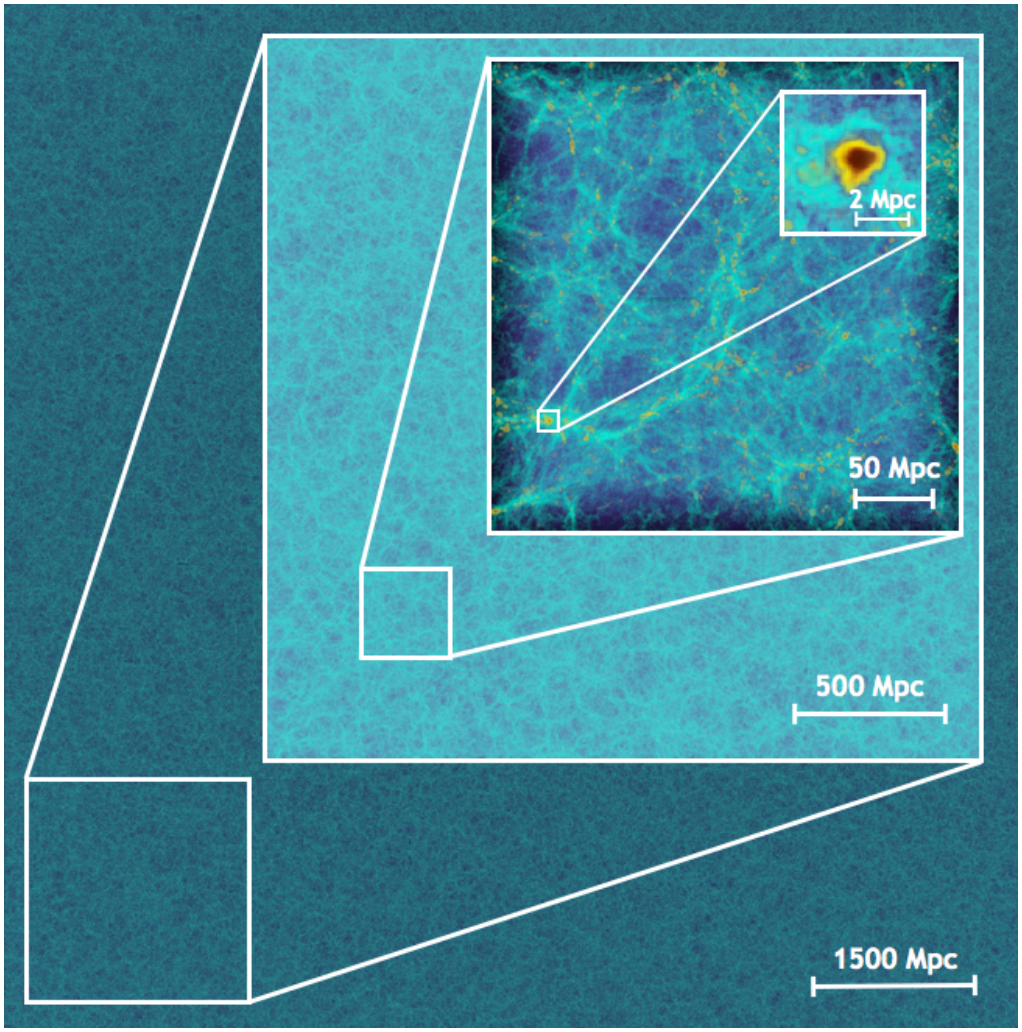


1.1 trillion particle test simulation carried out on 1/3 of Mira

HACC: Hardware Accelerated Cosmology Code Framework

Katrin Heitmann
Argonne National Laboratory

Collaborators

Analysis Team

- Juliana, Suman, Sudeep, Sanghamitra, Joe, Jon, Nikhil, Martin, Joanne, Rahul, Amol, Eve, Uliana,...

HACC Development

- ANL: H. Finkel, N. Frontiere, S. Habib, K. Heitmann, V. Morozov, T. Peterka, A. Pope, V. Vishwanath
- LANL: J. Ahrens, D. Daniel, P. Fasel
- Other institutions: Z. Lukic (LBNL), S. Serish, W.-K. Liao (NW), P. Sathre (Virginia Tech), J. Takle (Rutgers), G. Zagaris (Kitware)

HACC Development Areas

- Code Developemt/Design/Testing
- Hardware-specific implementions (current focus BG/Q, GPU)
- In-situ analysis
- I/O

The HACC Story Begins ...

Andrew White Dec 7, 2007 + What if you had a petaflop/s

- **... with an email:** Los Alamos National Lab offers the opportunity to run open science projects on the fastest supercomputer in the world for the first six months of the machine's existence: **Roadrunner**
- **Roadrunner:** First machine to achieve petaflop performance via Cell-acceleration, CPU/Cell hybrid architecture (more details later)
- **The Challenges:**
  - The machine has a "~~crazy~~" architecture, requiring major code re-designs and rewrites (we ended up writing a brand **new code**)

    Andy White: "forward-looking"
  - Roadrunner probably **one of a kind**, code-design needs to be flexible and **portable** to other future architectures (Cerrillos now exists, an open, but smaller version of Roadrunner)
  - File systems on new machines are usually unstable and we would not have had much time to transfer the data off the machine; the file system was going to be wiped after open science period, hence the need for **on-the-fly analysis capabilities** to avoid writing to disk as much as possible
- **Cosmologists are poor -- so we took on the challenge!**
- **Outcome: MC³ (Mesh-based Cosmology Code on the Cell) which later morphed into HACC, N-body framework to simulate large-scale structure formation in the Universe**

The Story Continues … and makes it into “Die Süddeutsche”

sueddeutsche.de

Politik | Wirtschaft | Geld | Kultur | Sport | Leben | Karriere | München & Region | Bayern | M

Home > Digital Supercomputer - Rasend schnell

Supercomputer

Rasend schnell

Er ist der schnellste Rechner der Welt: der amerikanische Supercomputer "Roadrunner" hat die Petaflop-Grenze geknackt. Sein Job: die Simulation von Atombombenexplosionen.

Twittern 0 Empfehlen Senden +1 0

Ein Rechner der US-Regierung schafft erstmals mehr als eine Billiarde Operationen in der Sekunde (Petaflops) und ist damit nun der schnellste Computer der Welt. Das berichten das US-Energieministerium und der Hersteller IBM am Montag.

