GRAPE and GRAPE-DR

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

GPU-SMP 2010 July 27-29, 2010, Harbin, China

Summary

- GRAPEs, special-purpose computer for gravitational N-body system, have been providing 10x - 100x more computational power compared to general-purpose supercompuers.
- GRAPE-DR, with programmable processors, has wider application range than traditional GRAPEs.
- Peak speed of a GRAPE-DR card with 4 chips is 800 Gflops (DP).
- DGEMM performance 640 Gflops,
  LU decomposition $>$ 400Gflops
- Achieved the best performance per W (Top 1 in the Little Green 500 list, 815Mflops/W)
- Accelerators require new algorithms, not just porting and tuning

Talk structure

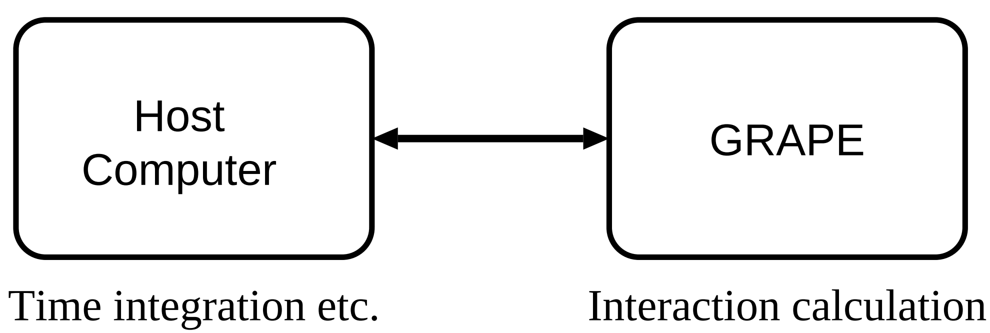
- Short history of GRAPE
- GRAPE-DR
  - Architecture
  - Comparison with other architecture
  - Application tuning/performance examples
- Summary
- (Next-generation GRAPE)

Short history of GRAPE

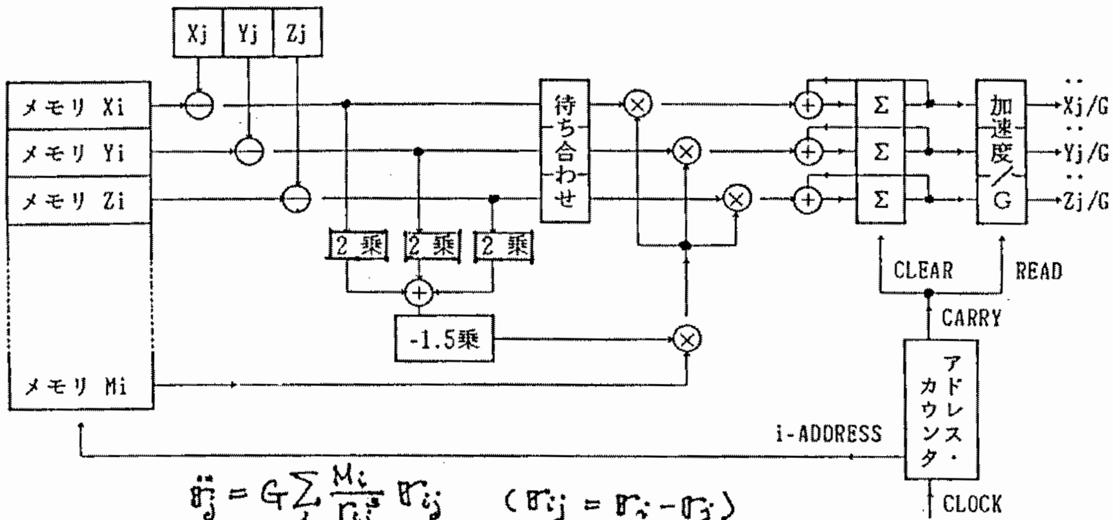
- Basic concept
- GRAPE-1 through 6
- Software Perspective

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)**



Chikada's idea (1988)



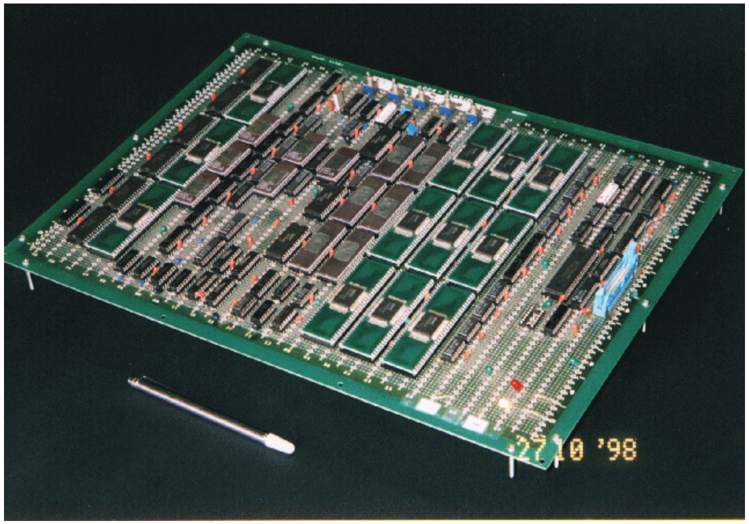
$$\ddot{\mathbf{r}}_j = G\sum_i \frac{M_i}{r_{ij}^3}\mathbf{r}_{ij} \quad (\mathbf{r}_{ij} = \mathbf{r}_i - \mathbf{r}_j)$$

