# Internals of the CUNBODY-1 library: particle/force decomposition and reduction

**Tsuyoshi Hamada,**
**Yousuke Ohno, Gentaro Morimoto, Makoto Taiji**
High-Performance Molecular Simulation Team
Genomic Sciences Center, **RIKEN**
**Toshiaki Iitaka,**
Computational Astrophysics Laboratory,
**RIKEN**
**Keigo Nitadori**
Department of Astronomy, University of Tokyo

# GPU computing at RIKEN

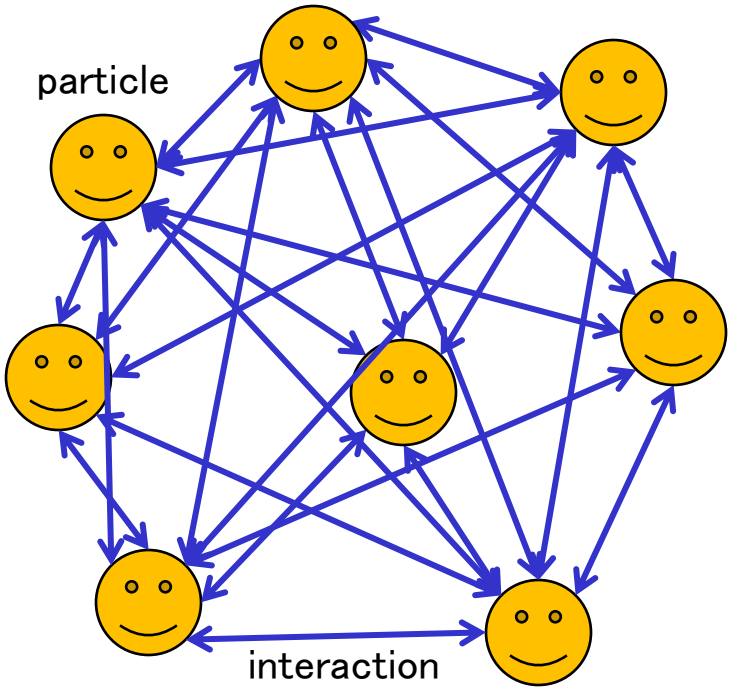
- Motivation
  - Accelerating billions of particle simulations such as
    - ✓ Cosmological N-body simulation
    - ✓ Large-scale molecular dynamics simulation

    using cost-effective hardware.

- Target code
  - PPPM (Particle-Particle Particle-Mesh)
  - PME (Particle Mesh Ewald)
  - TreePM (Tree Particle-Mesh)

# 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

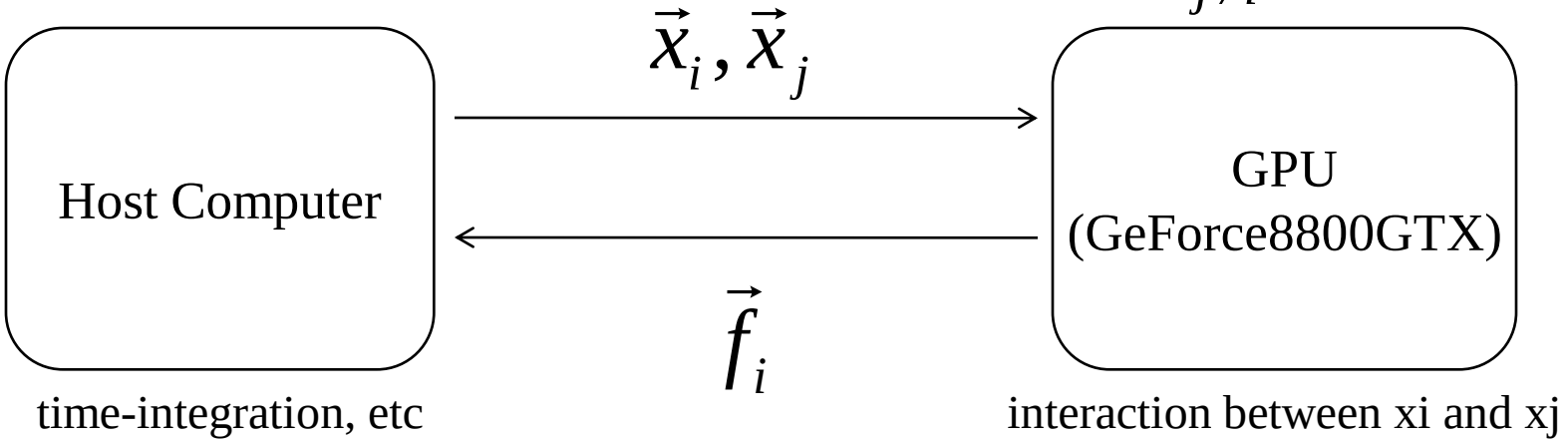


# CUNBODY-1 library

(Hamada & Iitaka, Astro-ph 2007)