Der Computer namens Roadrunner wurde am Los Alamos National Laboratory (LANL) in New Mexico installiert. Er wird zuvorderst für die Forschung an US-Atomwaffen rechnen. Der neu konstruierte Roadrunner ist auf einen Schlag mehr als doppelt so schnell wie der bisherige Spitzenreiter der "Top 500"-Liste der Supercomputer.

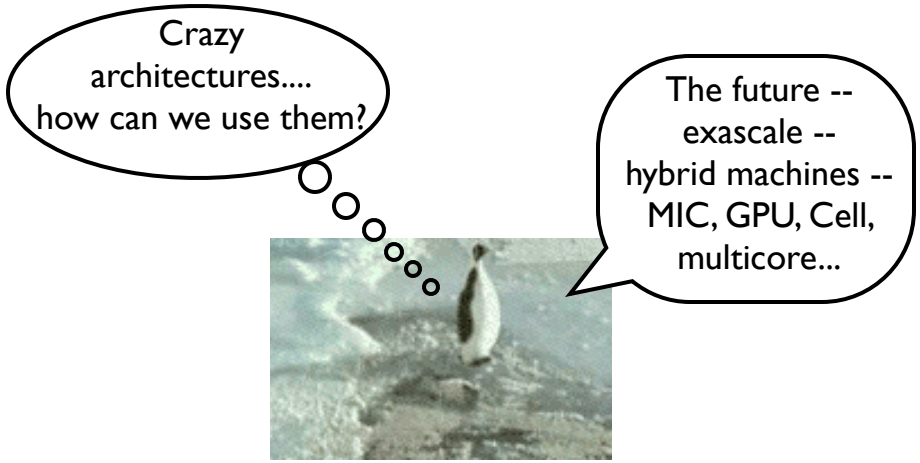
Anfangs soll der Roadrunner aber vor allem wissenschaftliche Probleme lösen. Beispielsweise sind Tests von Klimamodellen vorgesehen, doch rechnet das LANL mit Anwendungen in diversen Bereichen, darunter die Kosmologie, die Entwicklung von Antibiotika oder die Astrophysik. Danach wird der Supercomputer laut LANL militärischen Aufgaben zugeteilt und unter Geheimhaltung Explosionen nuklearer Waffen simulieren, um physikalische Modelle zu verbessern und das Vertrauen in das nukleare Arsenal der USA ohne tatsächliche Atomtests zu erhalten.

Supercomputer mit Vorbildfunktion

"Für uns und die HPC-Community ist es hoch erfreulich, dass es ein System gibt, das diese Marke geknackt hat", sagt Thomas Lippert, Leiter des Jülich Supercomputing Centre. Dadurch werde dem Supercomputing berechtigte Aufmerksamkeit zuteil.

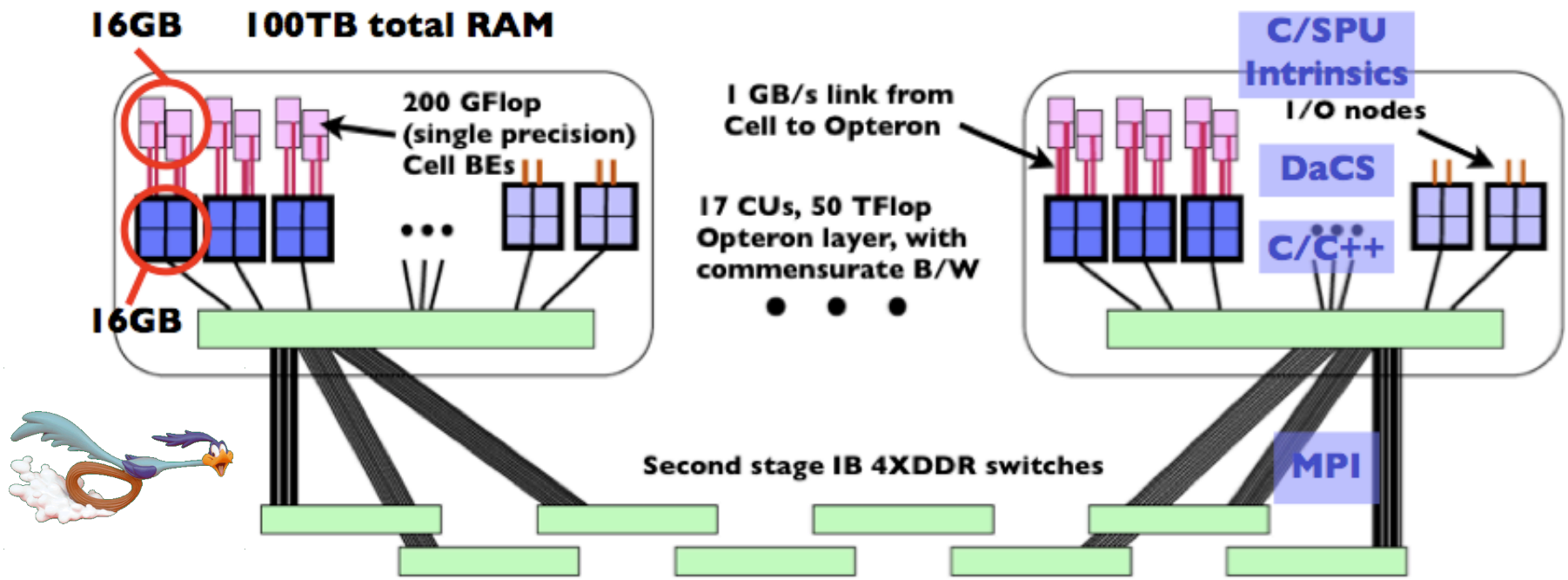
Technologisch dürfte Roadrunner Vorbildwirkung haben. "Es zeichnet sich ab, dass Hybrid-Technologie auf jeden Fall Zukunft haben", meint Lippert. Damit sind Systeme gemeint, die klassische CPUs mit Beschleunigern wie beispielsweise den Cell-Chips oder Grafikprozessoren kombinieren.

