Hardware/Hybrid Accelerated Cosmology Code

Nicholas Frontiere, JD Emberson, Salman Habib, Adrian Pope, Hal Finkel, Katrin Heitmann, Vitali Morozov

Goal

In response to the plethora of data from current and future large-scale structure surveys of the Universe, sophisticated simulations are required to obtain commensurate theoretical predictions. We have developed the Hardware/Hybrid Accelerated Cosmology Code (HACC) capable of sustained performance on powerful and architecturally diverse HPC machines to address this numerical challenge. Presented here are preliminary performance results from multiple implementations of HACC designed for optimal performance on the Summit and Theta supercomputers.

Code

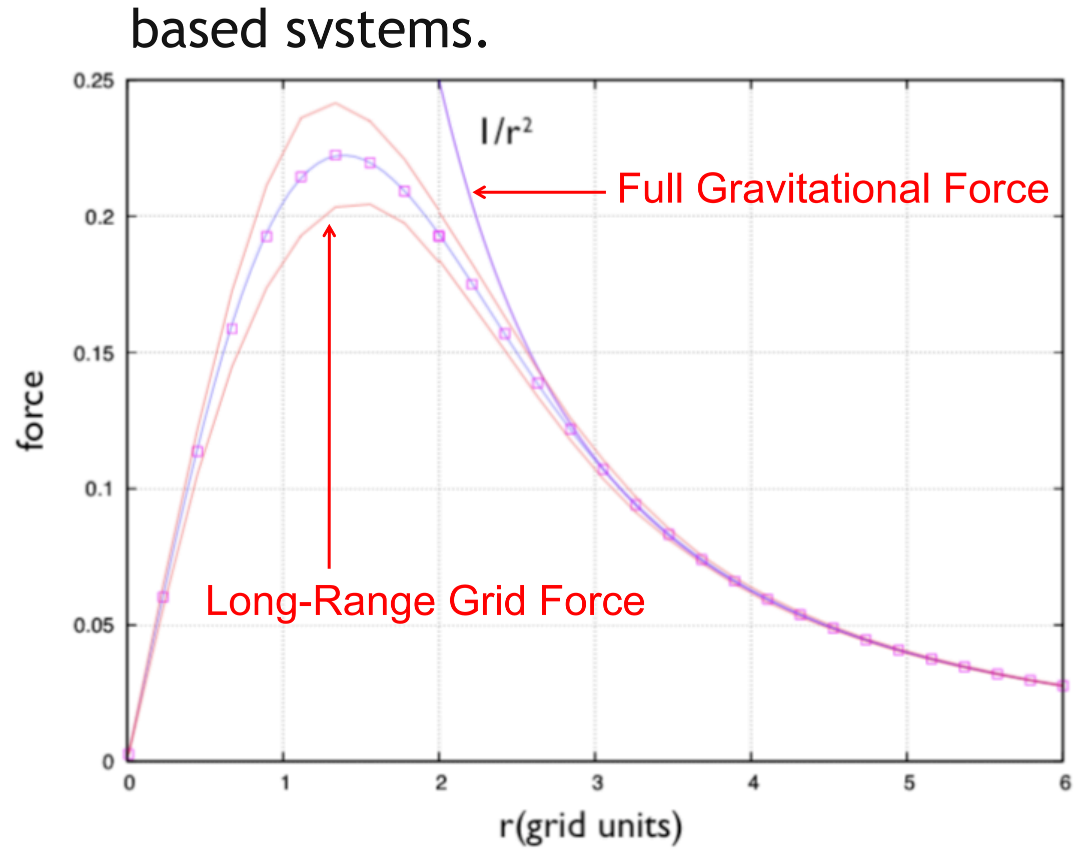
HACC follows the evolution of (collisionless and non-relativistic) dark matter under gravity in an expanding universe by solving the Vlasov-Poisson equations:

$$\frac{\partial f}{\partial t} + \dot{\mathbf{x}} \cdot \frac{\partial f}{\partial \mathbf{x}} - \nabla\phi \cdot \frac{\partial f}{\partial \mathbf{p}} = 0, \quad \mathbf{p} = a^2\dot{\mathbf{x}} \tag{1}$$

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

Particle tracers and N-body methods are used to numerically solve the 6 dimensional PDE. The force calculation is split into long and short-range components.

- The long-range force is computed with distributed-memory Fast Fourier Transform (FFT) methods that use the standard Message Passing Interface (MPI).
- The short-range force calculation is designed to be modular and tuned to the underlying hardware. Versions of HACC have been developed for IBM Cell/GPU-accelerated systems, the Blue Gene Q architecture, and for Intel Xeon Phi (KNL)-based systems.