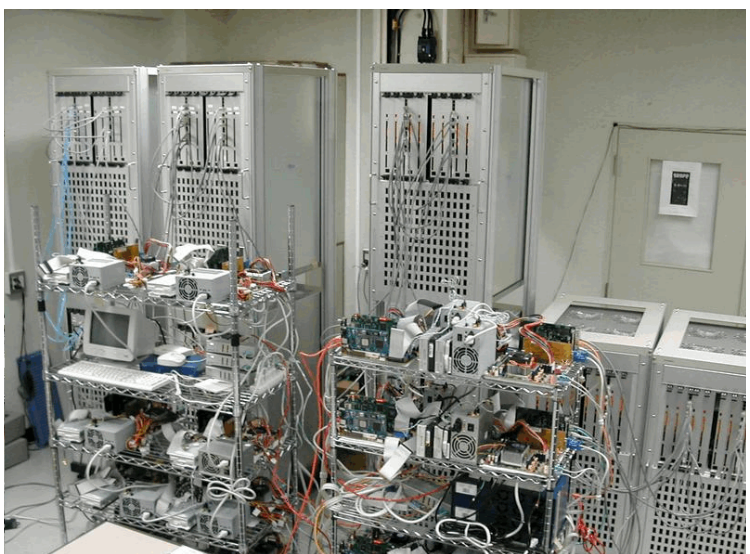
+，－，×，２乗は１operation， -1.5乗は多項式近似でやるとして10operation 位に相当する．
総計24operation.
各operation の後にはレジスタがあって，全体がpipelineになっているものとする．
「待ち合わせ」は２乗してMと掛け算する間の時間ズレを補正するためのFIFO(First-In First-Out memory)．
「Σ」は足し込み用のレジスタ．N回足した後結果を右のレジスタに転送する．

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)

GRAPE-1 to GRAPE-6



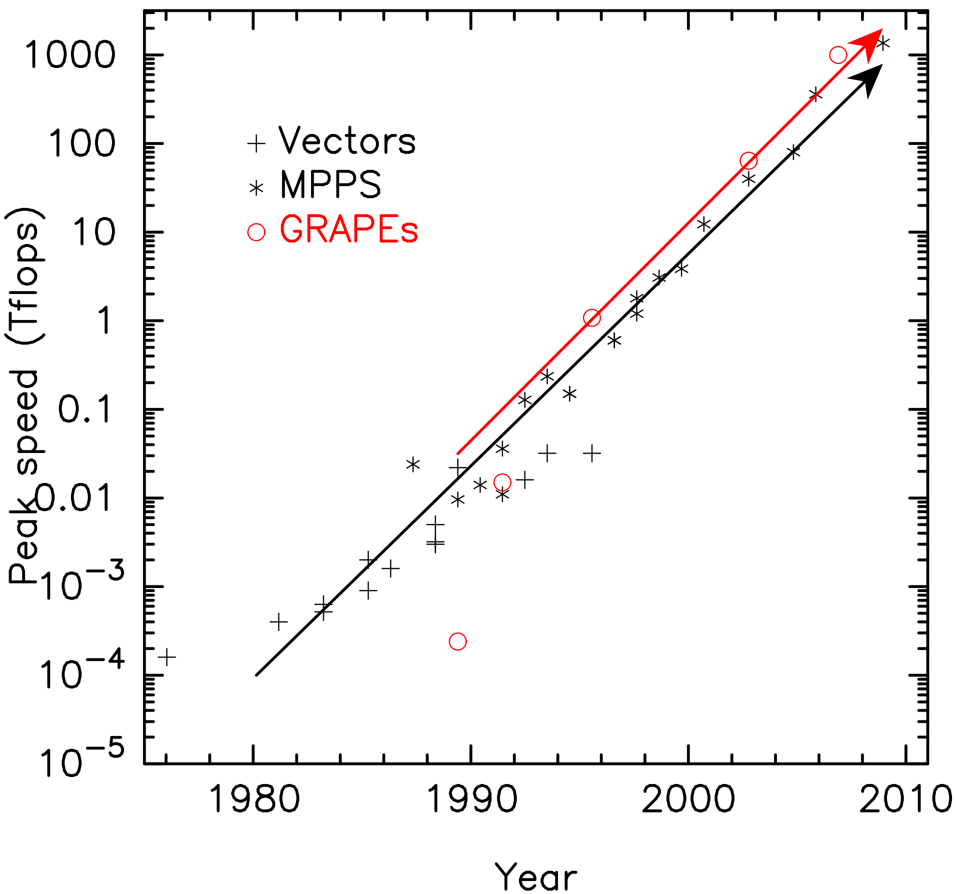


GRAPE-1: 1989, 308Mflops

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.

Science on GRAPEs

- **Pure N-body**
  - **Planetary formation (Kokubo, Ida, ...)**
  - **Star clusters (JM, Baumgardt, Portegies Zwart, Hurley, ...)**
  - **Galactic Dynamics (Athanassoula, Fujii, ...)**
  - **Galaxies with central BH (JM, Iwasawa, ...)**
  - **Cosmology (Fukushige, Yoshikawa)**
- **SPH**
  - **Galaxy Formation (Steinmetz, Susa, Saitoh)**
  - **Star formation (Klessen)**

“Problem” with GRAPE approach

- Chip development cost has become too high.