So we started thinking --



(from S. Furlanetto)

The Roadrunner Architecture



- **Opterons** have little compute but half the memory and balanced communication: For N-body codes, memory is the limiting factor, so want to make best use of CPU layer
- **Cells dominate** the compute but communication is poor, 50-100 times out of balance (also true for CPU/GPU hybrid systems)
- **Multi-layer programming model:** C/C++/MPI for Opterons, C/Cell-intrinsics for Cells

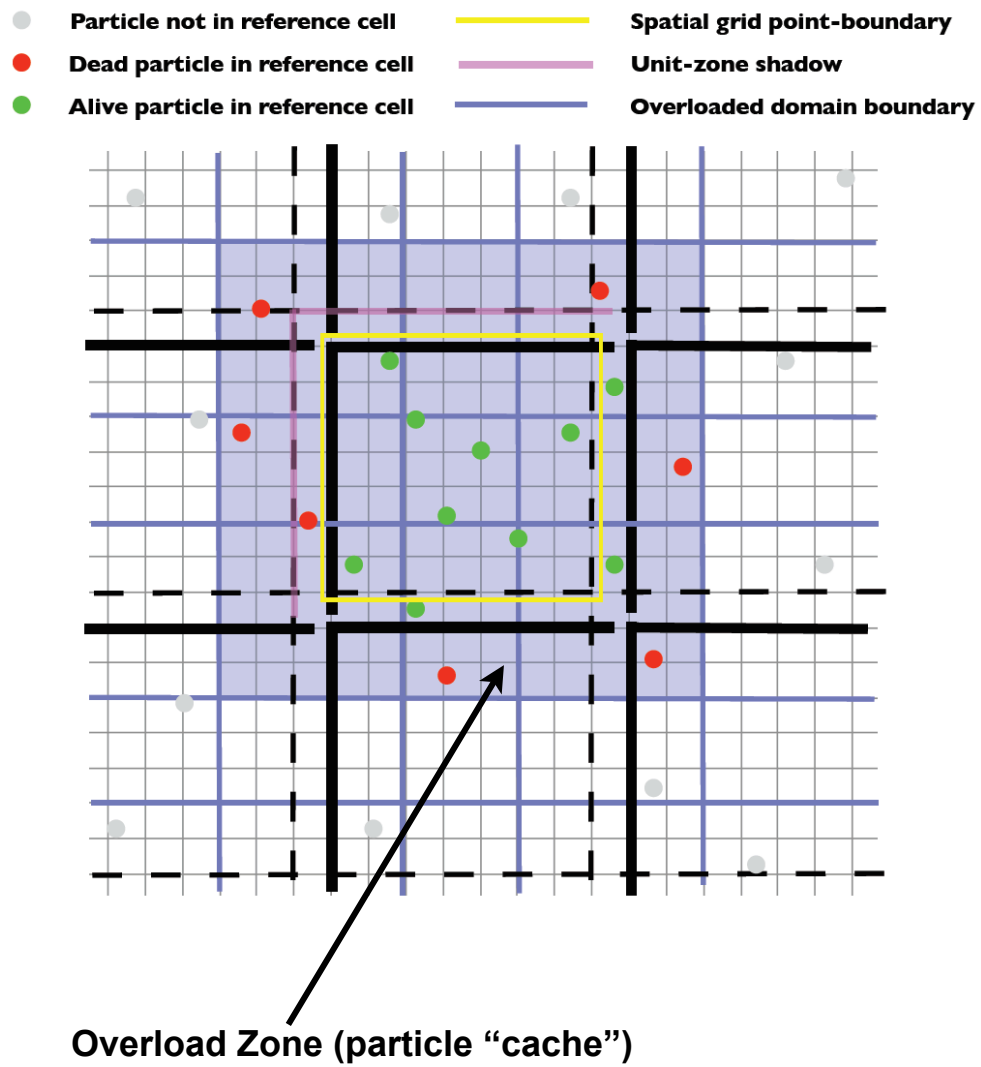
Design Challenges and Solutions for MC^3

- **Challenges (summarized from last slide):**
  - Opterons have half of the machine's memory, balanced communication, but not much compute, standard programming paradigm, C/C++/MPI
  - Cells have other half of machine's memory, slow communication to Opteron layer, lots of compute, new language required
- **Design desiderata:**
  - Distribute memory requirements on both parts of the machine
  - Give the Cell lots of (communication limited) work to do, make sure that Cell part is easy to code and later on easy to replace by different programming paradigm
- **Our Solution:** P^3M algorithm (long range - short range split)
  - Particle Mesh (PM) solver for long-range force, FFT based, grid lives on the Opterons, all coarse-grained parallelism here, base grid is maximized
  - Direct Particle-Particle solver for short range force, particles live on the Cells, lots of compute, simple data structure, easy to implement, can be replaced by tree for different architecture
  - Overloading trick to minimize communication needs, only simple grid information flows between Cells and Opterons (more details next); enables node level short-range force plug-ins

