# Performance evaluation and tuning of GRAPE-6 — towards 40 “real” Tflops

Jun Makino, Eiichiro Kokubo, Toshiyuki Fukushige, and Hiroshi Daisaka,

University of Tokyo
National Astronomical Observatory of Japan

## Talk overview

- GRAPE Project
- GRAPE-6
- Method of parallelization
- Mesured Performance
- Performance tuning
- Conclusion

## GRAPE project

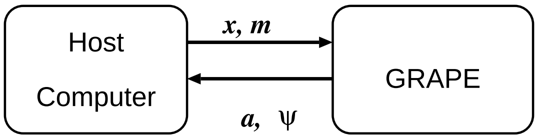
**GOAL:**
Design and build specialized hardware for simulation of astronomical self-gravitational systems.

**Rational:**
You can do larger simulations (better resolution) for same amount of money.

| | | |
|---|---|---|
| GRAPE-6 | (2002, 64 TF) | 4M$ |
| ASCI White | (2001, 12 TF) | 200M$? |
| ASCI Q | (2002?, 30 TF) | 200M$? |
| Earth Simulator | (2002, 40 TF) | 300M$ |

## Basic idea of GRAPE

Special-purpose hardware for force calculation
General-purpose host for all other calculation

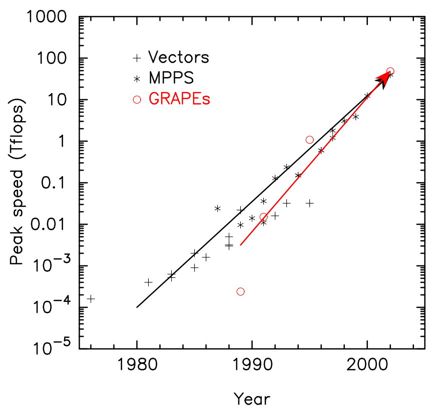



$$\frac{d^2\mathbf{x}_i}{dt^2} = -\sum_{j\neq i} Gm_j \frac{|\mathbf{x}_i-\mathbf{x}_j|}{(\mathbf{x}_i-\mathbf{x}_j)^3}$$

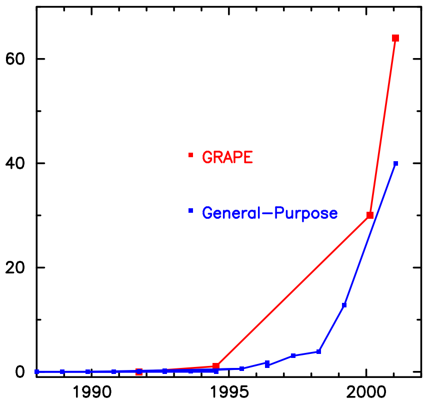
## Advantages of GRAPE

- Special-purpose hardware
  - Hardwired logic → No software needed
  - Pipelined processor with many arithmetic units → High performance
- General-purpose host
  - Programmable in high-level languages
  - can be used with Barnes-Hut tree/Aarseth scheme

## Evolution of GRAPE systems



## A different way to show the same data




## Why not TREE/FMM?

We do use TREE/FMM whenever they work

- modest accuracy requirement
- small range of timestep

Many astrophysical applications are not well suited to FMM

- high accuracy requirement
- very wide range of timestep

## GRAPE-6

- Peak speed 63 Tflops.
- Project started in summer 1997.
- Total budget $\sim$ 4M\$.
- Full machine completed April 2002.
- won GB Prize 2000, 2001
- 2002 GB Prize finalist (29.5 Tflops)

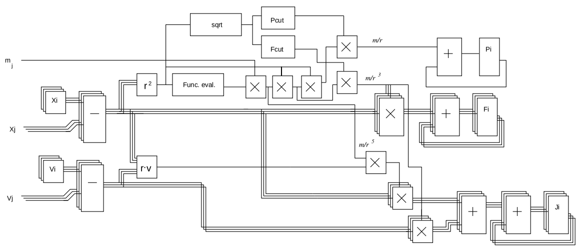
(2002 GB winner: AGCM on ES, 26.6 TF peak/23.9 TF sustained)

## How 63 TF is achieved

- 4 clusters
- 16 processor boards per cluster
- 8 processor modules per board
- 4 chips per modules
- 6 pipelines per chip
- $\sim$ 60 arithmetic units per pipeline

In total, $7.0 \times 10^5$ Arithmetic units operating at 90 MHz = 63 Tflops

## GRAPE-6 Processor pipeline

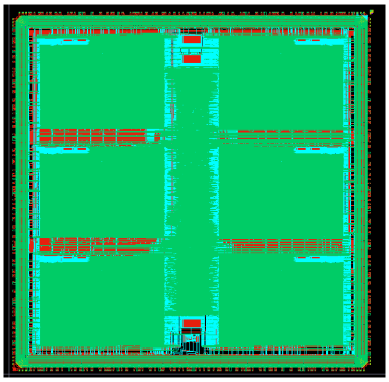



Calculates gravitational force, its first time derivative and potential.

