

Astrophysical N-body Simulation on a cluster of GPUs

NACC

Nagasaki Advanced Computing Center

Tsuyoshi Hamada,
NACC, Nagasaki University



長崎大学
NAGASAKI UNIVERSITY



The GPU evolution



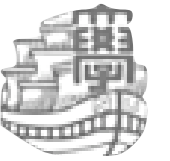
- The **Graphic Processing Unit (GPU)** is a processor that was **specialized** for processing graphics.
- The GPU has recently **evolved** towards a **more flexible** architecture.
- **Opportunity**: We can implement ***any algorithm***, not only graphics.
- **Challenge**: obtain **efficiency** and **high performance**.



Tesla card

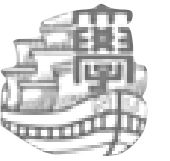


Tesla S1070: 4 cards



Overview of the presentation

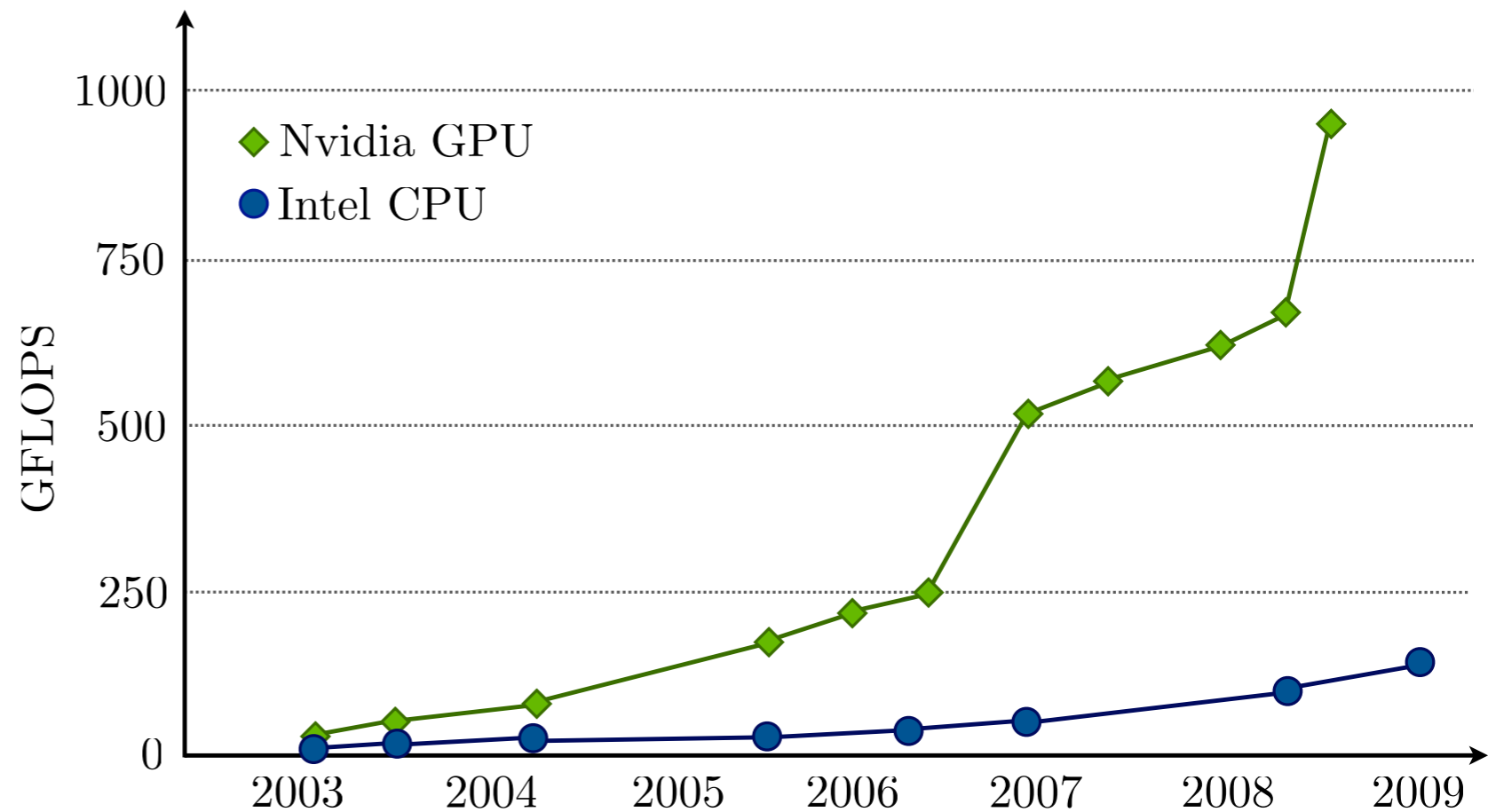
- Motivation
- The Buzz: GPU, Teraflops, and more!
- The reality (**my** point of view)
- Astrophysical N-body simulation on a GPU cluster

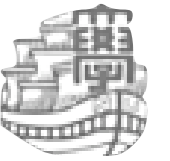


The motivation

GPU computing - key ideas:

- Massively parallel.
- Hundreds of cores.
- Thousands of threads.
- Cheap.
- Highly available.
- Programable: CUDA





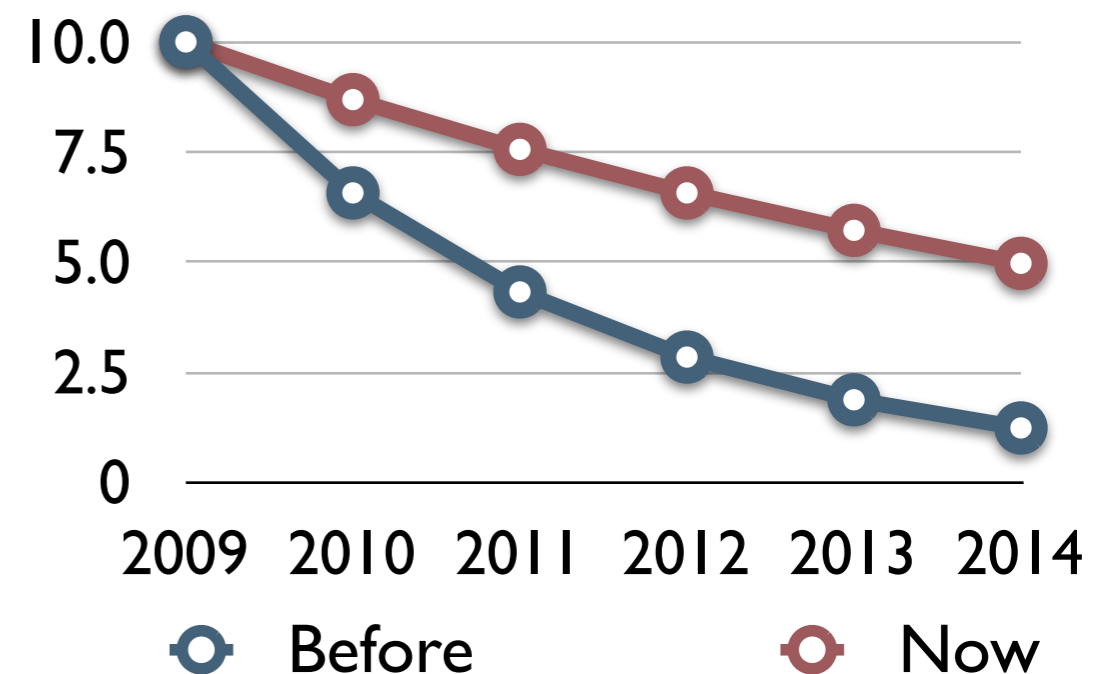
Ok... after the buzz

- Question 1: **Why accelerator technology today?** If it has been around since the 70's!
- Question 2: **Can I really get 100x in my application?**



Why accelerator technology today?