- **CU**da **NBODY** version 1 library
  - The first implementation of accelerating particle-particle interaction in N-body simulation using
    - CUDA
    - GeForce8800GTX

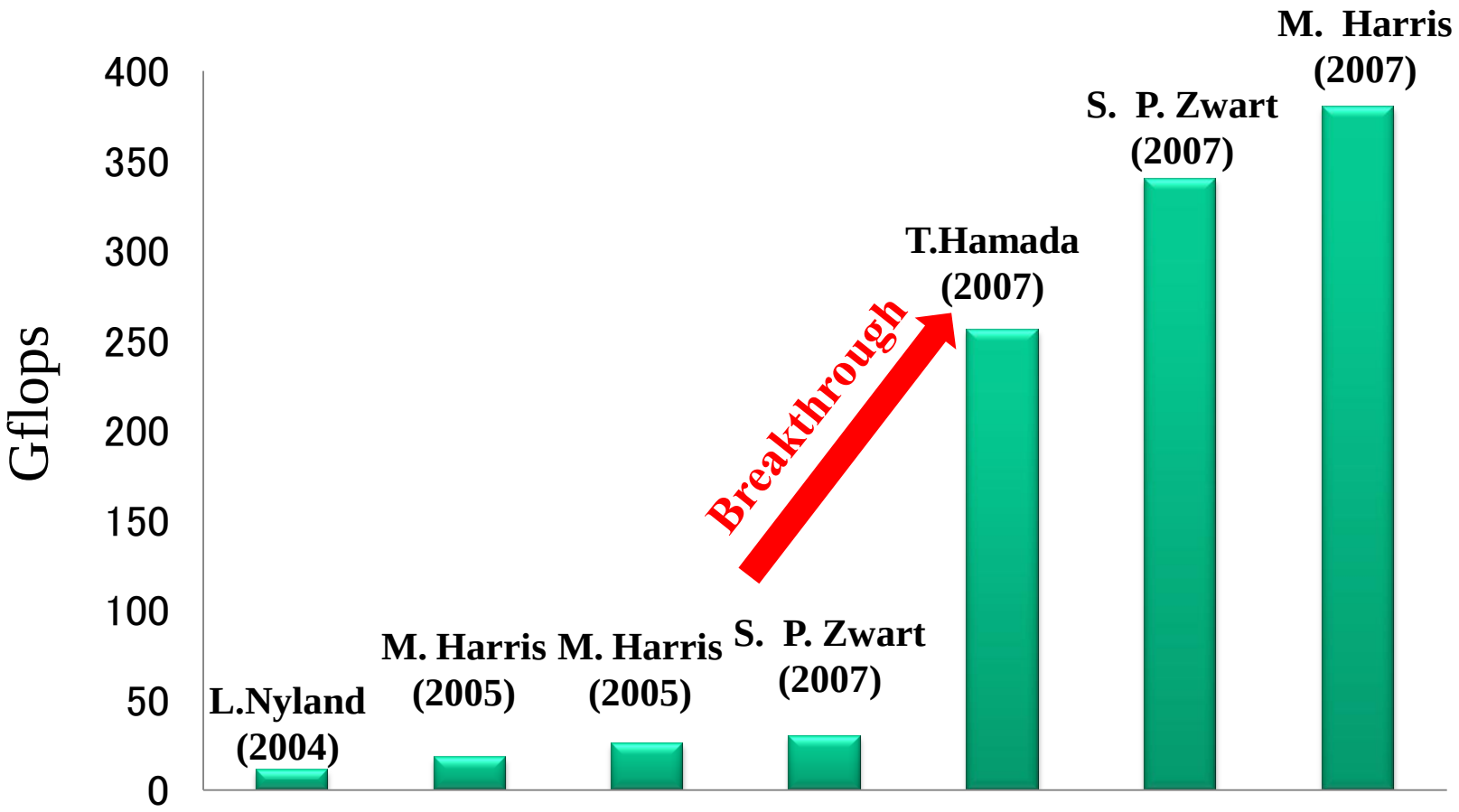
$$\vec{f}_i = \sum_{j \neq i} \vec{g}(\vec{x}_i, \vec{x}_j)$$