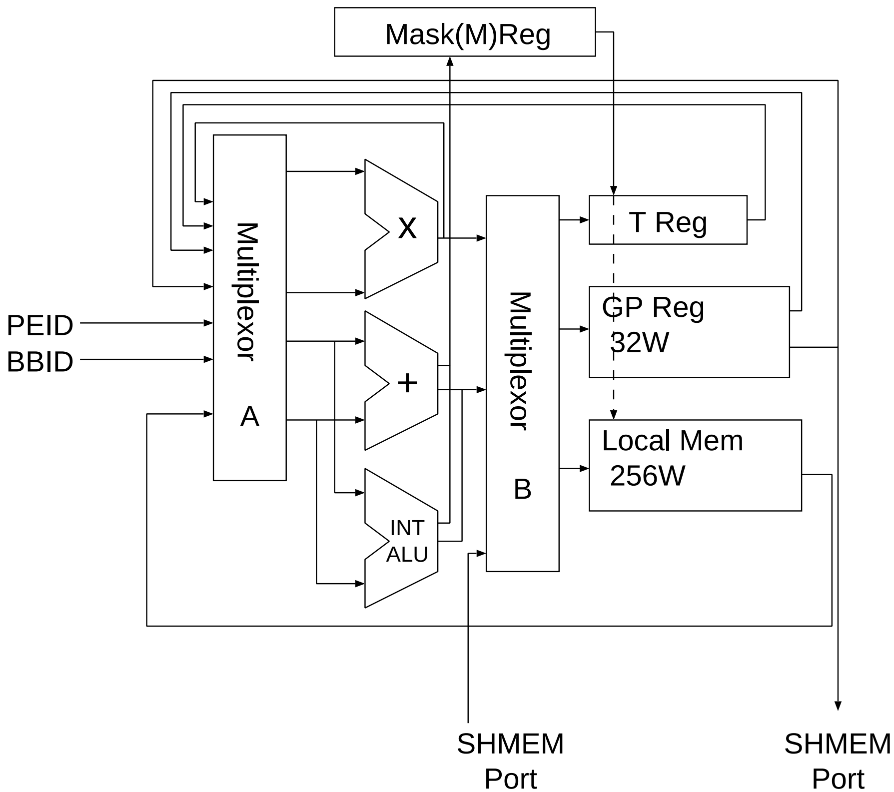
Year	Machine	Chip initial cost	process
1992	GRAPE-4	200K$	1μm
1997	GRAPE-6	1M$	250nm
2004	GRAPE-DR	4M$	90nm
2010?	GDR2?	$>$ 10M$	45nm?

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

Current Generation— GRAPE-DR

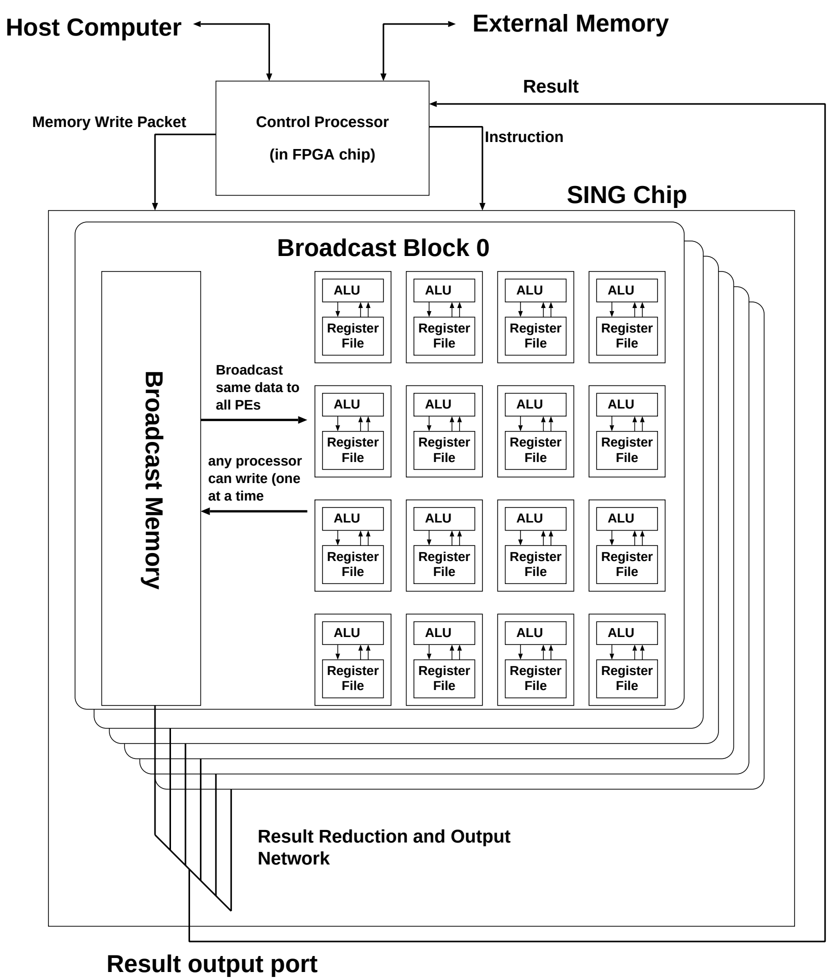
- New architecture — wider application range than previous GRAPEs
- primarily to get funded
- No force pipeline. SIMD programmable processor
- "Parallel evolution" with GPUs.
- Developent: FY 2004-2008

Processor architecture



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

Chip architecture



- 32 PEs organized to "broadcast block" (BB)
- BB has shared memory.
- Input data is broadcasted to all BBs.
- Outputs from BBs go through reduction network (sum etc)

Computation Model

Parallel evaluation of

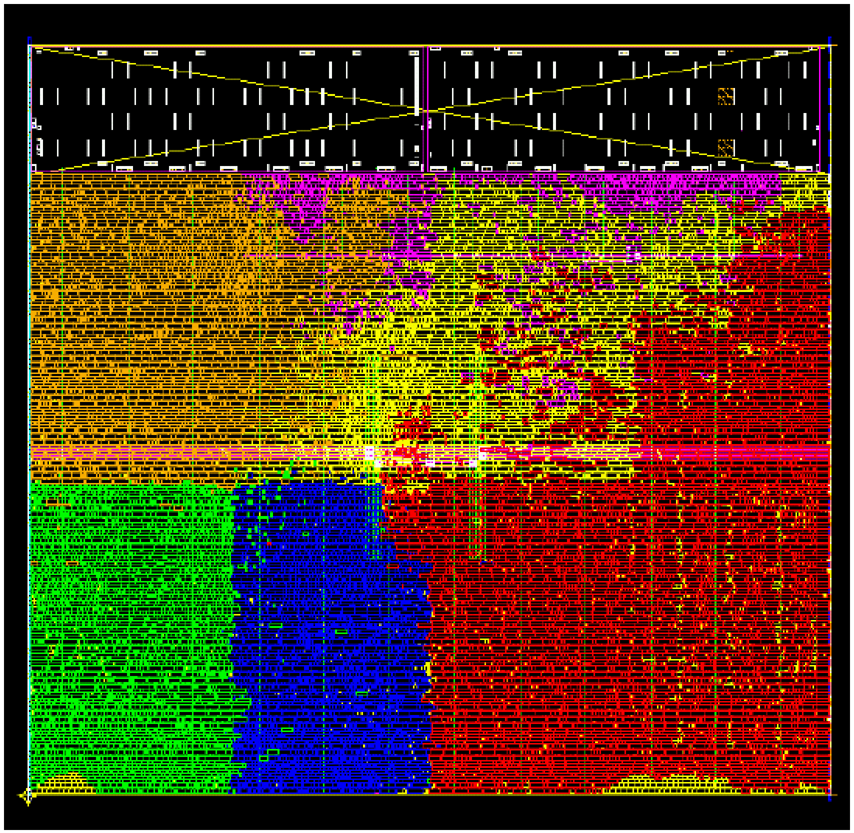
$$\boldsymbol{R_i = \sum_j f(x_i, y_j)}$$