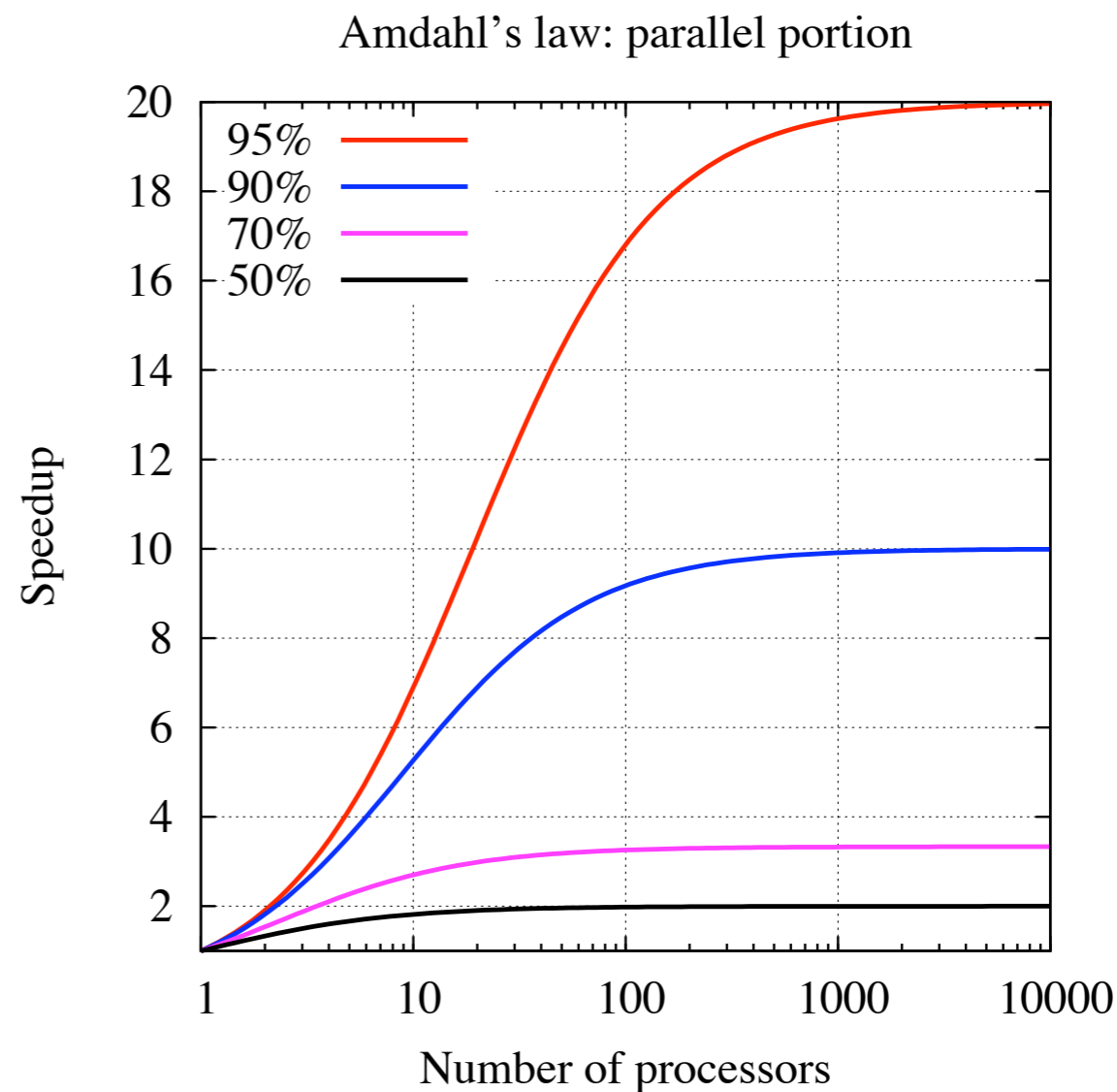
- Investment on GPU technology makes **more sense today** than in 2004.
- CPU uni-processor speed is not doubling every 2 years anymore!
- Case: investing in an accelerator that gives a ~10x speedup:
 - **2004** speedup **1.52x** per year: 10x today would be **1.3x** acceleration in 5 years.
 - **TODAY** speedup **1.15x** per year: 10x today would be **4.9x** acceleration in 5 years.
 - Consider the point that **GPU parallel performance is doubling** every 18 months!





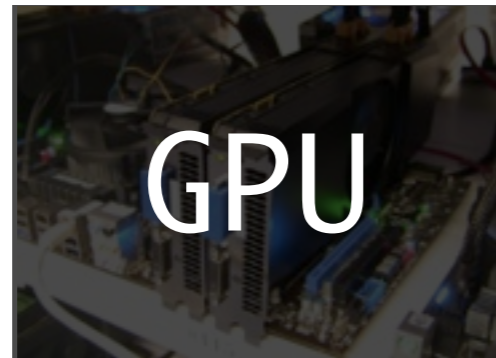
Can I get 100x speedups?

- **You can** get hundred-fold speedup for **some** algorithms.
- It depends on the non-parallel part: **Amdahl's law**.
- Complex application normally make use of many algorithms.
- Look for **alternative ways** to perform the computations that are more parallel.
- **Significance**: An accelerated program is going to be as fast as its serial part!



Amdahl's Law
Maximum speedup

Summary



+

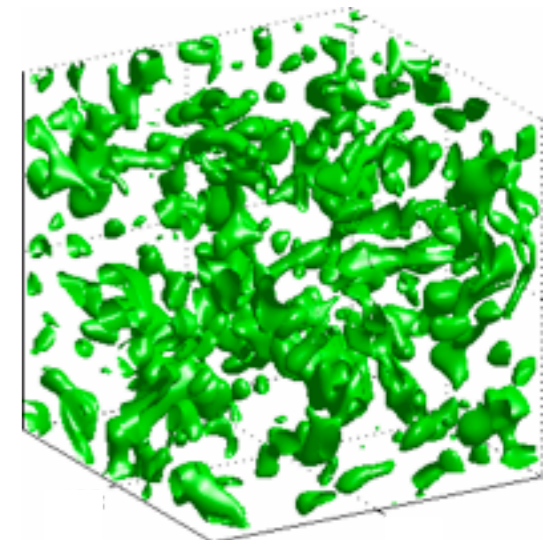
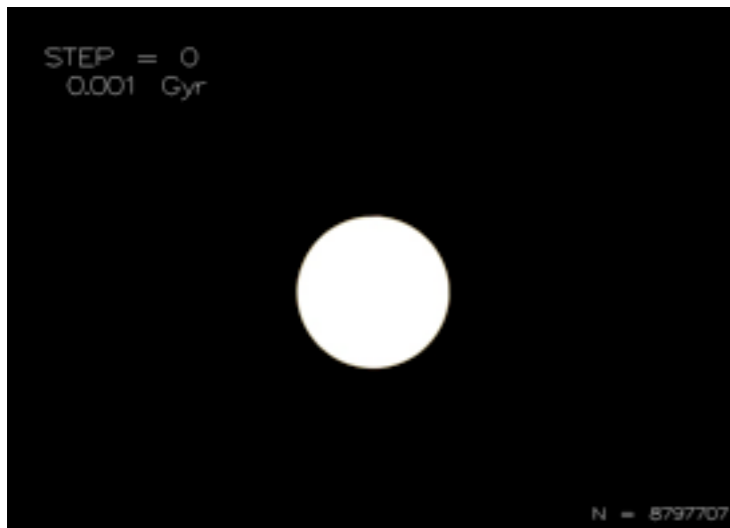


🌐 Achieved Price/Performance

🌐 \$ 3.9 / Gflops

🌐 15 times better than 2006' GB Finalist

🌐 2.5 times better than 2009' GB winner



Challenges

- To get a high efficiency on GPUs for hierarchical $O(N \log N)$ or $O(N)$ method (not on brute $O(N^2)$ method)
- using large amount of GPUs
414,720 = 240 * 3 * 576 FP units
- To get a good scalability on commodity network (GbE)

Challenges

- To get a high efficiency on GPUs for hierarchical $O(N \log N)$ or $O(N)$ method (not on brute $O(N^2)$ method)

Treecode, FMM

- using large amount of GPUs
414,720 = 240 * 3 * 576 FP units
- To get a good scalability on commodity network (GbE)

Challenges

- To get a high efficiency on GPUs for hierarchical $O(N \log N)$ or $O(N)$ method (not on brute $O(N^2)$ method)

Treecode, FMM

- using large amount of GPUs
414,720 = 240 * 3 * 576 FP units

Multiple walk

- To get a good scalability on commodity network (GbE)

Challenges

- To get a high efficiency on GPUs for hierarchical $O(N \log N)$ or $O(N)$ method (not on brute $O(N^2)$ method)

Treecode, FMM

- using large amount of GPUs
414,720 = 240 * 3 * 576 FP units

Multiple walk

- To get a good scalability on commodity network (GbE)

Delegated Alltoallv

N-body simulation

☉ Particles interact with each other

☉ Stars, Galaxies, Atoms, etc.

