GRAPE-DR: An accelerator for scientific computing

Jun Makino
Center for Computational Astrophysics
and
Division Theoretical Astronomy
National Astronomical Observatory of Japan

Who am I?

Current position: Director, Center for Computational Astrophysics (CfCA), National Astronomical Observatory of Japan
CfCA computers: Cray XT4 (812 quad-core nodes), NEC SX-9, several GRAPE hardwares....

What I have been doing for the last 20 years:
Developing GRAPE and similar hardwares for astrophysical N-body simulations, using them for research.

Talk structure

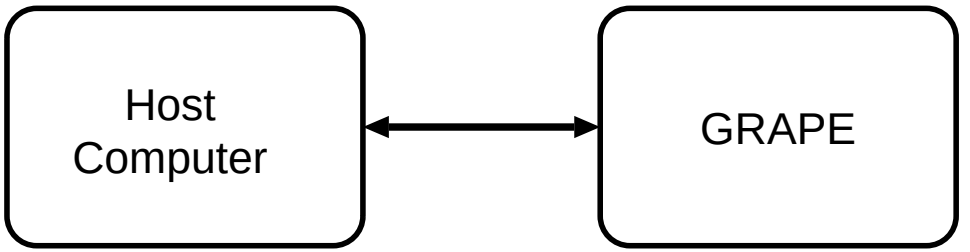
- GRAPE hardwares
  - GRAPE machines
  - Software Issues
- GRAPE-DR
  - Architecture
  - Comparison with other architecture
  - Development status

Short history of GRAPE

- Basic concept
- Application Example
- GRAPE-1 through 6
- Software Perspective

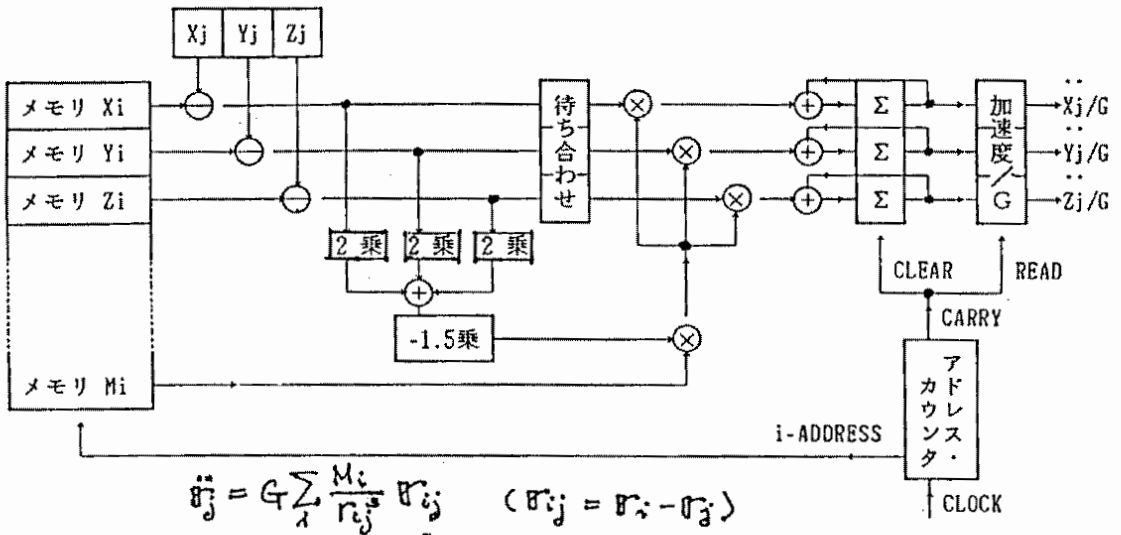
Basic concept (As of 1988)

- With N-body simulation, almost all calculation goes to the calculation of particle-particle interaction.
- This is true even for schemes like Barnes-Hut treecode or FMM.
- A simple hardware which calculates the particle-particle interaction can accelerate overall calculation.
- Original Idea: Chikada (1988)



Hybrid Architecture Computing

Chikada's idea (1988)



$$\ddot{\vec{r}}_j = G\sum_i \frac{M_i}{r_{ij}^3}\vec{r}_{ij} \qquad (\vec{r}_{ij} = \vec{r}_i - \vec{r}_j)$$

+, −, ×, 2乗は1 operation, -1.5乗は多項式近似でやるとして10operation 位に相当する.
総計24operation.
各operation の後にはレジスタがあって, 全体がpipelineになっているものとする.
「待ち合わせ」は2乗してMと掛け算する間の時間ズレを補正するためのFIFO(First-In First-Out memory).
「Σ」は足し込み用のレジスタ. N回足した後結果を右のレジスタに転送する.

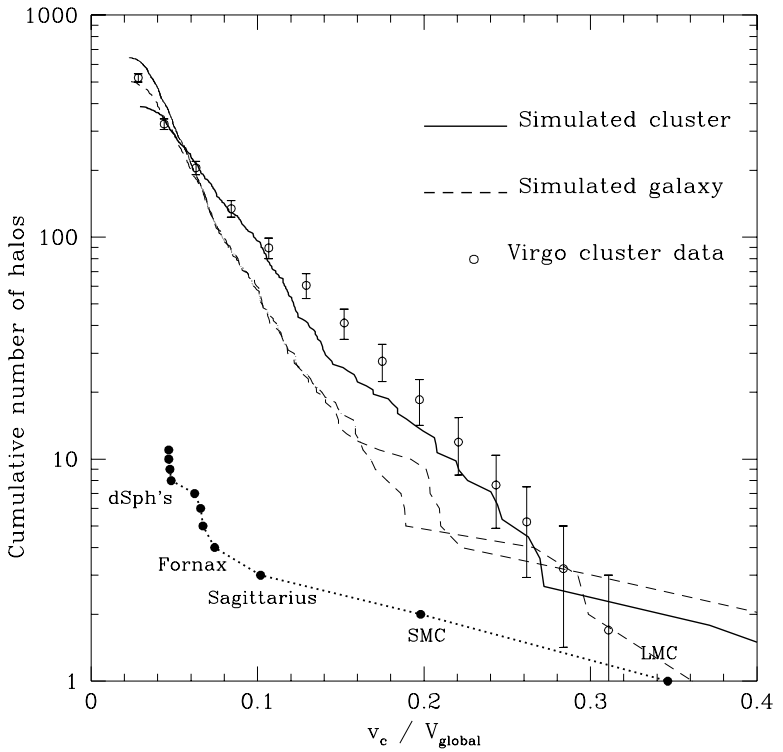
図2. N体問題のj-体に働く重力加速度を計算する回路の概念図.