The basic idea is the same as GRAPE systems

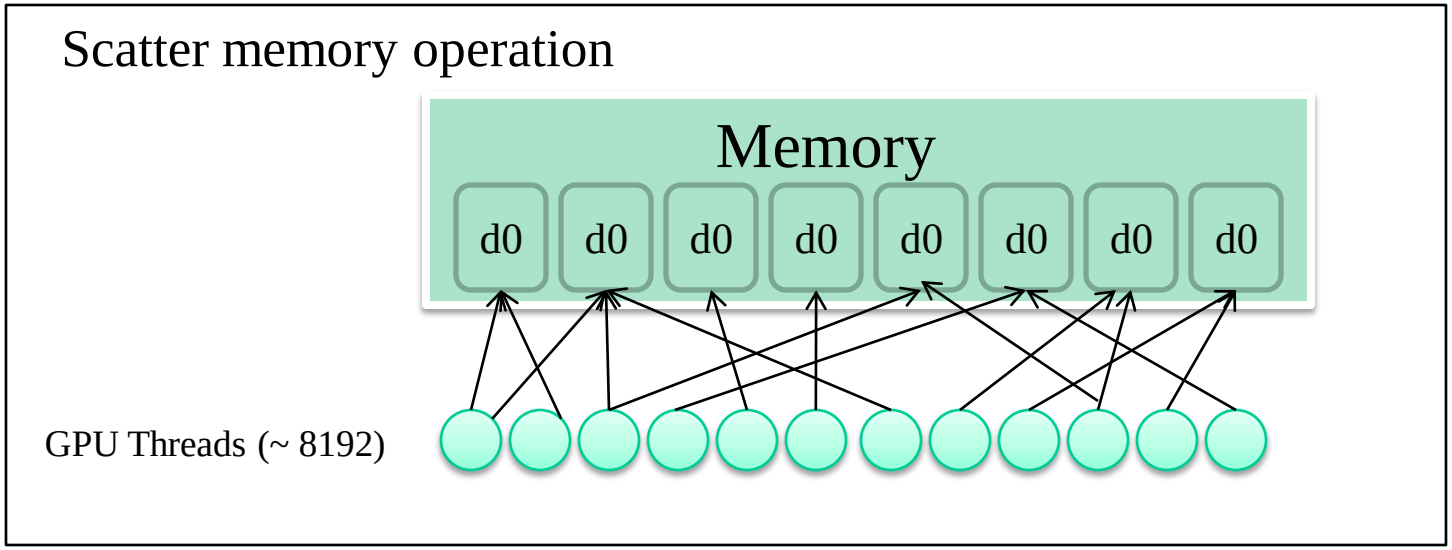
You can get **source code** from http://progrape.jp/cs/

# History of N-body with GPU



# The key to breakthrough – GeForce8800GTX + CUDA

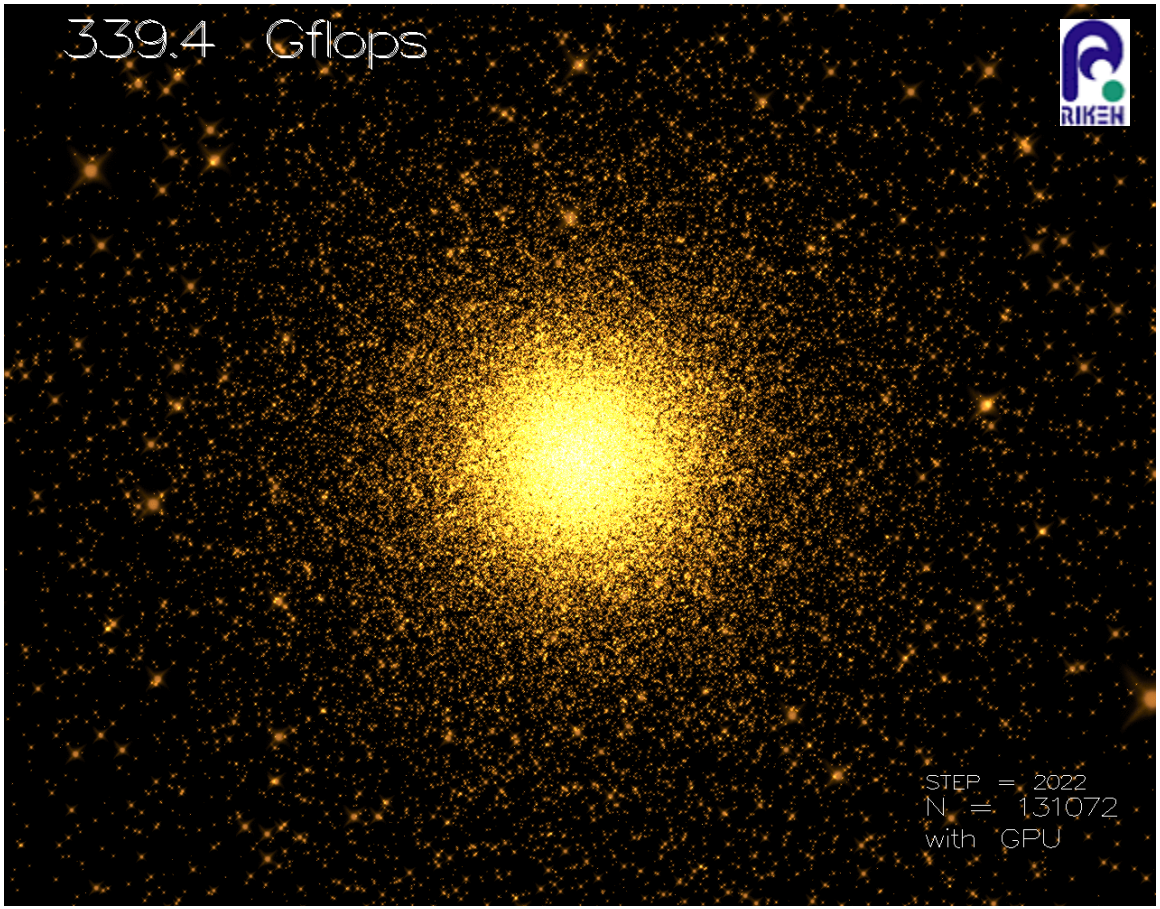
- GeForce8800GTX
  - **8192** thread s in maxmum
  - **345** Gflops peak
- Using CUDA software
  - We can use **scatter memory operation** on GPU.



