国立大学法人
長崎大学
NAGASAKI UNIVERSITY

A Novel Multiple-Walk Parallel Algorithm for the Barnes-Hut Treecode on GPUs

- Towards Cost Effective, High Performance N-Body Simulation

Tsuyoshi Hamada (Nagasaki University, Japan),
Keigo Nitadori (RIKEN, Japan)

K. Benkrid(Edinburgh Univ.), Y. Ohno, G. Morimoto, (RIKEN)
T. Masada, Y. Shibata, K. Oguri, (Nagasaki Univ.) ,
M. Taiji (RIKEN)

Introduction

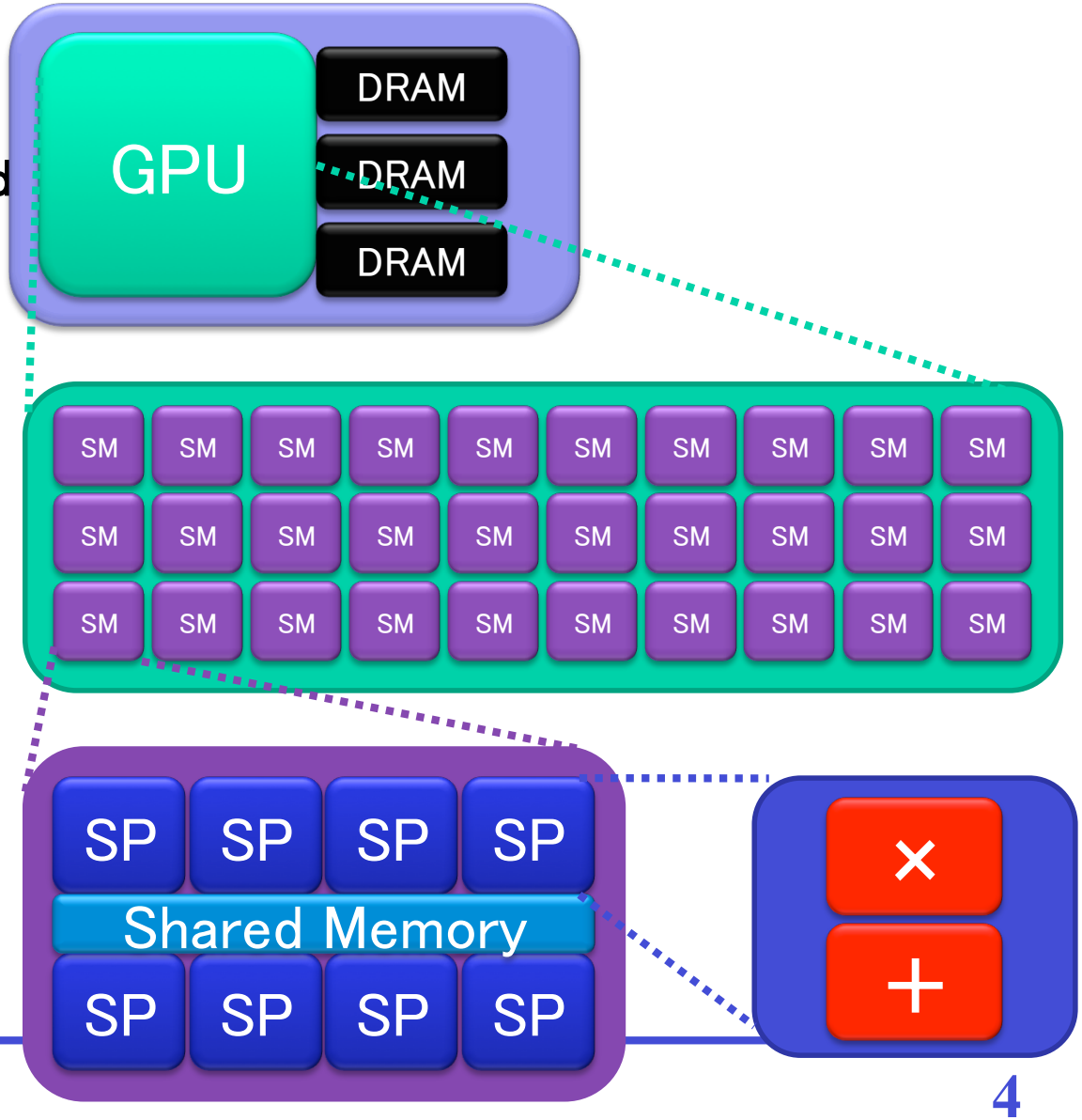
Why GPU ? Why N-body ? Why GPU cluster ?

国立大学法人 長崎大学 NAGASAKI UNIVERSITY

- GPU computing will be a trend for HPC.
  - Transistors are cheap
  - Network is expensive
- N-body simulation
  - Convolution between particle-particle interactions
- N-body simulation is a killer application for GPU Cluster.
  - Compute intensive
  - Network bandwidth is not so important (relatively)
- N-body simulation is not for Astrophysics, for general-purpose !!
  - Fluid Dynamics (SPH, Vortex method, etc)
  - Acoustics, electromagnetic (BEM)

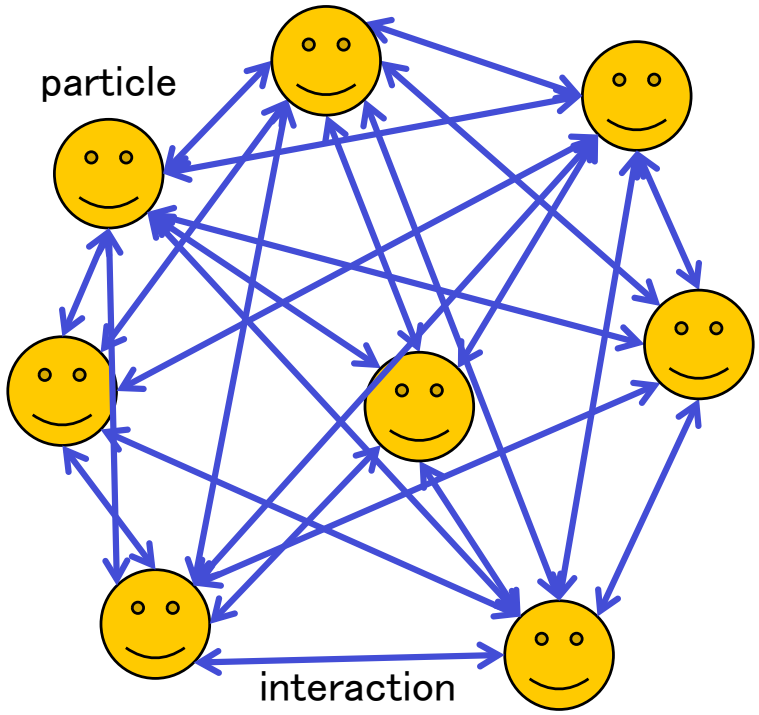
NVIDIA GPU(GT200) Architecture

- **GPU board**
  - 1 ～2 GPUs per board
- **GPU**
  - 30 SM (Streaming Multiprocessor) per GPU
- **SM, SP**
  - 8 SPs per SM (SP: Stream Processor)
  - FMAD per SP



N-body simulation

- Particles are interacting with each other
  - Particle
    - stars (star cluster)
    - galaxies (cluster of galaxies)
    - atom s(protein, metal, crystal, etc)
- Computation cost
  - $O(N^2)$ in naive algorithm



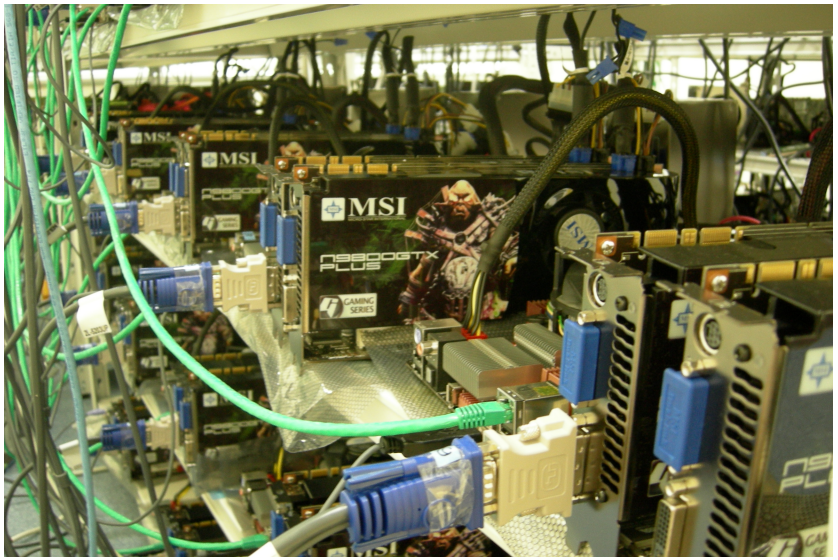
GPU computing at Nagasaki University

- Motivation
  - Accelerating billions of particle simulations such as
    - ✓ Large-scale Cosmological N-body simulation
    - ✓ Large-scale molecular dynamics simulation

    using cost-effective hardware.

国立大学法人
長崎大学
NAGASAKI UNIVERSITY

- Target code
  - PPPM (Particle-Particle Particle-Mesh)
  - PME (Particle Mesh Ewald)
  - TreePM (Tree Particle-Mesh)
  - FMM (Fast Multipole Method)
  - BEM (Boundary Element Method)



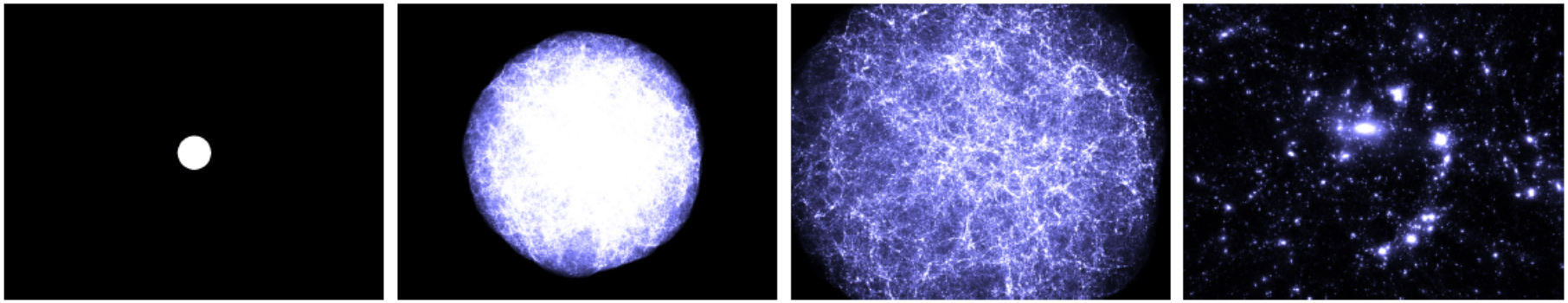
NGC 190T (Nagasaki GPU cluster)

国立大学法人 長崎大学 NAGASAKI UNIVERSITY

- PC cluster of 256 GPUs
  - 190 Tflops (SP) peak (Nov 2008)
  - Cheap network (48 port GbE switch x4)
  - Running from May 2008
  - DIY

Application I on the Nagasaki GPU cluster

- Cosmology (large scale structure of Universe)

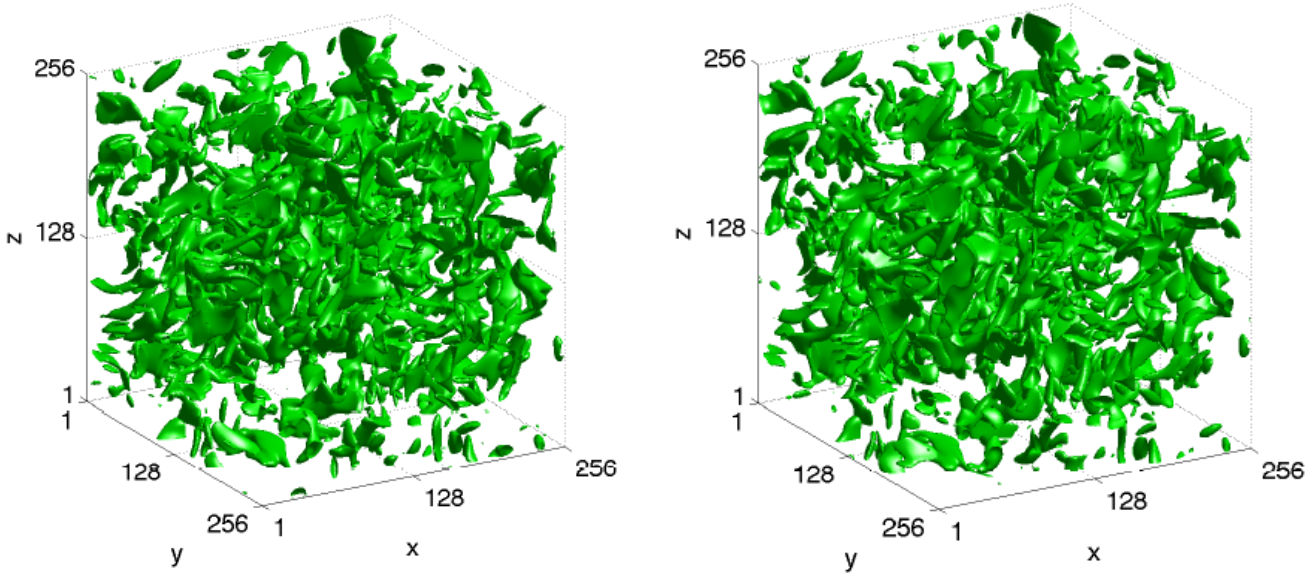


  - Barnes-Hut tree using 256 GPUs
  - 1.5 Billion particles
  - 17 sec / time-step

Hamada et al., ISC09 (in press)

Application II

- Simulation of turbulence



  - Vortex method
  - FMM(Fast Multipole Method) using 256 GPUs

Hamada et al., SC09(submitted)

Application III

- Simulation of Electromagnetic
  - BEM (Boundary Element Method) for Helmholtz' equation on GPU
  - Quasi double-precision technique