- Hardwired pipeline for force calculation (similar to Delft DMDP)
- Hybrid Architecture (things other than force calculation done elsewhere)

Application: Dark Matter Halos

Problem:

Moore et al 1999



Galaxy-size Simulated Dark-matter halos contain far too many subhalos

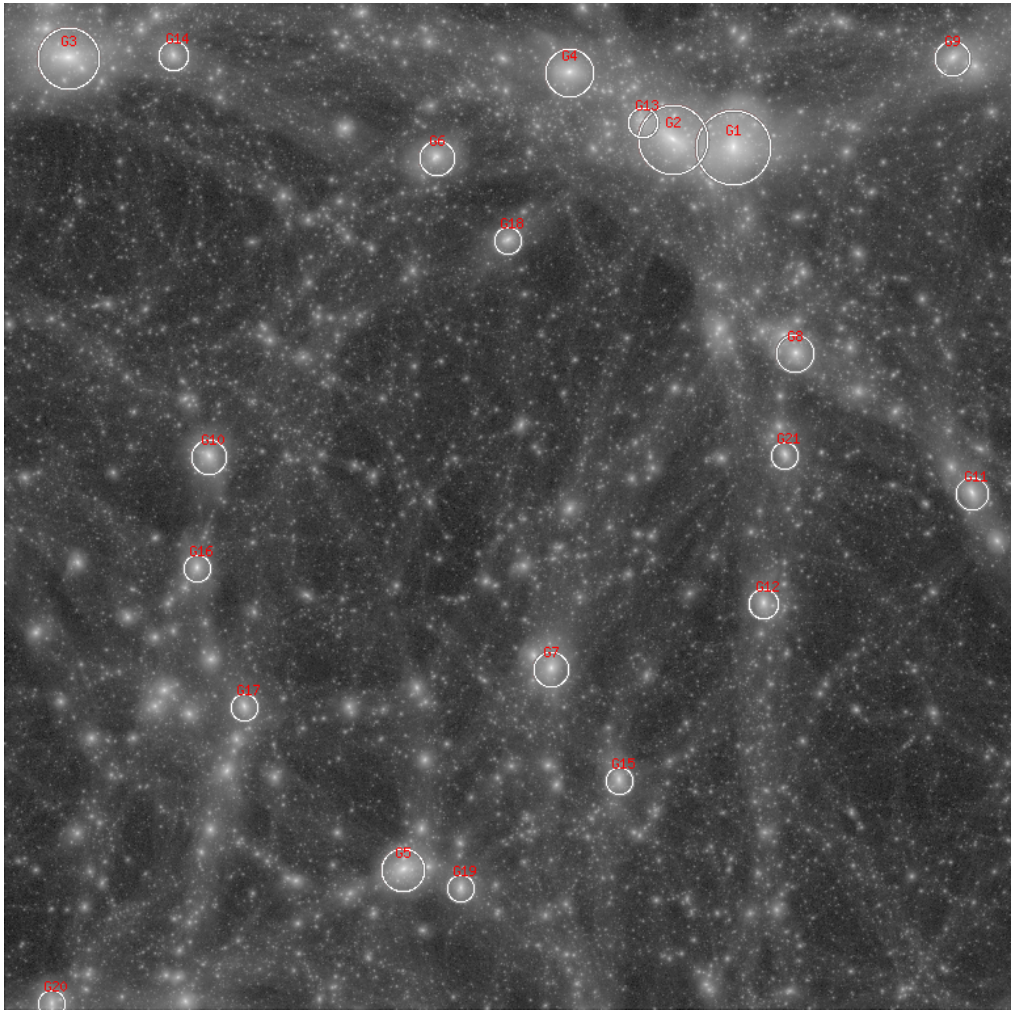
Our galaxy contain only ~ 10 satellite galaxies

Why?

Our calculation

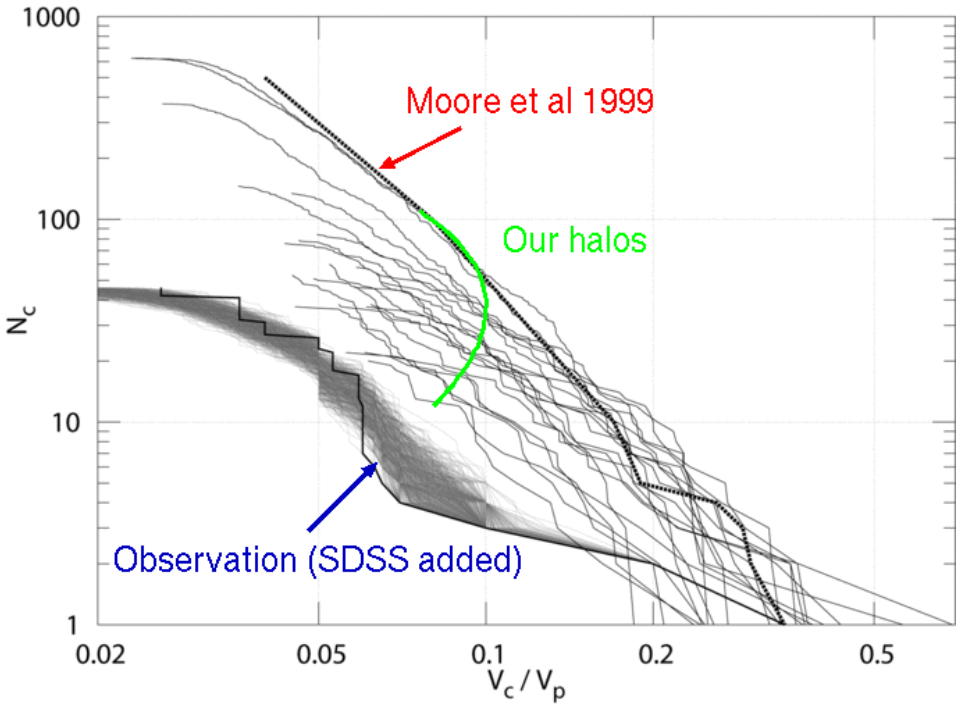
- **"Observe" all simulated halos in one simulation box**
- **GRAPE-6A cluster/PC Cluster/Cray XT4**
- **512^3 — 1600^3 particles**