## GRAPE-6 Processor Chip

- 0.25 $\mu$m custom design (Toshiba TC240E, 1.8M gates)
- 2.5 V power supply, 100 MHz operation (design goal)
- 6 pipelines in a chip
- $>$ 30 Gflops (equivalent) per chip
- system clock: 25 MHz
- Single chip equivalent to single GRAPE-4 board

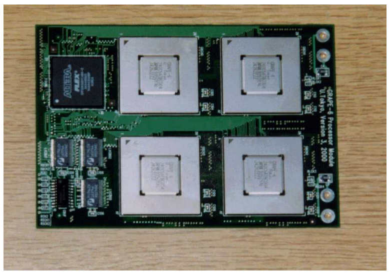
## GRAPE-6 Processor Chip



## GRAPE-6 vs Intel P4

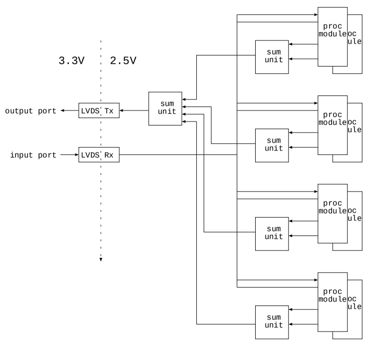
| | G6 | P4 |
|---|---|---|
| Design Rule ($\mu$m) | 0.25 | 0.13 (NW) |
| # transistors | 7M | 40M |
| Clock freq. | 90MHz | 3.2GHz |
| Power Consumption | 15W | $\sim$ 100W |
| # ALUs | $\sim$ 400 | 2 (SSE2) |
| Peak speed(GF) | 31 | 6.4 |

## GRAPE-6 processor module



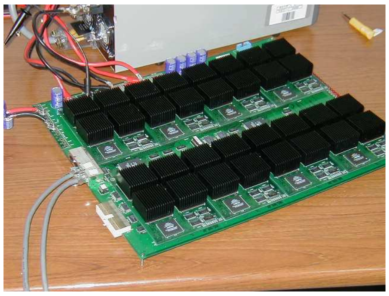
- 4 processor chips + Memory chips.
- Integrates all functions of GRAPE-4 processor board.

## The GRAPE-6 Processor board



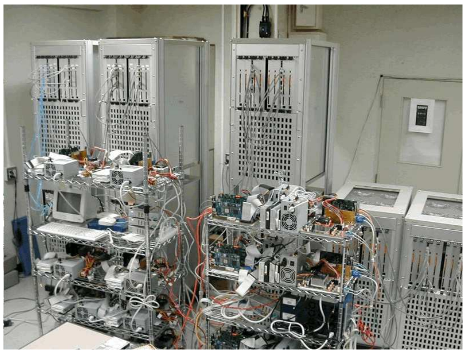

- Single board with 32 chips
- Semi-serial (LVDS) inter-face (350MHz clock, 4 wires)
- Two input ports and one output port
- Result from 32 chips are summed by a reduction tree

## The GRAPE-6 processor board



*Visit our booth (#622) to see the real one!*

## The 64-Tflops GRAPE-6 system

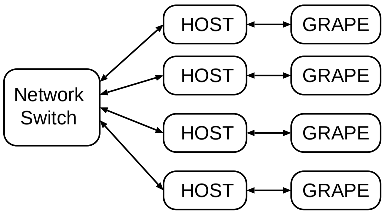


# Host Parallelization

Major challenge: We need parallel host to serve GRAPE-6
Single PC (or whatever) is not fast enough to serve a 100 Tflops machine.
$\sim$ 10 processors would be okay.

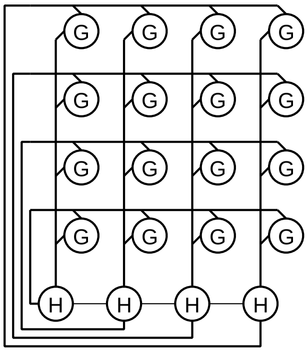
# A Naive approach

Connect multiple GRAPEs and multiple hosts




Works fine for treecode
Does not work for direct summation
Each host need to receive all data from all other processors:
Communication independent of # of nodes.

