Current status of GRAPE Project

Jun Makino

University of Tokyo

Talk overview

- **GRAPE Project**
- **Science with GRAPEs**
- **GRAPE-DR: Next-Generation GRAPE**

GRAPE project

- basic idea
- hardware
- performance — Direct, Tree, P^3M
- GRAPEs in the world

GRAPE project: Rationale

GOAL:
Design and build specialized hardware for simulation of stellar systems.

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

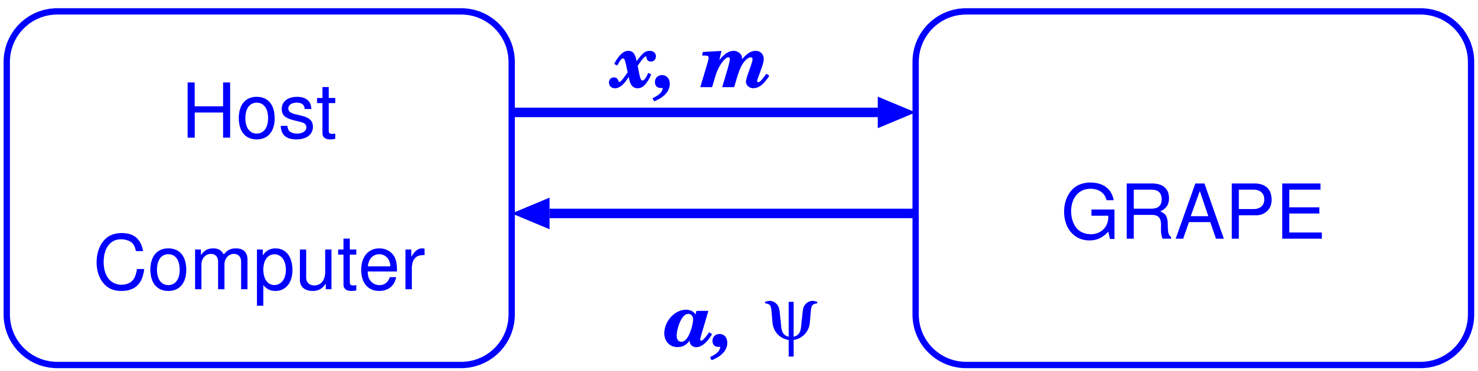
GRAPE-6	(2002, 64 TF)	4M$
ASCI White	(2001, 12 TF)	200M$
ASCI Q	(2002, 30 TF)	200M$
Earth Simulator	(2002, 40 TF)	300M$
BG/L	(2005?, 360 TF?)	??M$

Basic idea of GRAPE

Special-purpose hardware for force calculation

General-purpose host for all other calculation

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



Time integration, IO, etc

Force calculation

Flexibility

High performance

Special-purpose hardware

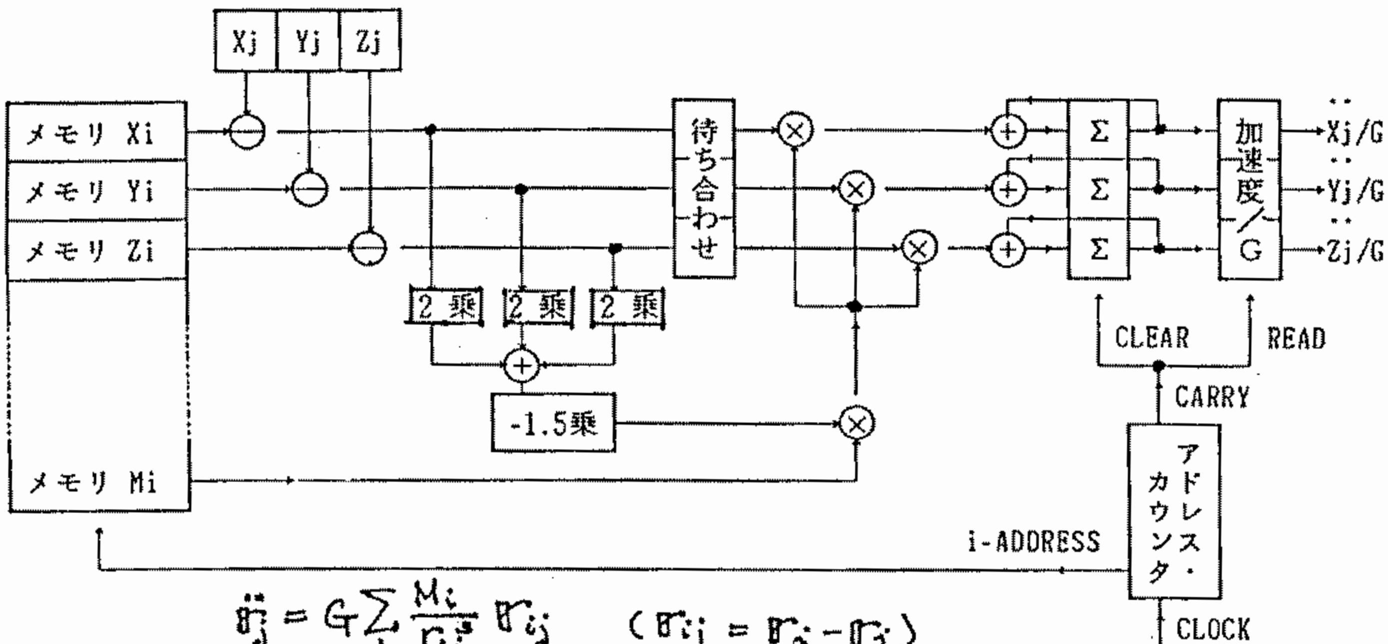
- Pipeline processor specialized for the interaction calculation
  - Can use large number of processors
  - All processors work in parallel