# Practice : a simple test – direct summation

- GeForce8800GTX + Core2Duo E4400
- 131 k particles
- 2000 shared time steps
- 1 hours
- about **300** Gflops

You can get **source code** from http://progrape.jp/cs/

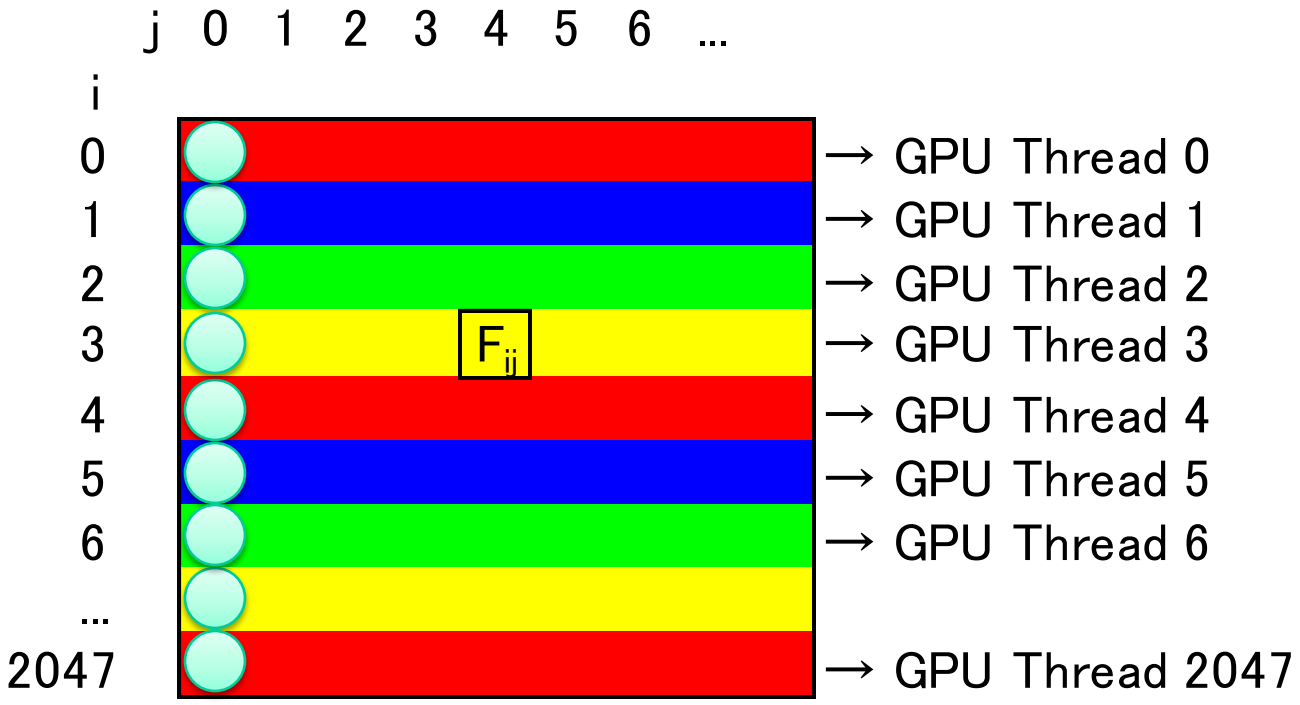




http://progrape.jp/cs/ or YouTube