512^3 and 1024^3 results



1024^3 movie

Result

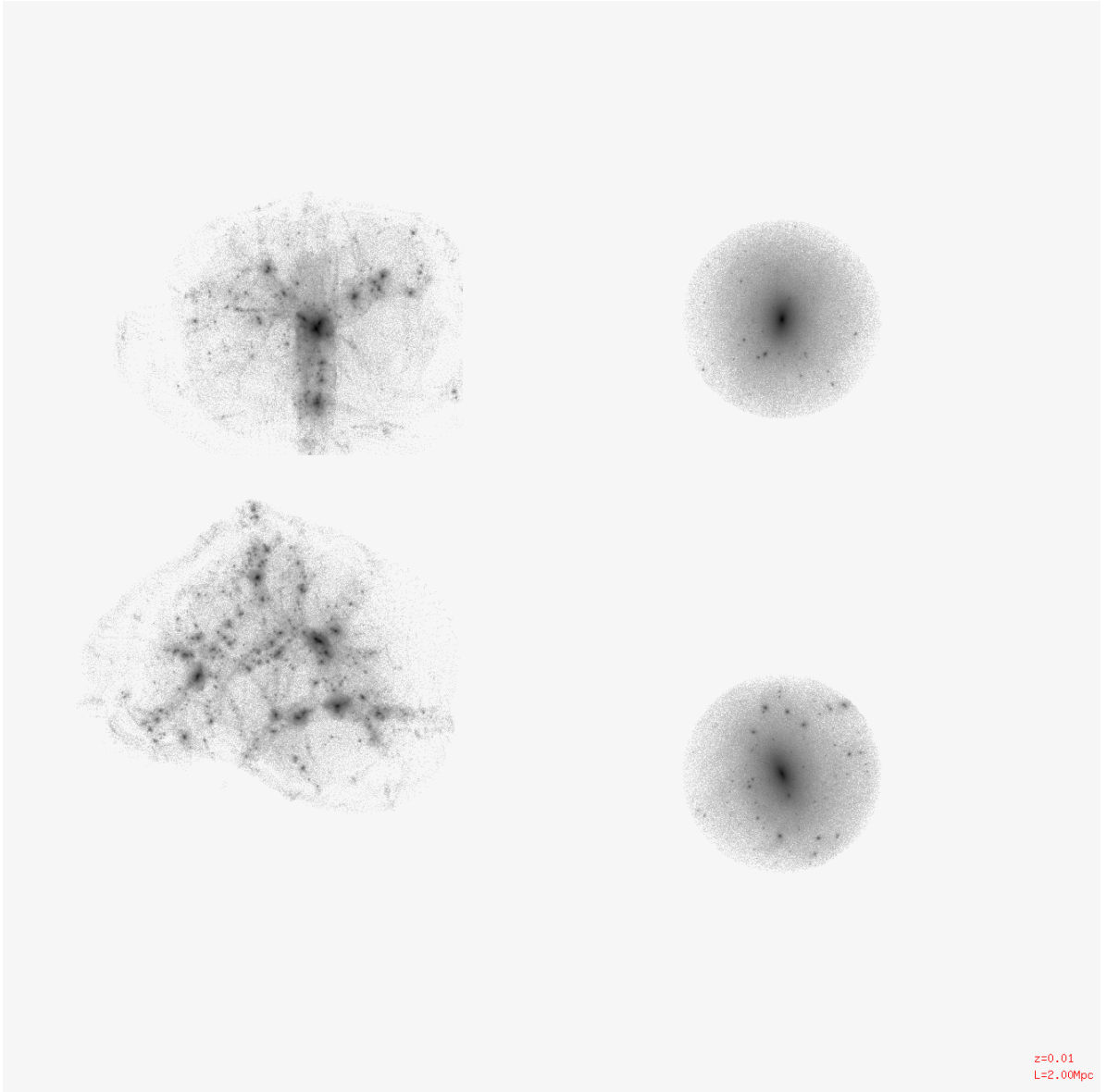


- Large variation in number of subhalos
- The richest ones agree with Moore's result

The poorest ones are within a factor of two with observations

= Dark CDM subhalos are not necessary

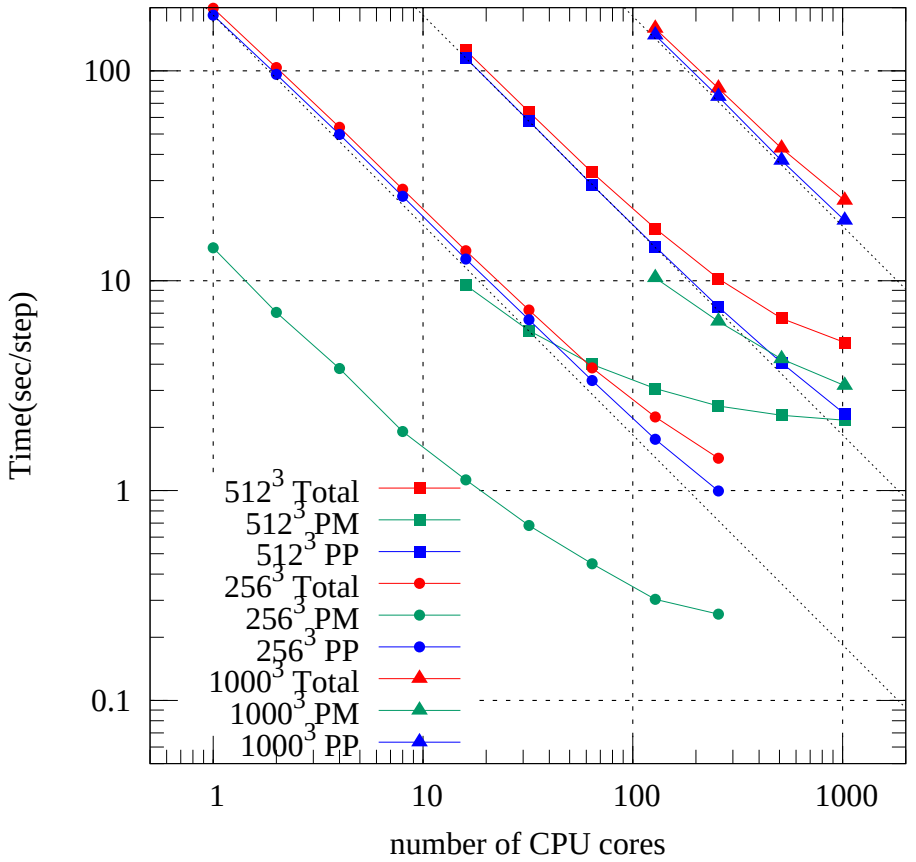
Poor and Rich halos



A poor halo at z=3 (left) and 0 (right)

A rich halo at z=3 (left) and 0 (right)

Performance (On Cray XT4)