Solutions for MC³: Some Details

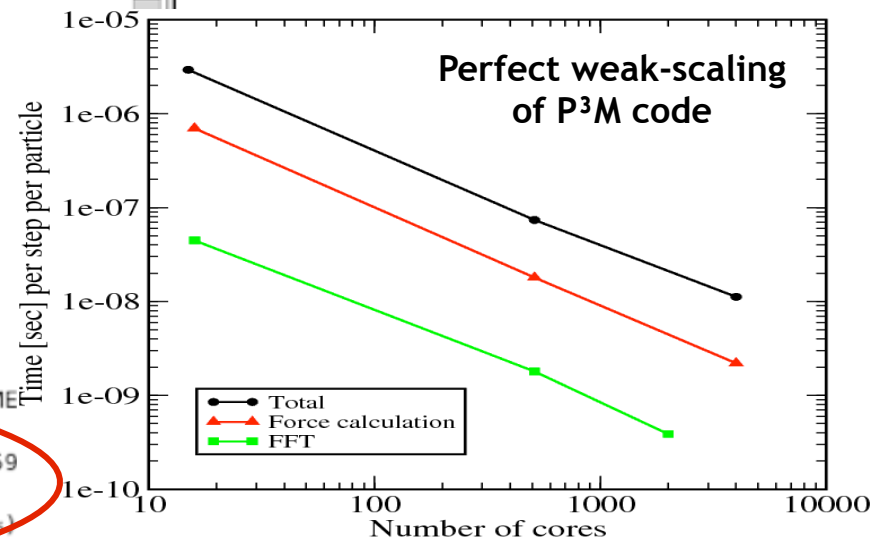
Habib et al. 2009, Pope et al. 2010

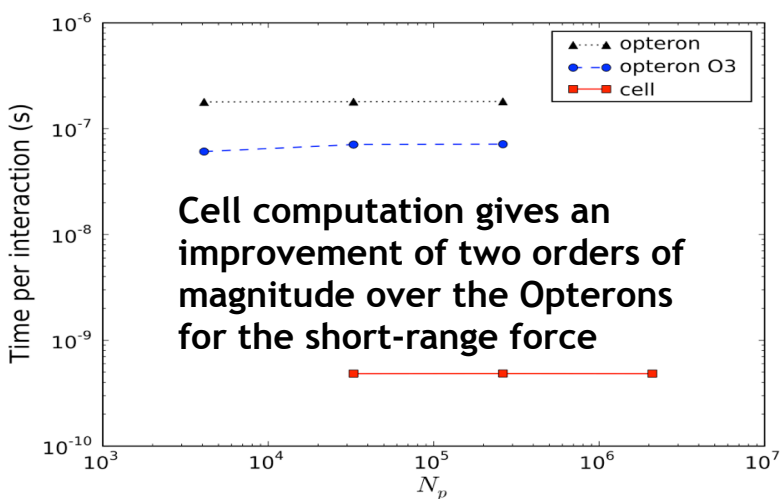
- **Overloading:** Each processor holds particle "cache", trade memory for communication; refresh of active and passive particles every 10-20 long time steps
- **Advantage:** Short-range solver implemented on node, easy to swap
- **Time-stepper:** Symplectic time-stepper, on Cell: sub-cycles, currently: no individual particle time steps, can be implemented in straightforward way; at every long-range force time-step: grid information goes from Cell to Opterons
- Spectral smoothing of the CIC density field allows 6-th order Green function and 4th order super-Lanczos gradients for high-accuracy Poisson-solves, allows force matching at 3 grid cells and hence reduces particle grid interaction

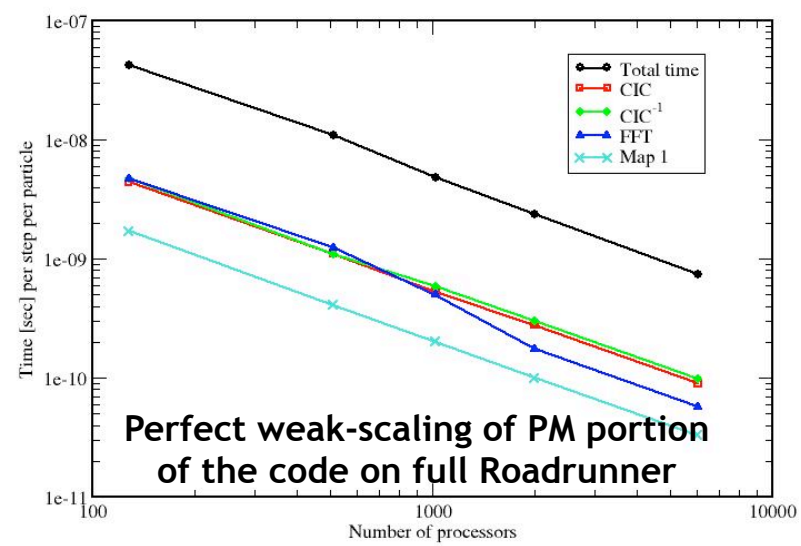


MC³ Performance

```
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%)
```

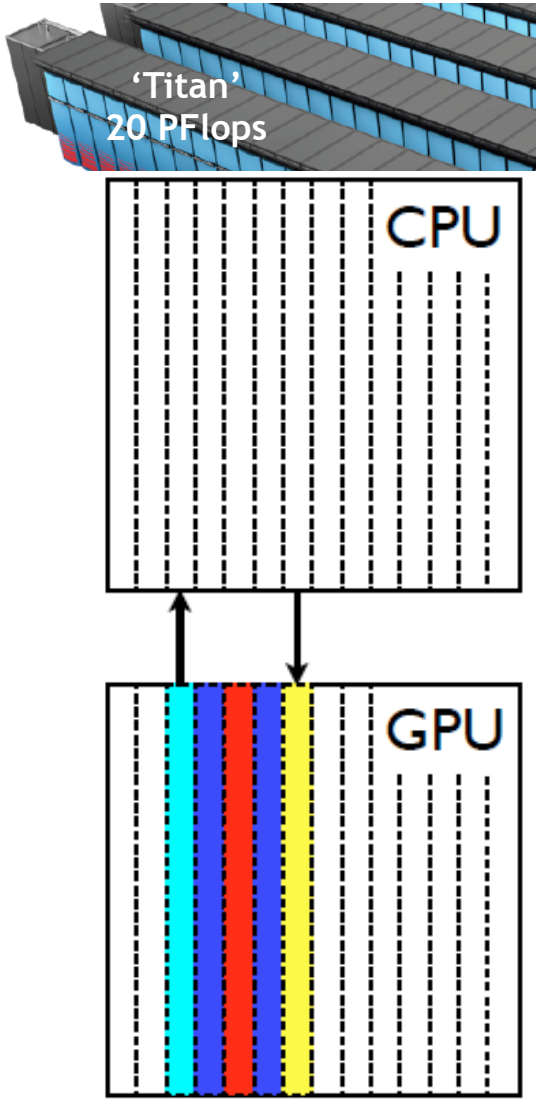






The Story continues, MC³ becomes HACC: CPU+GPU....

- **Proof of concept for easy portability:** replace Cell part by GPU implementation
- Paul Sathre (CS undergraduate) successfully ports code within weeks (with guidance from A. Pope), later, Nick Frontiere takes over and rewrites GPU code
- **New challenges:**
  - CPU/GPU performance and communication out of balance, unbalanced memory (CPU/main memory dominates)
  - New programming language on GPU
- **Solutions:**
  - Particles in CPU main memory, CPU does low flop/byte operations
  - Stream slabs through GPU memory (pre-fetches, asynchronous result updates)
  - OpenCL
- **Successful run on heterogenous system (Nvidia and AMD accelerated hardware) at supercomputing conference SC10, including in-situ visualization (movie played while code ran)**



.... and a very successful port to Mira, BG/Q

- **Proof of concept for portability II:** IBM Blue Gene (BG) systems
  - BG/P at ANL: Intrepid, 163,840 cores, 4 cores per node, 2GB per node
  - BG/Q: Mira, 10 PF/s, just arrived, 750,000 cores, 16GB per node
- **New challenges:**
  - BG systems have many more cores but no accelerators
  - FFT so far was slab-decomposed, does not scale well on large number of cores
- **Solutions:**
  - HACC stands now for “Hardly accelerated...” ;-)
  - Particle-particle interaction now replaced by tree
  - Pencil decomposed FFT (scales to full BG/Q)
- **Scalability of HACC is controlled only by FFT performance**

