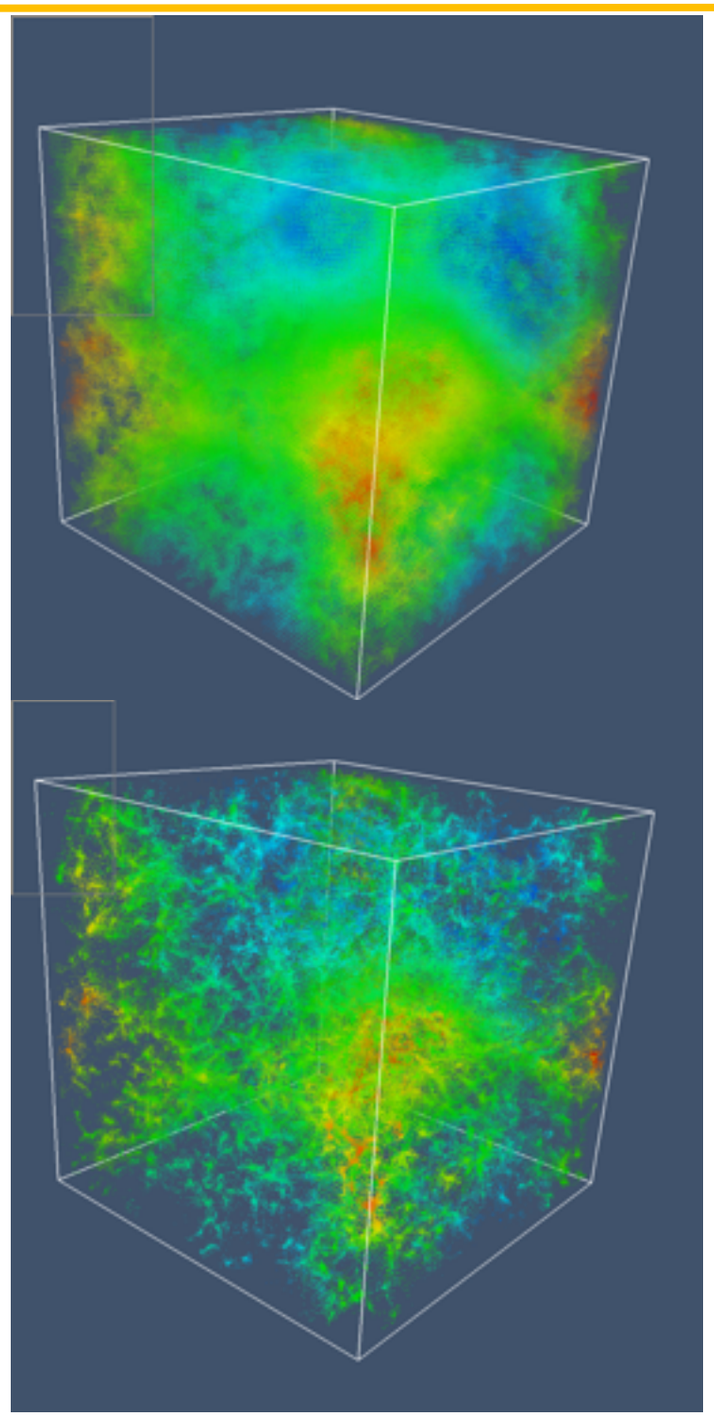
- **Code comparison (256^3) < 1% agreement**
- **Tests at scale**
  - Roadrunner
    - 4 Gpc/h side length
    - 64 billion particles
    - 1000 nodes (1/3)
  - Cerrillos (360 nodes, open network)
    - 2 Gpc/h side length
    - 8 billion particles
    - 128 nodes
  - Both
    - ~5-10 kpc/h force resolution
    - 500x3 time steps
    - ~Week (+queue)
  - Verifying results



1/512 of 8 billion particle run

OpenCL

- **Initial port of Particles class by summer student**
  - Quicker/easier development than Cell (data transfers)
- **SC10 demo**
  - Calculation in real time (small problem size)
  - Mix of NVIDIA and ATI hardware
  - Interactive 3D visualization in real time
- **Initial performance not awful**
  - Fast on NVIDIA
  - ATI needs improvement (we have some ideas)
- **Kernels**
  - Single kernel with optional vectorization?
  - Tune kernels for each hardware?
  - Settle datastructures



Future

- **Cell**
  - Debugging speed improvements (3x faster)
  - Clean up code from beta to 1.0
- **OpenCL**
  - Improve code from demo to production
  - Should soon have access to a machine large enough for real tests
- **Local tree solver (CPU)**
  - Data structures in place (threaded tree)
  - Need to implement force solve walk
- **Early Science Program on Argonne BG/Q**
  - OpenMP thread some operations (planning)
  - P^3M? Tree? Both?
- **Baryon physics**
  - Exploring methods