GPU Implementation

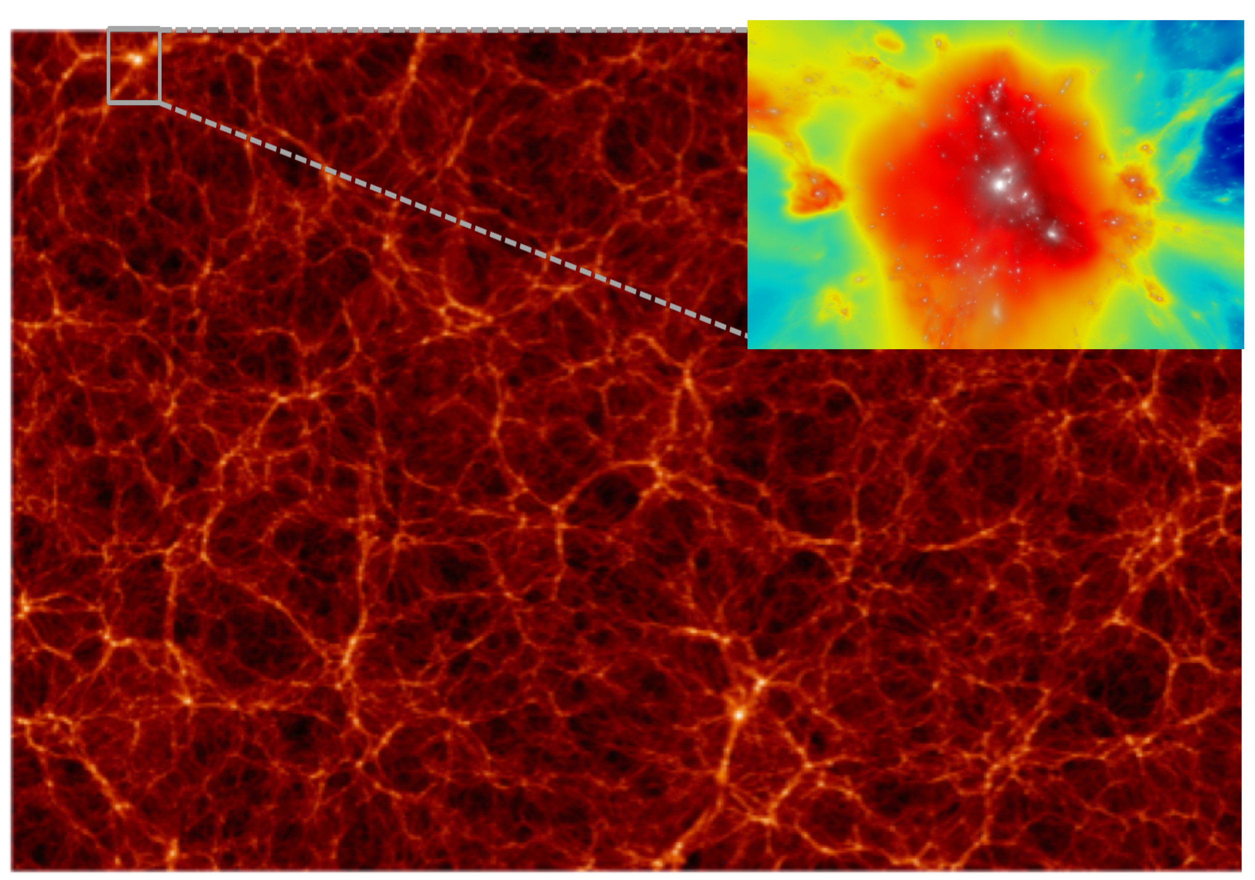
The local force kernel for the GPU implementation is a direct particle-particle interaction, utilizing the high throughput of accelerators. Data is partitioned into large blocks determined by the memory limit of the GPU, and are asynchronously pushed and calculated on the GPU to hide latency. The original GPU implementation was optimized on the Titan supercomputer and achieved 99.2% parallel efficiency on 16,384 nodes, on 90% of the machine, where one of our biggest runs evolved $10{,}240^3$ (~1 trillion) particles. Our preliminary Summit performance results demonstrate a factor of **6.7x** speed-up when comparing our gravity kernel on a Titan vs Summit GPU. Note: The hardware flop ratio between the K20 and Volta devices is ~ 4x.

RCB-Tree Implementation

For many-core systems such as the BG/Q or the Intel Xeon Phi, the GPU strategy is not optimal, and tree-base methods yield much higher efficiency. HACC employs a recursive coordinate bisection (RCB) tree, where we make full usage of spatial locality of our data, and the ability to optimize tree-walk minimization.