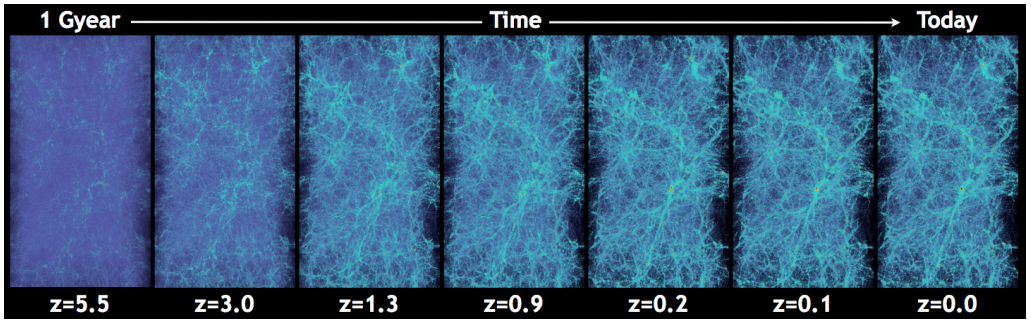
.... and a very successful port to Mira, BG/Q

- HACC port to Mira complete
- Very good performance achieved with PPTreePM algorithm
- Test run on 16 racks on Mira:

```
Creating rho(x,y,z)  3        39.77        17.78
Poisson solve        3        51.94        23.23
Particle move        3        2.04         0.9123
Output               3        0.53         0.237
==========================================

PXCH buffer = 2*1943590 = 3.707104 MB
InitialExchange TotalAliveParticles 1073741824000
```

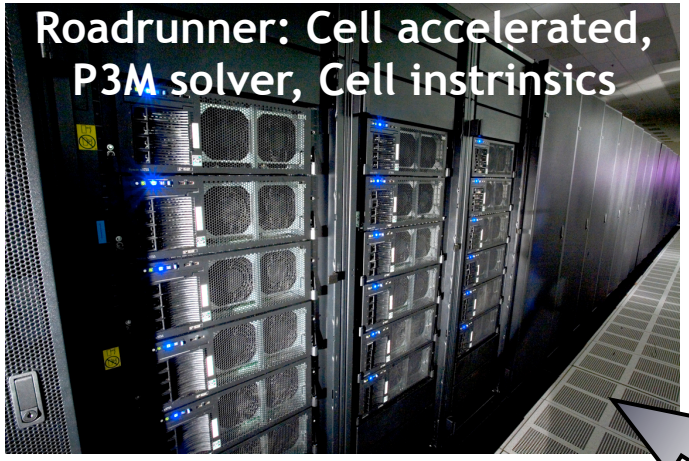
One trillion+ particles!

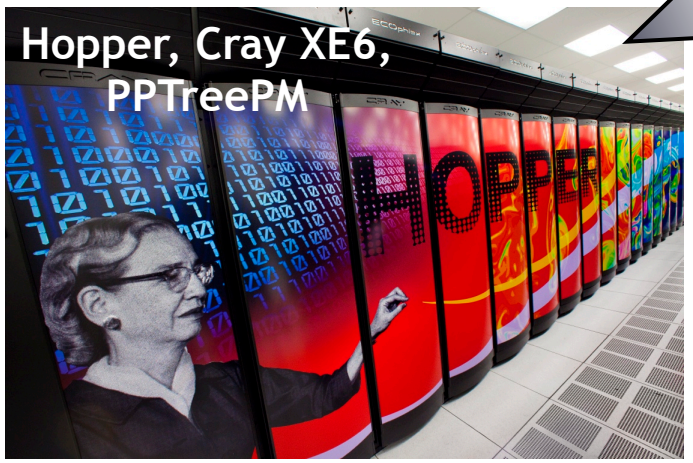
- Scaling tests on up to 72 racks (more than a million cores; 4.718.592 parallel execution threads, almost 5 Mill!) on Sequoia
- One of five finalists for this year's Gordon Bell award at SC12