  $\longrightarrow$ High performance

General-purpose host computer

- “High-level” language (Fortran, C, C++...)
- Existing codes with “minor” modifications
- Individual timestep, Tree algorithm

GRAPE Pipeline processor



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

+, −, ×, 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.

Chikada 1988

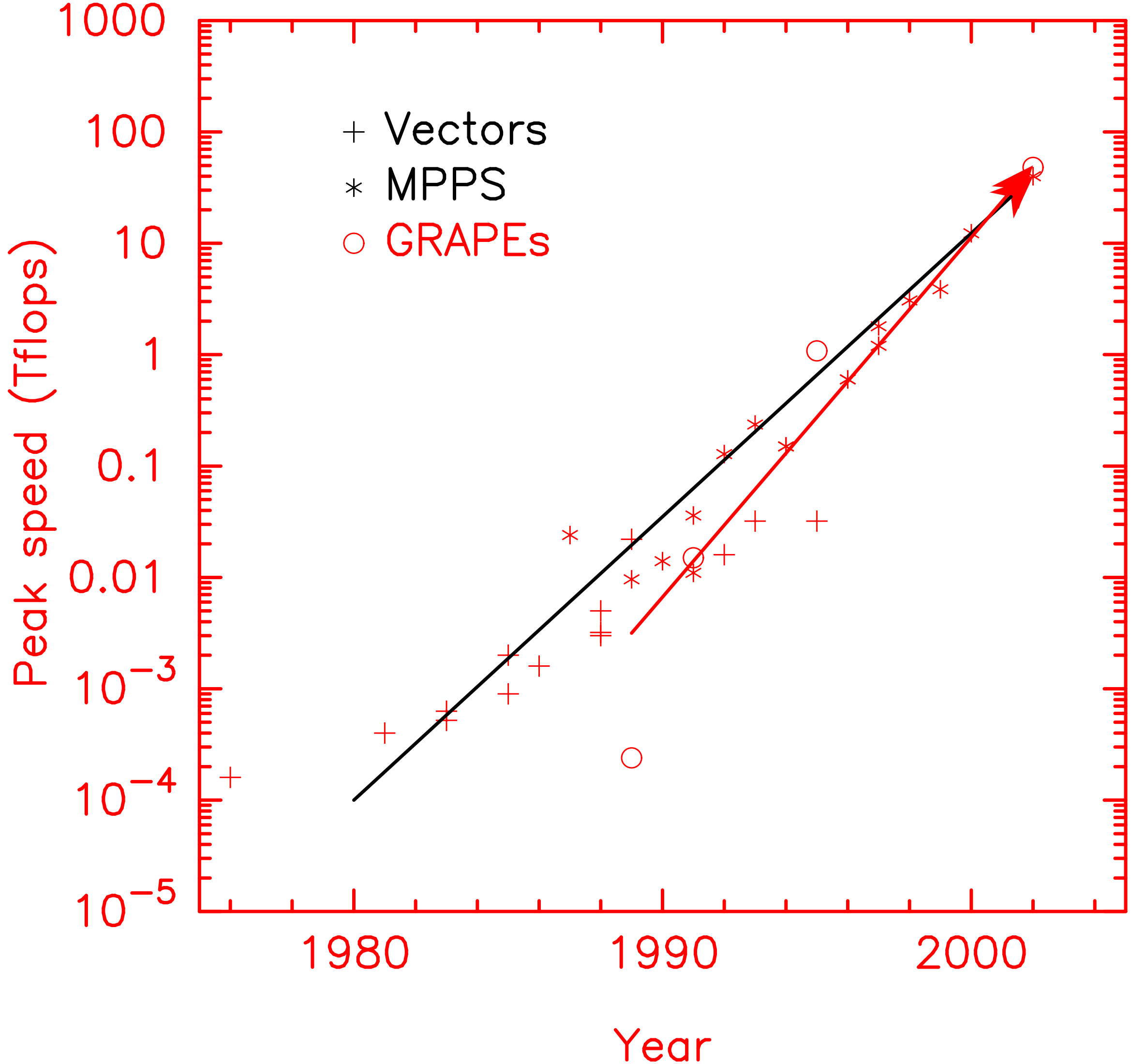
GRAPE machines

1989	GRAPE-1	240 MF	Low accuracy(LA)
1990	GRAPE-2	40 MF	High accuracy(HA)
1991	GRAPE-3	15 GF	LA, custom chip
1995	GRAPE-4	1.08 TF	HA, custom chip
1998	GRAPE-5	40*n GF	LA, 2 pipelines in a chip
2001	GRAPE-6	64 TF	HA, 6 pipelines in a chip

Molecular Dynamics

1992	GRAPE-2A	120MF	
1996	MD-GRAPE	2.4GF	custom chip
2001	MDM	75 TF	RIKEN