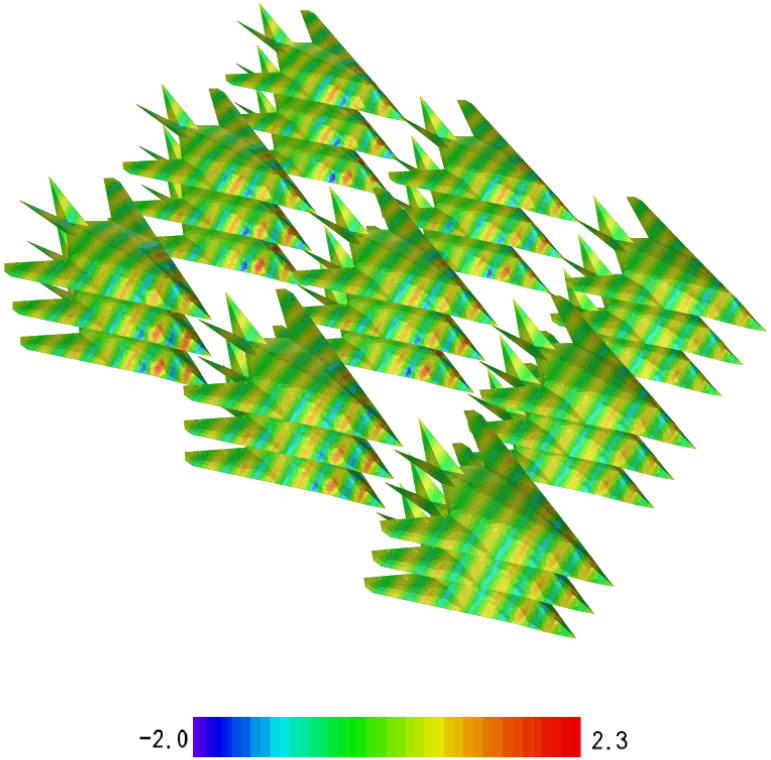
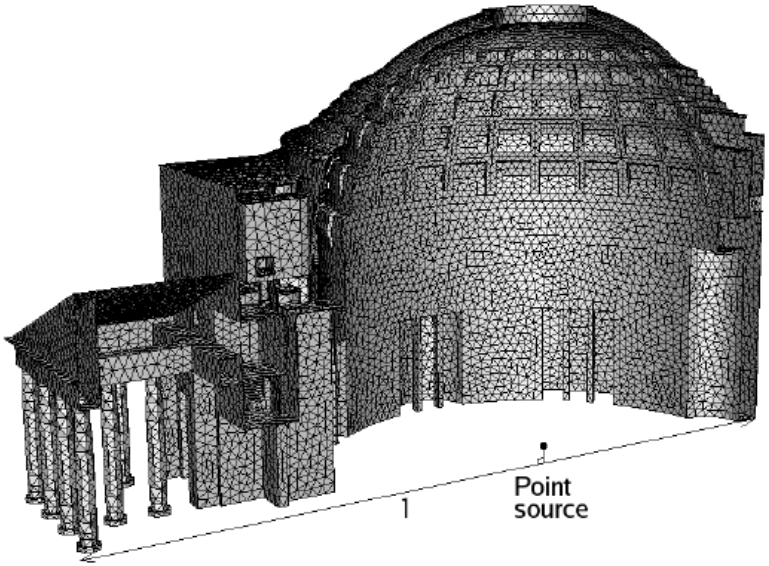


Figure 7. Distribution of Im u on fighters

Takahashi and Hamada, Journal for Numerical Methods in Engineering(in press)

Application IV

- **Acoustics analysis**



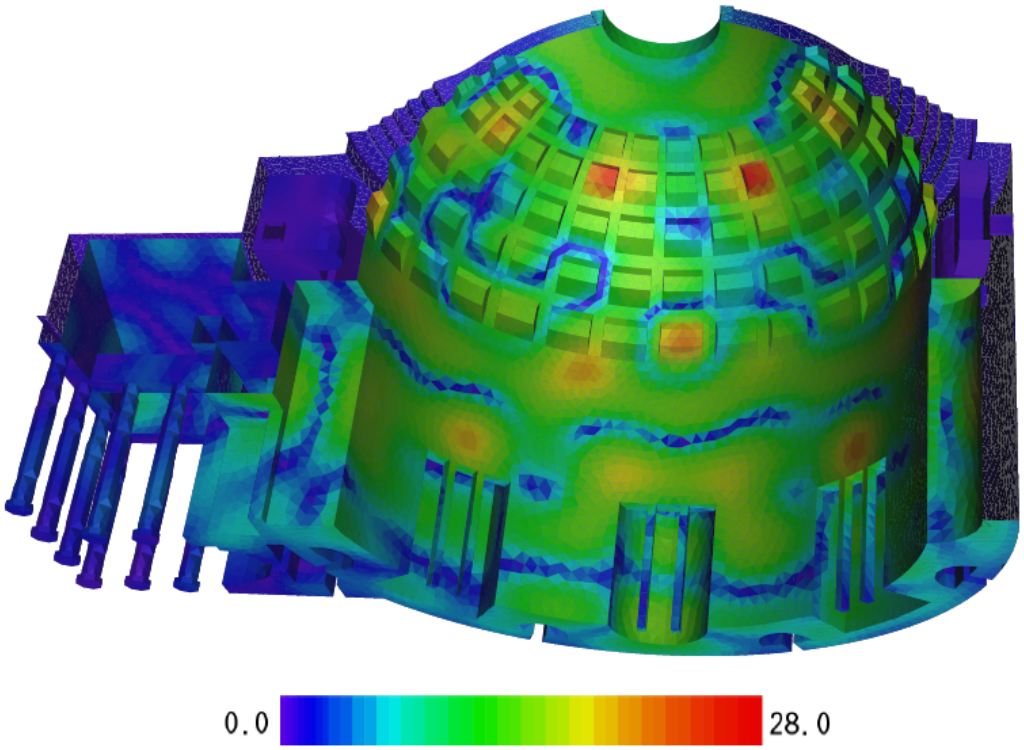
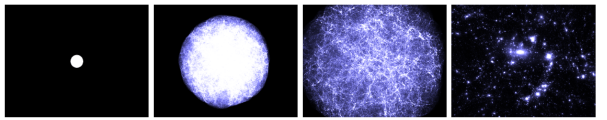


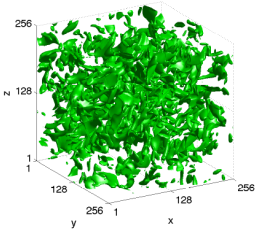
Figure 9. Distribution of $|u|$ on rotunda (half part; view from bottom side)

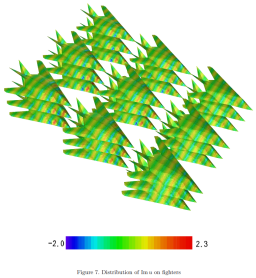
Takahashi and Hamada,
Journal for Numerical Methods in Engineering(in press)

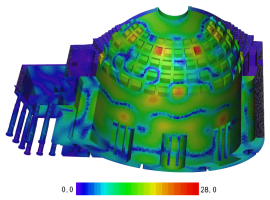
N-body simulation is not only for Astro.

- Application I : Cosmology
  - Barnes-Hut treecode … N-body
- Application II: Turbulence
  - Vortex Method with FMM … N-body
- Application III: Electromagnetic
  - Boundary Element Method … N-body
- Application IV: Acoustic analysis
  - Boundary Element Method … N-body









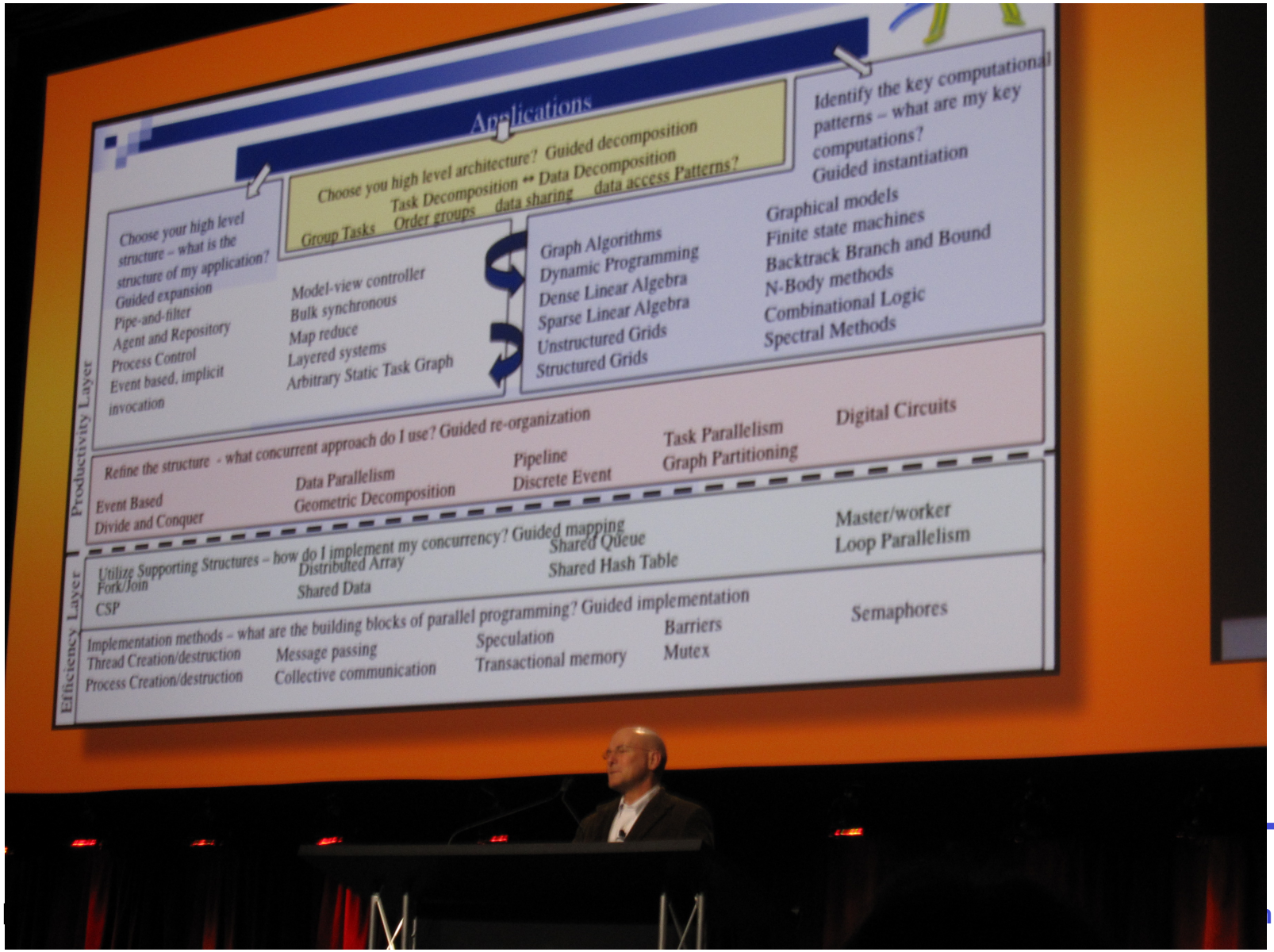
Same Mathmatics, Analogy, Scheme can be applied !

N-body is important for HPC

- Analogy for N-body is general-purpose
  - Mathmatics for N-body
  - Implementation Scheme for N-body
  - Hardware/Software tuning for N-body

I'm not only the man says N-body is important ^∞

- **David Patterson also said same thing**



Patterson
SC08

And I'm one of the Dwarfs in HPC.

- D. Patterson(left) and me(right) at SC08

Why astrophysics.

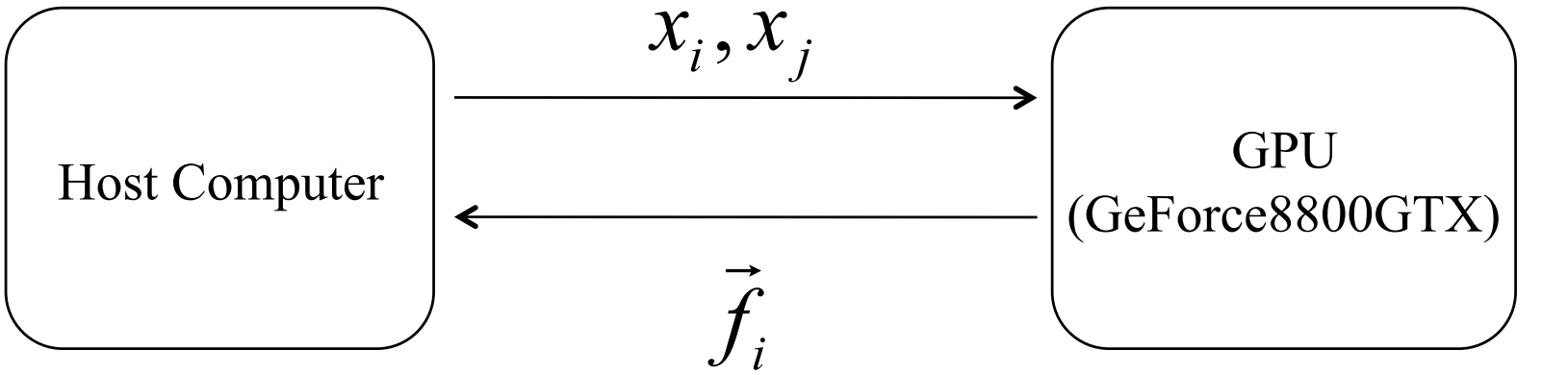
- N-body methods is important for HPC
- Astrophysics tells us many important things
  - List of schemes used in astrophysical N-body simulations
    - Individual Time-step
    - Block Time-step
    - Neighbor scheme
    - Barnes-Hut Tree codes
    - Fast Multipole Methods
    - Particle-mesh codes
    - Adaptive Mesh Refinement method
    - Self consistent field methods
    - P3M and PM-Tree codes
    - Celestial mechanics codes
    - Grid based solvers for the Collisionless Boltzmann Equation
    - Fokker-Planck and Monte Carlo methods