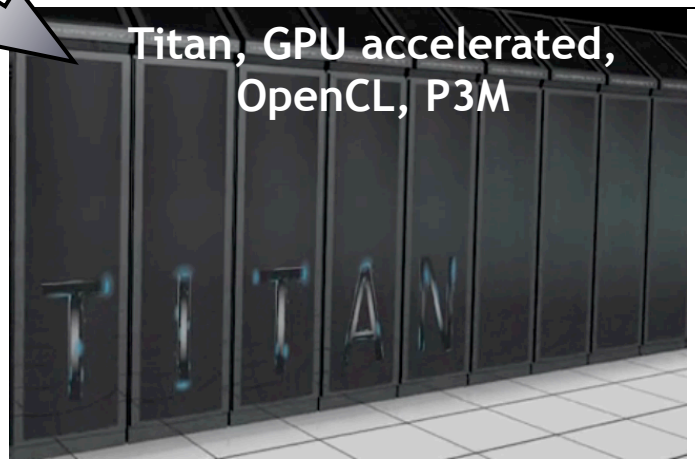


1.1 trillion particle test simulation

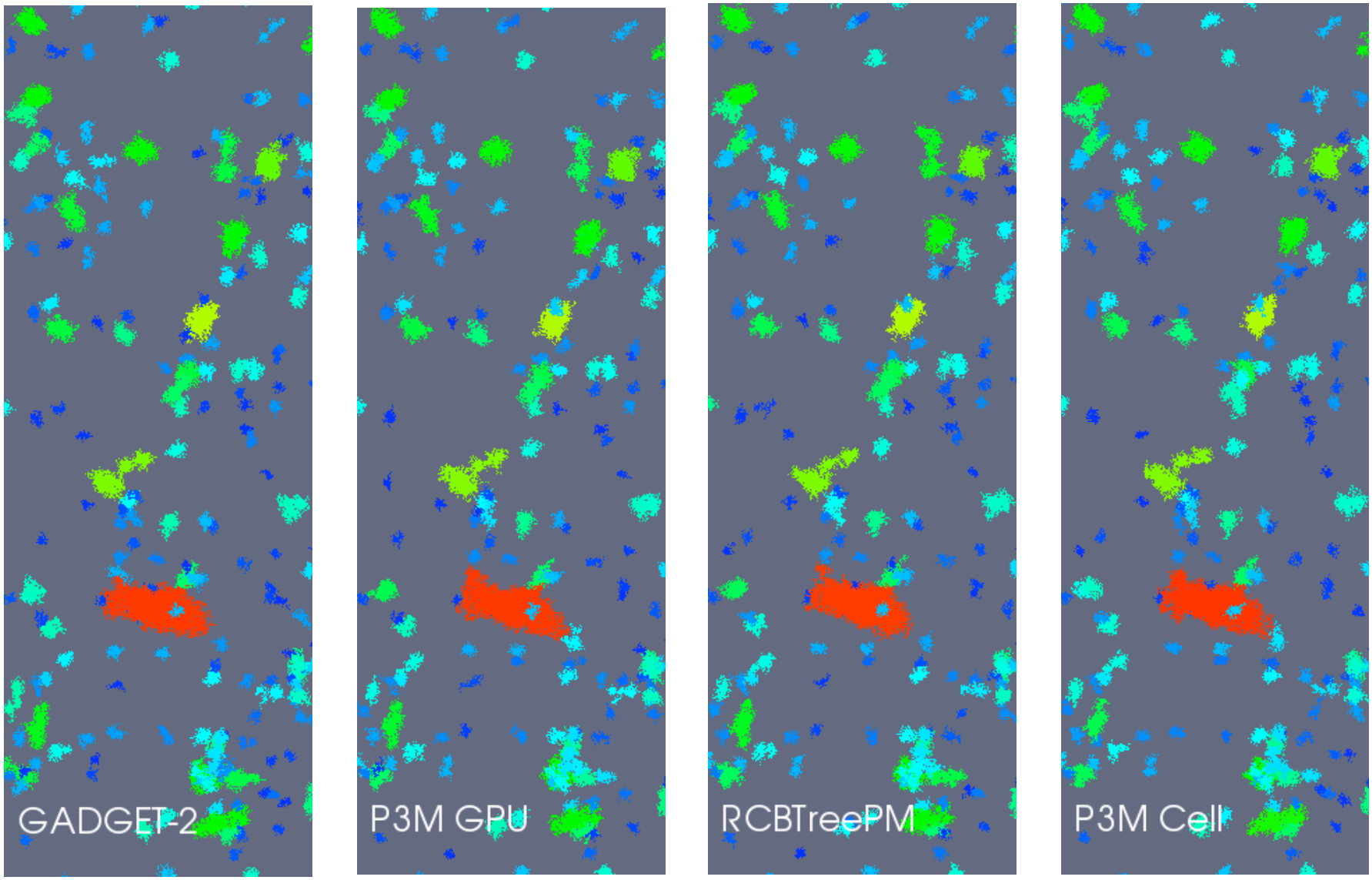
HACC: A Code for All Architectures



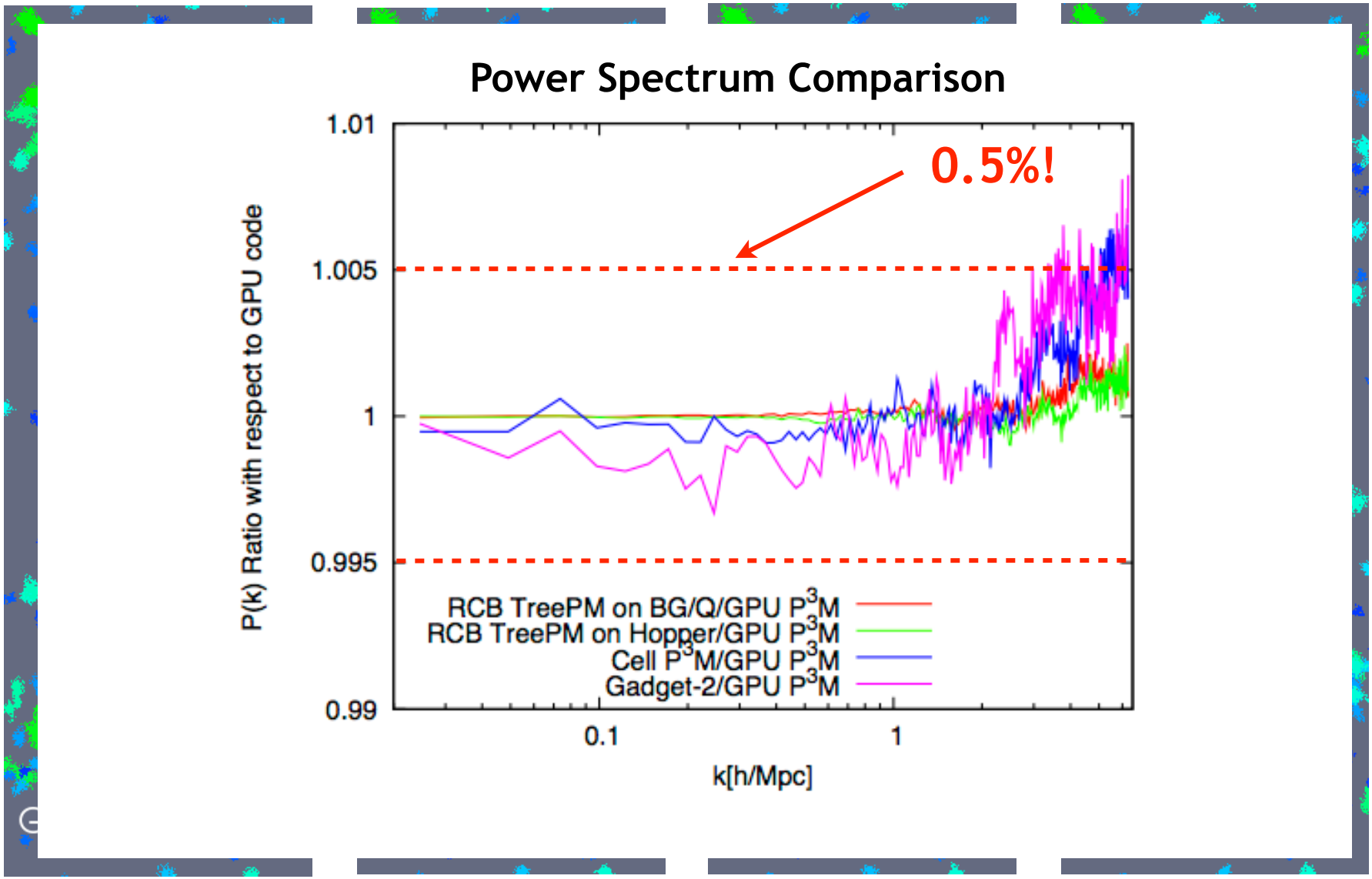




HACC: A Code for All Architectures

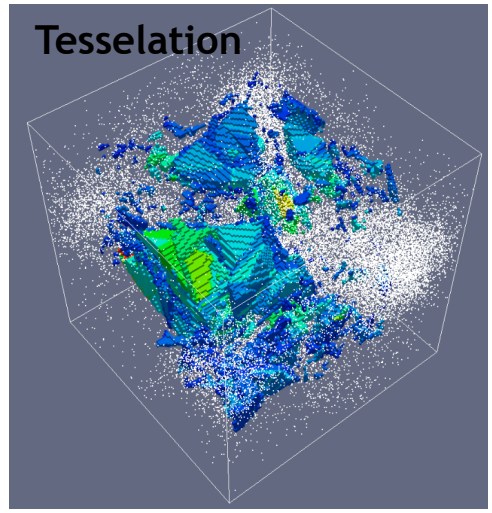


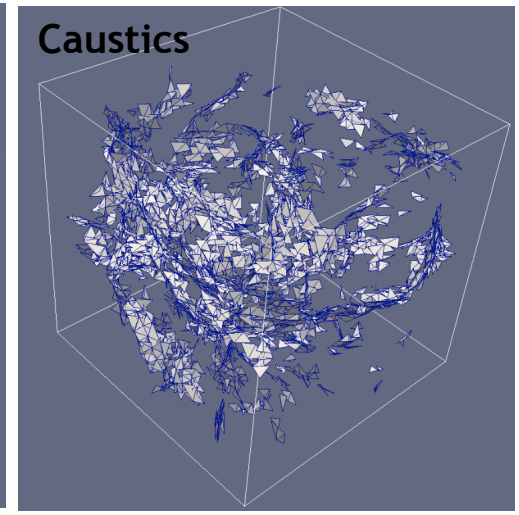
HACC: A Code for All Architectures

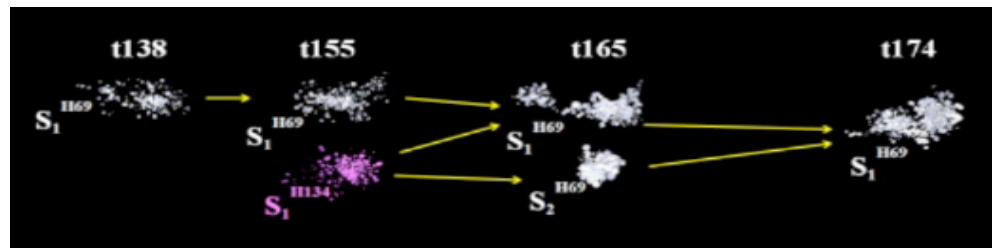


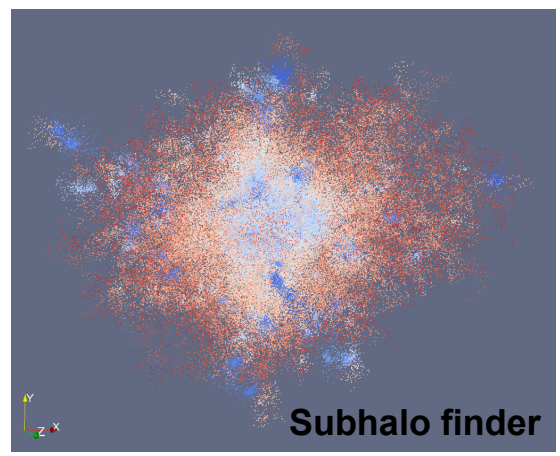
HACC Analysis Tools

- **Development of "CosmologyTools", a library of tools coupled to HACC**
- **On-the-fly halo finder**
  - Friends-of-friends (FoF) finder
  - Spherical overdensity finder
  - Diverse halo properties