2006?	PE	0.6PF	RIKEN

Evolution of peak performance



Why GRAPEs can do better than microprocessors?

Intel, AMD and IBM are spending 100s or 1000s of M$ to develop processors.

How a small group of astronomers can possibly outperform them?

Why GRAPEs can do better than microprocessors?

Intel, AMD and IBM are spending 100s or 1000s of M$ to develop processors.

How a small group of astronomers can possibly outperform them?

Answer:

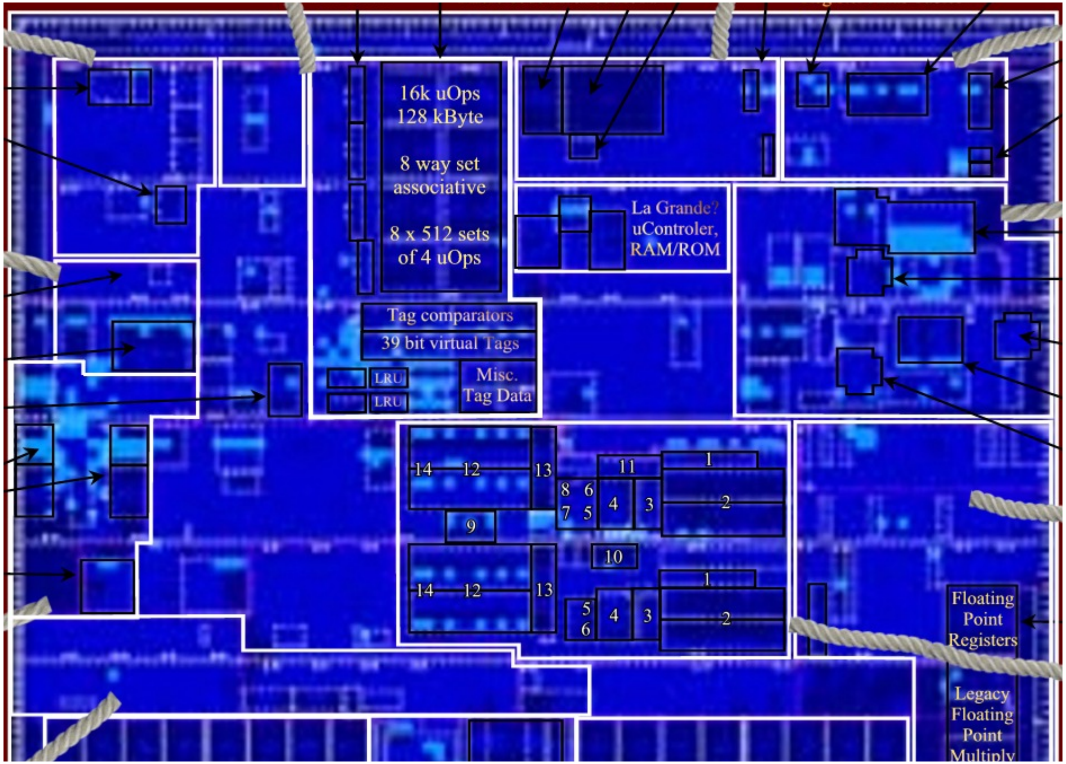
Intel is not designing their chip for N-body problem.
In fact, not for scientific computing in general...

Architecture of modern processors

Cache
Cache prefetch
Branch prediction
Speculative execution
Out-of-order execution

…… and all other stuff you don't want to get into.