So, we should start the study from astrophysics

Astrophysics N-body with GPUs

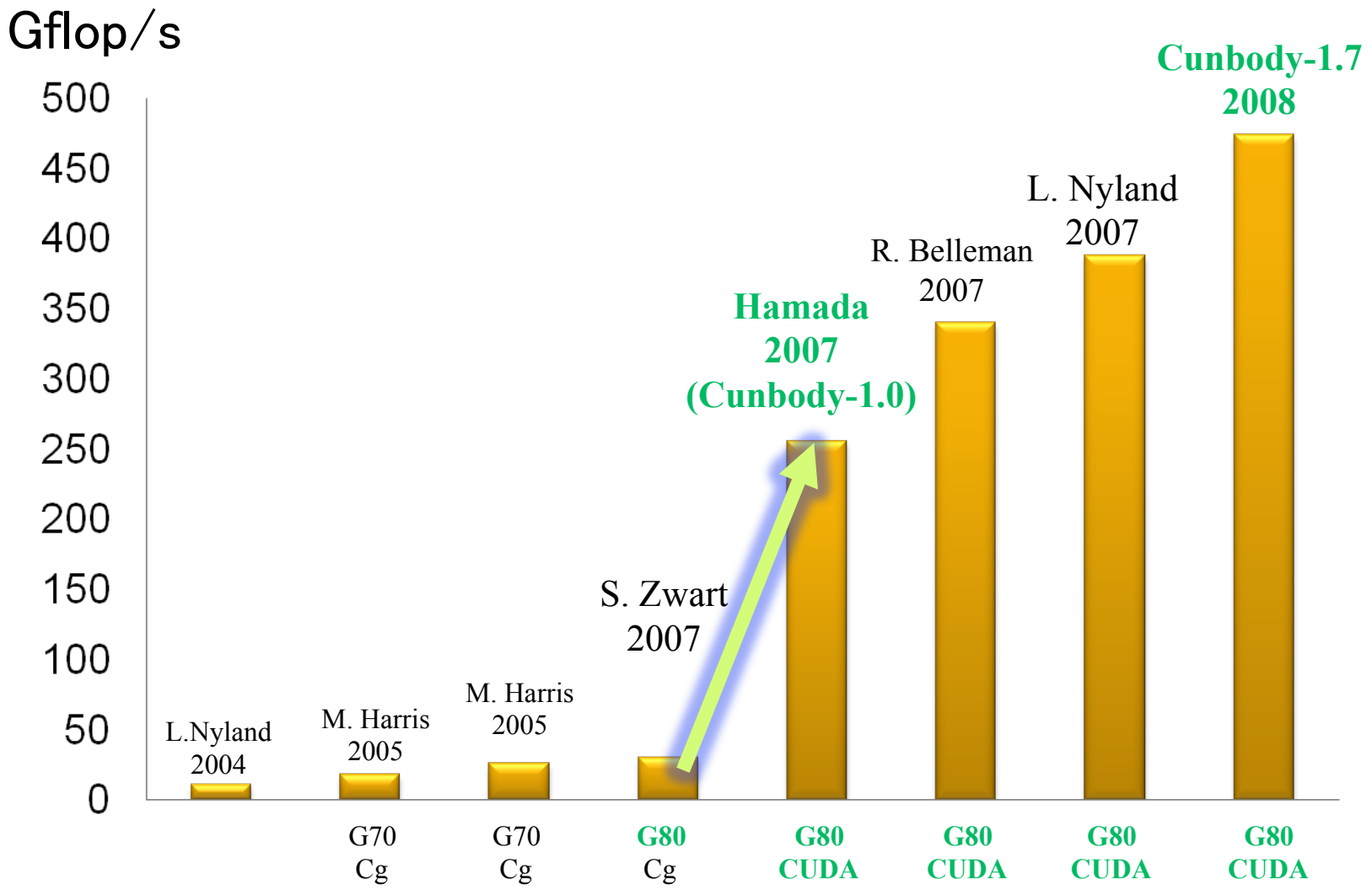
The analogy for GPU N-body for Astro.

$$\vec{f}_i = \sum_{j \neq i} \vec{g}(\vec{x}_i, \vec{x}_j)$$



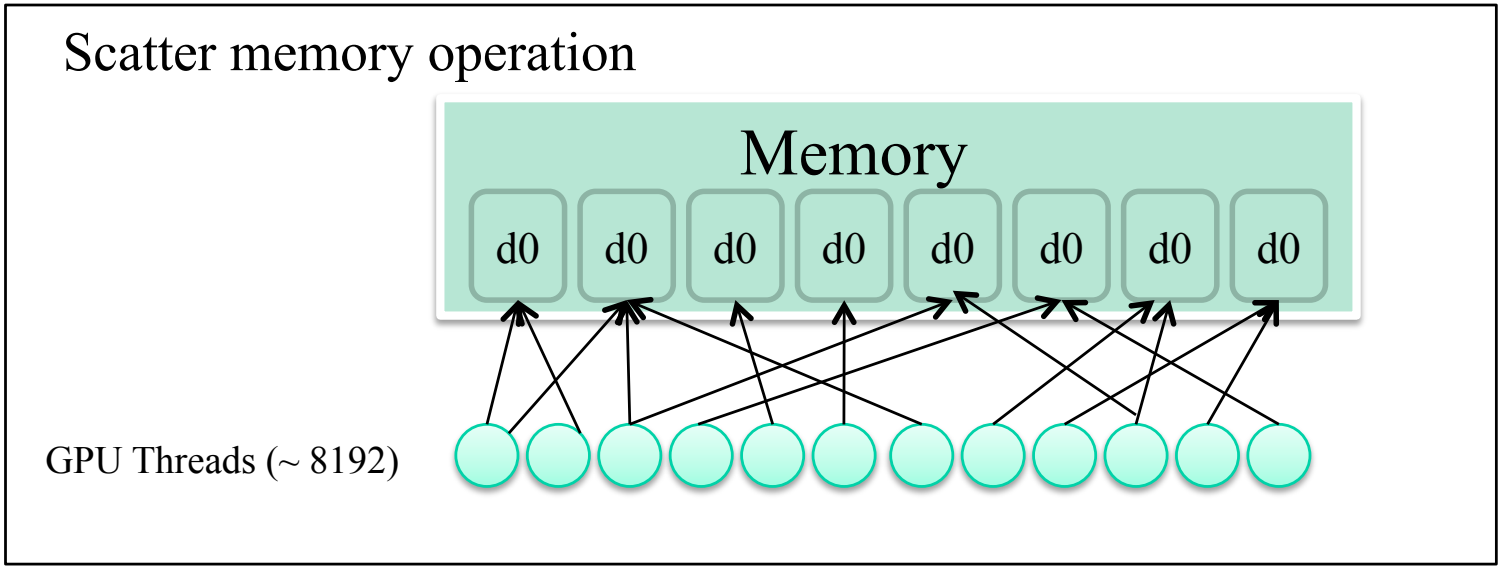
The basic analogy is almost the same as GRAPE systems (but a little different – today's talk)

Brute force N-body with GPU



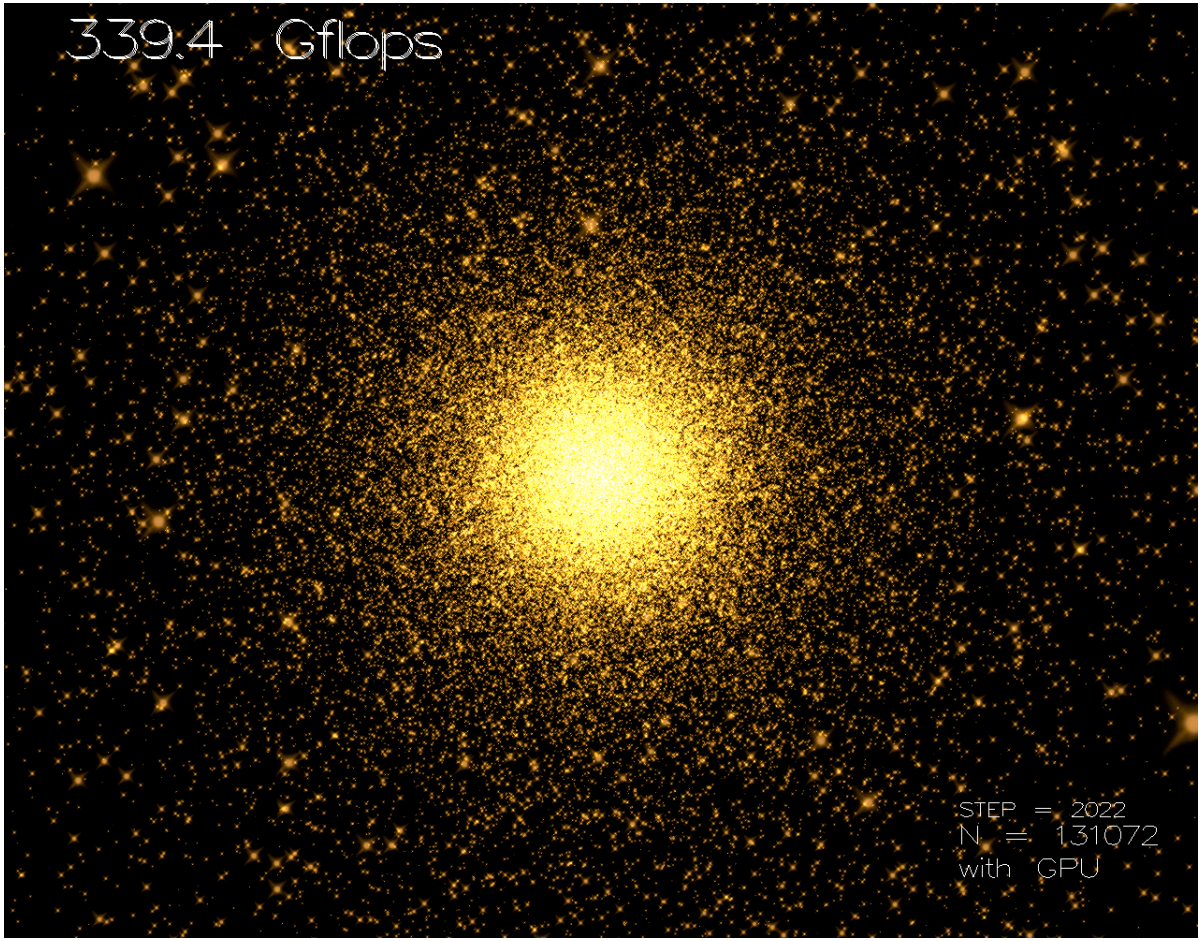
The key to breakthrough – GeForce8800GTX + CUDA

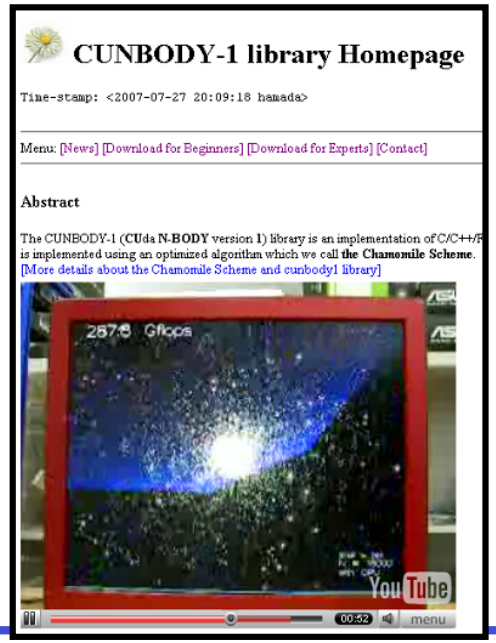
- GeForce8800GTX
  - **~10000** threads
  - **518** Gflops peak
- Using CUDA software
  - We can use **scatter memory operation** on GPU.



Practice : a simple test – direct summation

- GeForce8800GTX + Core2Duo E4400
- 131 k particles
- 2000 shared time steps
- 1 hours
- about **474** Gflops

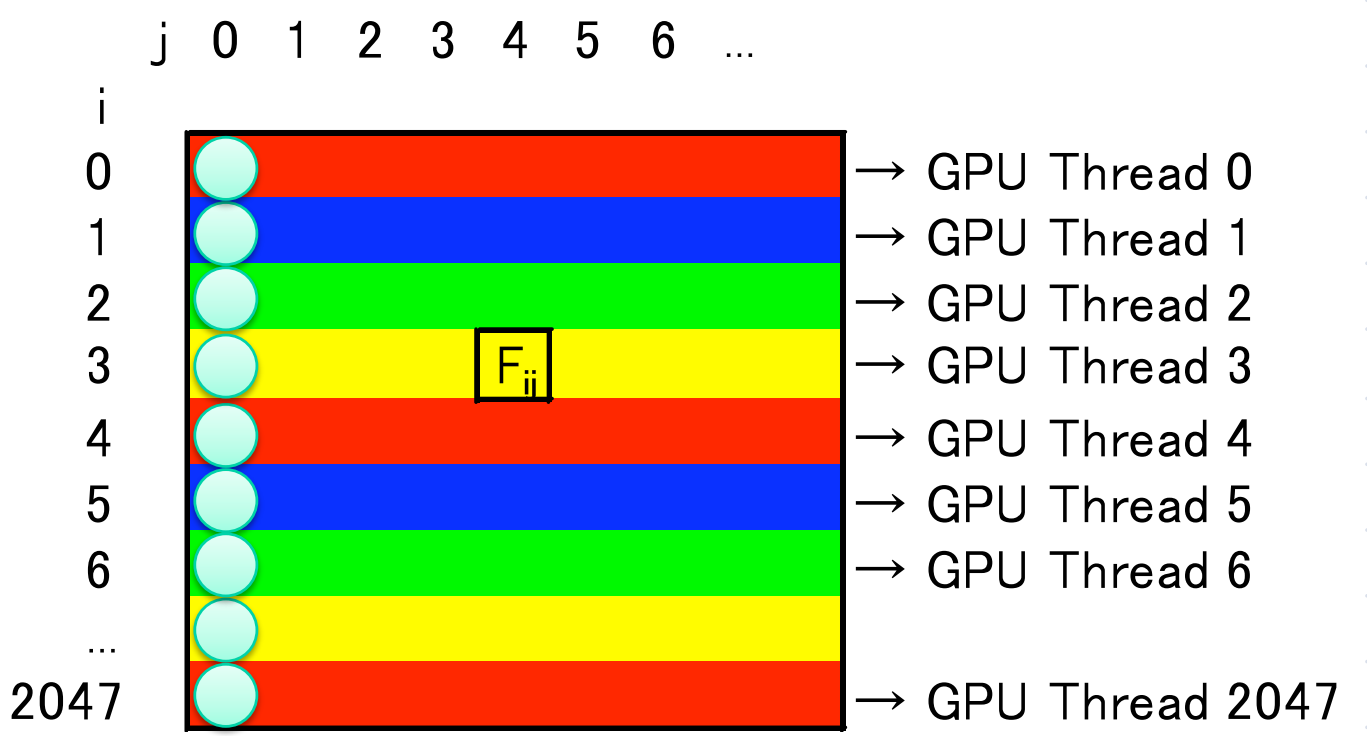




http://progrape.jp/cs/ or YouTube

Plummer sphere, 131072 particles

Details of parallelization (Brute-force)