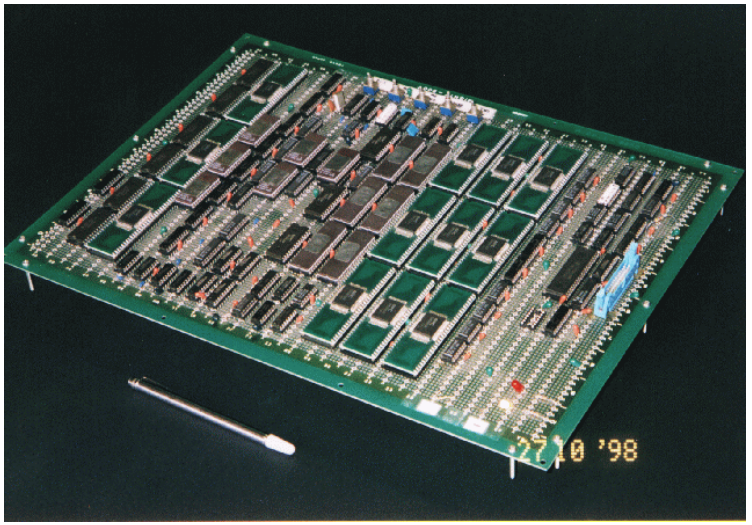


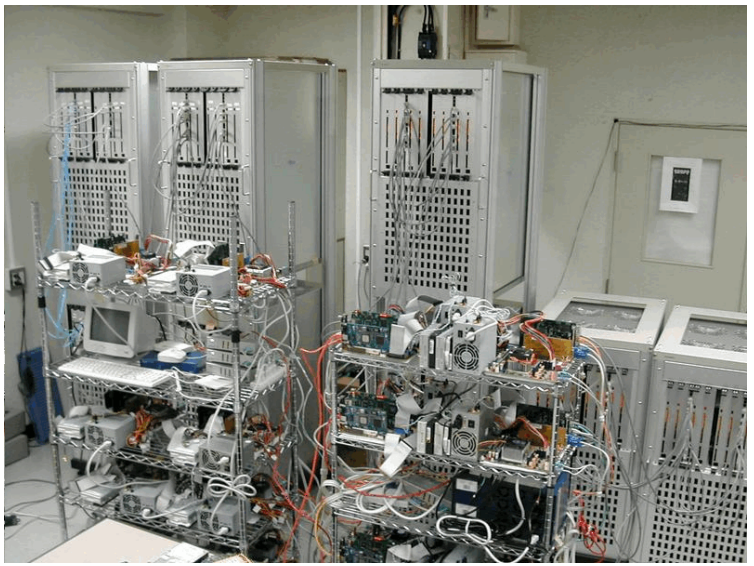
Practically linear scaling up to the size of machine we have (3000 cores)

Galaxy Formation/Merging

Galaxy Formation
Merging
***N*-body+SPH (Smoothed Particle Hydrodynamics)**

GRAPE-1 to GRAPE-6



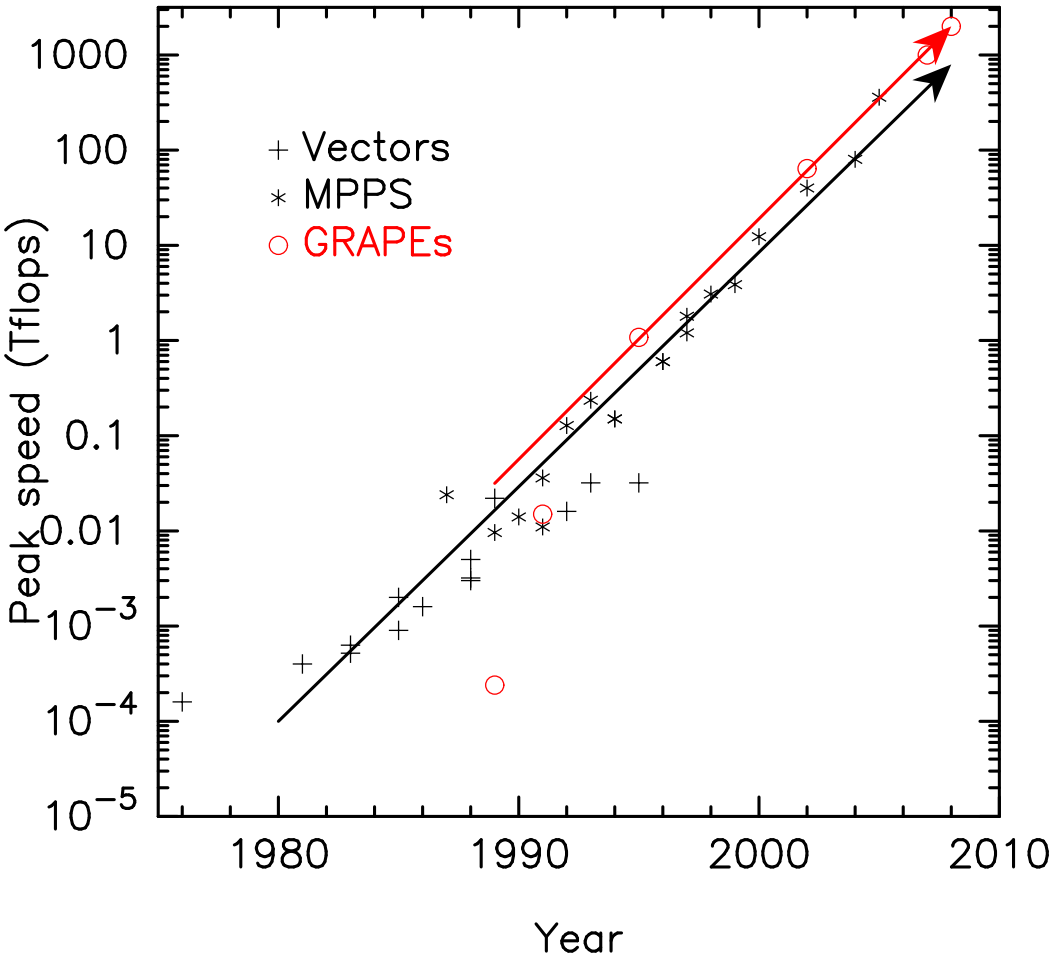


GRAPE-1: 1989, 308Mflops

GRAPE-4: 1995, 1.08Tflops

GRAPE-6: 2002, 64Tflops

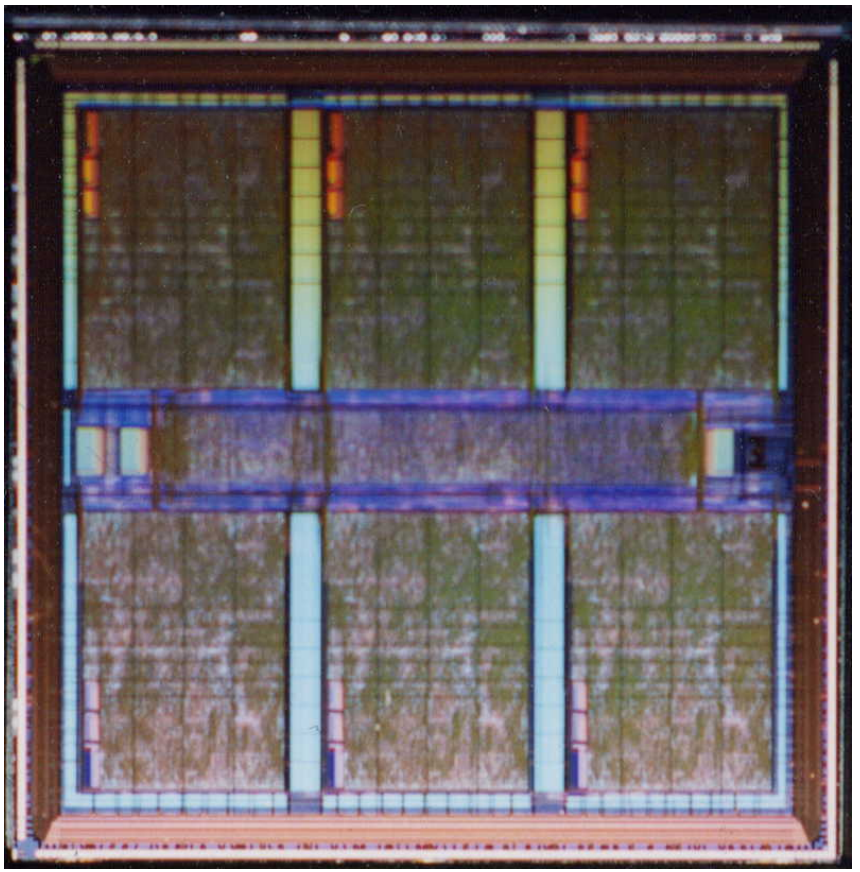
Performance history



Since 1995 (GRAPE-4), GRAPE has been faster than general-purpose computers.