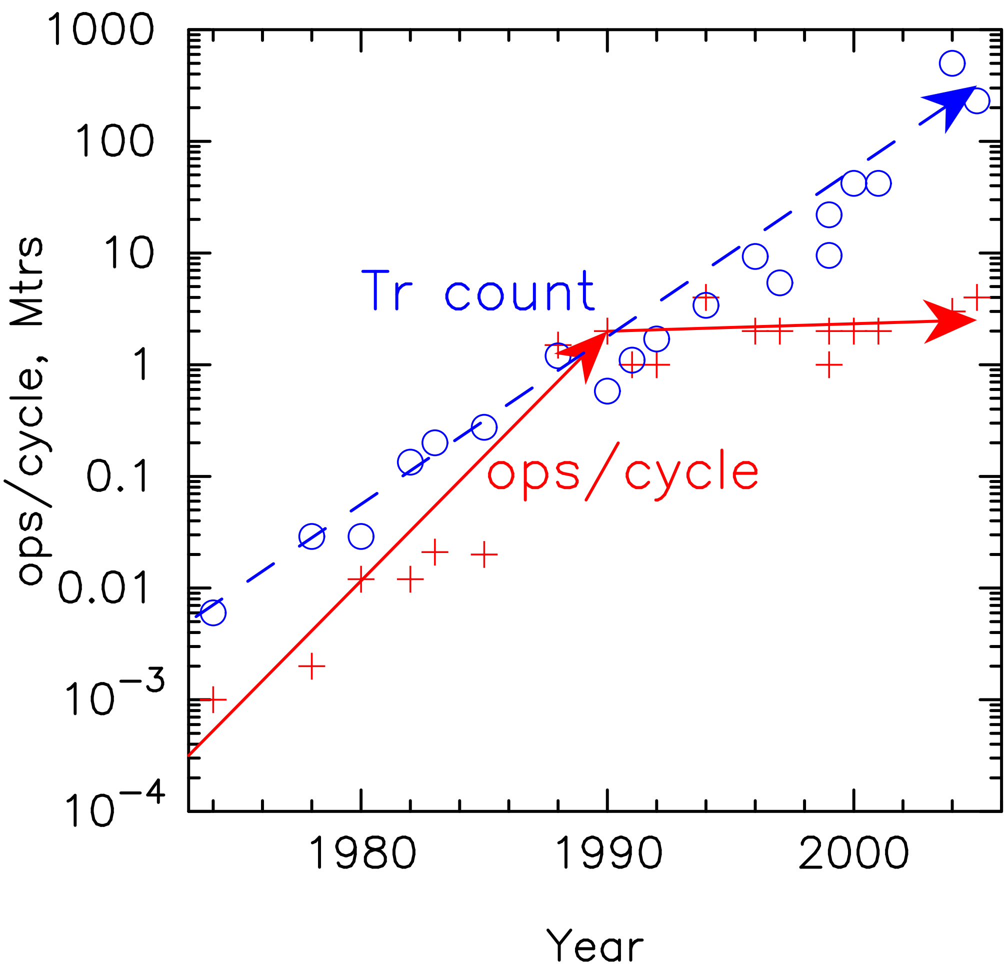
Intel Pentium 4 chip



A very small fraction of the chip is used for floating-point unit.

Total transistors $\sim 10^8$
Floating-point unit $\sim 10^5$
More than 99.9% of silicon is used for things other than real arithmetic operations.

Evolution of microprocessors



Number of transistors doubles every 18 months (*"Moore's Law"*)

Number of floating point units got stuck at $O(1)$. Never reached more than 4.

Why got stuck at 4?

Two “reasons”:

- **“superscalar” approach with more than 4 execution units gives very small increase in performance**
- **bandwidth to main memory is limited**

Superscalar?

- You write sequential program (single stream of instructions)
- the processor tries to figure out which instructions can be executed in parallel
- CDC 6600 is one of the first machines

as opposed to:
VLIW, in which the compiler tries to find parallelism (Multiflow, Intel Itanic)

Superscalar?

- Your program is sequential (single stream of instructions)
- the processor tries to figure out which instructions can be executed in parallel
- CDC 6600 is one of the first machines

as opposed to:
VLIW, in which the compiler tries to find parallelism (Multiflow, Intel Itanium)

Why not more than 4?

— partly because of the set of benchmark programs choosen.

Example:
SPECfp92 originally contained "matrix300"
At some point this was dropped, essentially because it was too easily parallelized.

Benchmark designers chose problems/programs which are difficult to parallelize, and conclude that problems are generally not parallelizable.

Memory bandwidth

This is a real problem.
$\sim 1,000$ processors can fit into a chip.
But how you can get data in and out?