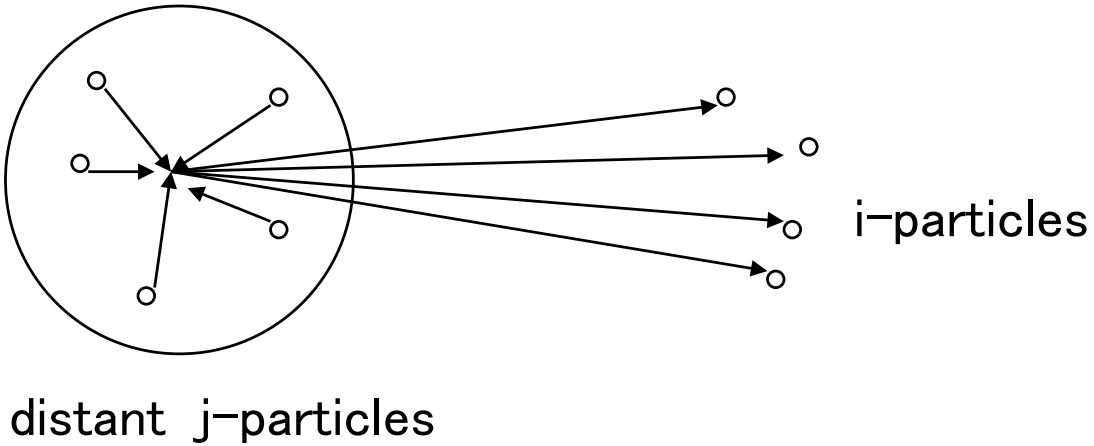


All 2048 threads are calculating forces on different i-particles

→ Particle Decomposition (i-parallelization)

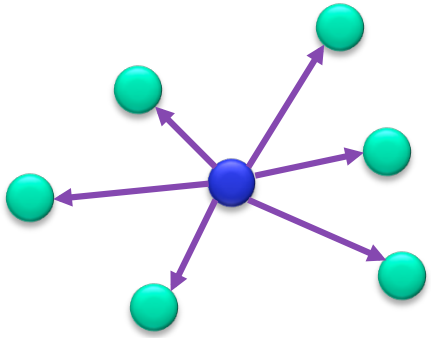
Practice: Hierarchical Tree Algorithm

- O(N log N) scaling with particle number
  - particles with a long distance are grouped
  - using hierarchical oct-tree structure
  - faster than direct summation for large number of particles



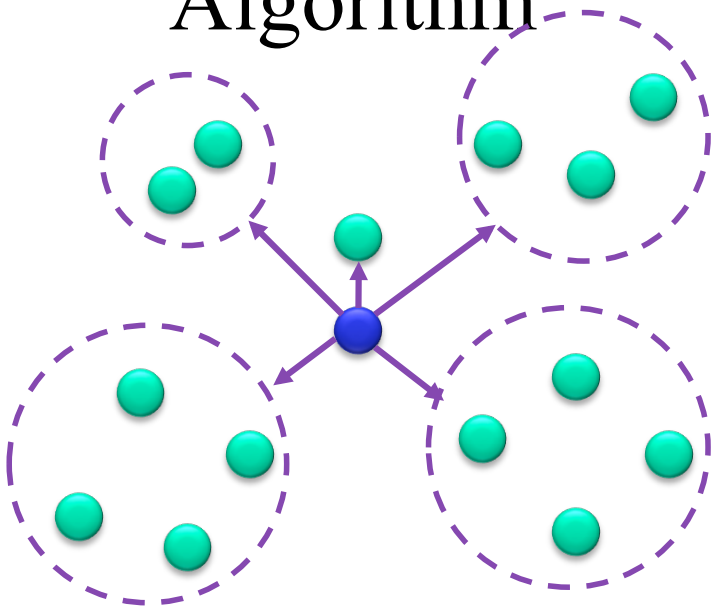
Calculation cost of N-body Simulation

Direct Summation Algorithm



$O(N^2)$: Small

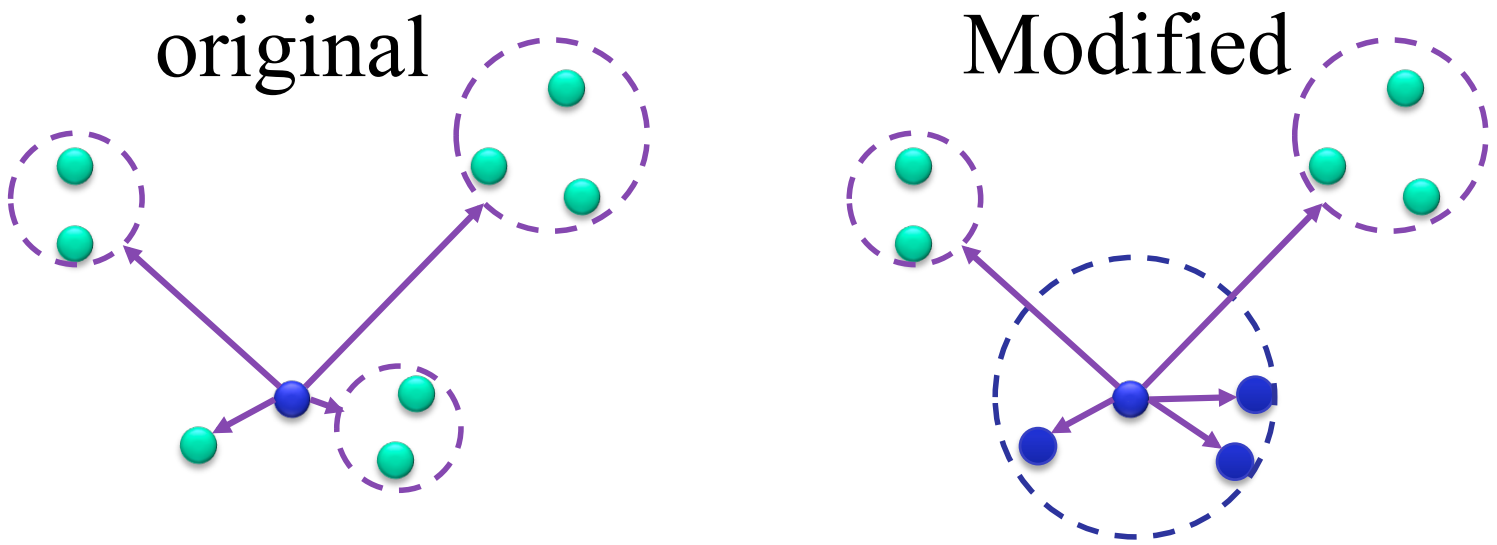
Tree Algorithm



O(N logN): Large

A “Grand Challenge” Problem in HPC

Modified Tree Algorithm
– Barnes' vectorization

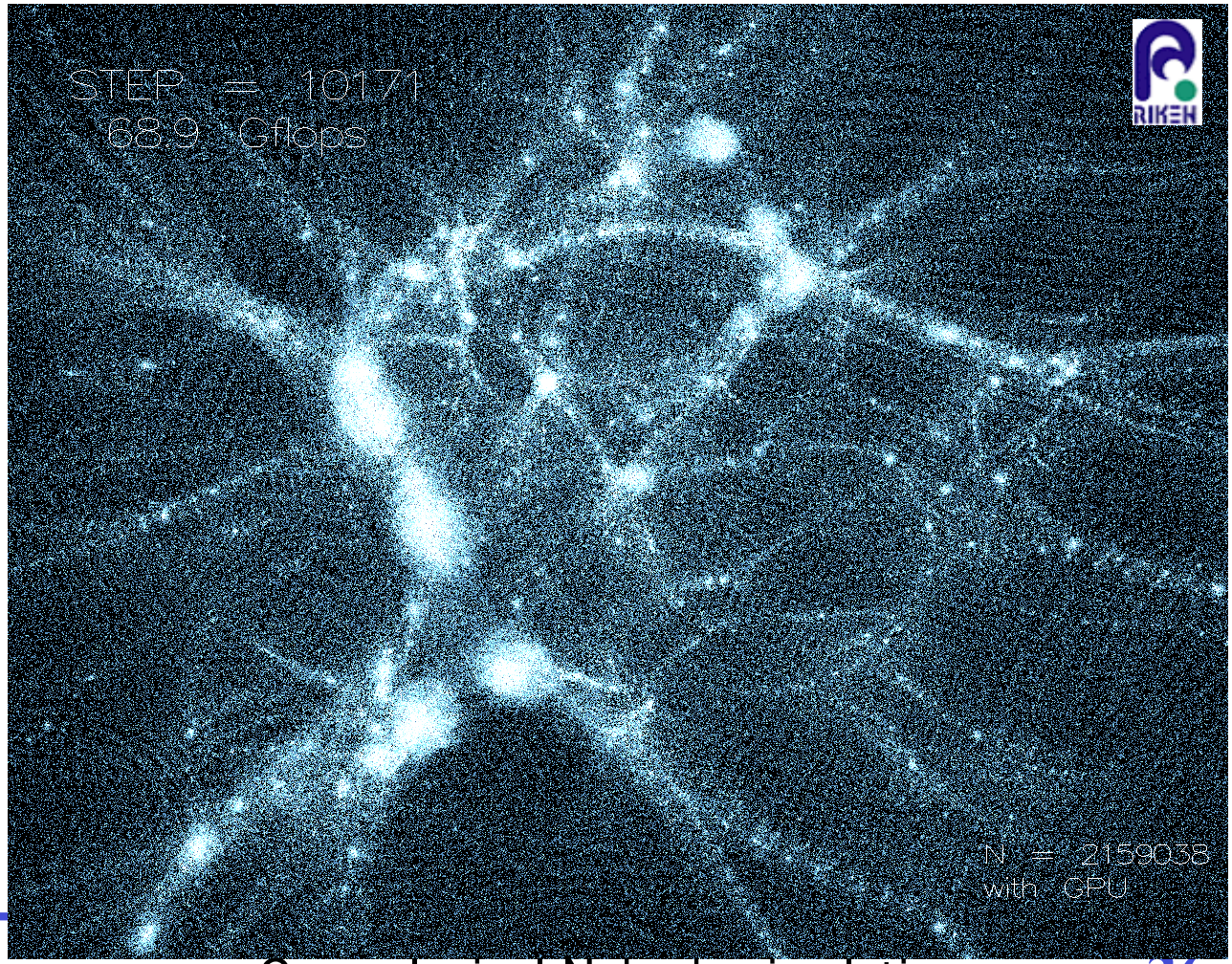


One interaction list is shared among n_g particles.

	Original	Modified
List creation cost on the host	N	N / ng
Interaction list length	shorter	longer

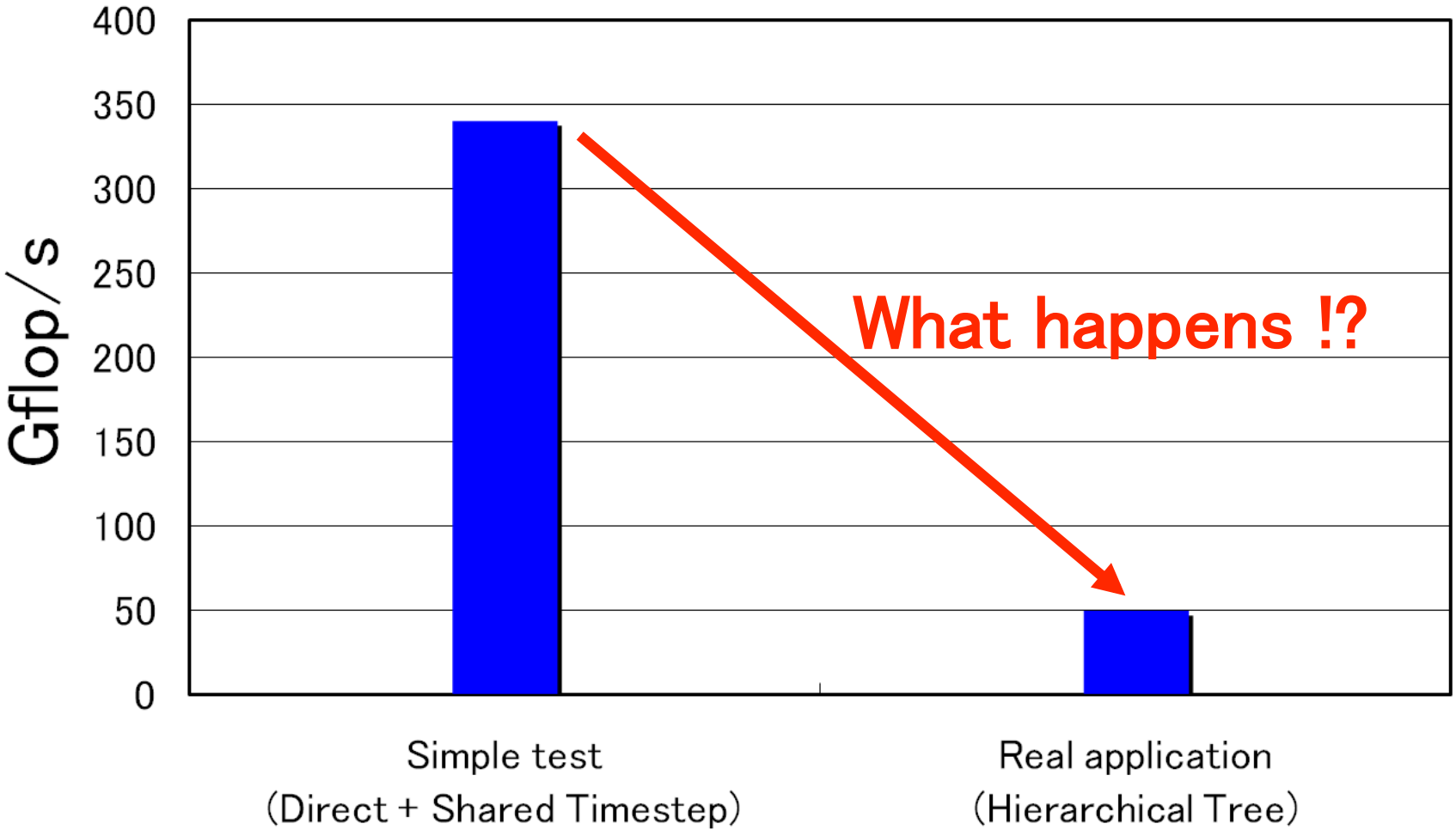
Practice : more complex test
– Hierarchical Tree Algorithm