- **parallel over both i and j**
- **y_j may be omitted (trivial parallelism)**
- **$S_{i,j} = \sum_k f(x_{i,k}, y_{k,j})$ also possible (matrix multiplication)**

The Chip

Sample chip delivered May 2006
90nm TSMC, Worst case 65W@500MHz

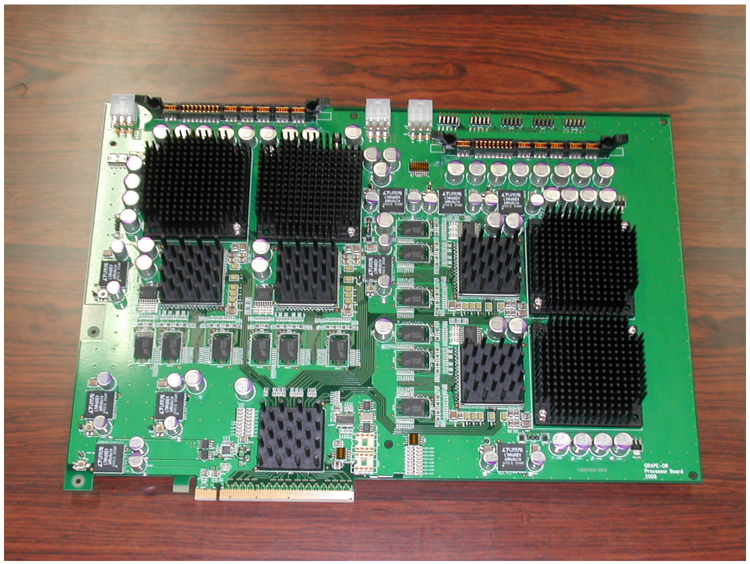
PE Layout



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

0.7mm by 0.7mm
800K transistors
0.13W@500MHz
1Gflops/512Mflops peak (SP/DP)