  - Lives on CPUs (minimize code rewrites)
- **Subhalofinder**
  - Features of sub-find and phase-space finders, different tree-walk
  - Test on large halo (1M+ particles) from Gadget run, comparison with other codes shows excellent agreement (thanks to Michal Maciejewski and Volker Springel for providing the data)
- **Tess:** Tesselation tool based on QHull
- **Stream finder:** Tesselation tool under development
- **Merger tree:** under development



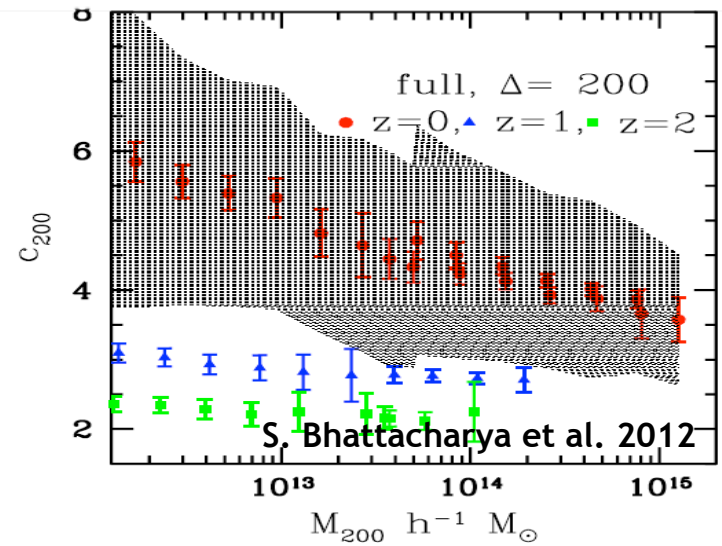




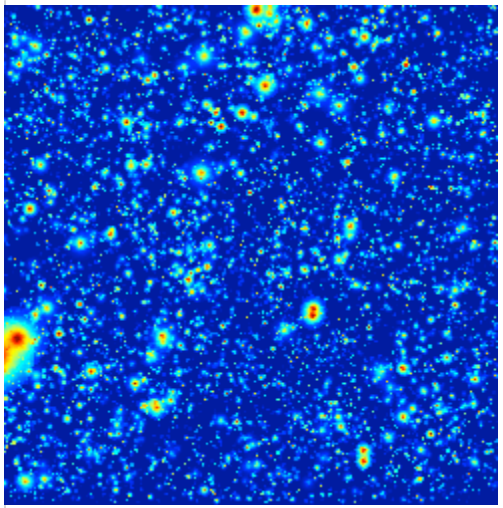


Some Science Results --

Roadrunner view (halos) of the Universe at z=2 from a 64 billion particle run (9 runs on one weekend)