- GeForce8800GTX + Core2Duo E4400
- **2M** particles
- 1000 time steps
- 2 hours
- about **70** Gflops

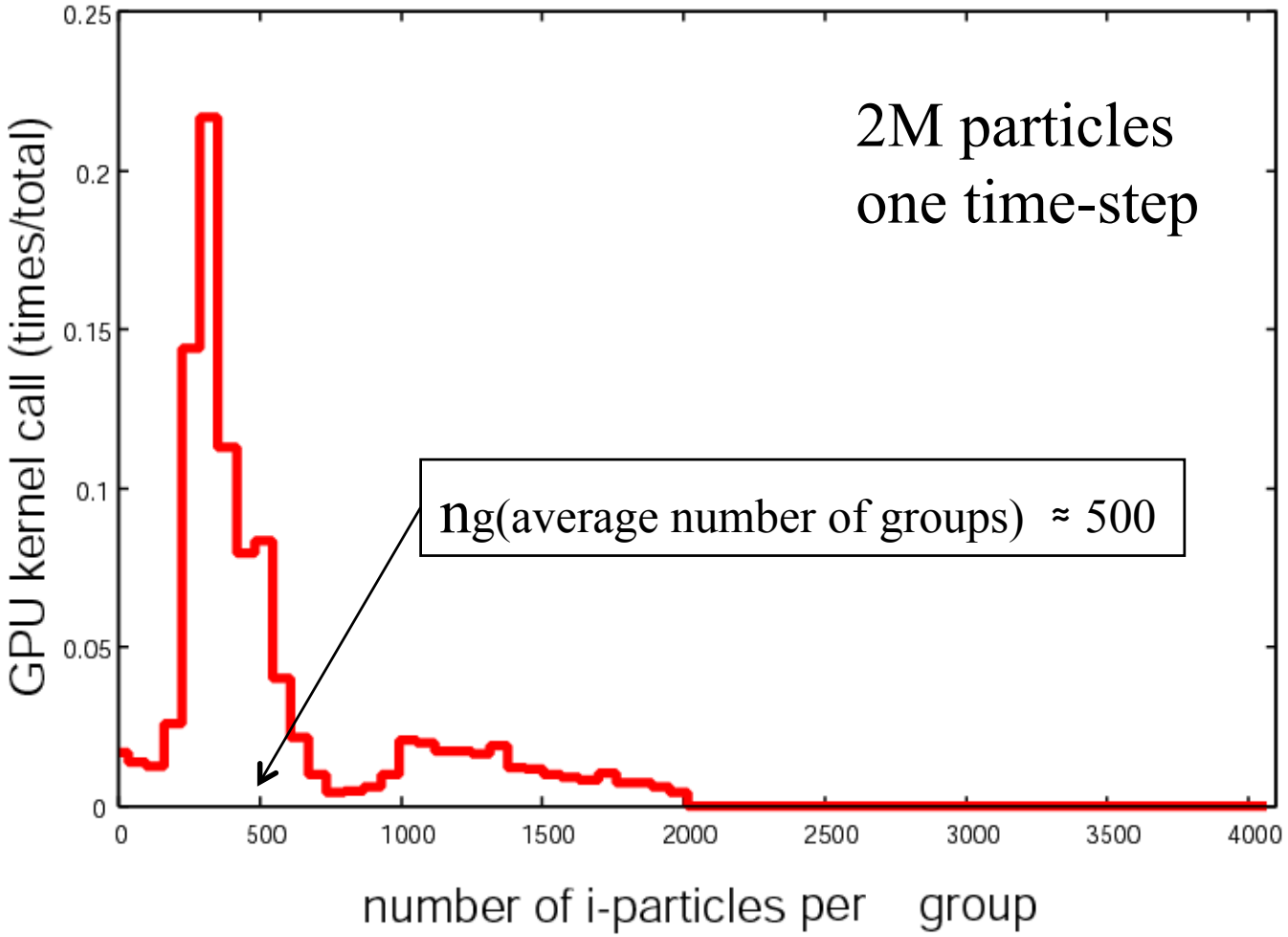


Cosmological N-body simulation

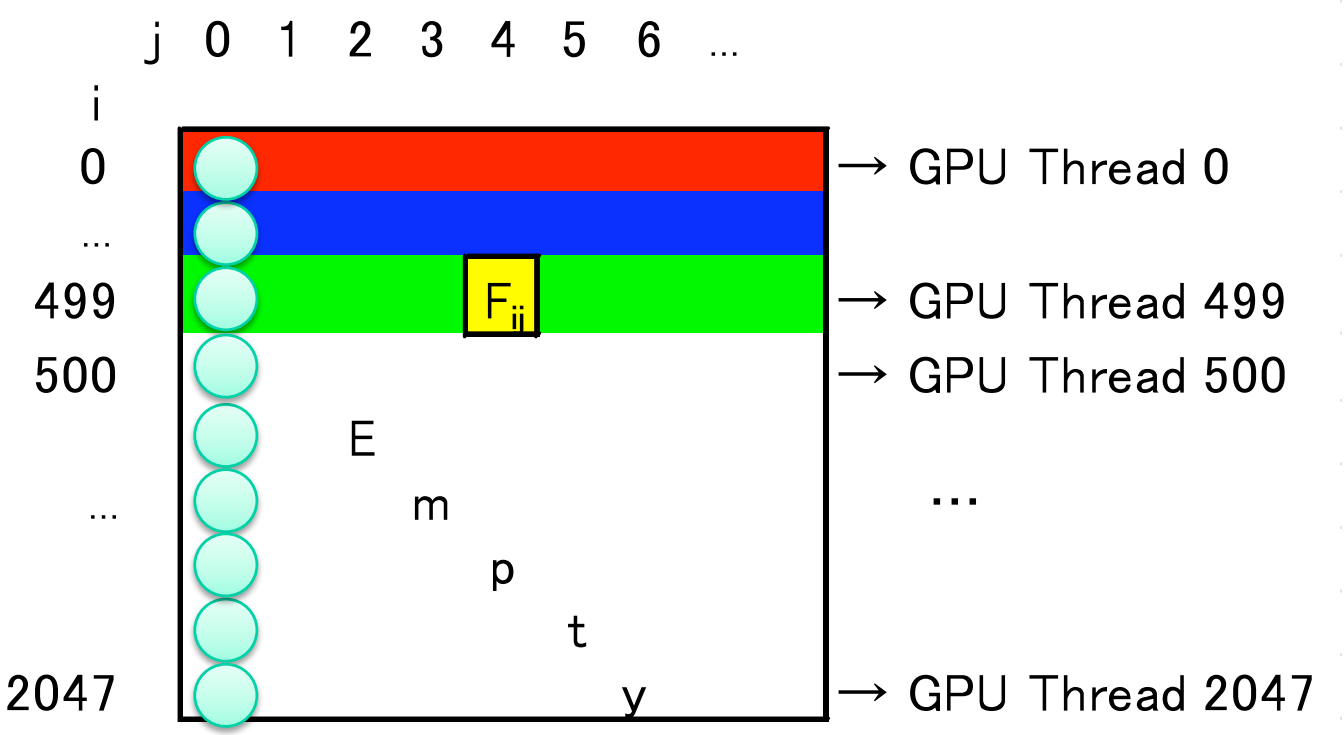
A problem in a real application



The performance problem in GPU treecode



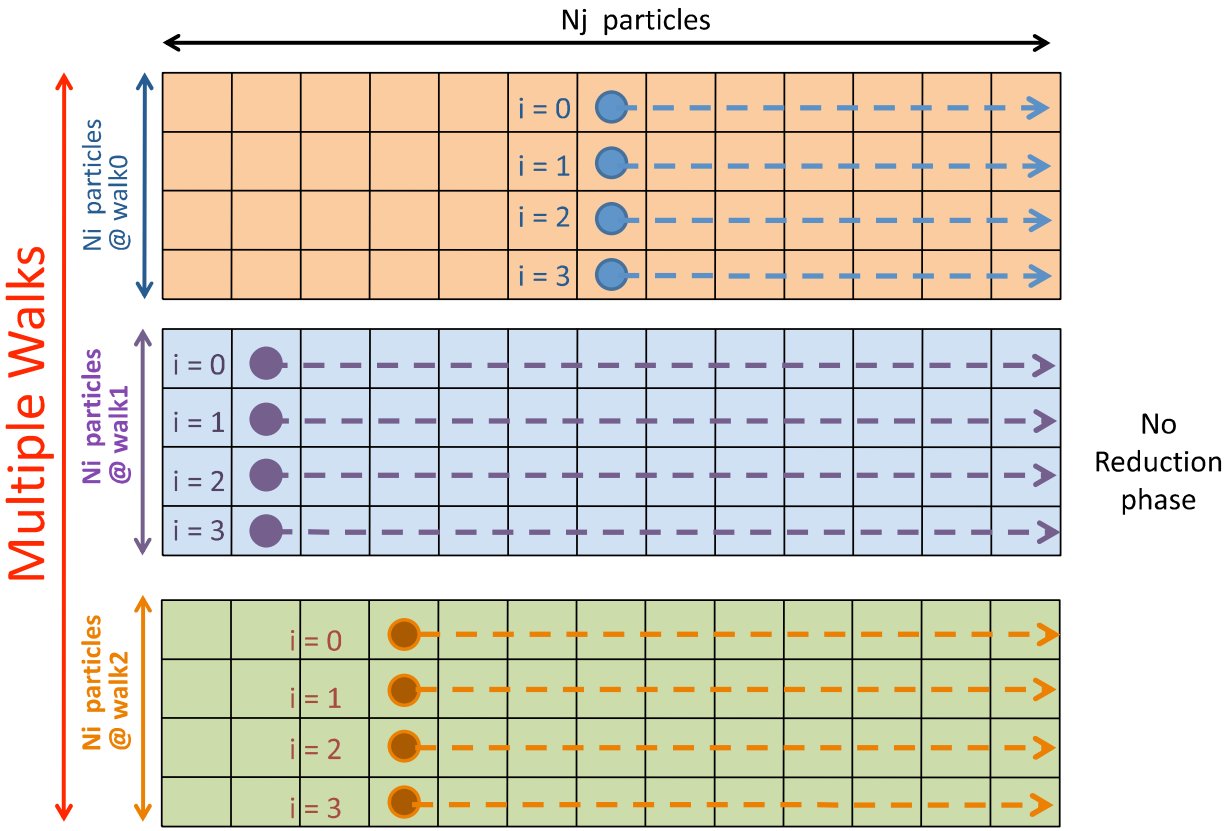
of i particles ≈ 500, # of threads = 2048



About 1500 threads are sleeping

The Multiple-Walk Scheme

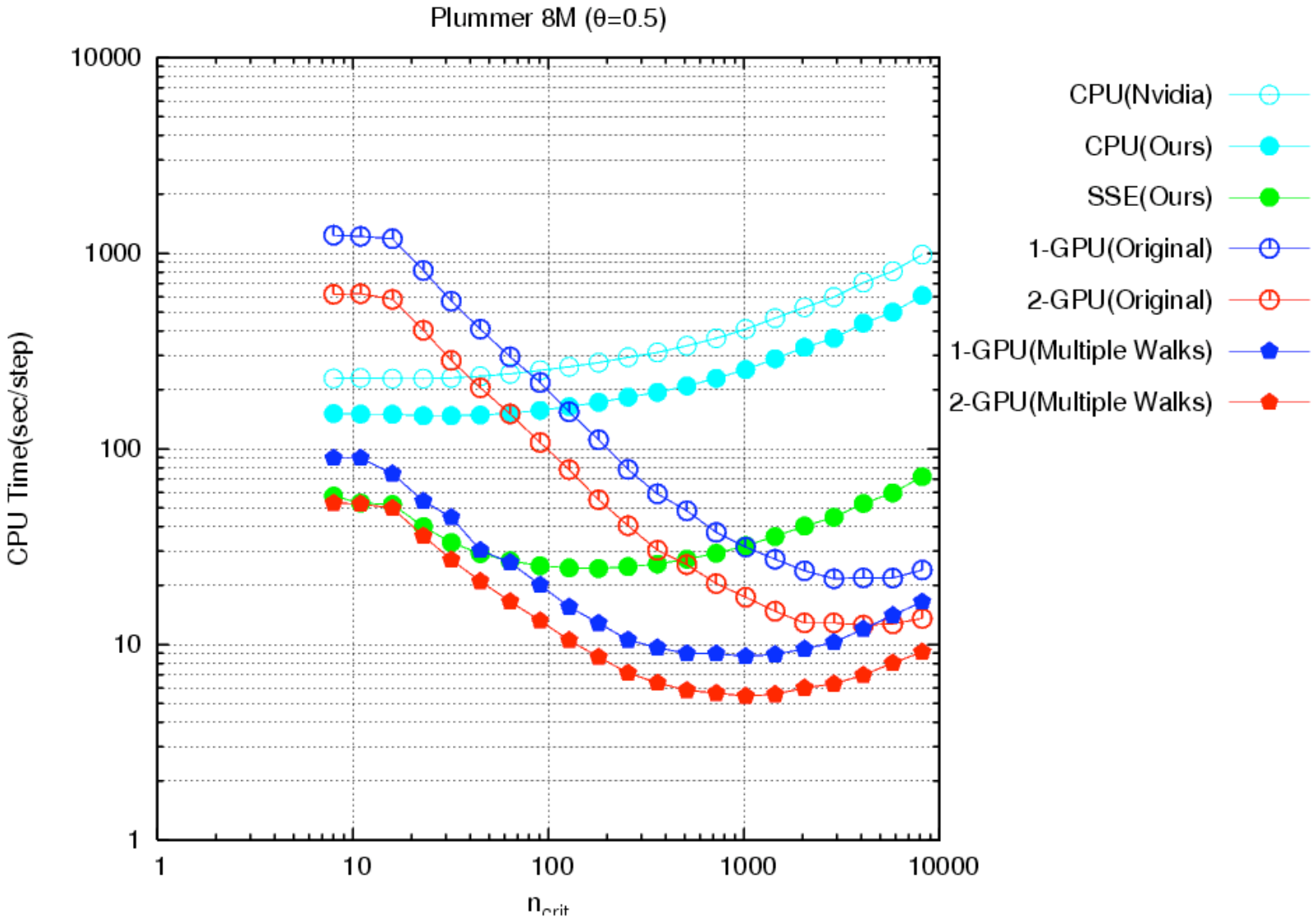
Our Novel GPU Implementaion for Tree Algorithm