1,000 1GHz processors
$\longrightarrow$ 24 TB/s of memory bandwidth.

Intel Pentium 4 : 6.4 GB/s. Less than the need of processor.
(This is why you need cache)

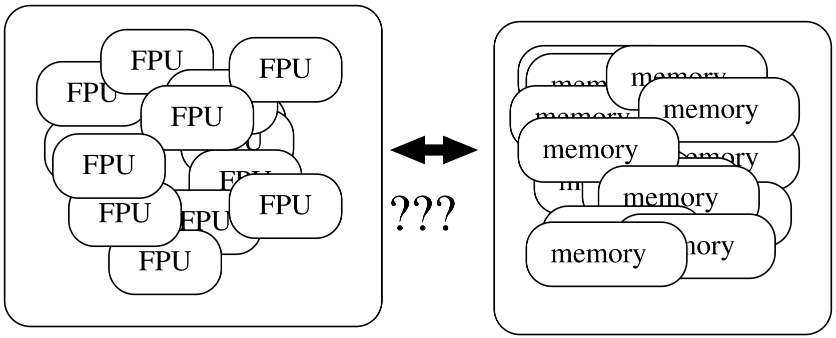
The GRAPE approach

- **parallelism: All of N^2 (or $N \log N$ for treecode) interactions can be evaluated in parallel: There is much more parallelism than you can possibly use.**
- **Memory bandwidth:**
  - **pipeline processor: needs 3 words for 30 operations. Reduction of a factor of 30.**
  - **(real/vitual) multiple pipelines calculate the forces from one particle to many particles: Reduction of a factor of 50 (in GRAPE-6)**

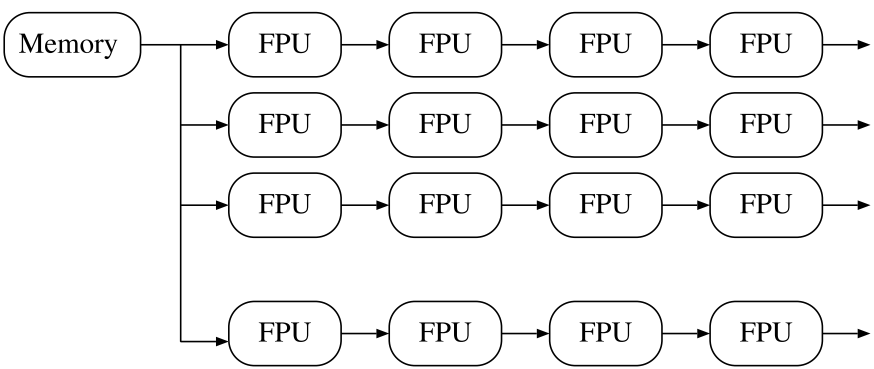
In total, reduction by a factor $>$ 1,000