# GRAPE-6 Network structure

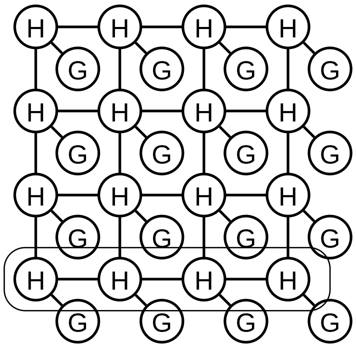



GRAPE at $(i,j)$ calculates the forces from particles in host $j$ on particles in host $i$.

No host-host communication required (except for synchronization)

Actual network design allows partitioning of the system.

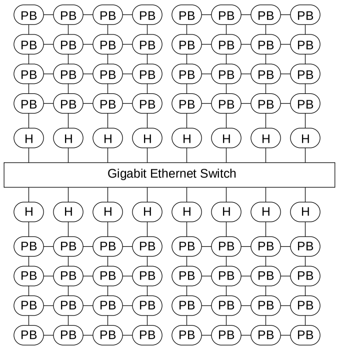
# An Alternative — $n^2$ hosts




This approach can be used for parallelization on MPPs without GRAPE.

Low-latency communication protocol is crucial for good performance.

# The full 64 Tflops GRAPE-6 system

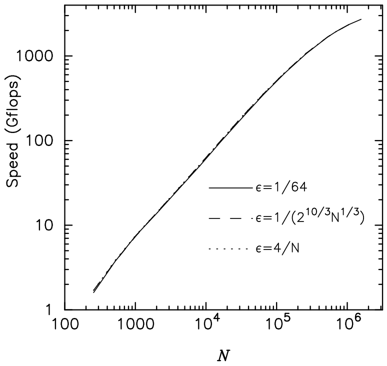



- 4-host, 16-board "block" with dedicated network
- 4 "blocks" connected through GbE network

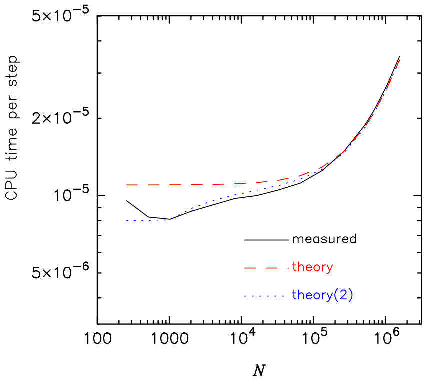
Combination of host network solution and dedicated network solution.

# Performance of GRAPE-6

Single host



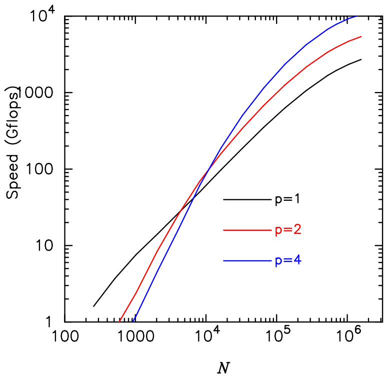
## Performance model




$T = T_{host} + T_{comm} + T_{GRAPE}$,

"theory (2)" include the effect of cache size

## Parallel performance (1)

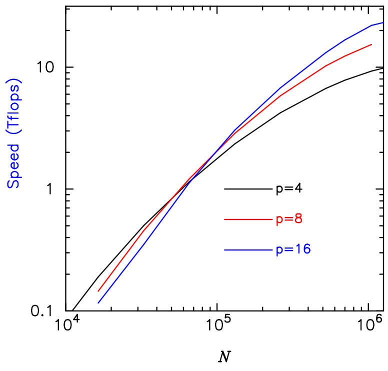



1, 2, 4-host systems

Communication through hardware network

Crossover point: $N \sim 3000$

## Parallel performance (2)

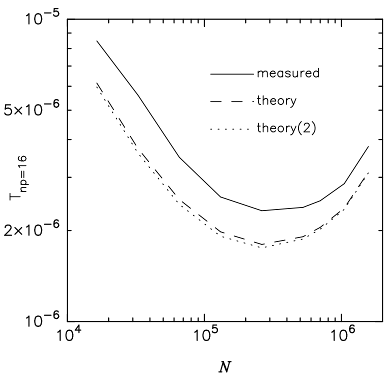



4,8,16-host systems

Communication partly through GbE host network

Crossover point: $N \sim 50,000$

## Parallel Performance model




$T = T_{host} + T_{comm} + T_{GRAPE}$,

Communication (syncronization) cost dominates the communication time (and total time.)