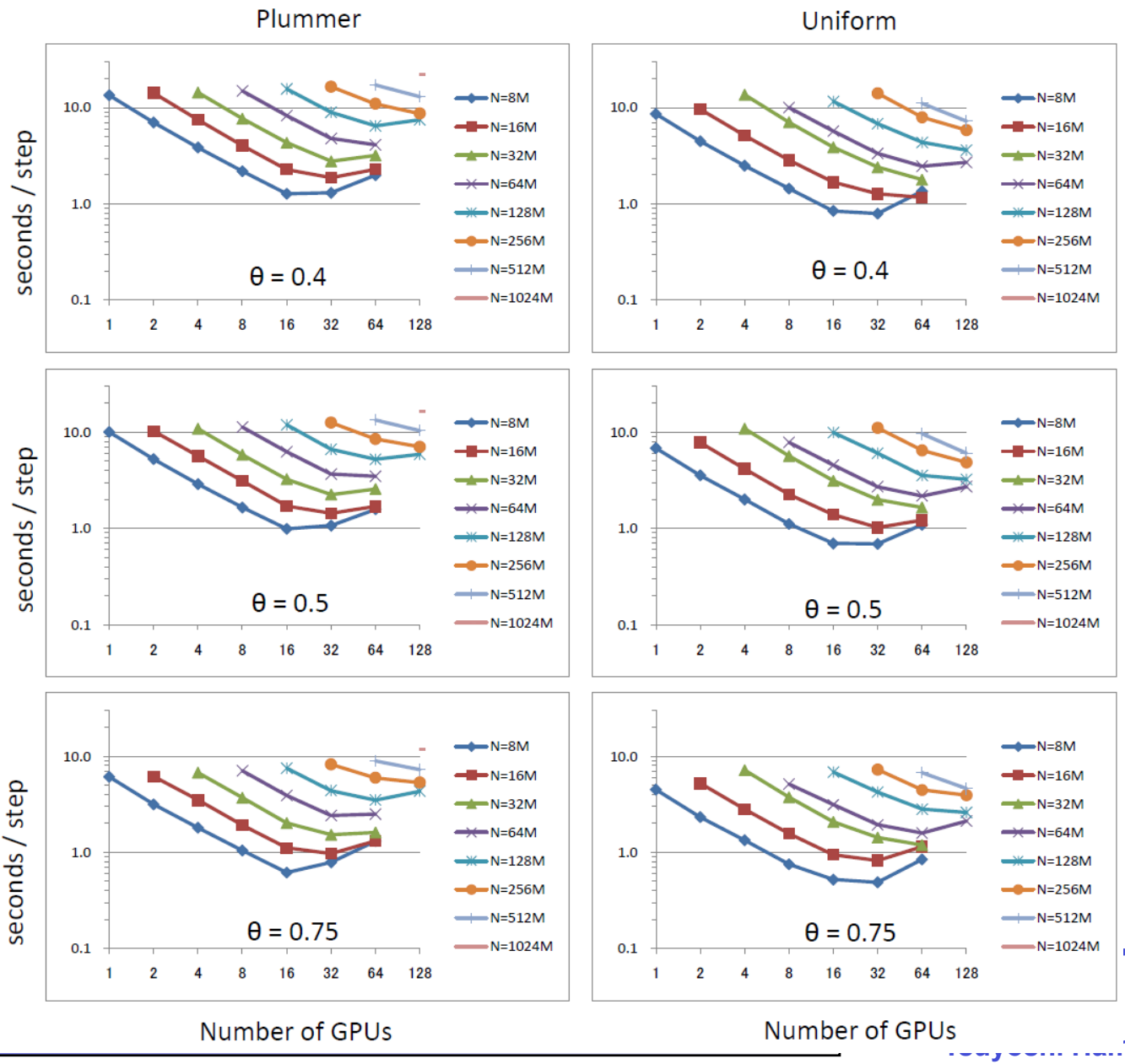
Algorithm (Multiple Walks)

```
extern "C" void vforce_mp(
        float4 xi[], float4 xj[],
        float4 accp[],
        int ioff[], int joff[],
        int nwalk){
    for(int iw=0; iw<nwalk; iw++){
        int ni = ioff[iw+1] - ioff[iw];
        int nj = joff[iw+1] - joff[iw];
        for(int i=0; i<ni; i++){
            int ii = ioff[iw] + i;
            float x = xi[ii].x; float y = xi[ii].y; float z = xi[ii].z;
            float eps2 = xi[ii].w;
            float ax = 0; float ay = 0;  float az = 0; float pot = 0;
            for(int j=0; j<nj; j++){
                int jj = joff[iw] + j;
                float dx = xj[jj].x - x;
                float dy = xj[jj].y - y;
                float dz = xj[jj].z - z;
                float r2 = eps2 + dx*dx + dy*dy + dz*dz;
                float r2inv = 1.f / r2;
                float rinv = xj[jj].w * sqrtf(r2inv);
                pot += rinv;
                float r3inv = rinv * r2inv;
                ax += dx * r3inv; ay += dy * r3inv; az += dz * r3inv;
            }
            accp[ii].x = ax; accp[ii].y = ay; accp[ii].z = az;
            accp[ii].w = -pot;
        }
    }
}
```

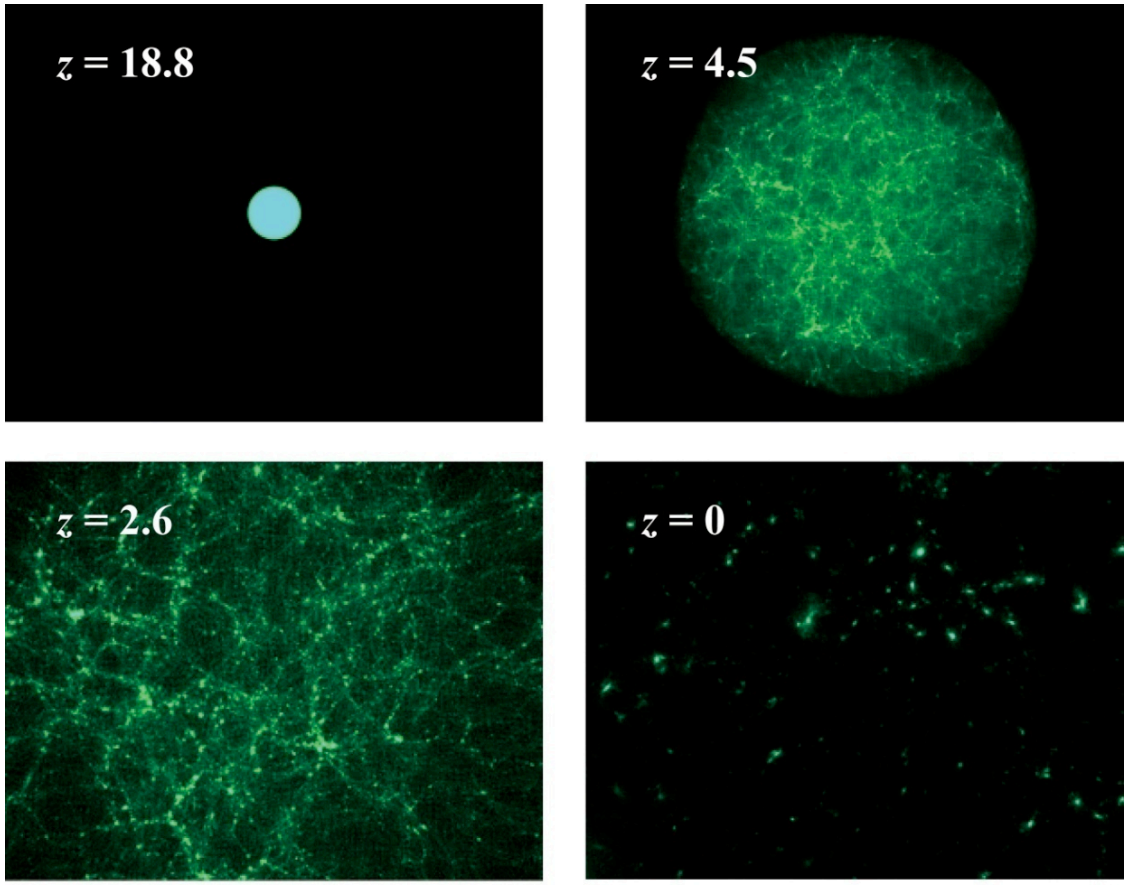
Performance with single node
– Comparative work for Multiple-walks