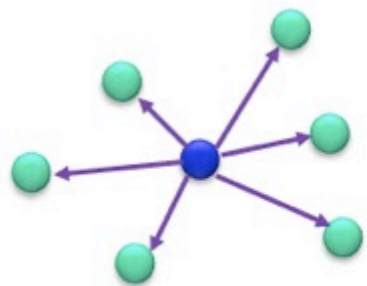
☉ Computational cost

☉ Direct sum - $O(N^2)$

☉ Tree algorithm - $O(N \log N)$

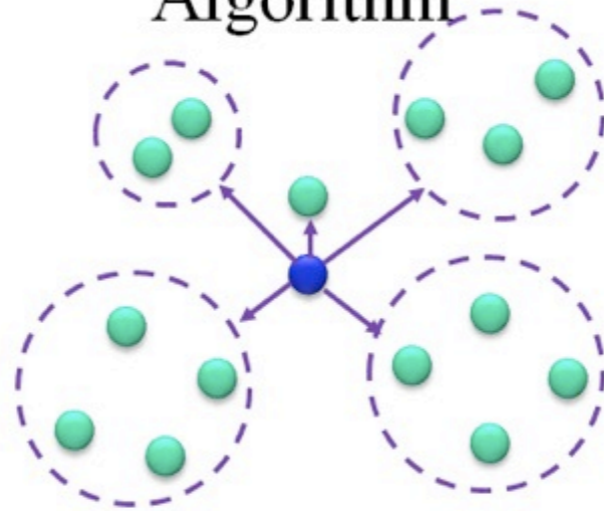
☉ Fast Multipole Method - $O(N)$

Direct Summation
Algorithm

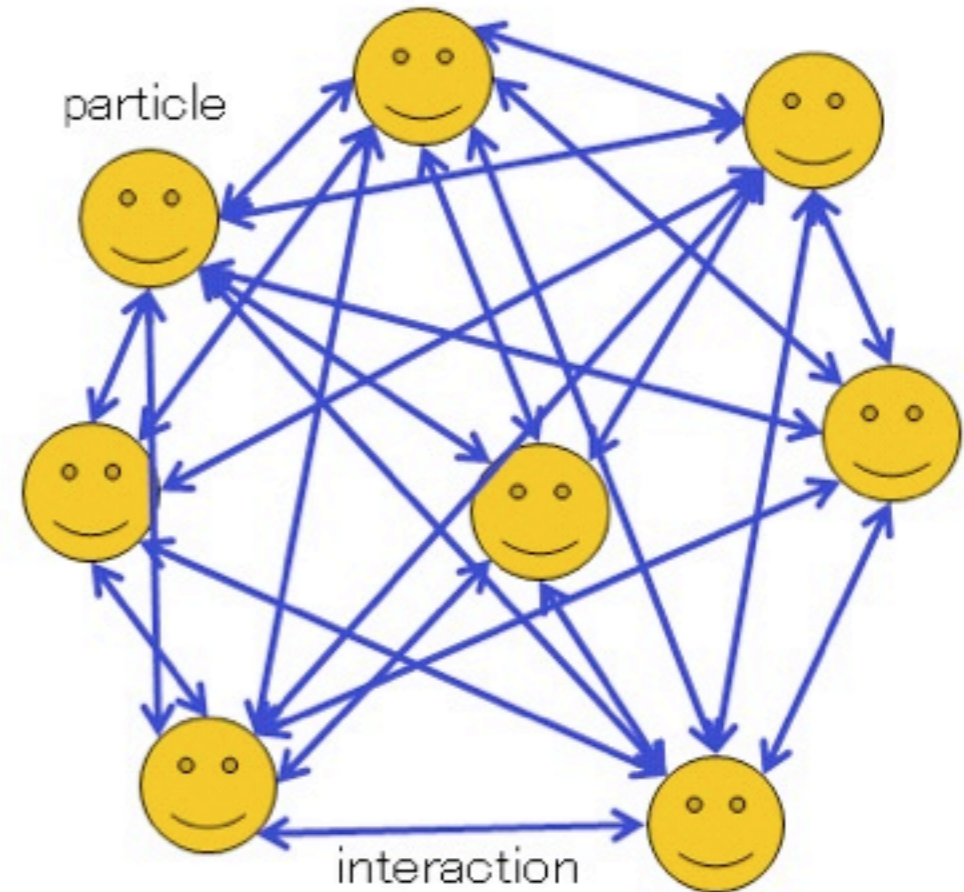


$O(N^2)$

Tree
Algorithm



$O(N \log N)$



Applications of N-body

Poisson
 $\nabla^2 u = -f$

Astrophysics
Electrostatics
Fluid Mechanics

$$\nabla^2 \phi = 4\pi GM$$
$$\nabla^2 \phi = -\frac{q}{\epsilon_0}$$
$$\nabla^2 p = -\nabla \cdot \{\mathbf{u} \cdot (\nabla \mathbf{u})\}$$
$$\nabla^2 \mathbf{u} = -\nabla \times \boldsymbol{\omega}$$

Helmholtz
 $\nabla^2 u + k^2 u = f$

Acoustics
Electromagnetics

$$\frac{1}{c^2} \frac{\partial^2 \phi}{\partial t^2} = \nabla^2 \phi$$
$$\mu_0 \epsilon_0 \frac{\partial^2 \mathbf{E}}{\partial t^2} = \nabla^2 \mathbf{E}$$
$$\mu_0 \epsilon_0 \frac{\partial^2 \mathbf{H}}{\partial t^2} = \nabla^2 \mathbf{H}$$

Quantum Mechanics

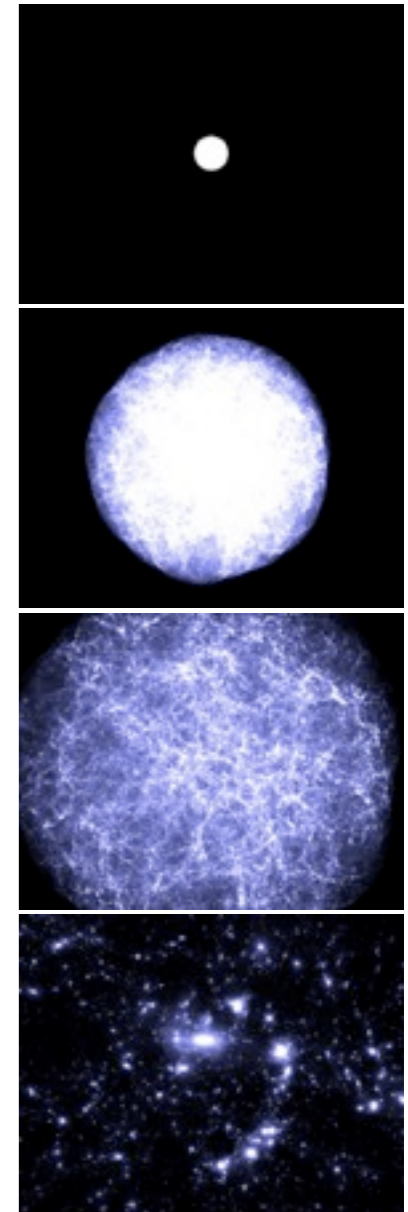
$$\frac{1}{c^2} \frac{\partial^2 \phi}{\partial t^2} = \nabla^2 \phi - \frac{m^2 c^2}{\hbar^2} \phi$$

Run

- Standard cold dark-matter cosmological simulation

$$\frac{d^2 \vec{r}_i}{dt^2} = \sum_{j \neq i} -\frac{G m_j \vec{r}_{ij}}{r_{ij}^3}$$

- Barnes & Hut tree method
- 576 MPI processes
 - 9 x 8 x 8 decomposition
- 3.3 Giga particles
 - 3,278,982,596 particles



Past Approaches (treecode in Gordon Bell)

Massively-parallel system

 Warren et al. (1997, winner)

Dedicated hardware, single node

 Kawai et al. (1999, Winner)

FPGA, single node

 Kawai et al. (2006, finalist)

GPU, massively-parallel

 Hamada et al. (Our work)