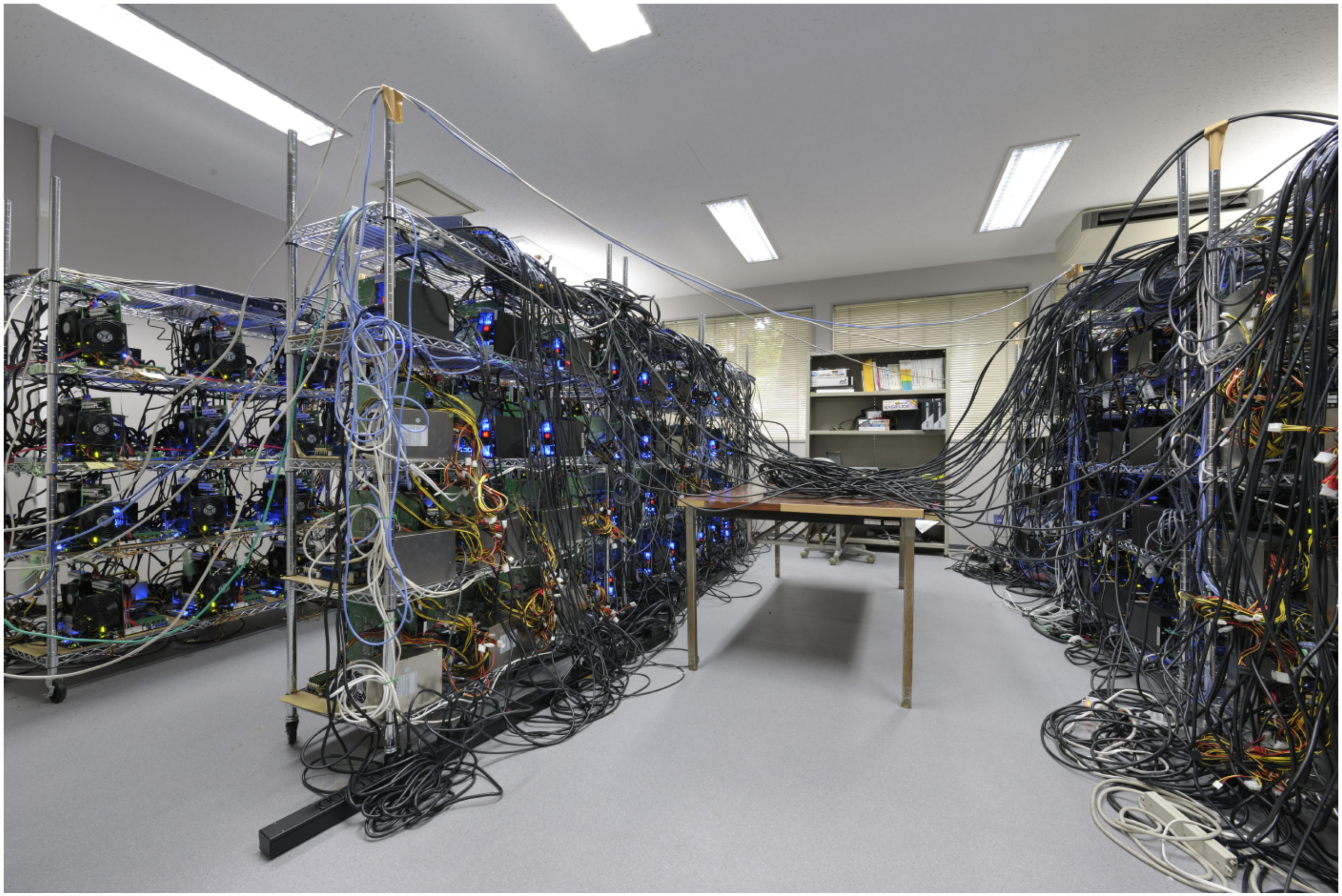
Processor board



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

- Around 200W power consumption
- Not quite running at 500MHz yet... (FPGA design not optimized yet)
- 819Gflops DP peak (400MHz clock)
- Available from K&F Computing Research (www.kfcr.jp)

GRAPE-DR cluster system



GRAPE-DR cluster system

- 128-node, 128-card system (105TF theoretical peak @ 400MHz)
- Linpack measured: 360 Gflops/node
- Gravity code: 340Gflops/chip
- Host computer: Intel Core i7+X58 chipset, 12GB memory (some are 18GB nodes)
- network: x4 DDR Infiniband
- plan to expand to 384-node system.

Software Environment

- Assembly Language
- Kernel libraries
  - matrix multiplication
    - BLAS, LAPACK
  - Particle-Particle interaction
- Compiler Language
- OpenMP-like interface

Idea based on PGDL (Hamada, Nakasato)
— pipeline generator for FPGA

Compiler language example

Nakasato (2008), based on LLVM.

```
VARI xi, yi, zi;
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;
rinv = rsqrt(r2);
mr3inv = rinv*rinv*rinv*mj;
fx+=  mr3inv*dx;
fy+=  mr3inv*dy;
fz+=  mr3inv*dz;
```

Translated to assembly language and API calls. Achieves near-peak performance, without further tuning. (also available for NVIDIA and AMD GPUs)

Driver functions

Generated from the description in the previous slide

```
int SING_send_j_particle(struct grape_j_particle_struct *jp,
                         int index_in_EM);
int SING_send_i_particle(struct grape_i_particle_struct *ip,
                         int n);
int SING_get_result(struct grape_result_struct *rp);
void SING_grape_init();
int SING_grape_run(int n);
```

OpenMP-like compiler

Goose compiler (Kawai 2009)

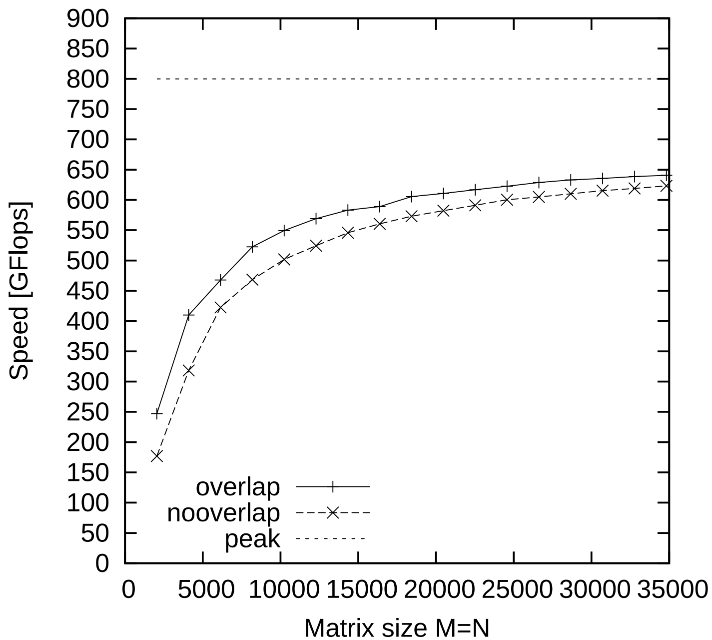
```
#pragma goose parallel for icnt(i) jcnt(j) res (a[i][0..2])
    for (i = 0; i < ni; i++) {
        for (j = 0; j < nj; j++) {
            double r2 = eps2[i];
            for (k = 0; k < 3; k++) dx[k] = x[j][k] - x[i][k];
            for (k = 0; k < 3; k++) r2 += dx[k]*dx[k];
            rinv = rsqrt(r2);
            mf = m[j]*rinv*rinv*rinv;
            for (k = 0; k < 3; k++)  a[i][k] += mf * dx[k];
        }
    }
```

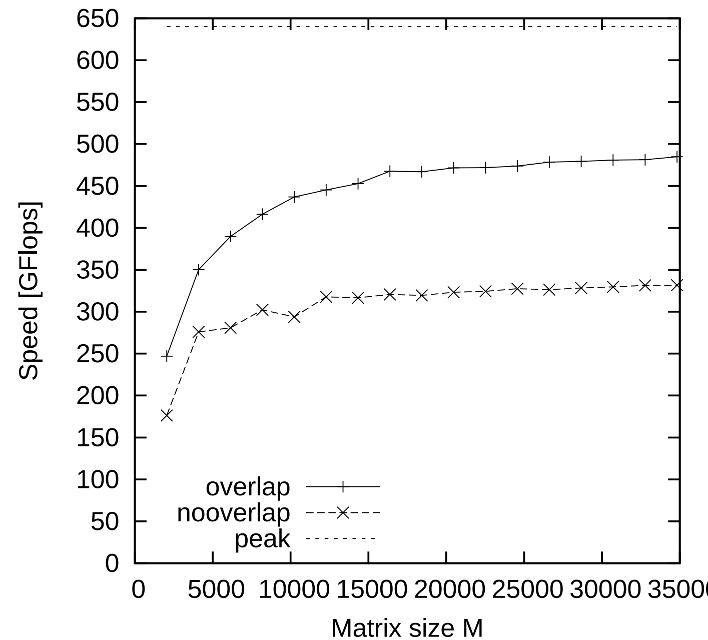
Generates code for single- and double-loops
(Translates to Nakasato's language)