## Communication system

- Hardware
  - GbE network
  - NS83820-based PCI card
  - AMD K7+SiS 745
- Software
  - MPICH/p4 over TCP/IP (1.2.2)
  - Direct socket call for sync/reduction/point-to-point communication (5x throughput and 1/3 latency for the above card)

Max throughput: 70MB/s

## Improving network performance

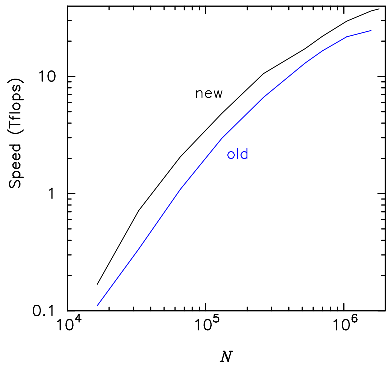
1. Replace GbE with something else
2. Replace TCP/IP with something else
3. Replace GbE chip/PCI chipset

We took option #3 (less expensive than option #1)

## New NIC (and new host)

| | Old | New |
|---|---|---|
| CPU | Athlon XP 1800+ | Intel P4 2.53GHz |
| CPU Clock | 1.53 GHz | 2.83 GHz |
| PCI Clock | 33 MHz | 37 MHz |
| Chipset | SiS 745 | Intel E7205 |
| NIC chip | NS 83820 | Intel 82540EM |
| Throughput | 70MB/s | 105 MB/s |
| Syncronization | 250 $\mu$s | 80 $\mu$s |

## Performance after tuning




$>50\%$ speedup for quite wide range of $N$

Exceeding 30 Tflops for 1M particles or more.

## "Best" real application performance

Evolution of early Kuiper belt
1.8M particles
Full 64-board, 16-host system
21120 dynamical time units (4000 years)
16.3 hours of actual CPU time
Total number of (individual) timesteps: $1.91 \times 10^{10}$

Floating point operations per timestep: $1.8 \times 10^6 \times 57$

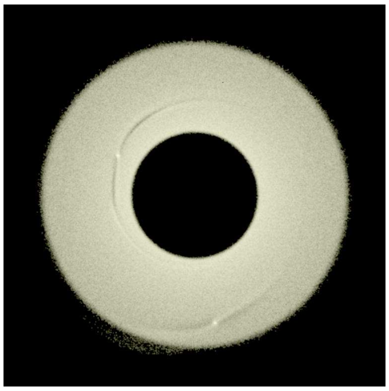
Total floating point operations: $1.96 \times 10^{18}$

The average speed during the run: 33.4 Tflops

## Planetesimal Simulation

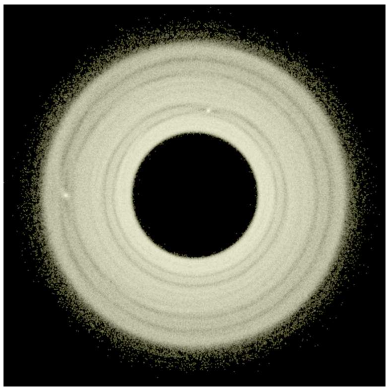
1. Planetesimals of the mass $\sim 10^{22}$g
2. Planetesimals of the mass $\sim 10^{22}$g
3. Radial range: 15—35 AU
4. Uranus at 20 AU and Neptune at 30 AU
5. Surface mass density $\propto r^{-1.5}$ (Hayashi model)

## Initial snapshot



Actually about 100 years after start.

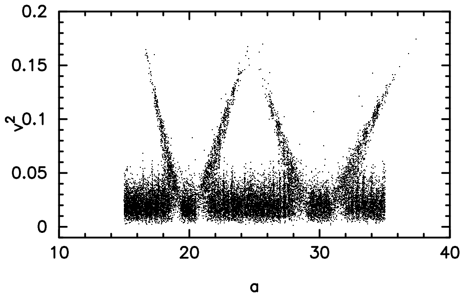
## Final snapshot



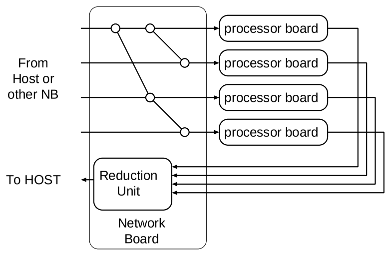
4800 years later

## Radius-random velocity relation

At 4800 years:




## A more flexible system




Can be configured for the broadcast from a single host, direct connections from $n$ hosts, and multicasts from $n/2$ hosts etc.

## Summary

- Single-host GRAPE-6 system has pretty good performancea characteristics, with getting 1 Tflops for $N = 10^5$.
- Up to 4 hosts, performance scaling is quite good.
- For larger systems, scaling is not as good as that for small systems.
- By improving the host-host communication throughput and latency, we improved ths speed of large systems by 50%.
- The best observed performance for real application is so far 33.4 Tflops.