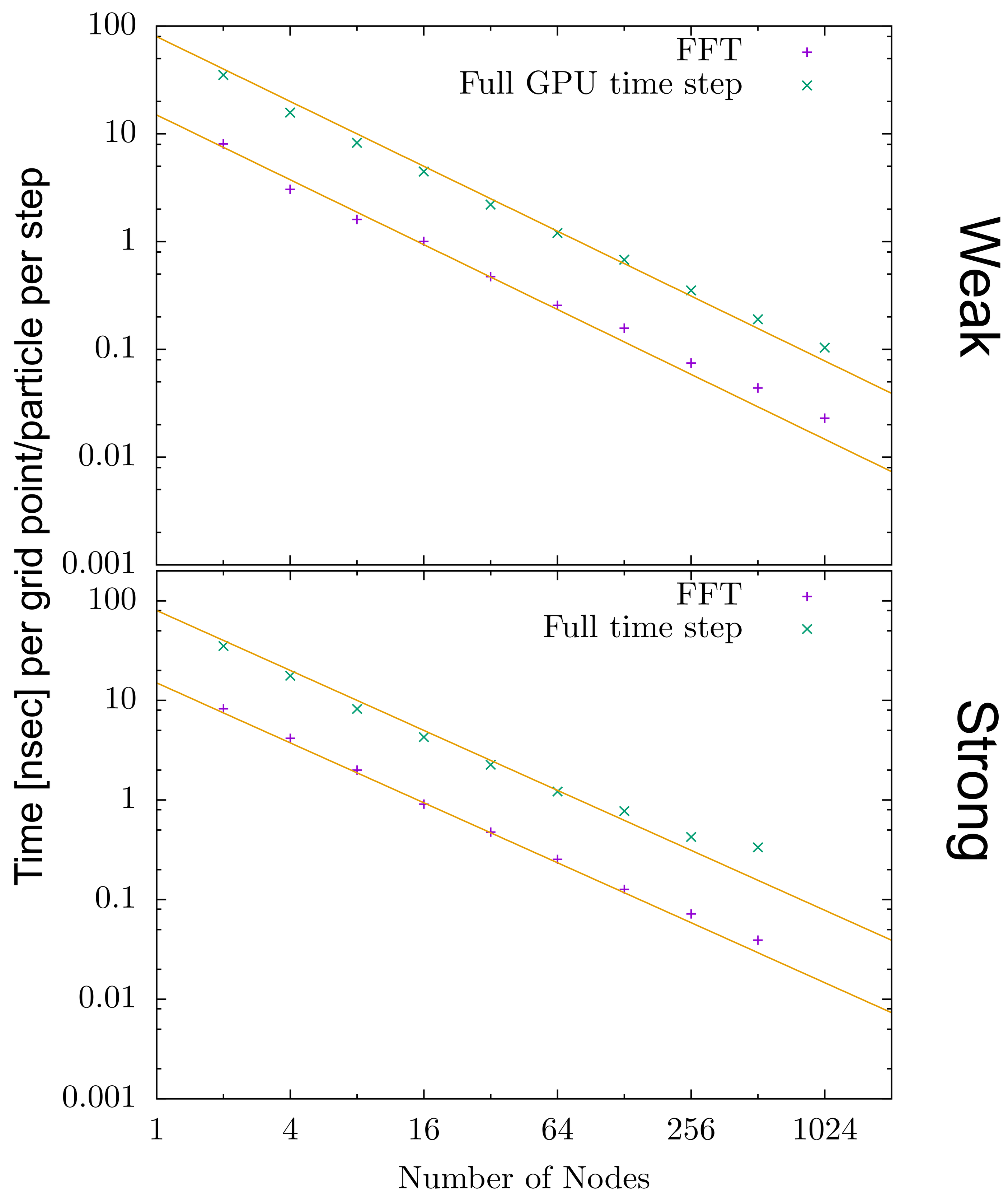
Hydro Implementation

HACC has additionally implemented a Lagrangian based hydrodynamic solver dubbed CRK-HACC. We couple our gravity solver to a variant of Smooth Particle Hydrodynamic called Conservative Reproducing Kernel SPH (Frontiere et al. 2017). We have both GPU and KNL implementations of CRK-HACC



BorgCube: First large-scale CRK-HACC simulation containing 2 X 2304^3 dark matter plus baryonic particles in a cubic volume of side length 800 Mpc/h. Run on Theta at ALCF as part of the Theta ESP (late December 2017).

Preliminary Summit Scaling



Theta CRK-HACC Strong Scaling