Performance and Tuning example

- HPL (LU-decomposition)
- Gravity

Matrix-multiplication performance





M=N, K=2048, 640 Gflops **N=K=2048, 450 Gflops**

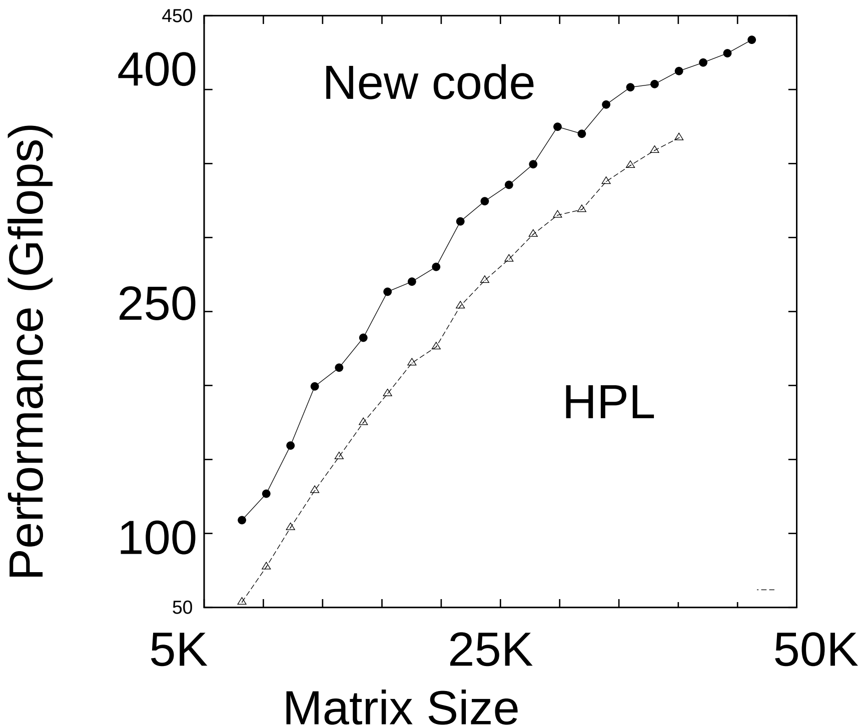
FASTEST single-card performance on the planet.

(Fermi: 3-400Gflops?)

LU-decomposition tuning

- **Almost every previously known techniques**
  - **Use large block (NB=2048)**
  - **right-looking form**
  - **TRSM converted to GEMM**
- **Several other "new" techniques (our new code)**
  - **Use row-major order to make row swapping fast**
  - **Transpose matrix during recursive column decomposition to make pivot search and narrow band matrix operation fast**
  - **Use recursive scheme for TRSM (calculation of L^{-1})**

LU-decomposition performance



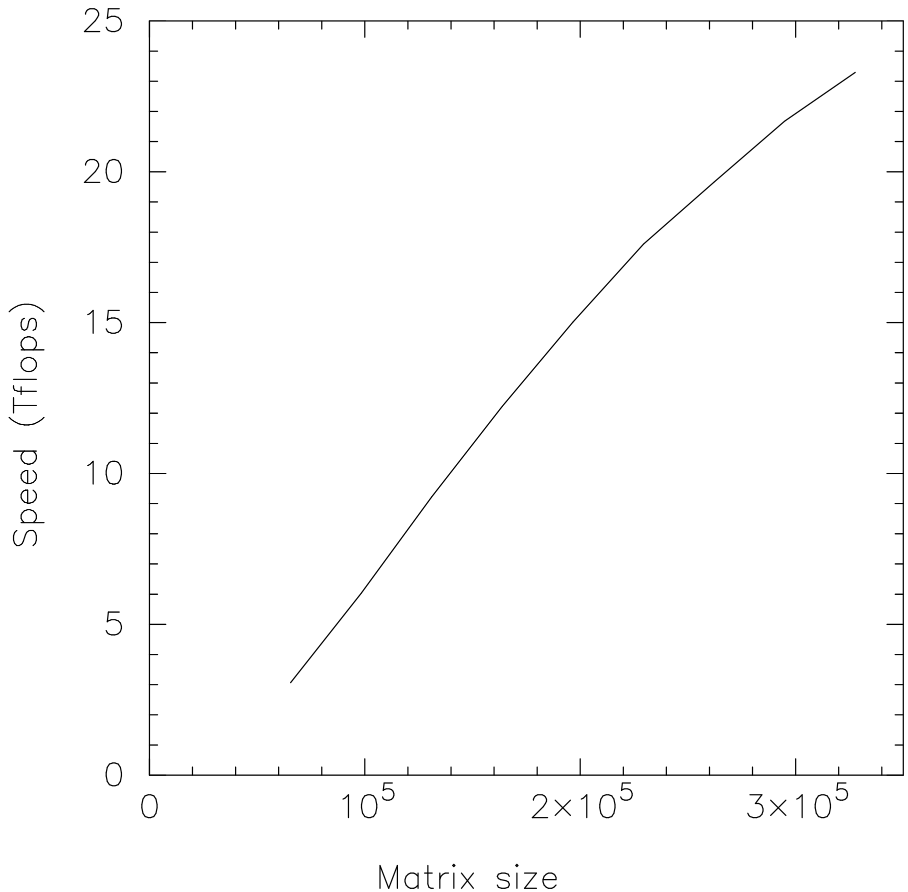
Speed in Gflops as function of Matrix size
Top: new code
Bottom: HPL 1.04a
430 Gflops (54% of theoretical peak) for N=50K

HPL (parallel LU)

- Everything done for single-node LU-decomposition
- Both column- and row-wise communication hidden
- TRSM further modified: calculate UT^{-1} instead of $T^{-1}U$
- More or less working, still lots of room for tuning

N=240K, 64 nodes: 24Tflops/29KW
x2 performance compared to HPL 1.04a
815Mflops/W: #1 in Little Green500 list

Dependence on Matrix size



Measurement on 64-node, 48-Tflops system
Matrix size limited by main memory size of host (18GB)
Efficiency limited by matrix size...