Plummer sphere, 131072 particles

# Details of parallelization

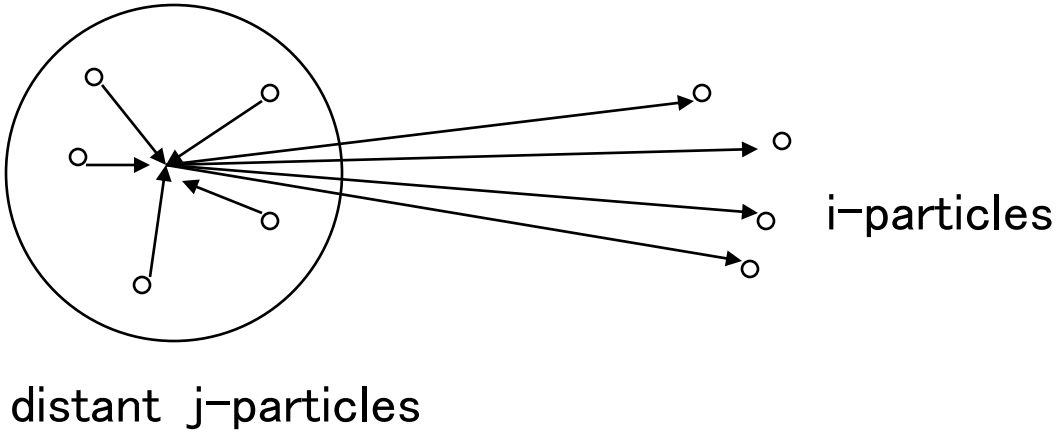



All 2048 threads are calculating forces on different i-particles

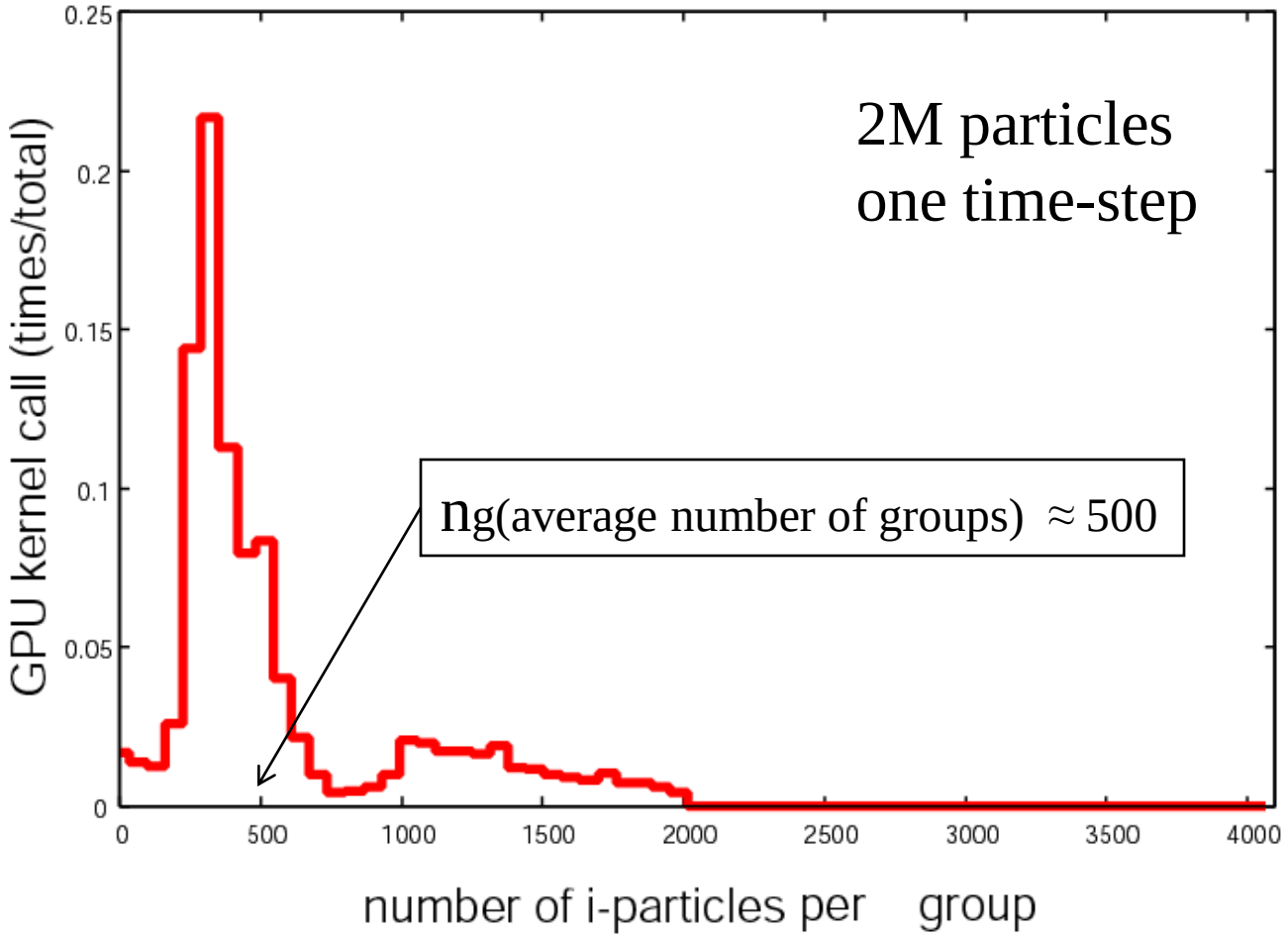
**→ Particle Decomposition (i-parallelization)**