Development cost was around 1/100.

GRAPE-6 Processor LSI



- 0.25 μm design rule (Toshiba TC-240, 1.8M gates)
- 90 MHz Clock
- 6 pipeline processors
- 32.4 Gflops / chip

Comparison with a recent Intel processor

	GRAPE-6	Intel Xeon X7460
Year	1999	2008
Design rule	250nm	45nm
Clock	90MHz	2.66GHz
Peak speed	32.4Gflops	64Gflops
Power	10W	130 W
Perf/W	3.24Gflops	0.49 Gflops

Even after 10 years...

Software/Algorithm perspective

- How we develop softwares for GRAPE?
- Is porting (for example from GRAPE-4 to GRAPE-6) difficult?
- Are programs developed for GRAPE "tied" to GRAPE hardware?

Software development for GRAPE

GRAPE software library provides several basic functions to use GRAPE hardware.

- **Sends particles to GRAPE board memory**
- **Sends positions to calculate the force and start calculation**
- **get the calculated force (asynchronous)**

User application programs use these functions.
Algorithm modifications (on program) are necessary to reduce communication and increase the degree of parallelism (essentially blocking).

Porting issues (within GRAPE hardwares)

- Libraries for GRAPE-4 and 6 (for example) are not compatible
- Even so, porting was not so hard. The calls to GRAPE libraries are limited to a fairly small number of places in an entire application code.

Porting issues (to other architectures)

- **Blocked algorithms were originally developed for a vector architecture (CDC Cyber 205).**
- **Tuning of these algorithm for GRAPE architecture resulted in extremely bandwidth-efficient programs.**
- **GPGPU, CELL, and SIMD features of microprocessors can be used efficiently once highly-optimized GRAPE-emulation library is developed for these architectures** (in practice things are more complex....)**.**
- **As a result, lots of good work on use of GPGPU for particle simulations (Hamada, Nitatori, Narumi, Yasuoka, Portegies Zwart ....)**

Real-World issues with "Porting"

— Mostly on GPGPU....

- Getting something run on GPU is not difficult
- Getting the good performance number compared with non-optimized, single-core x86 performance is not so hard.
- Making it faster than 10-year-old GRAPE or highly-optimized code on x86 (using SSE/SSE2) is *VERY, VERY HARD*
- This is *mostly* software issues
- Some of the most serious ones are limitations in the architecture (lack of good reduction operation over processors etc)

"Problem" with GRAPE approach

- Chip development cost becomes too high.

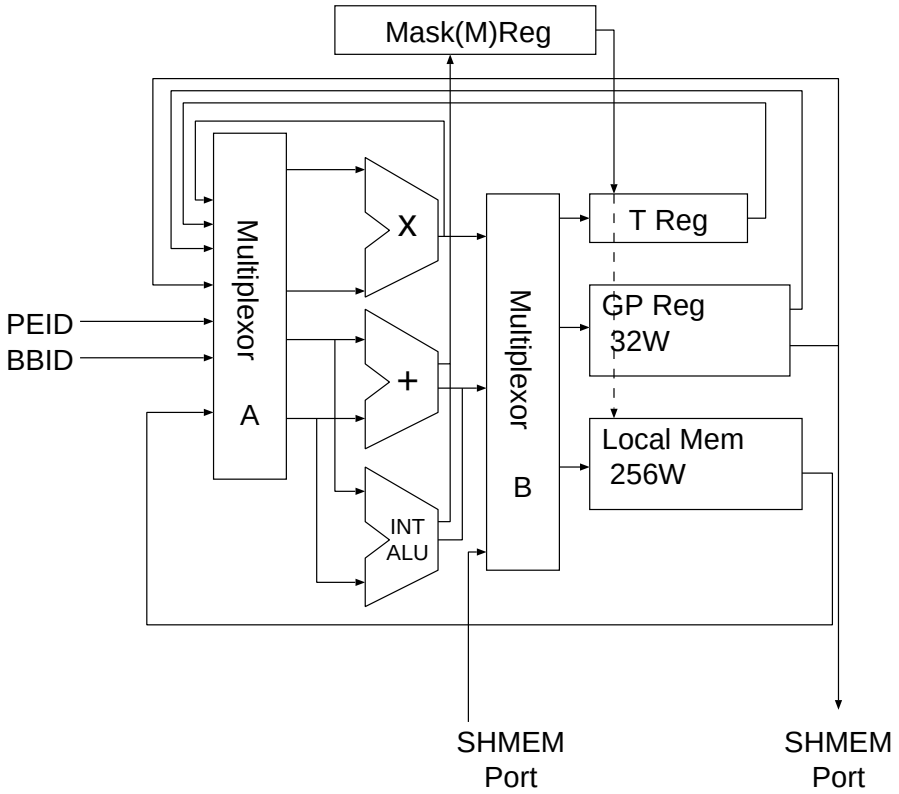
Year	Machine	Chip initial cost	process
1992	GRAPE-4	200K$	1μm
1997	GRAPE-6	1M$	250nm
2004	GRAPE-DR	4M$	90nm
2009?	GDR2?	$>$ 10M$	45nm?

Initial cost should be 1/4 or less of the total budget.
How we can continue?