Hardware Configuration

144 nodes of GPU cluster

- 576 NVIDIA GT200 chips

Host

- Core i7 920
- 12GB DDR3
- MSI X58 pro-E

GPU

- Dual GeForce GTX 295 (single PCB) per node

Network

- 36 port InfiniBand QDR x 6

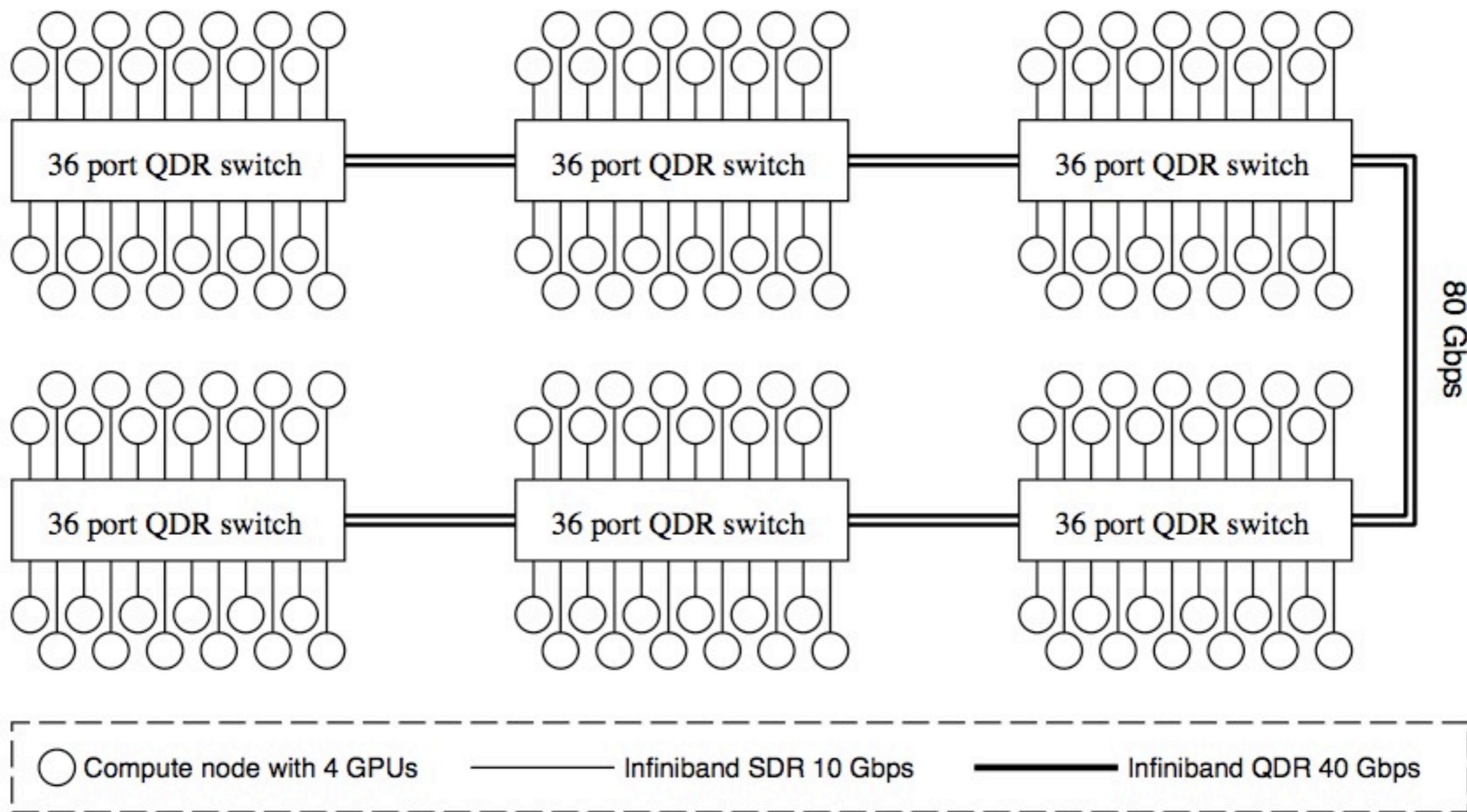


Price

- Host PCs: \$155,979
- GPUs: \$152,675
- InfiniBand Cards: \$20,646
- InfiniBand Cables: \$30,905
- InfiniBand Switches: \$51,717
- Total: \$411,921



System Configuration



Parallel implementation

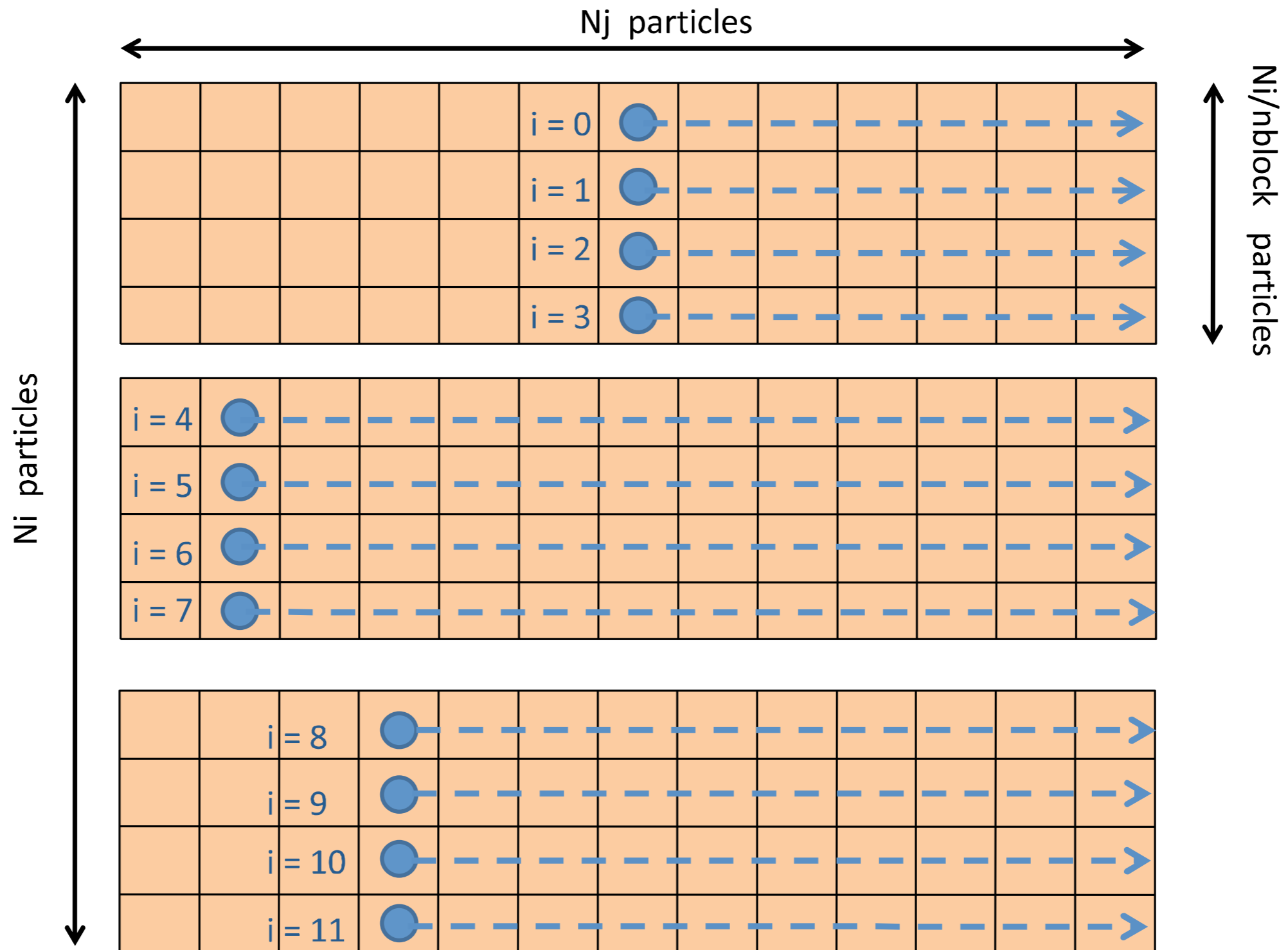
Domain decomposition

-  3-dimensional recursive multi-section

Force from other domain

-  Exchanging LET (local essential tree)