# Practice: Hierarchical Tree Algorithm

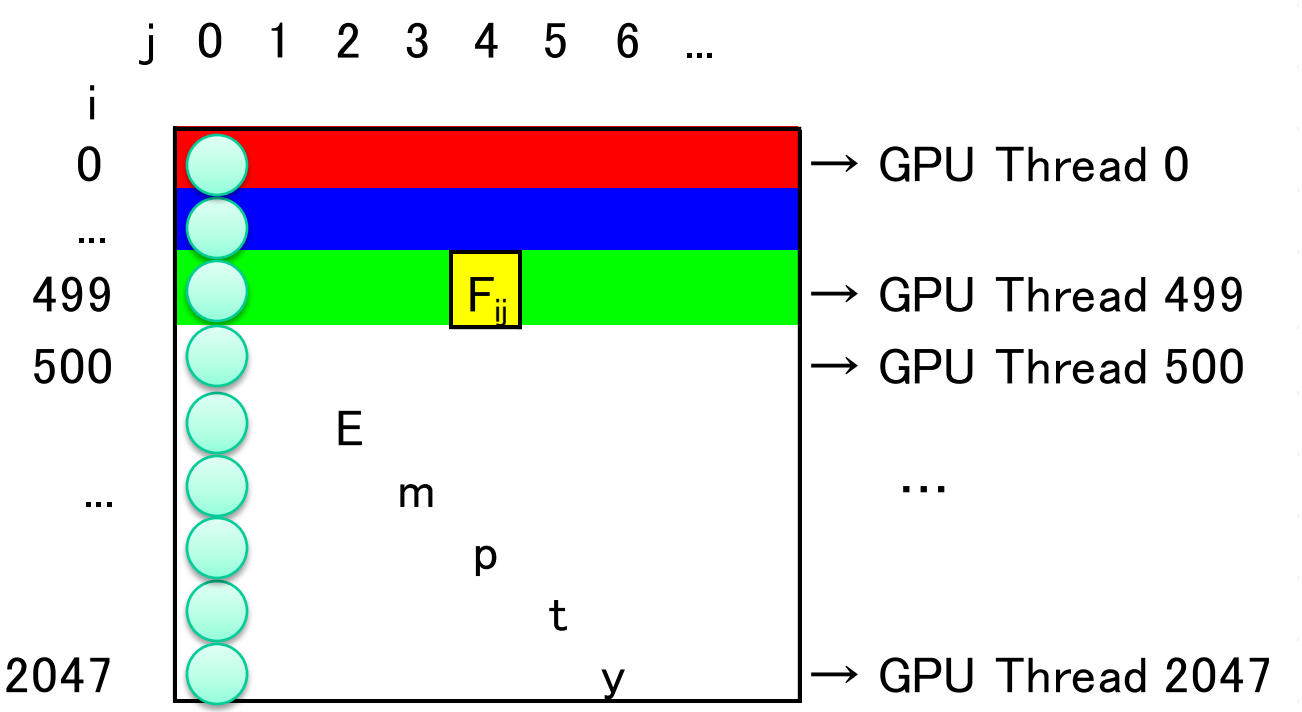
- O(N log N) scaling with particle number
  - particles with a long distance are grouped
  - using hierarchical oct-tree structure
  - faster than direct summation for large number of particles