Dark side of tuning...

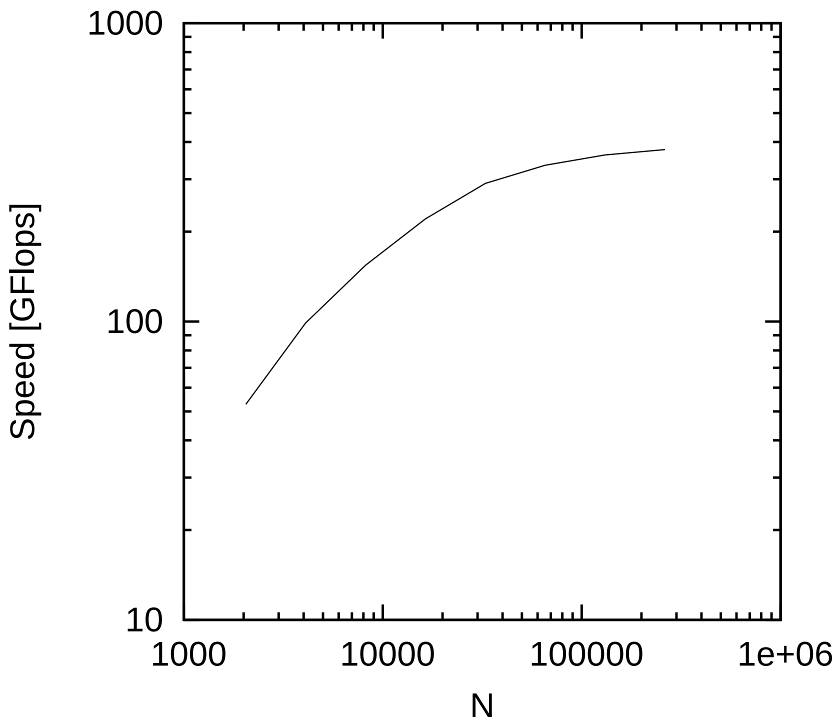
- X58 DMA performance seems to be limited to 6.4GB/s (sum of upstream and downstream, theoretical limit is 19.2GB/s)
- It starts to drop data silently when busy.
  - PIO write
  - DMA write

Workaround we used:

- Do not use PIOW
- Do not use DMA read and write concurrently

Gravity kernel performance

(Performance of individual timestep code not much different)



Assembly code (which I wrote) is not very optimized yet... Should reach at least 600 Gflops after rewrite.

Similarity and Difference with GPUs

	GRAPE-DR	GPU (Fermi)
SIMD	Yes	Yes
Accelerator	Yes	Yes
Design rule	90nm	40nm
# FPUs	512	448
Memory bandwidth	$\sim$ 5GB/s	$>$ 100GB/s
# transistors	400M	3G
Peak DP performance	200GF	515 Gflops
Power consumption	50W	250W
Performance per watt	4.0GF/W	2.1GF/W

Similarity and Difference with GPUs

- Both GRAPE-DR and GPUs achieved very high performance (and performance per watt) by going to SIMD many-core architecture
- The design of GRAPE-DR is much more extreme, with 1/10 transistors per FPU.
- Part of the reason of this difference is the limited memory bandwidth.
- Reduction in transistor count resulted in high performance/W.

GPGPU performance for N-body simulation

- x10 compared to a good SSE code for an N^2 code with shared timestep.
- $\sim$ x5 for production-level algorithms.
- $\sim$ x3 or less for the same price (if you buy GTX295, not Tesla).
- $<$ x2 if you are not using Keigo Nitadori's code.
- If you buy Tesla, no chance.

Keigo Nitadori(discussing the use of GPU)

Note: N-body simulations on GRAPE and GPGPUs

- Clusters of GPU-equipped PCs have achieved impressive performance on cosmological N-body simulations (Hamada et al 2009).
- This high performance is achieved by using the algorithm developed for parallel GRAPE systems (JM 2004, Ishiyama et al 2009).
- For accelerator architectures, the key to the high performance is new algorithms for things not done on the accelerator side.

“Porting” applications to accelerators

- You cannot get high performance by just “moving most expensive part to accelerators”
- Parameter tunings (such as the change of the block size etc) are not enough
- You need to develop new algorithms to reduce the calculation cost of the part not done on the accelerators
- In many cases, you can do this, because not much effort has been paid before

Quotes

From: Twelve Ways to Fool the Masses When Giving Performance Results on ~~Accelerators~~ Parallel Computers (D. H. Bailey, 1991)

1. Quote only 32-bit performance results, not 64-bit results.
2. Present performance figures for an inner kernel, and then represent these figures as the performance of the entire application.
6. Compare your results against ~~x87~~ scalar, unoptimized code on ~~Xeons~~ Crays.
7. When direct run time comparisons are required, compare with an old code on an obsolete system.
8. If MFLOPS rates must be quoted, base the operation count on the parallel implementation, not on the best sequential implementation.
12. If all else fails, show pretty pictures and animated videos, and don't talk about performance.

History repeats itself — Karl Marx

Summary

- GRAPEs, special-purpose computer for gravitational N-body system, have been providing 10x - 100x more computational power compared to general-purpose supercompuers.
- GRAPE-DR, with programmable processors, has wider application range than traditional GRAPEs.
- Peak speed of a GRAPE-DR card with 4 chips is 800 Gflops (DP).
- DGEMM performance 640 Gflops, LU decomposition $>$ 400Gflops
- Achieved the best performance per W (Top 1 in the Little Green 500 list, 815Mflops/W)
- Accelerators require new algorithms, not just porting and tuning

Next-Generation GRAPE

Question:
Any reason to continue hardware development?