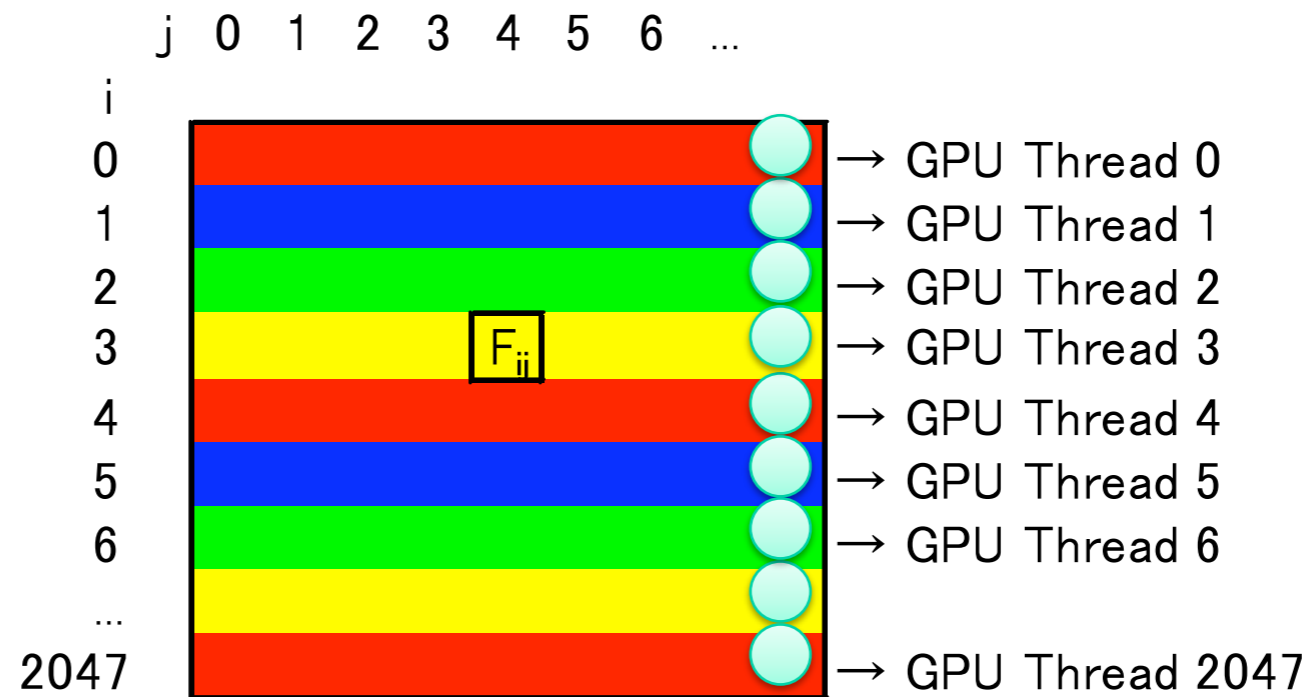
GPU implementation for $O(N^2)$



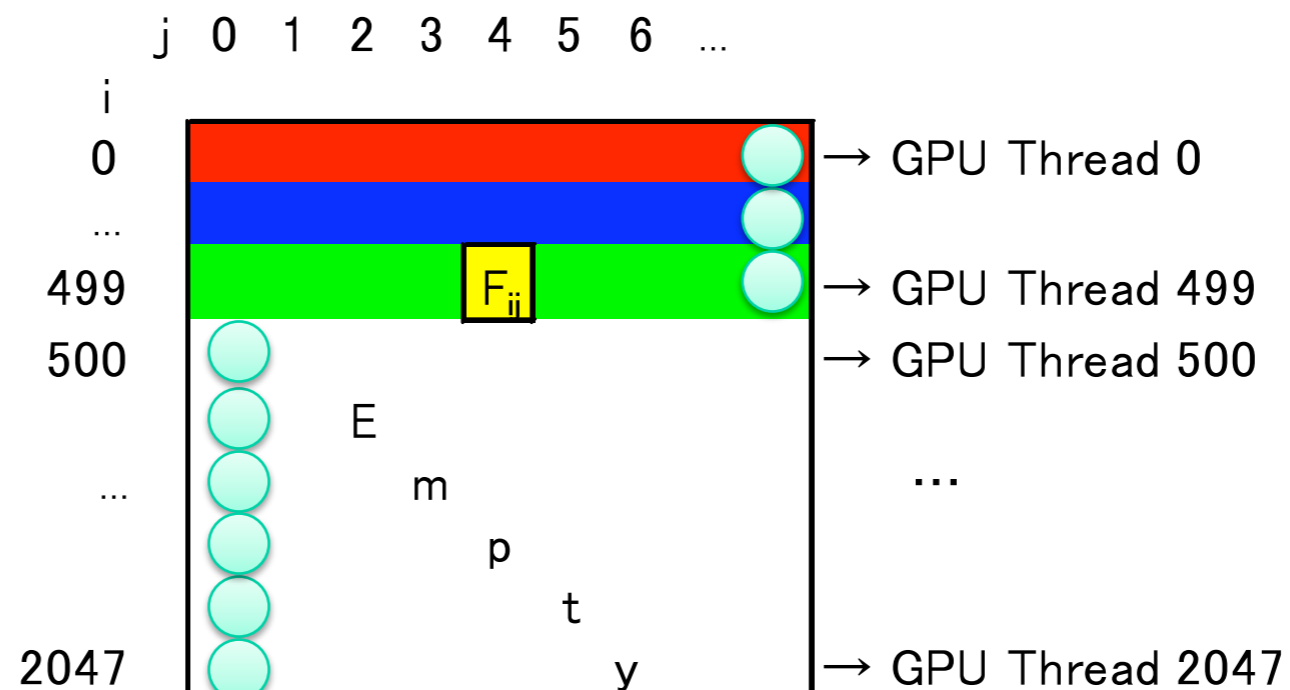
Hamada et al 2007, Belleman 2007, Nyland 2007, etc

Naive treecode implementation

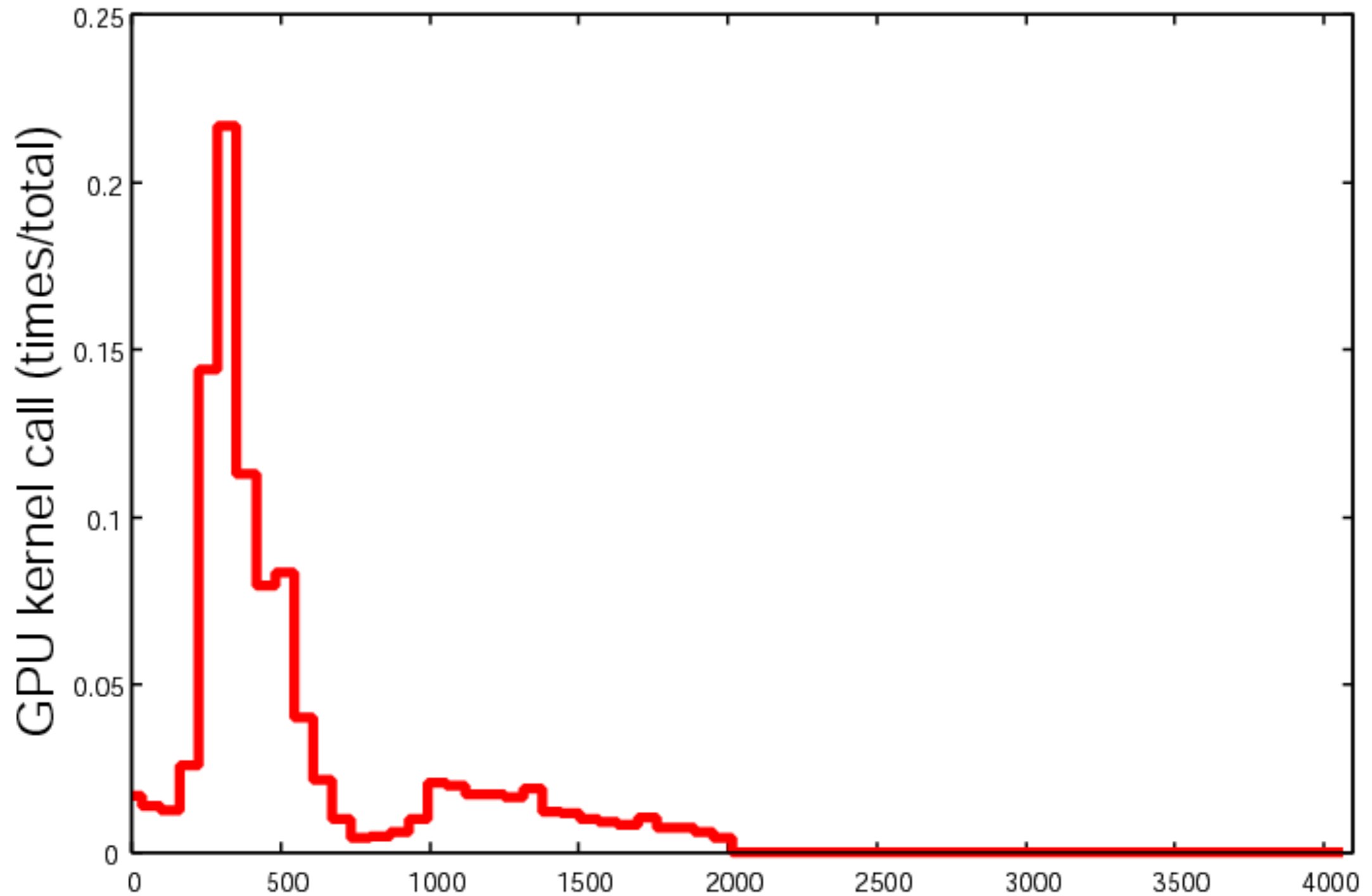
Direct sum.