Scalability

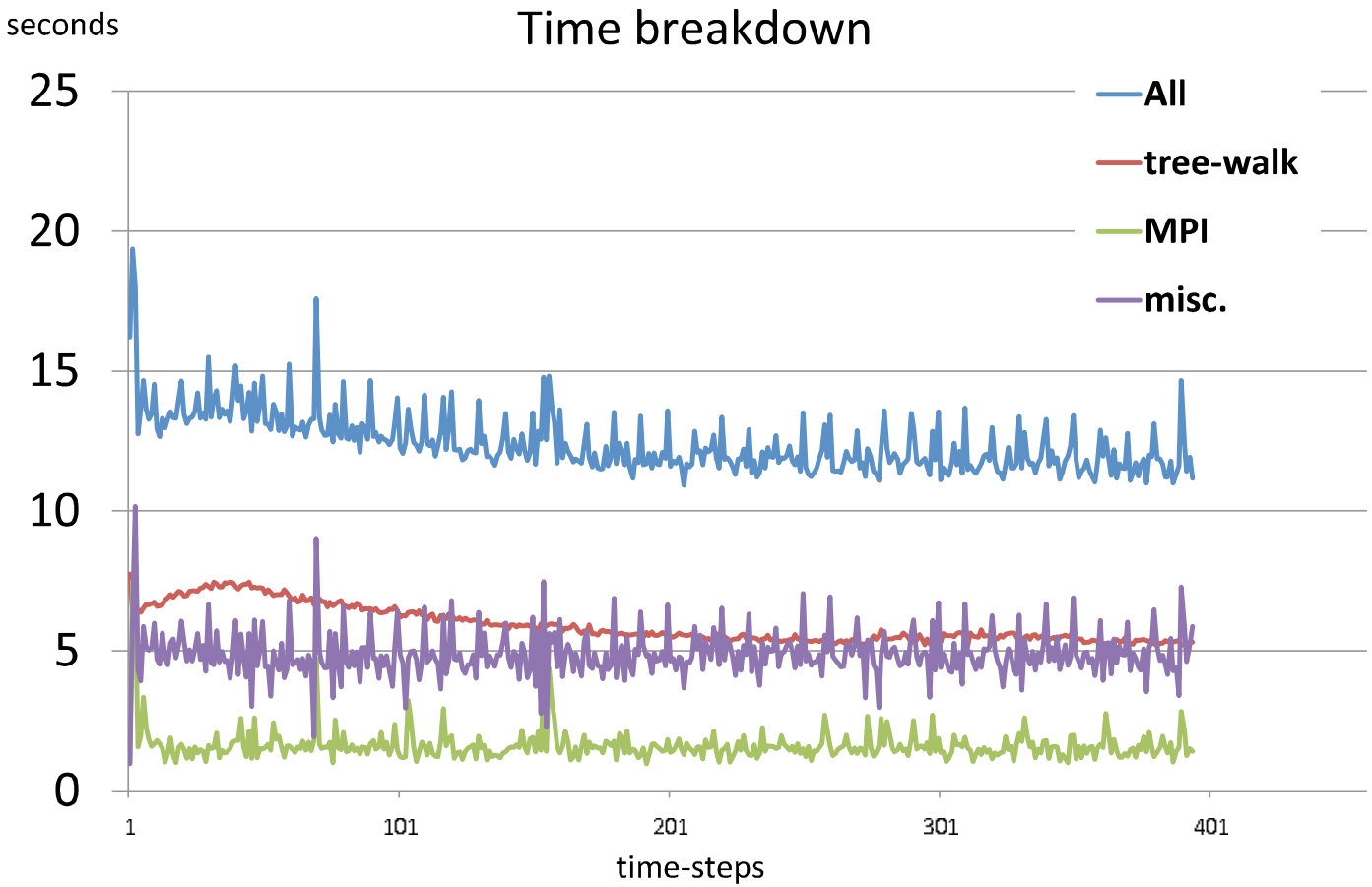


A Cosmological Simulation

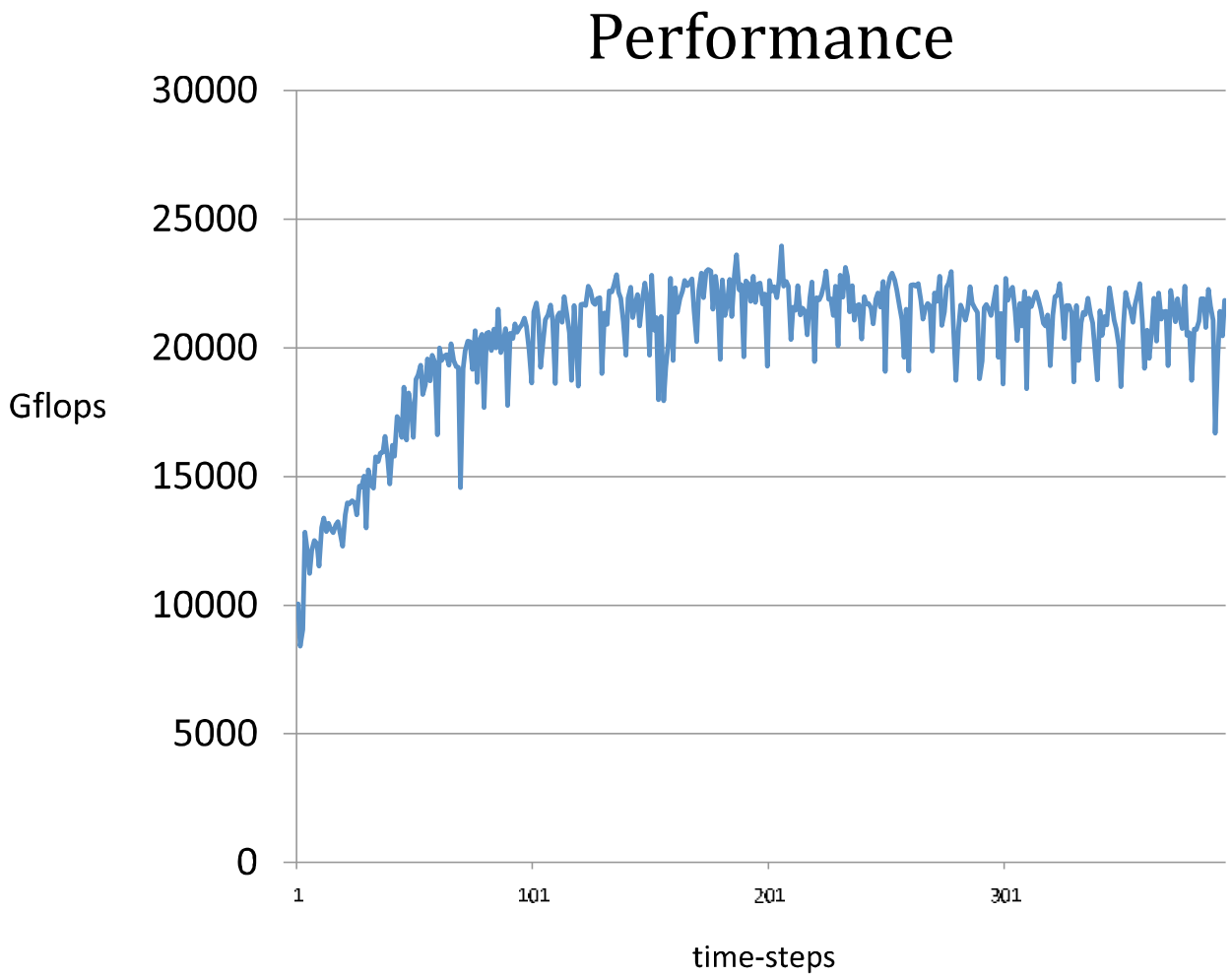


- N: 562 Mega Particles
- $\theta = 0.4$
- based on a standard cold dark matter scenario

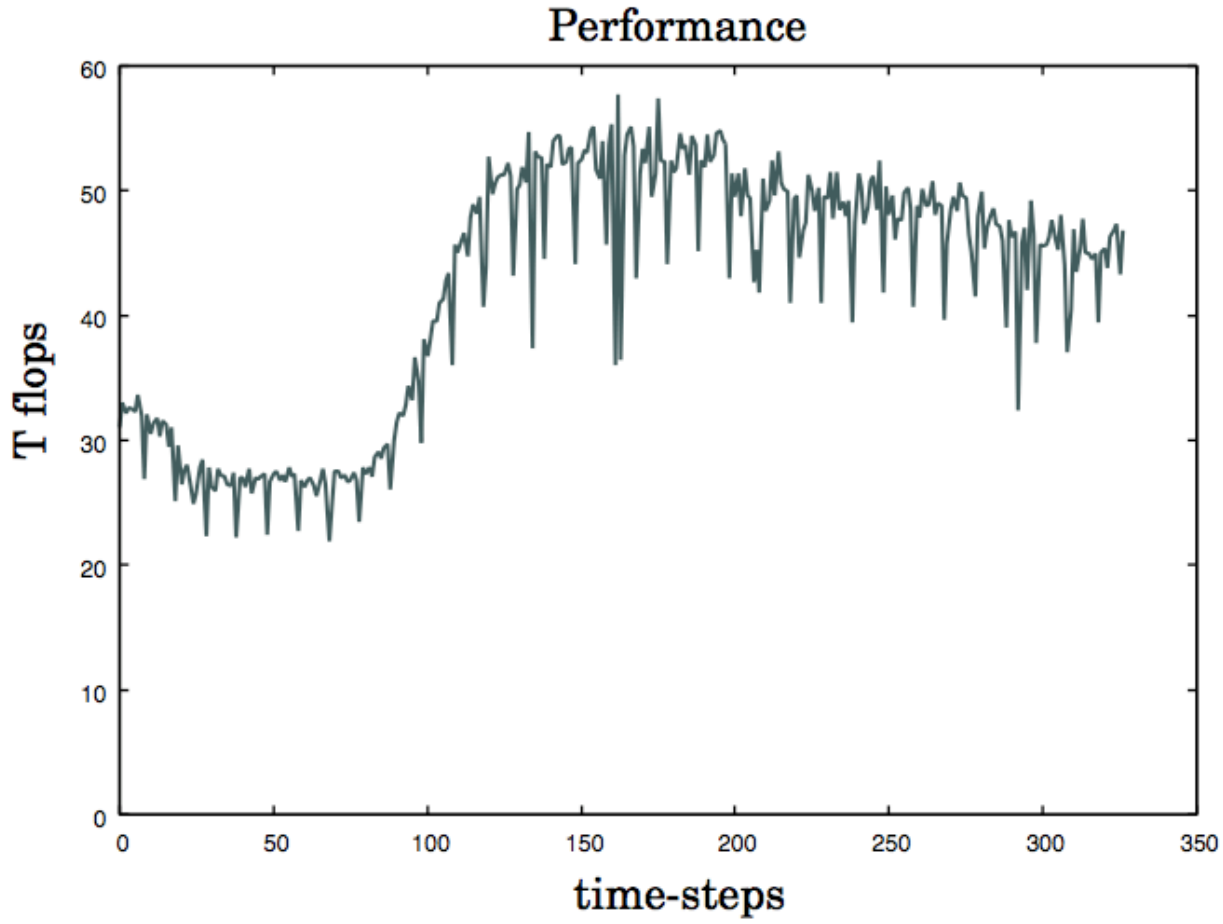
Time Breakdown



Sustained Performance (1 GPU/node)



Sustained Performance (2 GPUs/node)



Updated April 2009 !

1.5 Billion particles
$\theta = 0.4$

Price/Performance

1GPU/node (128 GPUs, 128 nodes)

Price of the GPU cluster

Elements	Number of elements	Price (JPY)	Price ($)
GPUs	128	6,080,000	$ 26,411
Host PCs	128	10,716,032	$ 70,429
Network switch	3	483,600	$ 2,918
Total		**17,279,632 JPY**	**$ 168,172**

$ 51 Mflops/$

2GPU/node (256 GPUs, 128 nodes)

Updated April 2009 !

Elements	Quantity	Price (JPY)	Price ($)
GPUs	256	12,160,000	$ 118,345
Host PCs	128	10,716,032	$ 104,292
Network switch	4	644,800	$ 6,275
Total		**23,520,832**	**$ 228,912**

124 Mflops/$

Conclusion

- N-body methods
  - Important not only astrophysics
  - Astrophysical N-body method is a basics for study
  - Killer application for the GPU cluster
- Nagasaki GPU cluster
  - 256 GeForce 9800GTX+ GPUs
  - Cheap network (GbE)
- Barnes Hut treecode on GPU cluster
  - Brute-force approach is not good .
  - We have proposed Multiple-Walks
    - 2 times faster than brute-fore approach
    - It uses like a process level parallelisms on a GPU.
    - And cost effective :