# Almost all GPU threads will sleep in hierarchical tree algorithm



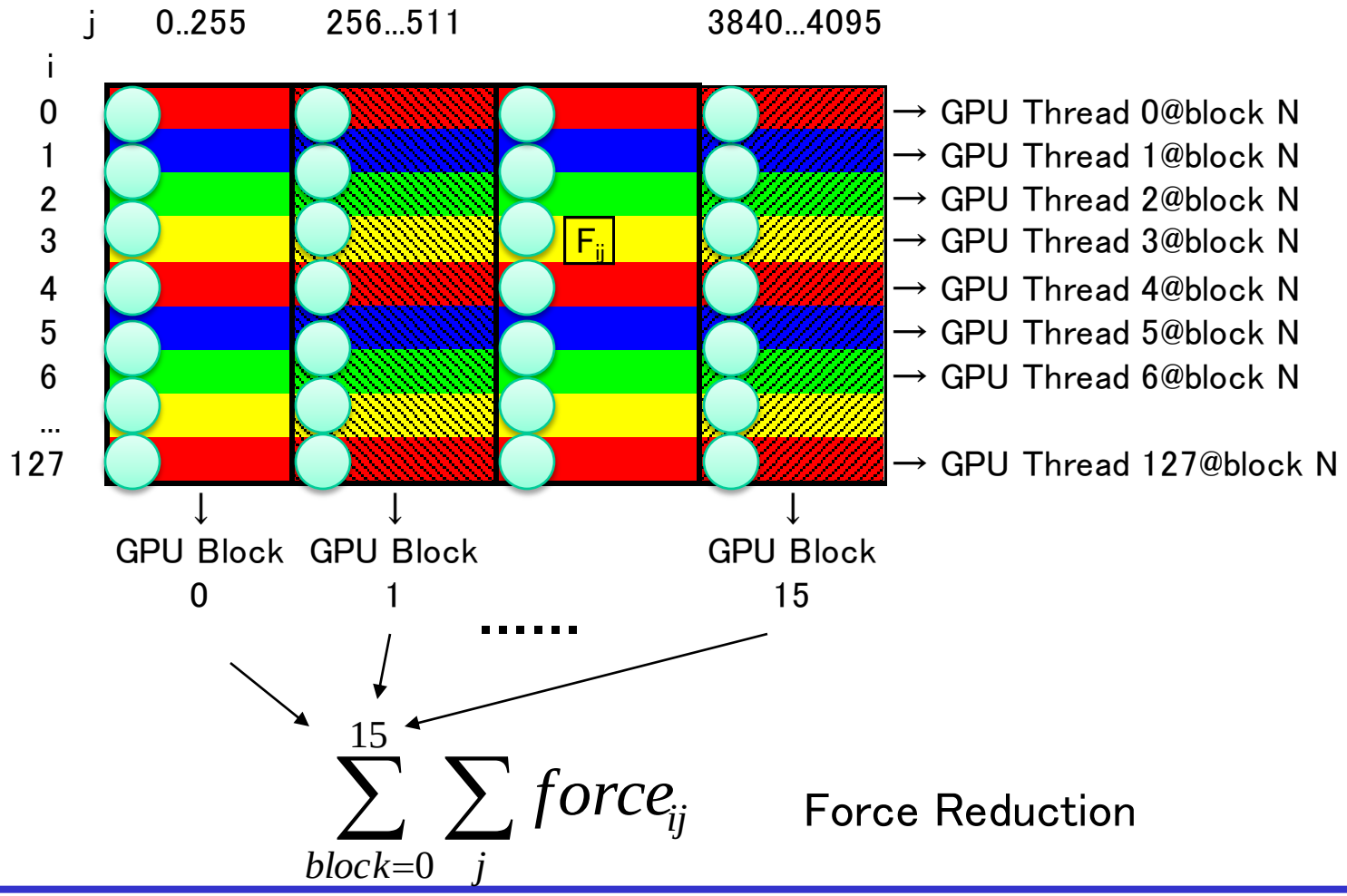
# # of i particles ≈ 500,  # of threads = 2048




About 1500 threads are sleeping

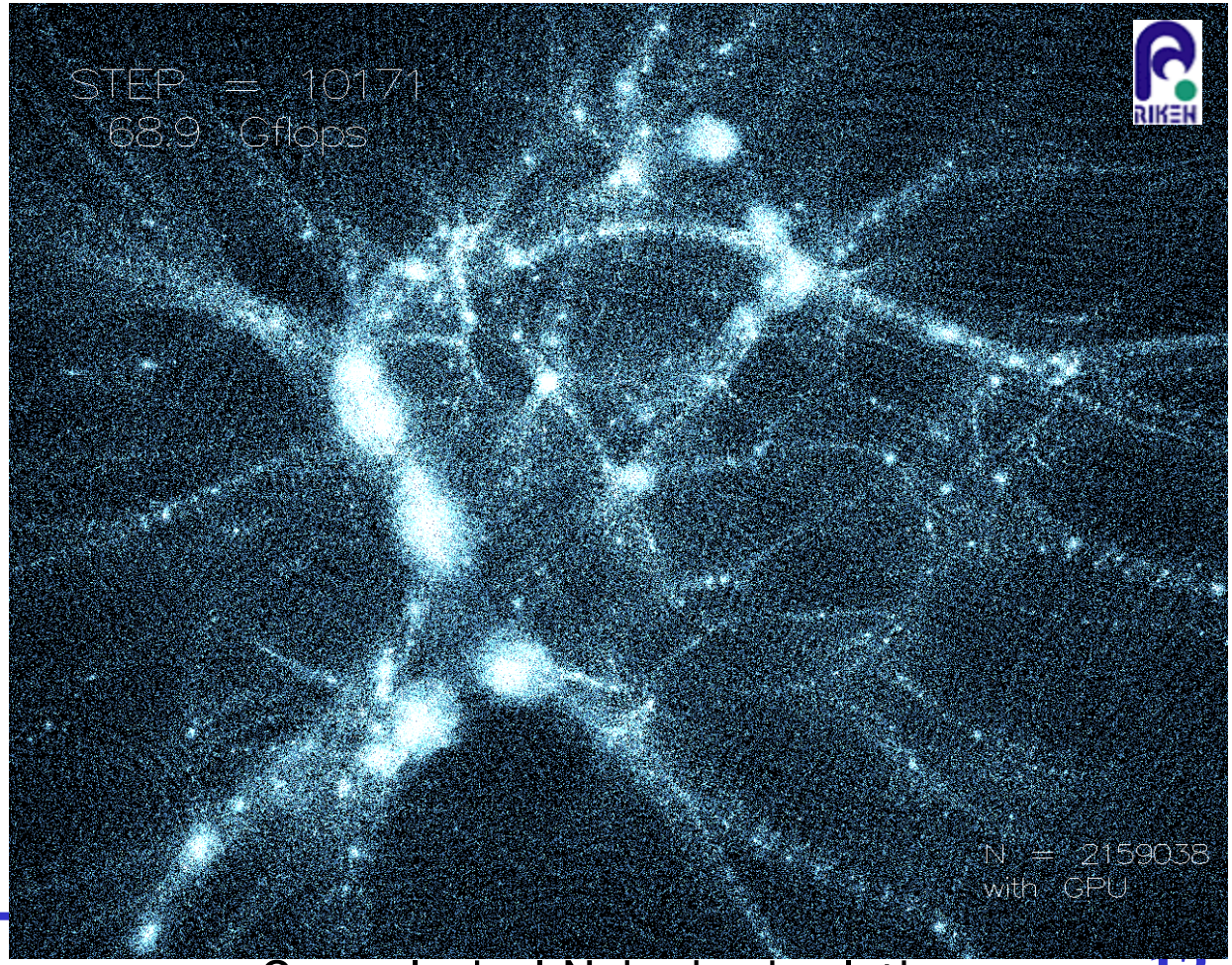
# Force decomposition (j-parallelization)