Treecode

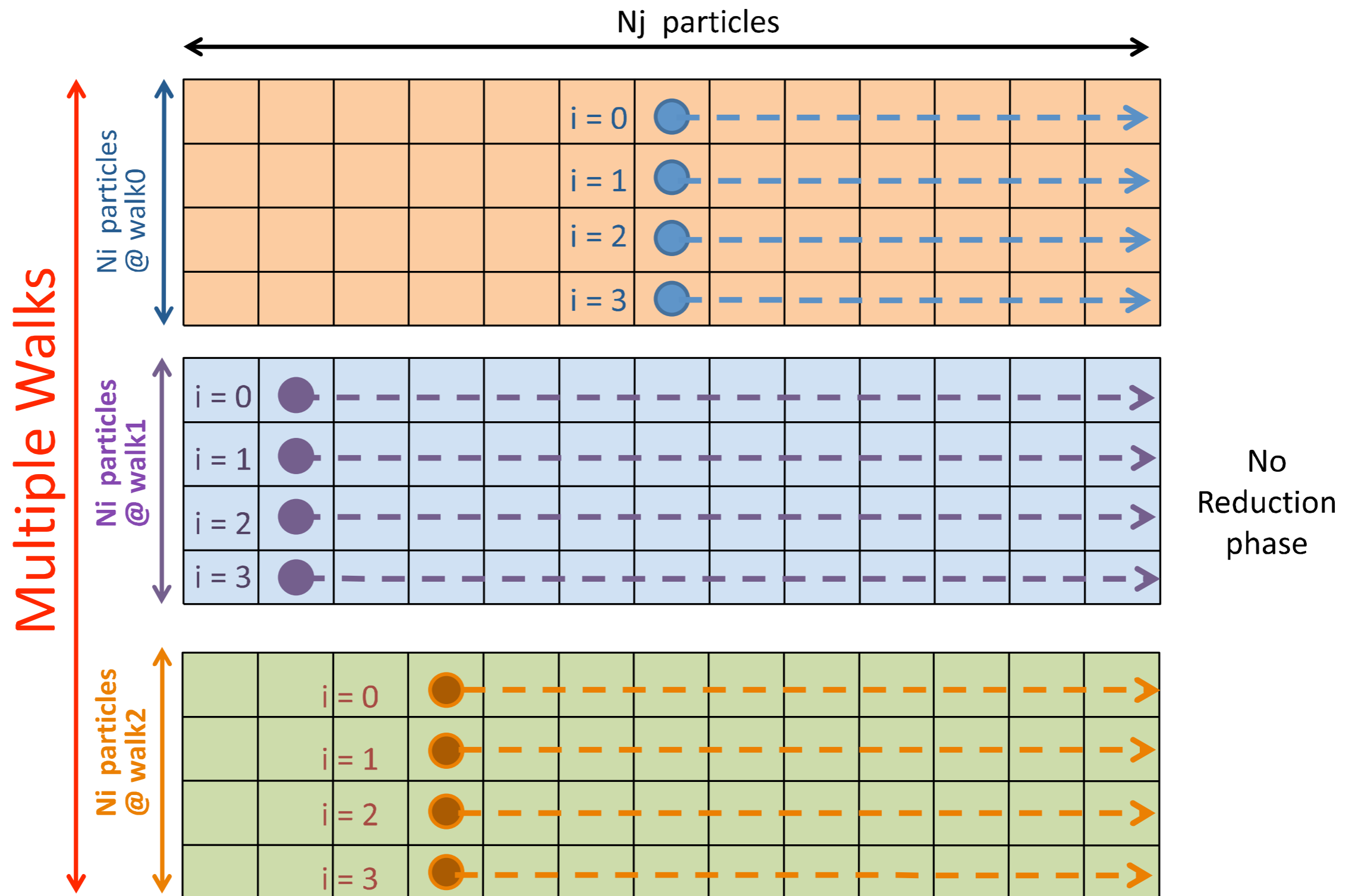


Number of working threads



number of particles per cell

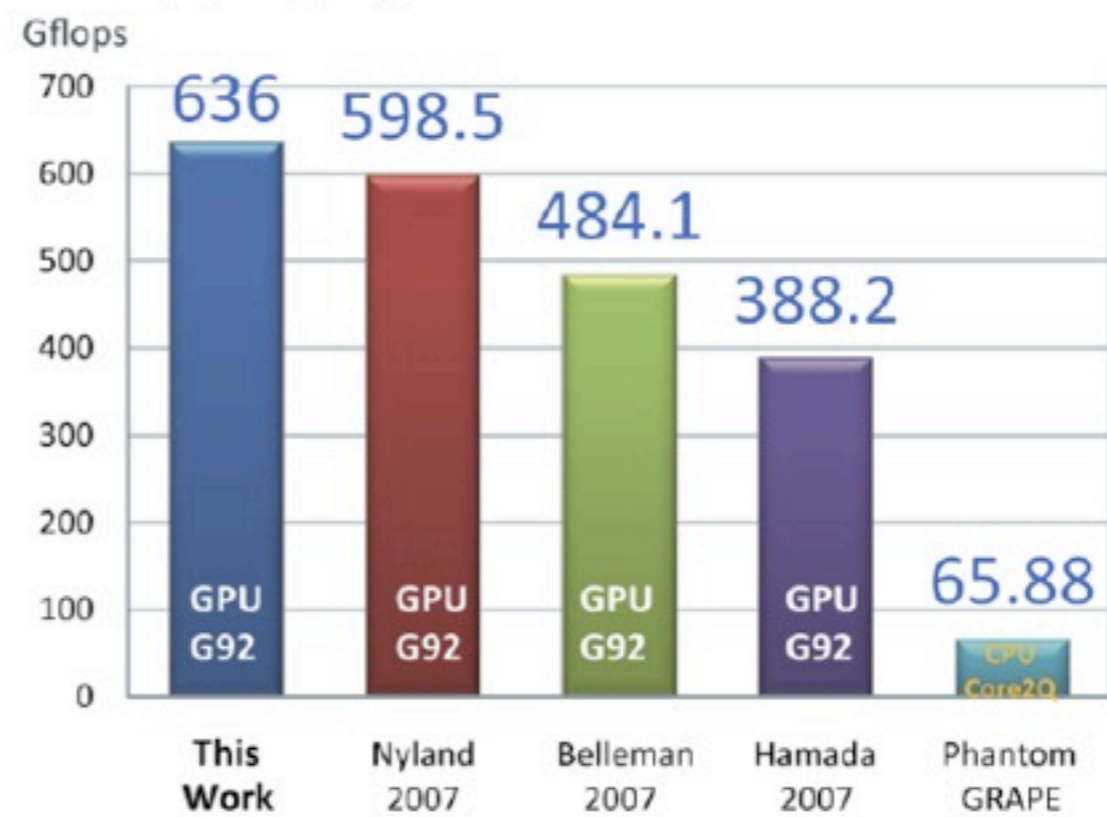
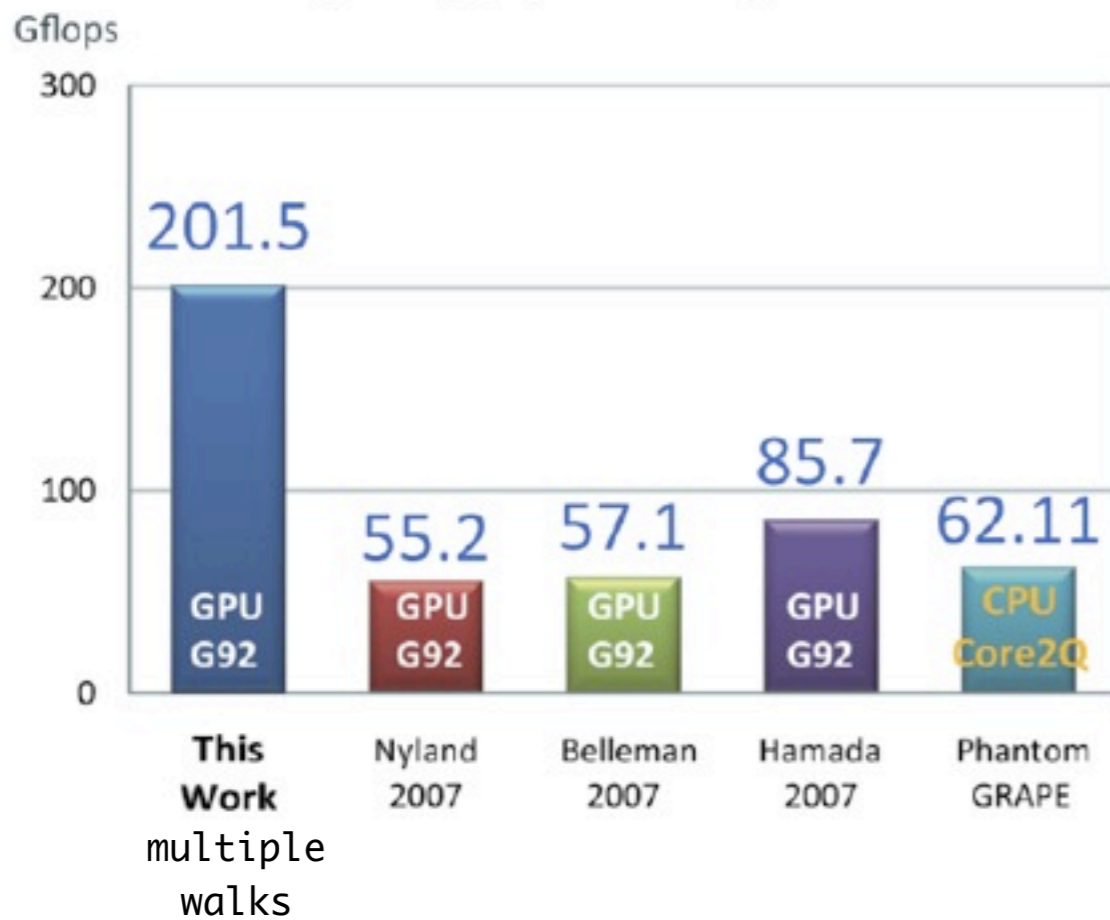
Multiple walks