The GRAPE approach

General-purpose



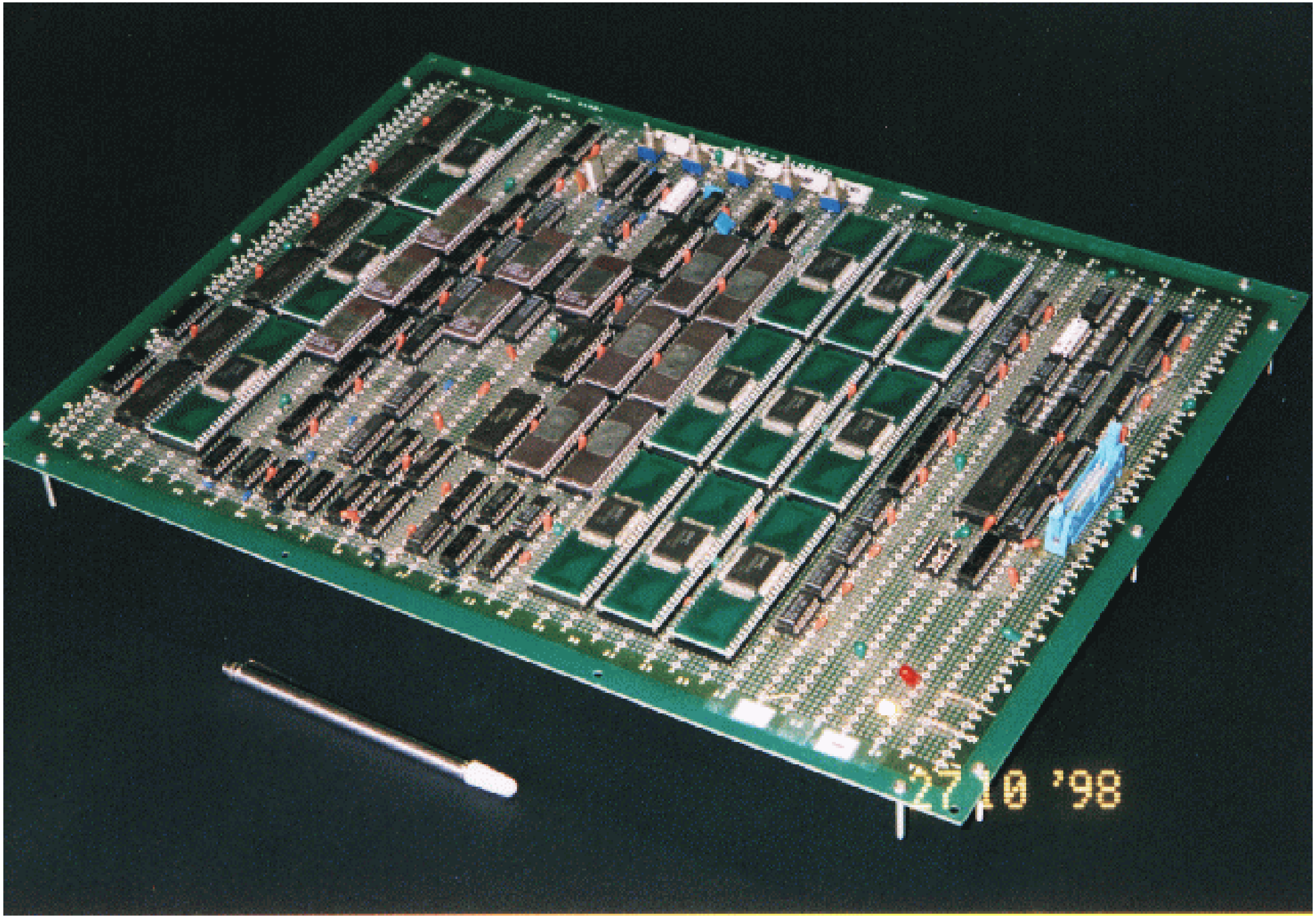
GRAPE



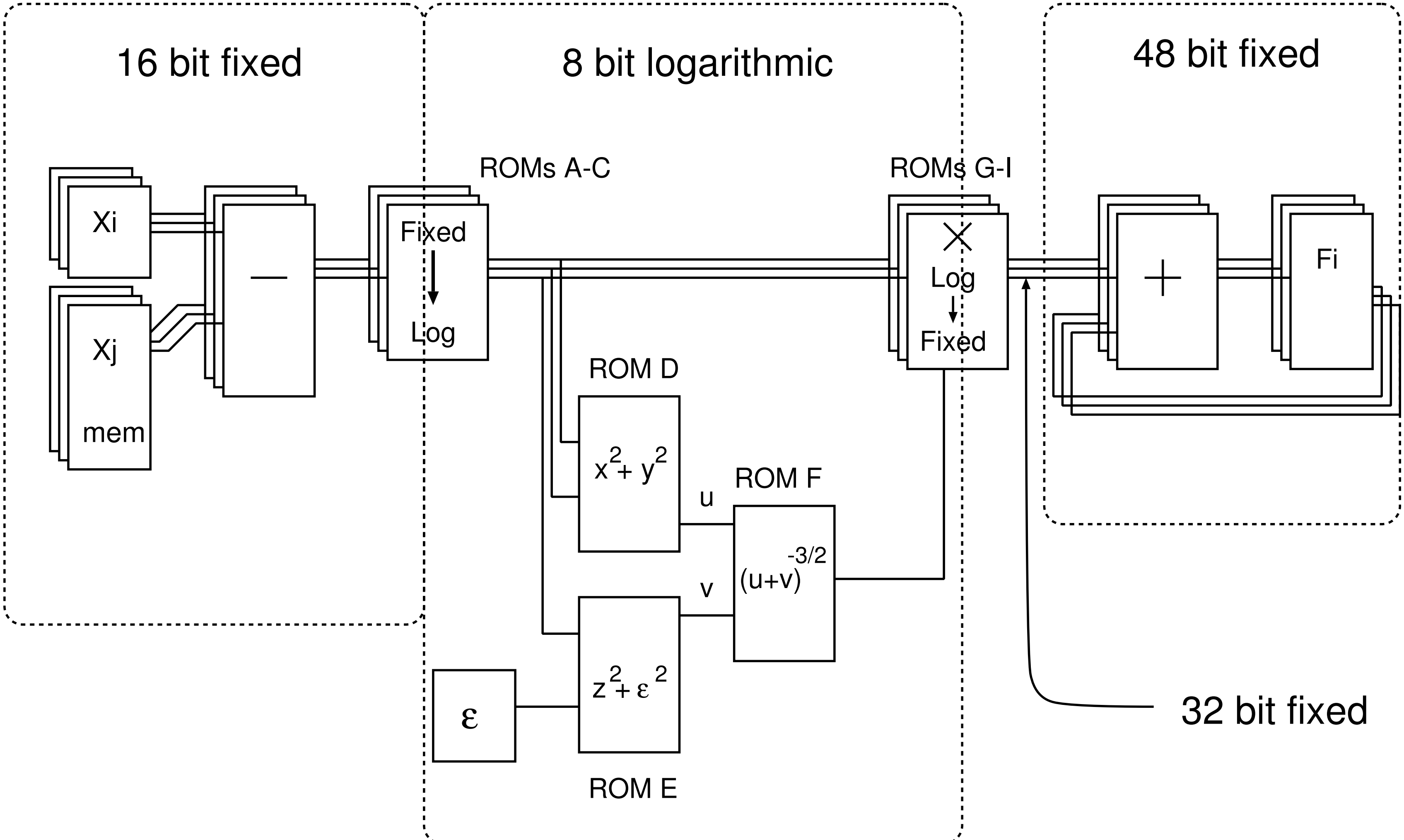