Next-Generation GRAPE — GRAPE-DR

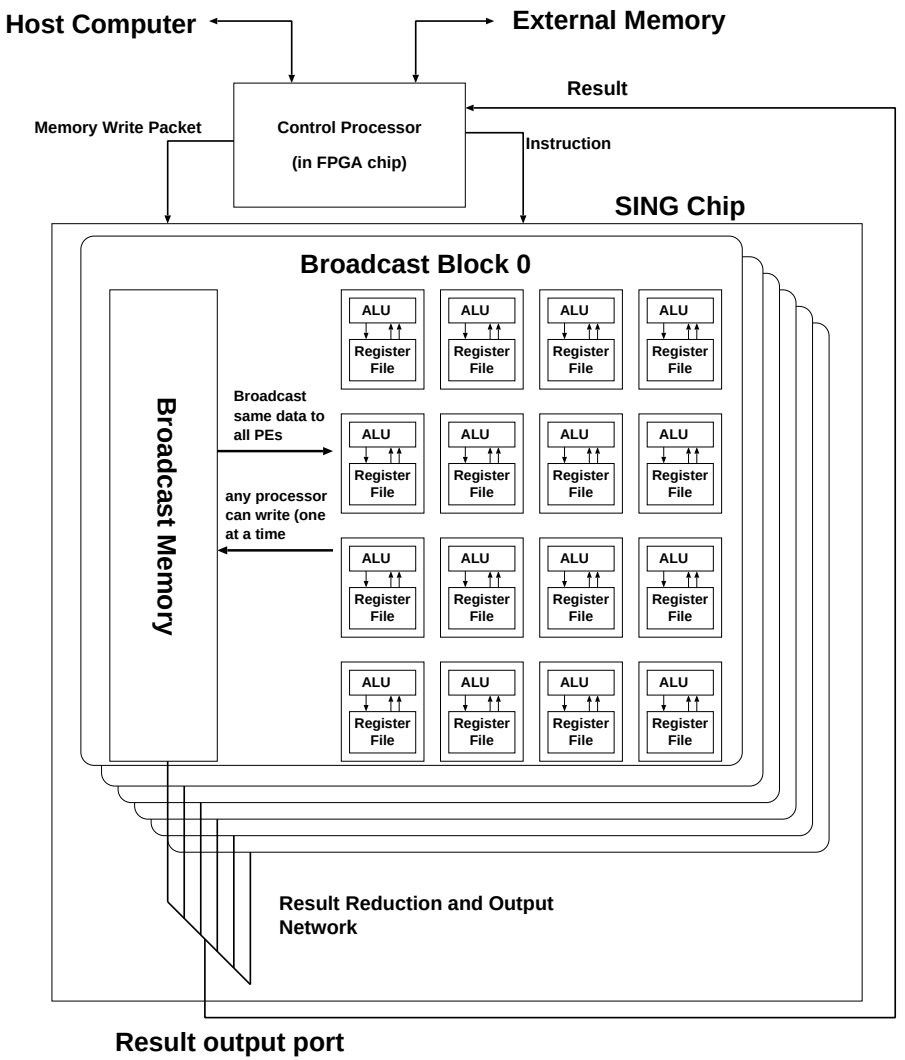
- Planned peak speed: 2 Pflops
- New architecture — wider application range than previous GRAPEs
- primarily to get funded
- No force pipeline. SIMD programmable processor
- Planned completion year: FY 2008 (early 2009)

Processor architecture



- Float Mult
- Float add/sub
- Integer ALU
- 32-word registers
- 256-word memory
- communication port

Chip structure



Collection of small processors.

512 processors on one chip
500MHz clock

Peak speed of one chip: 512Gflops (SP)
(256Gflops DP)

Hardware support for chip-wide reduction

Why we changed the architecture?

- To get budget (Theoretical Astrophysics is too narrow...)
- To make machine useful for a wider range of applications
  - Molecular Dynamics
  - Boundary Element method
  - Dense matrix computation *(LINPACK!)*
  - SPH
- To use a wider range of algorithms
  - FMM
  - Ahmad-Cohen

Design Decisions and range of applications

Major design decisions which limits the application range

- **Limited external memory bandwidth (4GB/s)**
- **Limited host communication bandwidth (PCIe x16 Gen 1)**
- **Limited On-chip memory (in total 1MB)**

These decisions are essential in reducing the hardware cost and power consumption.
Numbers are chosen to be able to run a fairly wide range of applications, including LINPACK (DGEMM).

Comparison with FPGA

- much better silicon usage (ALUs in custom circuit, no programmable switching network)
- (possibly) higher clock speed (no programmable switching network on chip)
- easier to program (no VHDL necessary; assembly language and compiler instead)

Comparison with GPGPU

Pros:

- **Significantly better silicon usage (512PEs with 90nm)**
- **Designed for scientific applications reduction, small communication overhead, etc**

Cons:

- **Higher cost per silicon area... (small production quantity)**
- **Longer product cycle... 5 years vs 1 year**

Good implementations of N-body code on GPGPU are coming (Hamada, Nitadori, Portegies Zwart, Harris, ...)

Comparison with GPGPU(2)

	GRAPE-DR	nV G92	AMD FS9170
Design rule	90	65	55
Clock(GHz)	0.5	1.5	0.8
# FPUs	512	112	320
SP peak(GF)	512	336	512
DP peak(GF)	256	—	?
Power(W)	65	70?	150?

Power Consumption Comparison

Single-node performance and power consumption including the host CPU.

	GRAPE-DR	ClearSpeed e710	IBM PowerXCell
chips/node	8	(2?)	4 (Tribrade)
DP Peak	2T	0.2T	0.41T
Power (W)	800	300?	700
GFlops/W	2.5	0.67	0.59

How do you use it?

- GRAPE replacement: The necessary software is now ready. Essentially the same as GRAPE-6.
- Matrix etc ... DGEMM implemented
- Other applications:
  - Primitive Compiler available
  - For high performance, you need to write the kernel code in assembly language (for now)

Computation Model

Parallel evaluation of

$$\boldsymbol{R_i} = \sum_{\boldsymbol{j}} \boldsymbol{f}(\boldsymbol{x_i}, \boldsymbol{y_j})$$

- parallel over both i and j
- x_j may be omitted (trivial parallelism)
- $S_{i,j} = \sum_{k} f(x_{i,k}, y_{k,j})$ also possible

Primitive compiler

(Nakasato 2006)

```
/VARI  xi, yi, zi, e2;
/VARJ  xj, yj, zj, mj;
/VARF  fx, fy, fz;
dx = xi - xj;
dy = yi - yj;
dz = zi - zj;
r2 = dx*dx + dy*dy + dz*dz + e2;
r3i= powm32(r2);
ff = mj*r3i;
fx += ff*dx;
fy += ff*dy;
fz += ff*dz;
```

- **Assembly code**
- **Interface/driver functions**
- **SIMD parallel data distribution**
- **Data reduction**

are generated from this "high-level description".
(Can be ported to GPUs)

Interface functions

```
struct SING_hlt_struct0{
  double xi;
  double yi;
  double zi;
  double e2;
};
int SING_send_i_particle(struct SING_hlt_struct0 *ip,
                         int n);
...

int SING_send_elt_data0(struct SING_elt_struct0 *ip,
                        int index_in_EM);
...
int SING_get_result(struct SING_result_struct *rp);

int SING_grape_run(int n);
```