Flops on GPUs (single node)

treecode

brute force



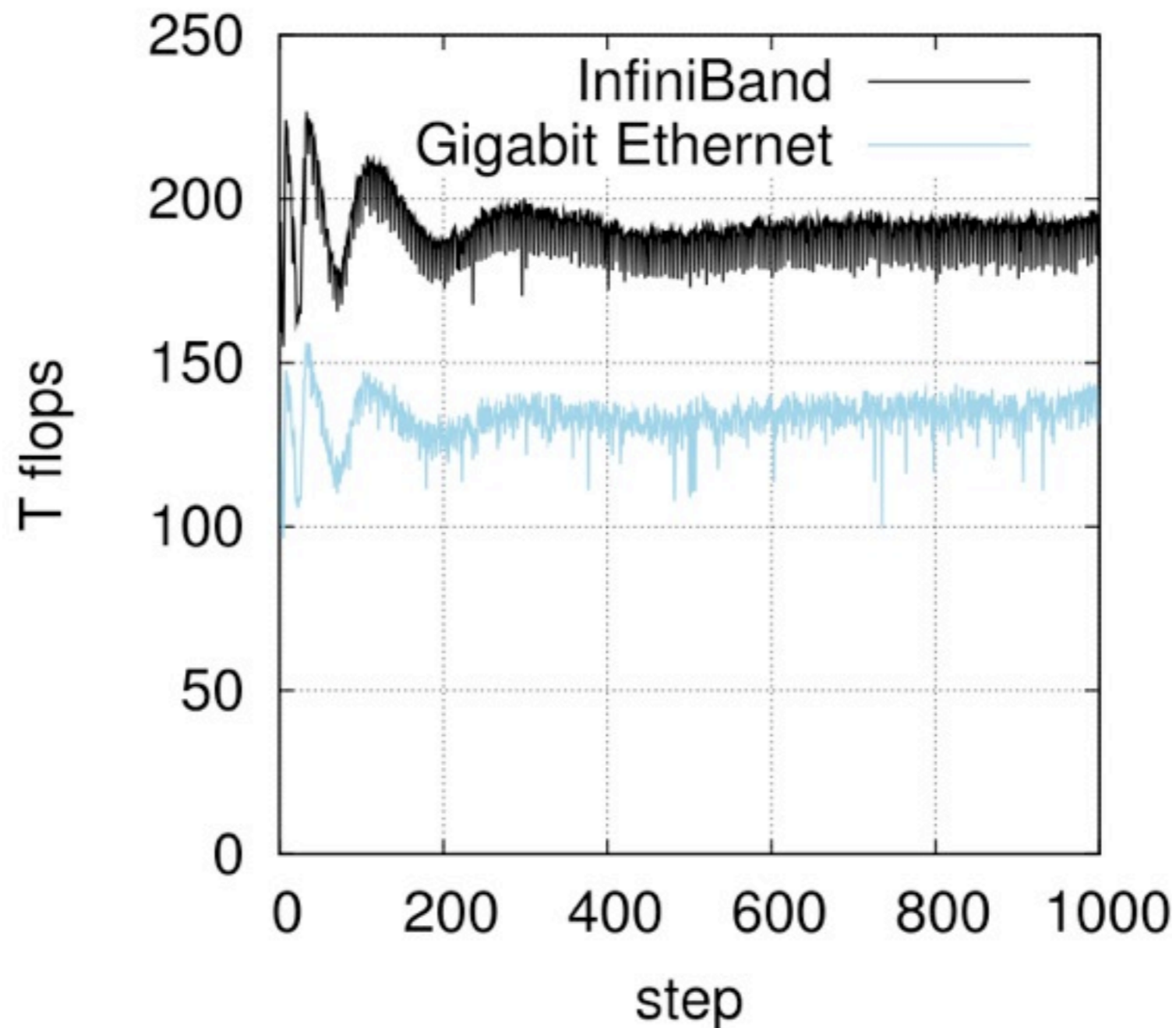
Parallel Performance

- Total # of particles
 - 3,278,982,596 particles
- Total # of interactions
 - $3.1e+16$
- Wall-clock time
 - 6,180 sec
 - time/step : 6.2 sec
- Particle advances per second
 - 528,868,160 particles step/sec