- **GPUs are fast, and getting faster**
- **FPGAs are also growing in size and speed**
- **Custom ASICs practically impossible to make**

Next-Generation GRAPE

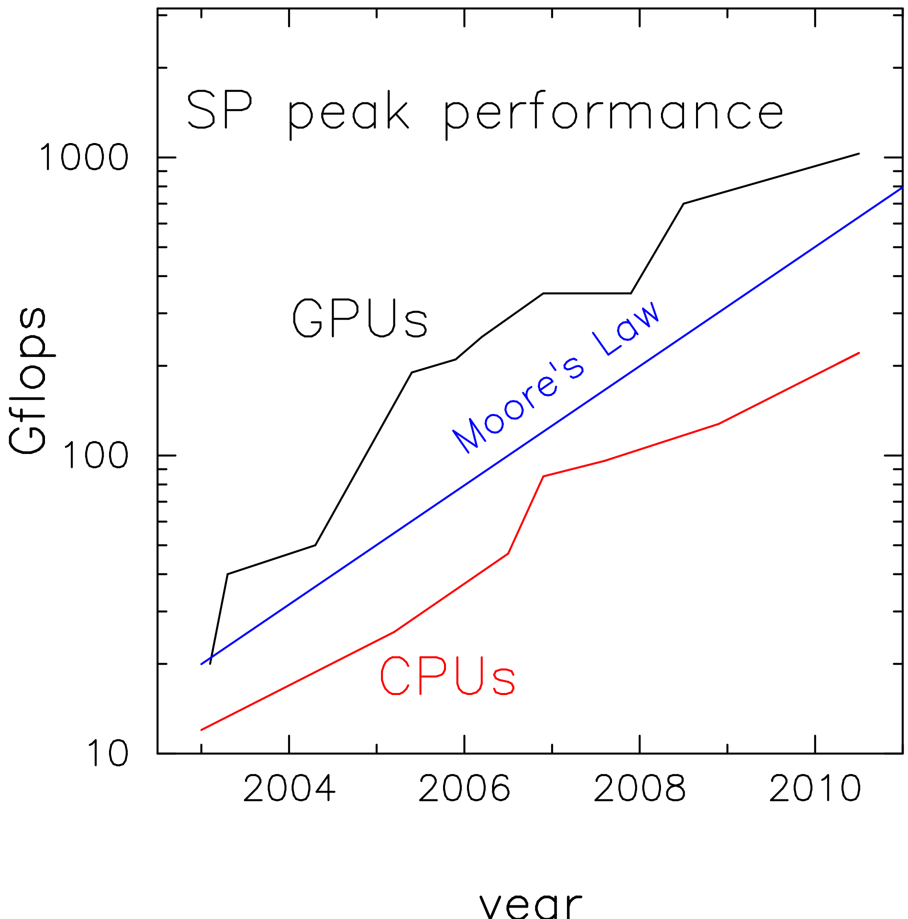
Question:
Any reason to continue hardware development?

- **GPUs are fast, and getting faster**
- **FPGAs are also growing in size and speed**
- **Custom ASICs practically impossible to make**

Answer?

- **GPU speed improvement might have slowed down**
- **FPGAs are becoming far too expensive**
- **Power consumption might become most critical**
- **Somewhat cheaper way to make custom chips**

GPU speed improvement slowing down?



Clear "slowing down" after 2006 (after G80)

Reason: shift to more general-purpose architecture

Discrete GPU market is eaten up by unified chipsets and unified CPU+GPU

But: HPC market is not large enough to support complex chip development

FPGA

“Field Programmable Gate Array”

- **“Programmable” hardware**
- **“Future of computing” for the last two decades....**
- **Telecommunication market needs: large and fast chips (very expensive)**

Power Consumption

1kW $\cdot$ 1 year $\sim$ 1000 USD

You (or your institute) might be paying more money for electricity than for hardware.

Special-purpose hardware is quite energy efficient.

Chip	Design rule	Gflops/W
GRAPE-7(FPGA)	65nm	> 20
GRAPE-DR	90nm	4
GRAPE-6	250nm	1.5
Tesla C2050	40nm	< 2
Opteron 6128	45nm	< 1.2

Structured ASIC

- **Something between FPGA and ASIC**
- **eASIC: 90nm (Fujitsu) and 45nm (Chartered) products.**
- **Compared to FPGA:**
  - **3x size**
  - **1/10 chip unit price**
  - **non-zero initial cost**
- **Compared to ASIC:**
  - **1/10 size and 1/2 clock speed**
  - **1/3 chip unit price**
  - **1/100 initial cost ($>$ 10M USD vs $\sim$ 100K)**

GRAPEs with eASIC

- Completed an experimental design of a programmable processor for quadruple-precision arithmetic. 6PEs in nominal 2.5Mgates.
- Started designing low-accuracy GRAPE hardware with 7.4Mgates chip.

Summary of planned specs:

- around 8-bit relative precision
- 100-200 pipelines, 300-400 MHz, 2-5Tflops/chip
- small power consumption: single PCIe card can house 4 chips (10 Tflops, 50W in total)

Will this be competitive?

Rule of thumb for a special-purpose computer project:

> **Price-performance goal should be more than 100 times better than that of a PC available when you start the project.**

— x 10 for 5 year development time
— x 10 for 5 year lifetime

Compared to CPU: Okay
Compared to GPU: ??? (Okay for electricity)

Will this be competitive?

Rule of thumb for a special-purpose computer project:

> **Price-performance goal should be more than 100 times better than that of a PC available when you start the project.**

— x 10 for 5 year development time
— x 10 for 5 year lifetime

Compared to CPU: Okay
Compared to GPU: ??? (Okay for electricity)

Will GPUs exist 10 years from now?