Some history

- GRAPE-1
- GRAPE-2
- GRAPE-3
- GRAPE-4
- GRAPE-6

GRAPE-1 — 1989

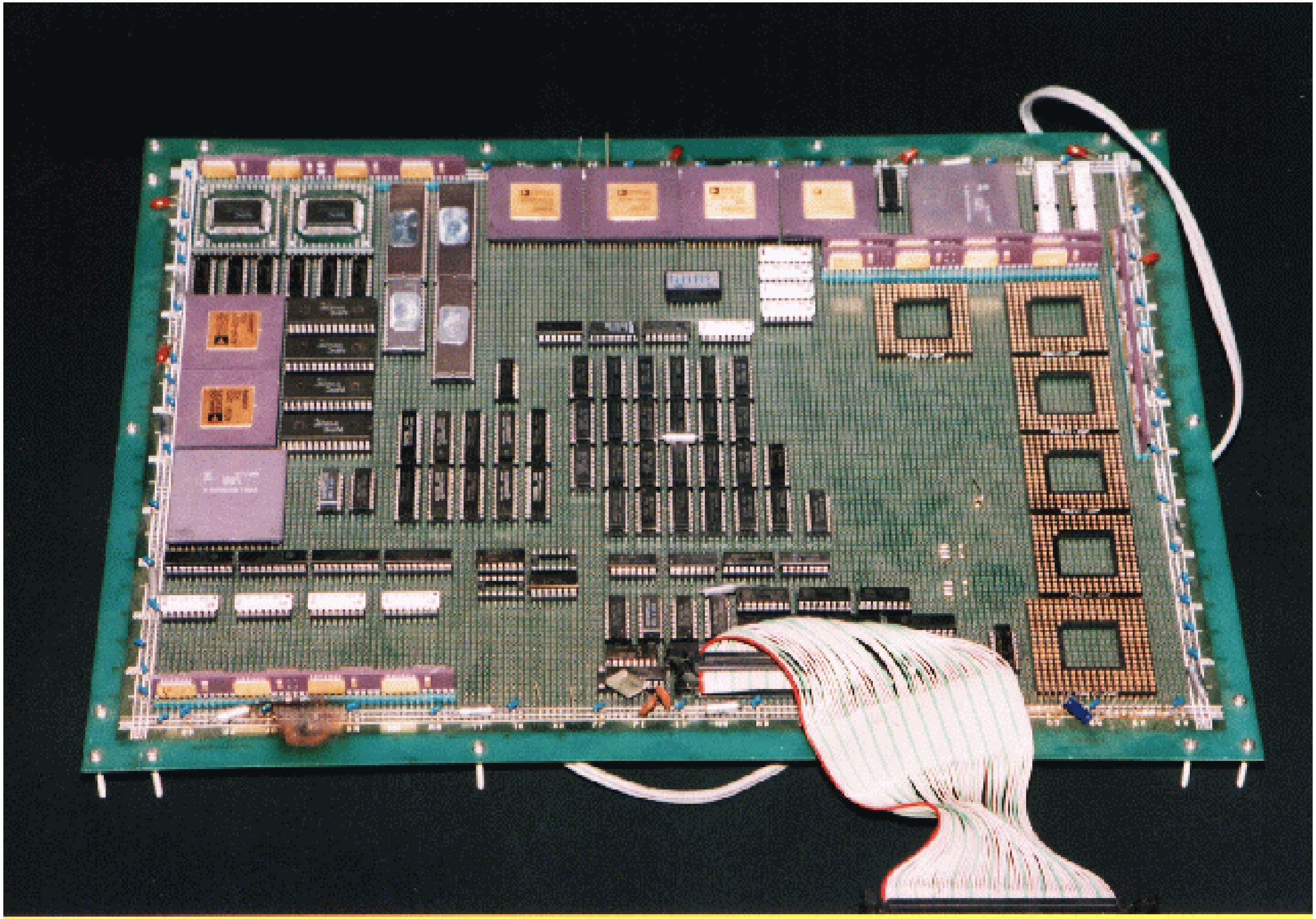