Reducing the i-parallelization using j-parallelization



# Practice : more complex test – Hierarchical Tree Algorithm

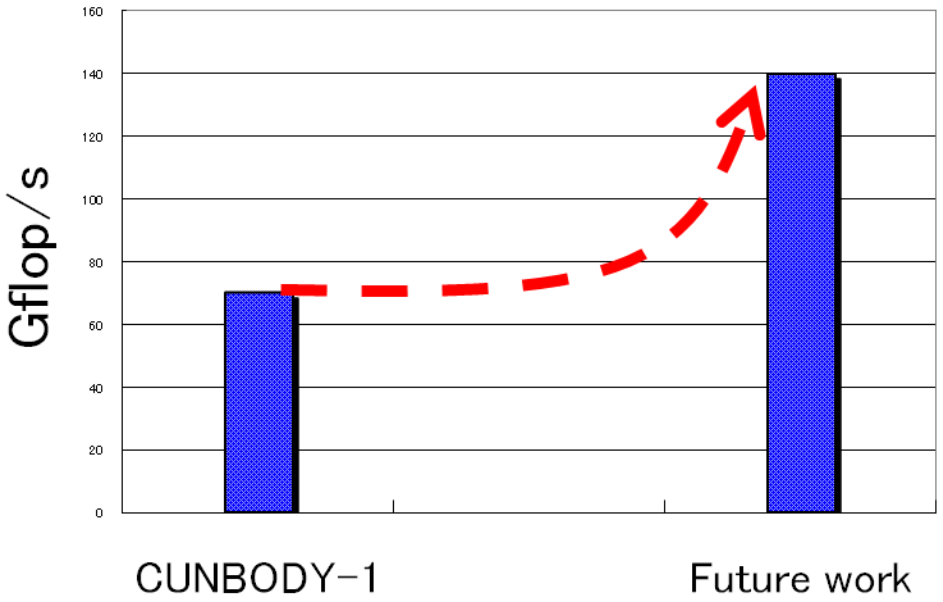
- GeForce8800GTX + Core2Duo E4400
- **2M** particles
- 1000 time steps
- 2 hours
- about **70** Gflops




Cosmological N-body simulation

# Breakdown of calculation time, and future work

| | CUNBODY-1 | | Future | |
|---|---|---|---|---|
| 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%) | 0.23 sec | (5%) |
| Total Time | 7.79 sec | | 4.44 sec | |
| Total Speed | 70 Gflop/s | | 140 Gflop/s | |



# 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)

# Acknowledgement


- H. Yahagi, J. Makino (NAOJ)
  - Constructing GPU cluster
  - Developing tree code/initial conditions
  - Many discussions
- T. Fukushige, A. Kawai (K&F co. Ltd)
  - Developing tree code/initial conditions
- K. Yasuda (Univ. Nagoya)
  - Constructing GPU cluster
- S. Fujikawa, T. Takahei, H. Matsubara, T. Ebisuzaki (RIKEN)
  - Constructing GPU cluster
- S. Nakamura (Mitsubishi Chemical Group)
  - Constructing GPU cluster
- T. Narumi (Keio Univ)
  - Many discussions
- N.Nakasato (Aizu Univ)
  - Initial conditions/tree code