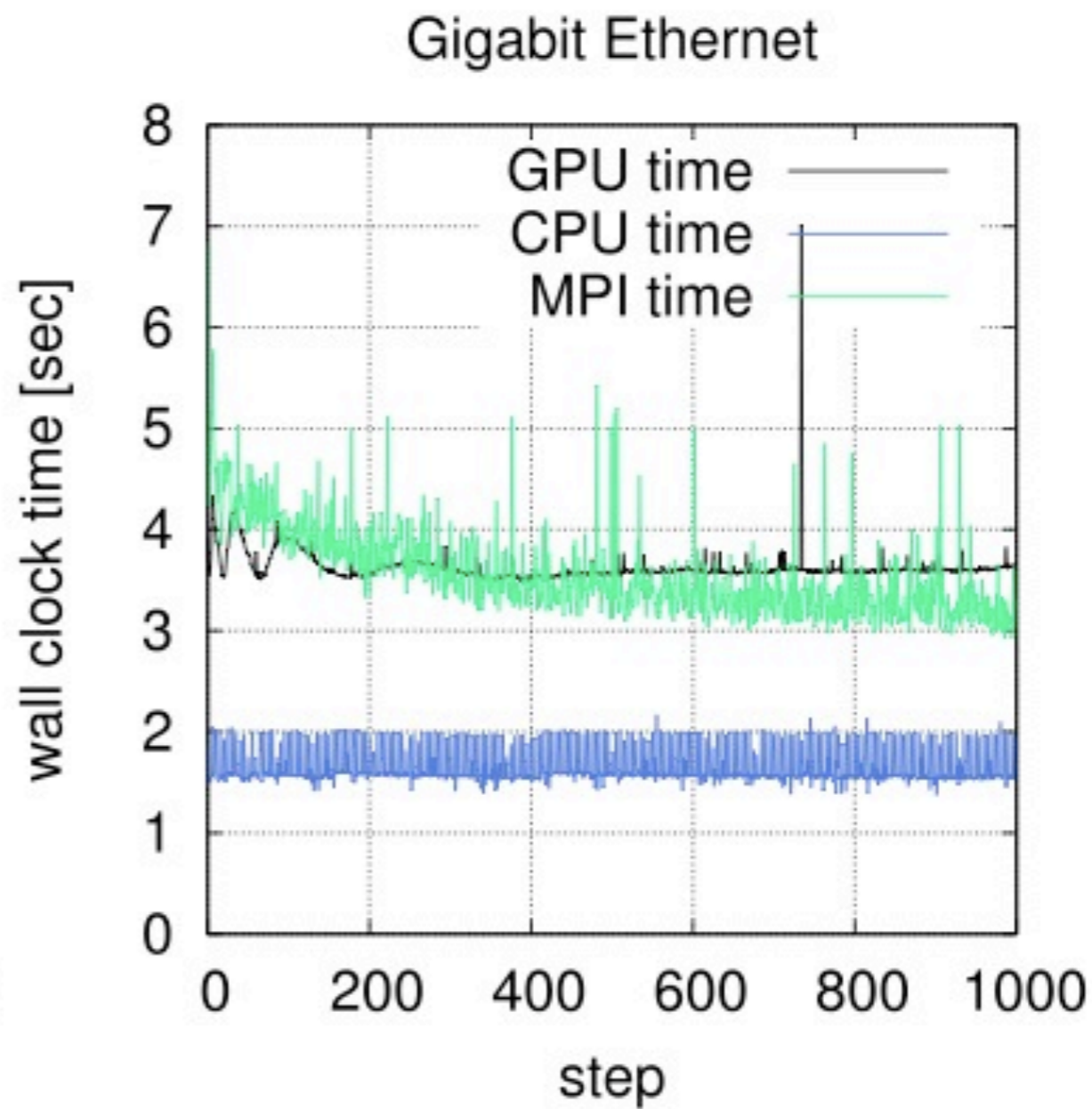
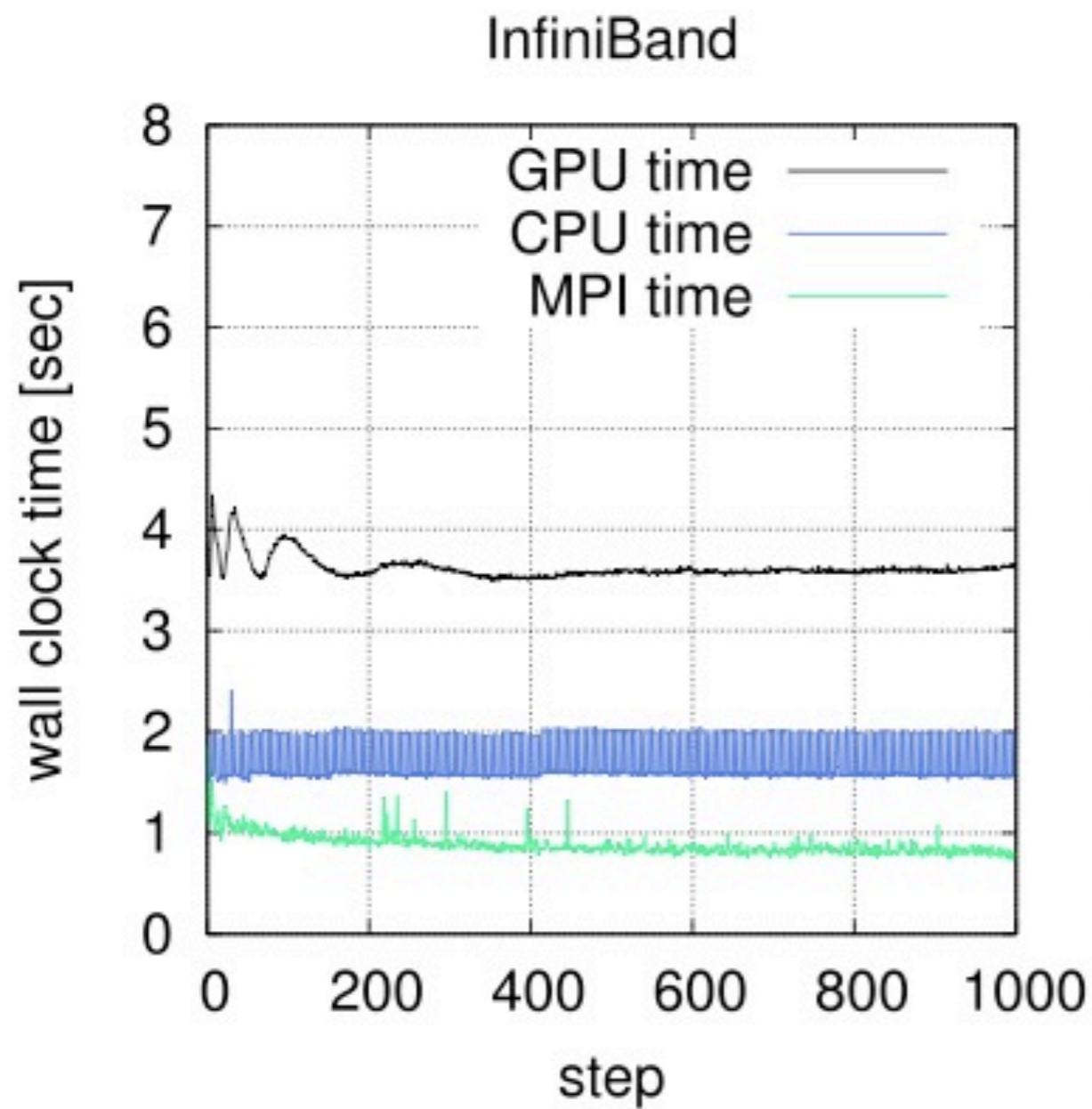
Raw (uncorrected) performance

🌐 190 Tflops average

🌐 Flop count per interaction : 38



Profile



Comparison to other works

- GreeM - Fine tuned treePM for X86(SSE) cluster/MPP
 - Ishiyama, Fukushige, Makino. 2009
 - Cray XT4
 - Particle advanced per second (per core) : 43,000
- Our treecode run
 - Particle advanced per second (144 node) : 528,868,160
 - Particle advanced per second (per node) : 3,672,696
- Our result is
 - **equivalent to Cray XT4 12,300 cores !**
 - 85 times faster (1 GPU node vs Cray XT4 1 core)
 - 12 times faster (144 GPU node vs Cray XT4 1k cores)



Mar 2007

Host: Xeon 3.0 GHz x 40

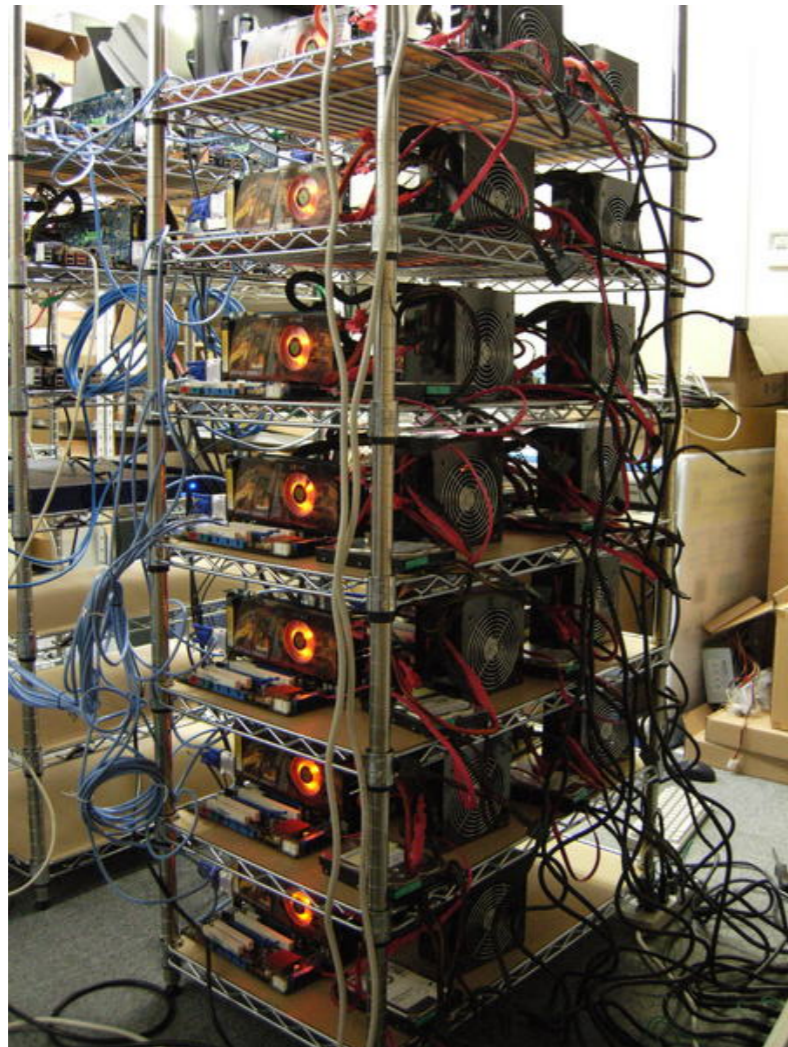
GPU: GeForce 8800GTX x 40



~ 1 Tflops in cosmology sim.

History of our GPU cluster

Mar 2008



Host: Core2Quad 2.4 GHz x 32
GPU: GeForce 8800GT x 32

History of our GPU cluster

April 2008

~ 20 Tflops in cosmology sim.



Host: Core2Quad 2.4 GHz x 128
GPU: GeForce 8800GTS x 128

History of our GPU cluster

Nov 2008



Host: Core2Quad 2.4 GHz x 128
GPU: GeForce 8800GTS x 256

~ 40 Tflops in cosmology sim.

History of our GPU cluster

Aug 2009



Power supply 600A -> 2000A


Host: Core2Quad 2.4 GHz x 166
GPU: GeForce 9800GTX+ x 256
GPU: GeForce GTX295 x 33

Never give up

● challenge, challenge, challenge



2010年11月11日木曜日

A server room with blue lighting and rows of server racks. The racks are filled with server units, and the room is dimly lit with blue light emanating from the racks and floor. The perspective is looking down a long aisle between the racks.

出島

DEGIMA
cluster



出島

DEGIMA

cluster



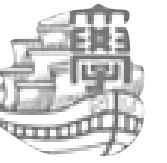
T. Hamada



T. Hamada



T. Hamada



Thank you