Mass function and c-M relation at z=0,1,2



SZ sky maps (this one was done on a Coyote run)

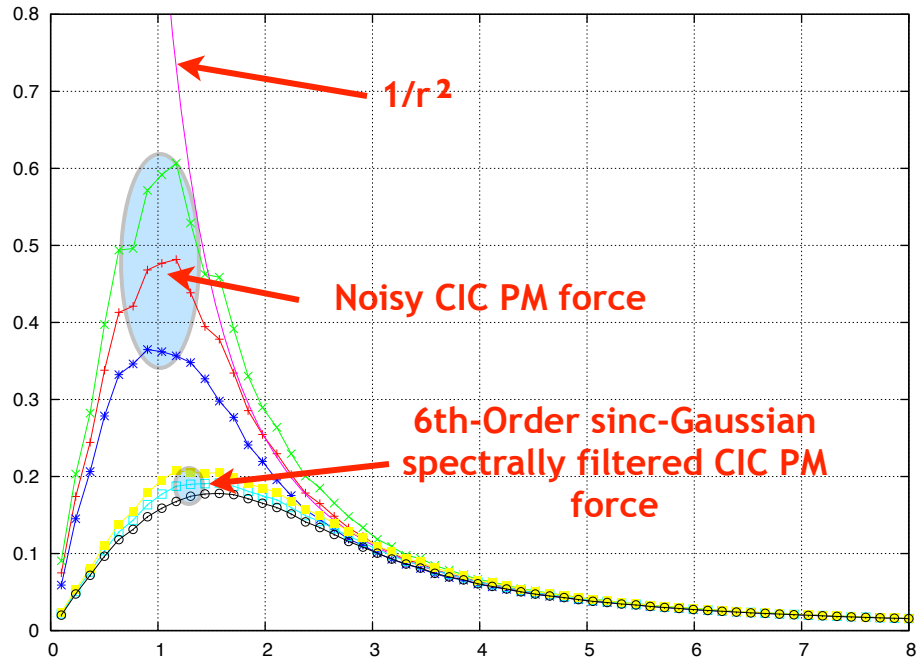
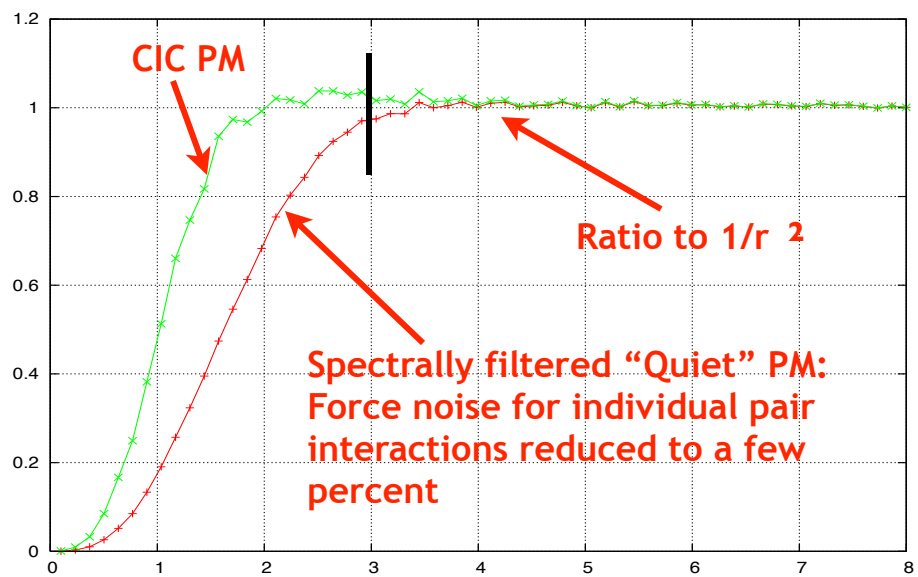
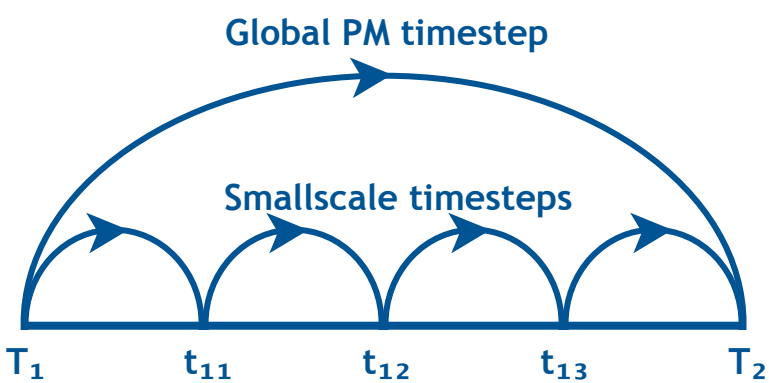
New simulation running on Hopper

Bhattacharya, Das et al. in prep.

The Outer Rim -- where thoughts, time, and space become one (Mira project)

Summary and Outlook

- Precision cosmology era and advent of large scale surveys demand large simulations
- High-performance computing landscape is rapidly changing, different architectures require flexibility in simulation codes, opportunity as well as “curse”
- With HACC we attempt to address this challenge
- Long-range force solved on CPU portion, short-range force (compute intensive) solved on accelerator when available, framework adapts to different architectures in a straightforward manner
- On-the-fly analysis important, helps with data reduction
- Major allocations currently:
  - Mira Early Science Project, 150 Mill. CPU hours, large set of different cosmologies
  - ALCC project on neutrino simulations
  - OLCF Early Science Project
  - NERSC SciDAC allocation

Solutions for MC³: Some Details

- Spectral smoothing of the CIC density field allows 6-th order Green function and 4th order super-Lanczos gradients for high-accuracy Poisson-solves
- Short-range force is fit to the numerical difference between Newtonian and long-range force (not conventional P³M)