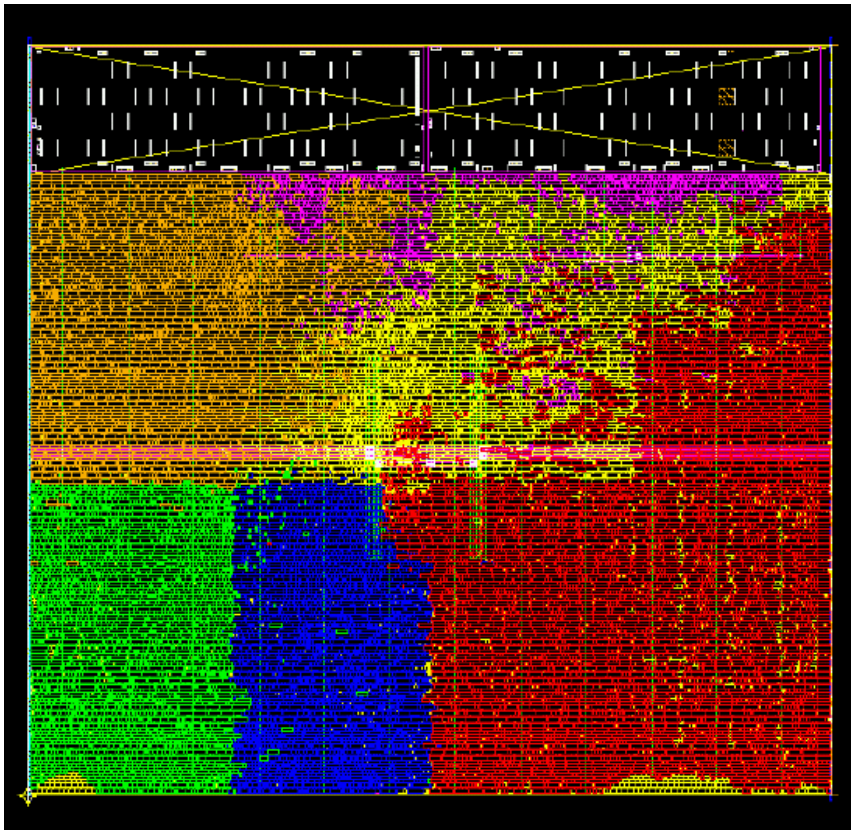
Some characteristic of software

- Parallelization on chip/board is automatic.
- Codes for communication (including reduction) are generated automatically.
- Data transfer and calculation are automatically overlapped.
- Same source can run on GPU, FPGA-based accelerator, etc.

The Chip

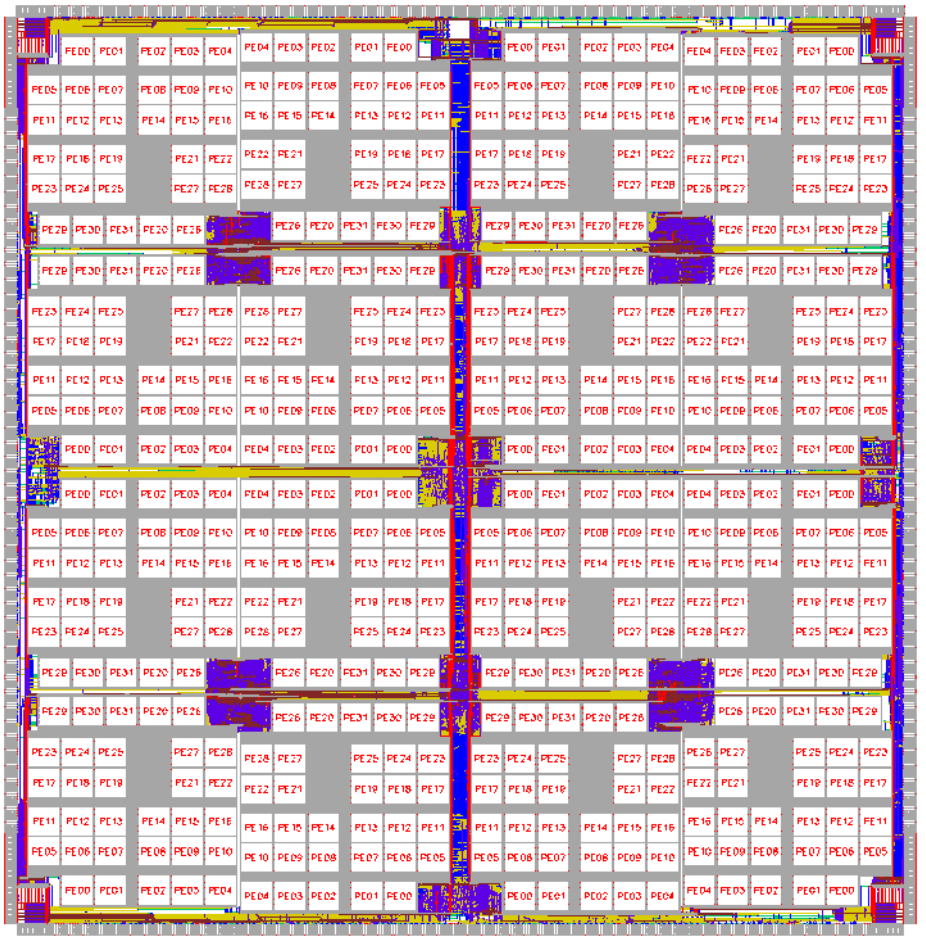
Sample chip delivered May 2006

PE Layout



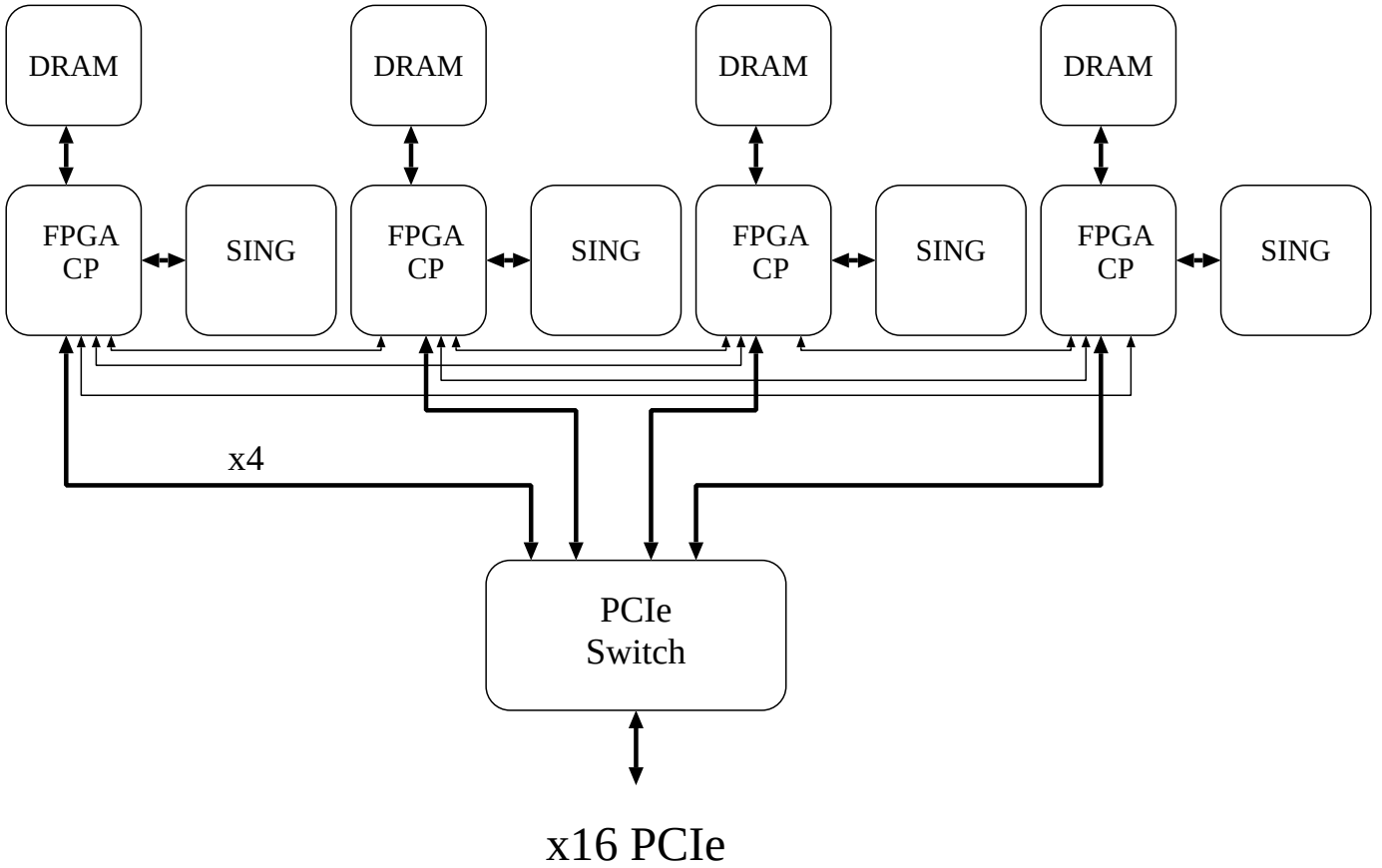
0.7mm by 0.7mm
Black: Local Memory
Red: Reg. File
Orange: FMUL
Green: FADD
Blue: IALU