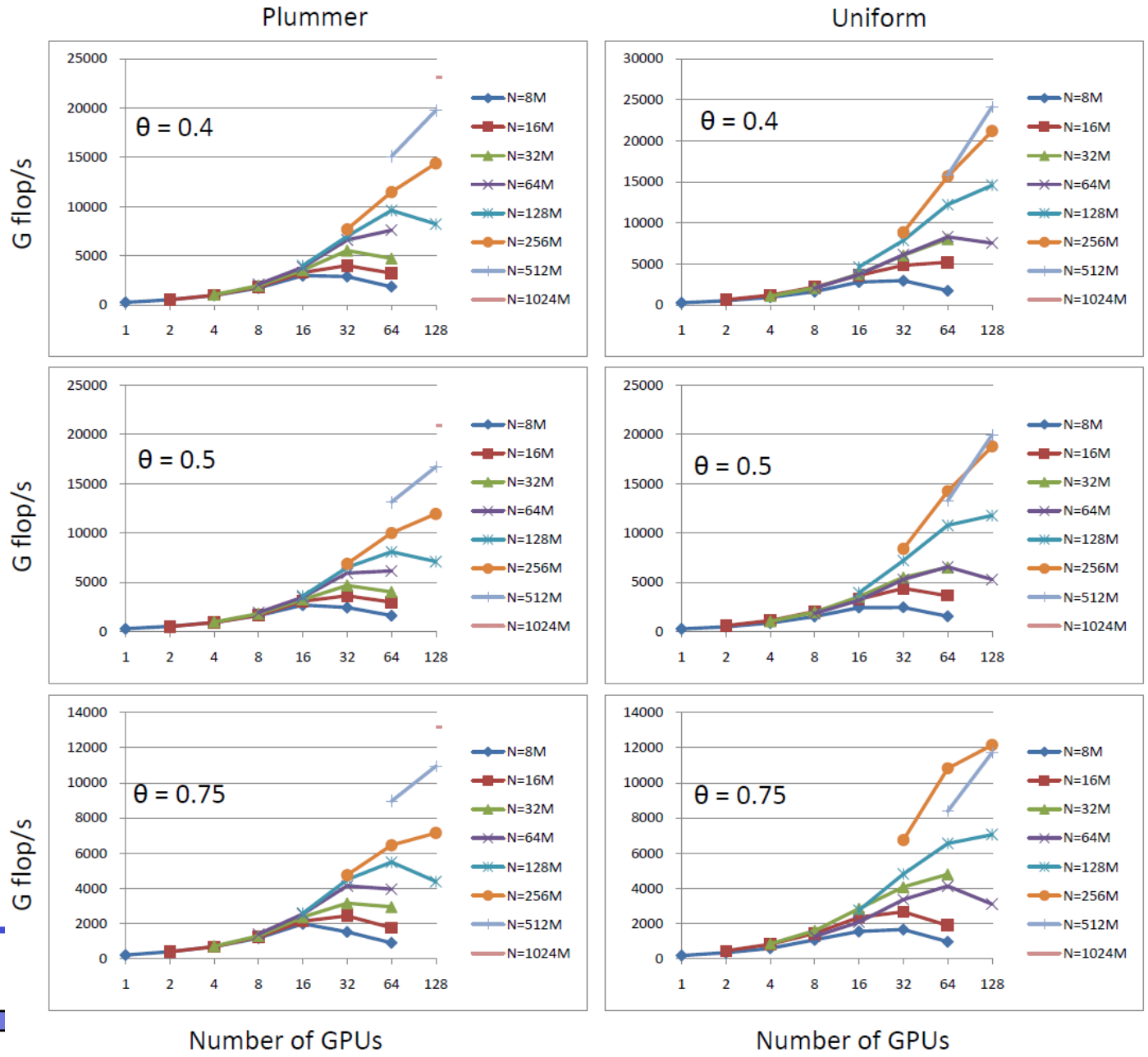
124 M flops/$

- Q & A

Conclusion

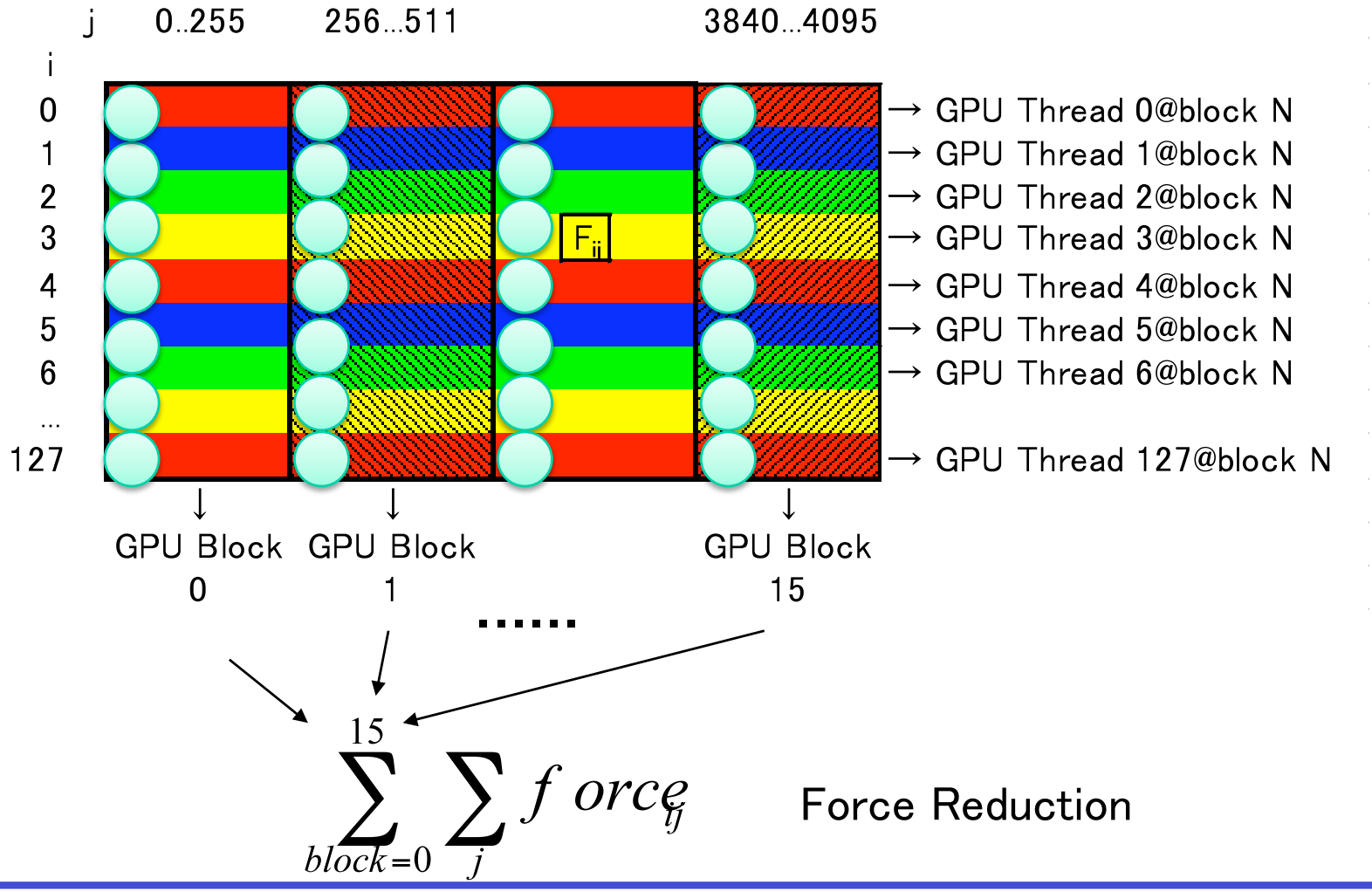
- CUNBODY-1 library
  - The first implementation of accelerating particle-particle interaction in N-body simulation using GeForce8800GTX
  - Performance in Direct sum. algorithm
    - 131072 particle
    - particle decomposition number = 2048
    - sustained speed ≈ **300 Gflops**
  - Performance in Hierarchical Tree Algorithm
    - 2M particle
    - particle decomposition number = 128
    - force decomposition number = 16
    - sustained speed ≈ 70 Gflops
    - there is still room for improvement. (→ 140 Gflops)

Scalability



Force decomposition (j-parallelization)

Reducing the i-parallelization using j-parallelization

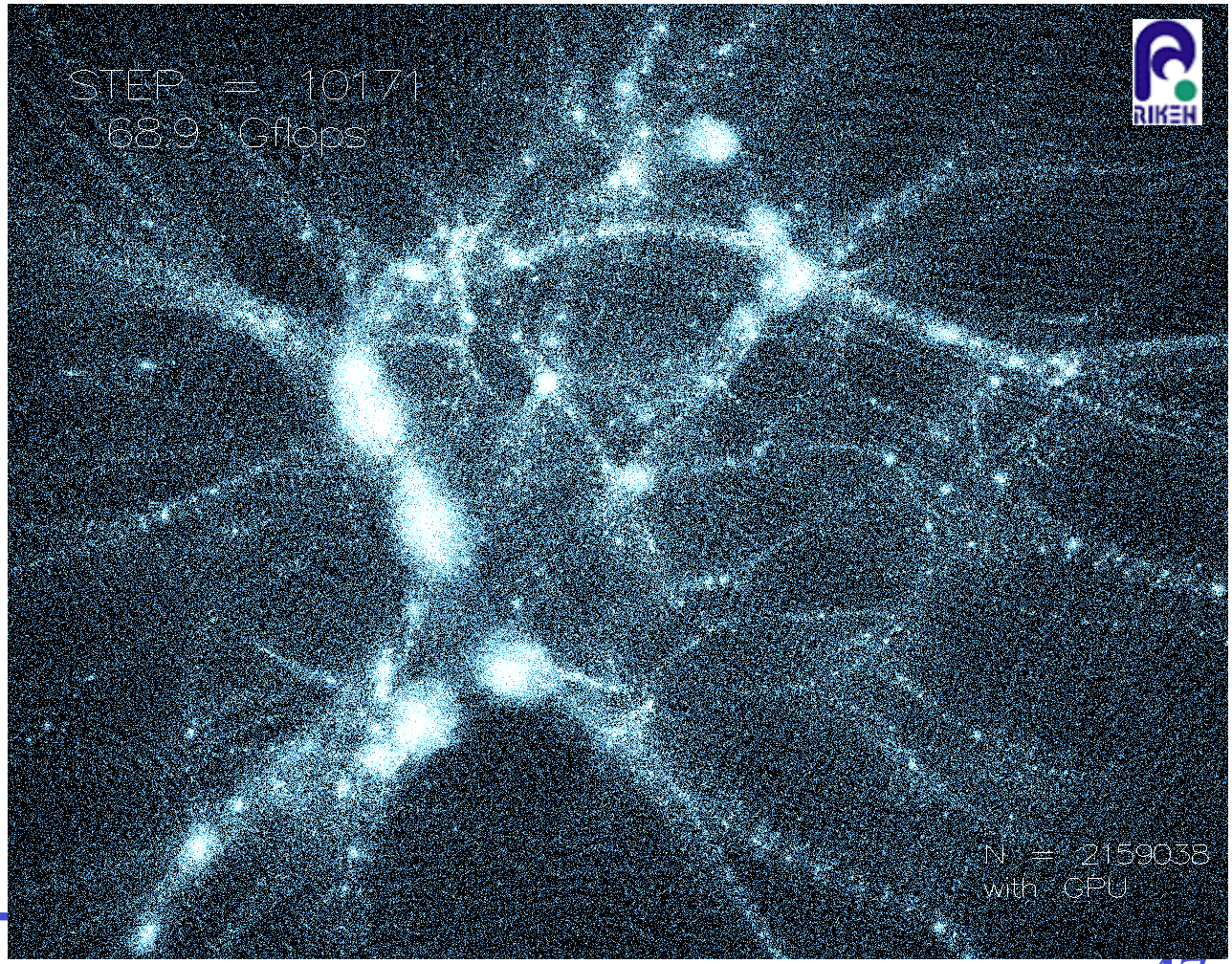


Conclusion

- GPU computing at RIKEN
  - Target application
    - ✓ N-body simulation with million/billion particles (Tree, TreePM, PPPM, PME)
  - Current understand
    - ✓ GPU has a good performance but a weakness at operation for reduction.
  - Our proposal
    - ✓ supporting the reduction in hardware level
  - Our prospect (the Hierarchical Tree Algorithm)
    - ✓ **80 Gflop/s** (G80) ⇒ **140 Gflop/s** (new G80)

Practice : more complex test
– Hierarchical Tree Algorithm

- GeForce8800GTX + Core2Duo E4400
- **2M** particles
- 1000 time steps
- 2 hours
- about **50** Gflops

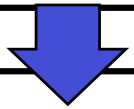


Cosmological N-body simulation

Force Reduction with current GPU(G80)

- Using j-parallelization
  - we can reduce the i-parallelization from 4096 to 128.
  - in stead, we need to sum up the forces between GPU blocks (**Force Reduction**)
    - Who does the "Force Reduction" ?

The GPU blocks on a GeForce8800GTX can't synchronize each other, then **the host computer should do it**..

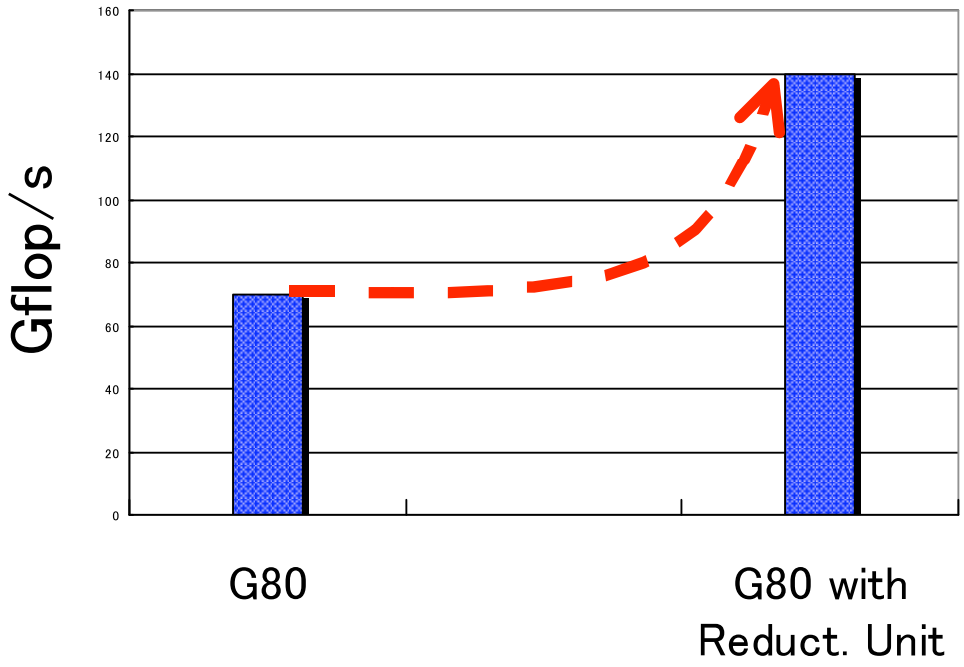


Traffic between host computer and GPU increases by a factor of # of blocks (x16 for G80)

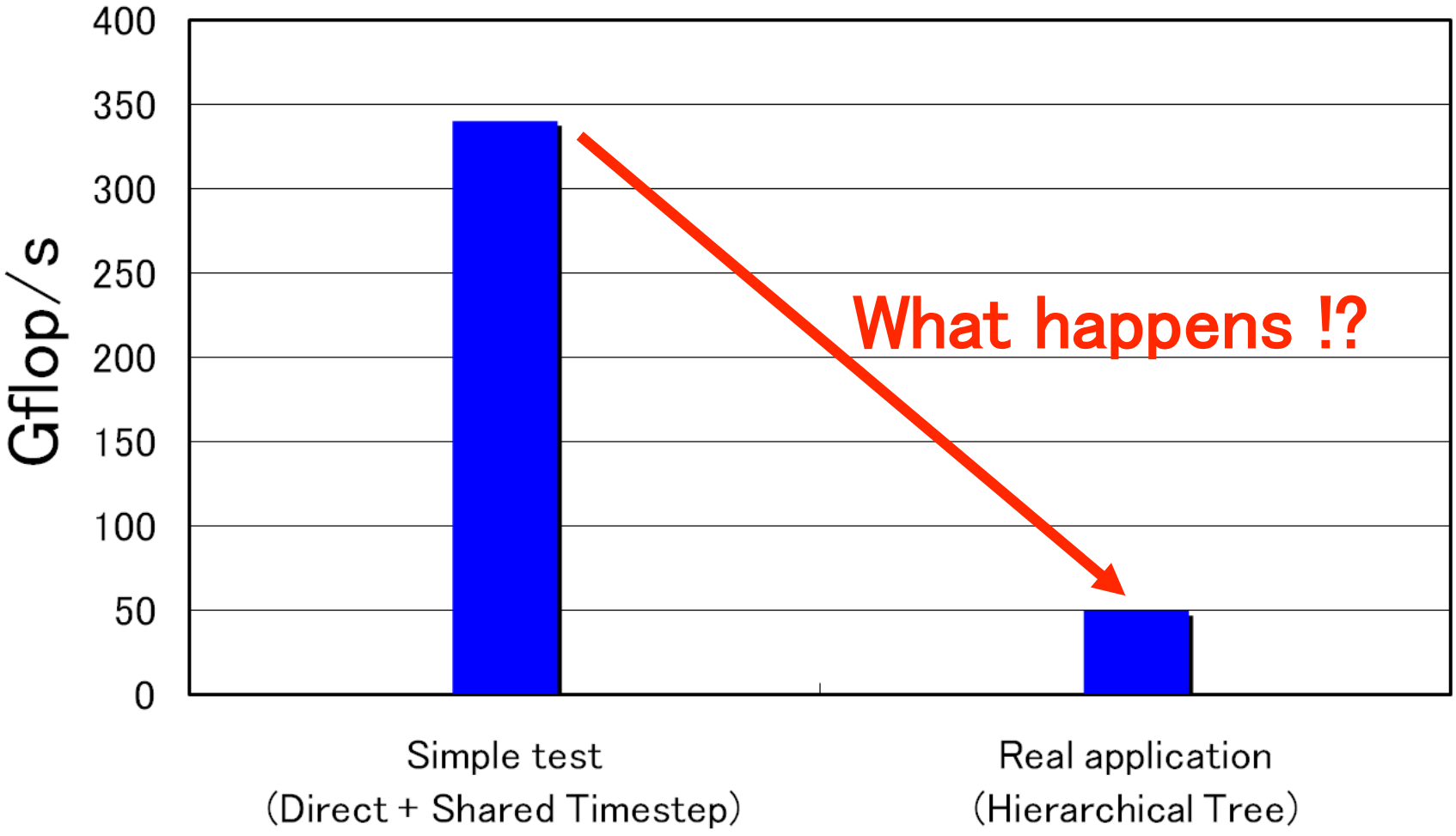
The traffic becomes bottleneck.

Our prospect (performance @ tree algorithm)

	G80		G80 with Reduct. Unit	
Host calc	3.7 sec	(47%)	3.7 sec	(83%)
Host->GPU	0.45 sec	(6%)	0.45 sec	(10%)
GPU calc	0.04 sec	(1%)	0.04 sec	(1%)
GPU->Host	3.6 sec	(46%)	0.25 sec	(6%)
Total Time	7.79 sec		4.44 sec	
Total Speed	80 Gflop/s		140 Gflop/s	



A problem in a real application



Our proposal to NVIDIA

- In next generation GPU,
  - we hope NVIDIA will add a hardware function for the reduction ,
  - rather than double precision and massively threads (such like Tesla)