Chip layout

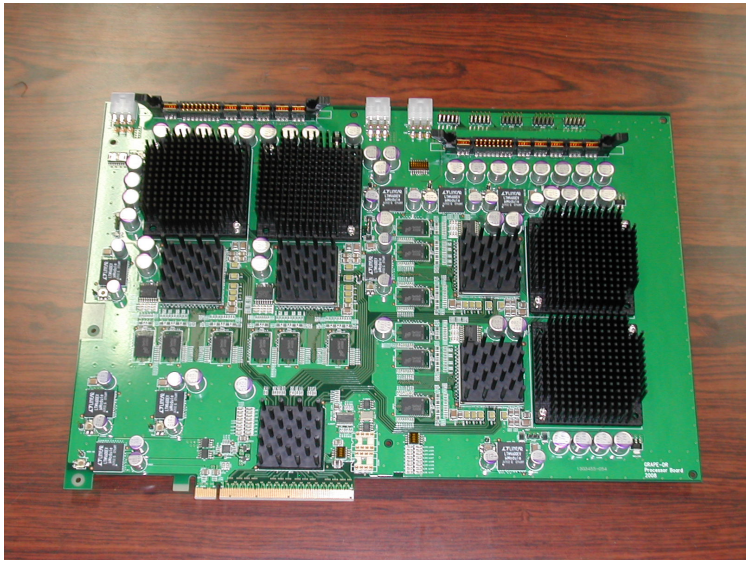


- 32PEs in 16 groups
- 18mm by 18mm

Processor board block diagram



Processor board

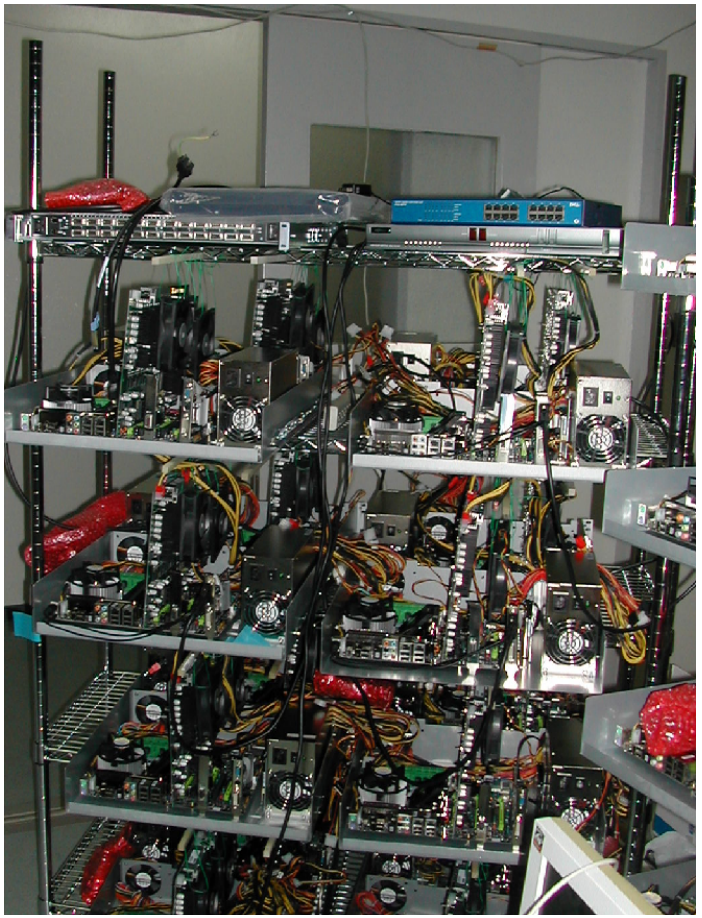


PCIe x16 (Gen 1) interface
Altera Arria GX as DRAM controller/communication interface

- Around 250W power consumption
- Not quite running at 500MHz yet... (FPGA design not optimized yet)
- 900Gflops DP peak (450MHz clock)
- Available from K&F Computing Research

GRAPE-DR cluster system

Just to show that the system exists...



Host computer: Intel Core 2 Quad Q6600 with nVidia 780i chipset

8GB memory

Network: IB (4x DDR)

HPC Linpack passed (not tuned yet....)

The system and (preliminary) performance numbers submitted to TOP500

Major concern: Effective host memory bandwidth

GDR cluster in early 2009

- Majority of board with Gen2 interface (new chip from PLX)
- Nehalem with 3way DDR3 memory should resolve potential bandwidth problem.
- IB network
- 800T-1P DP peak range.

GDR-2?

- Current design has rather large room for improvement, in many places.
- With 45nm, it is not difficult to achieve
  - 2 DP Gflops/chip
  - 4 SP Tflops/chip
  - On-chip memory (16-32MB)
- System cost will be much cheaper.

Summary

- GRAPE-DR, with programmable processors, will have wider application range than traditional GRAPEs.
- Small cluster of GDR system is now up and running
- Should be able to put some number for Nov 11 Top 500
- Peak speed of a card with 4 chips will be 1 Tflops (DP).
- The system to be completed in early 2009 will have the peak speed around 1Pflops (DP)