GRAPE-1 pipeline processor



240 Mflops peak speed

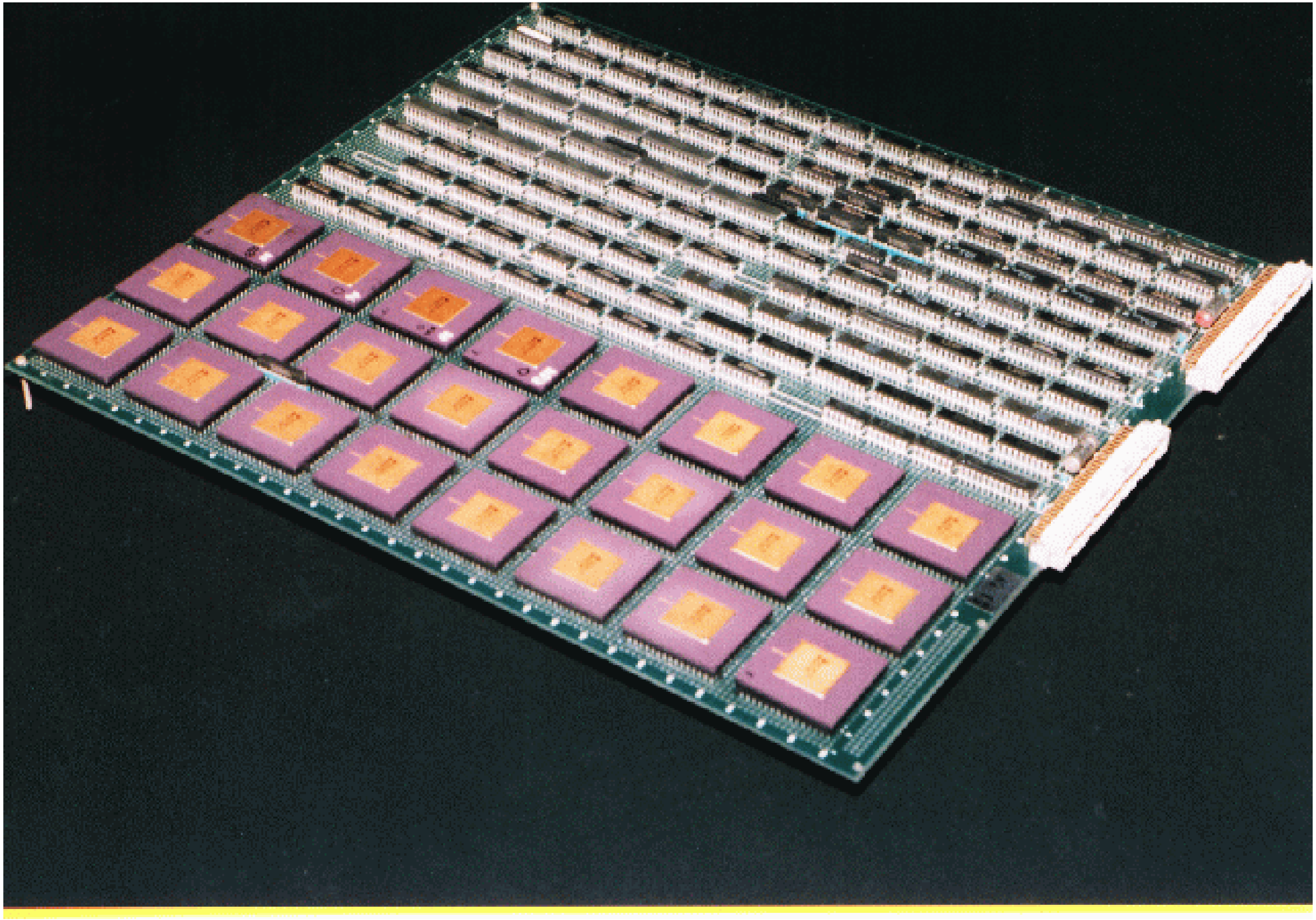
GRAPE-2 — 1990



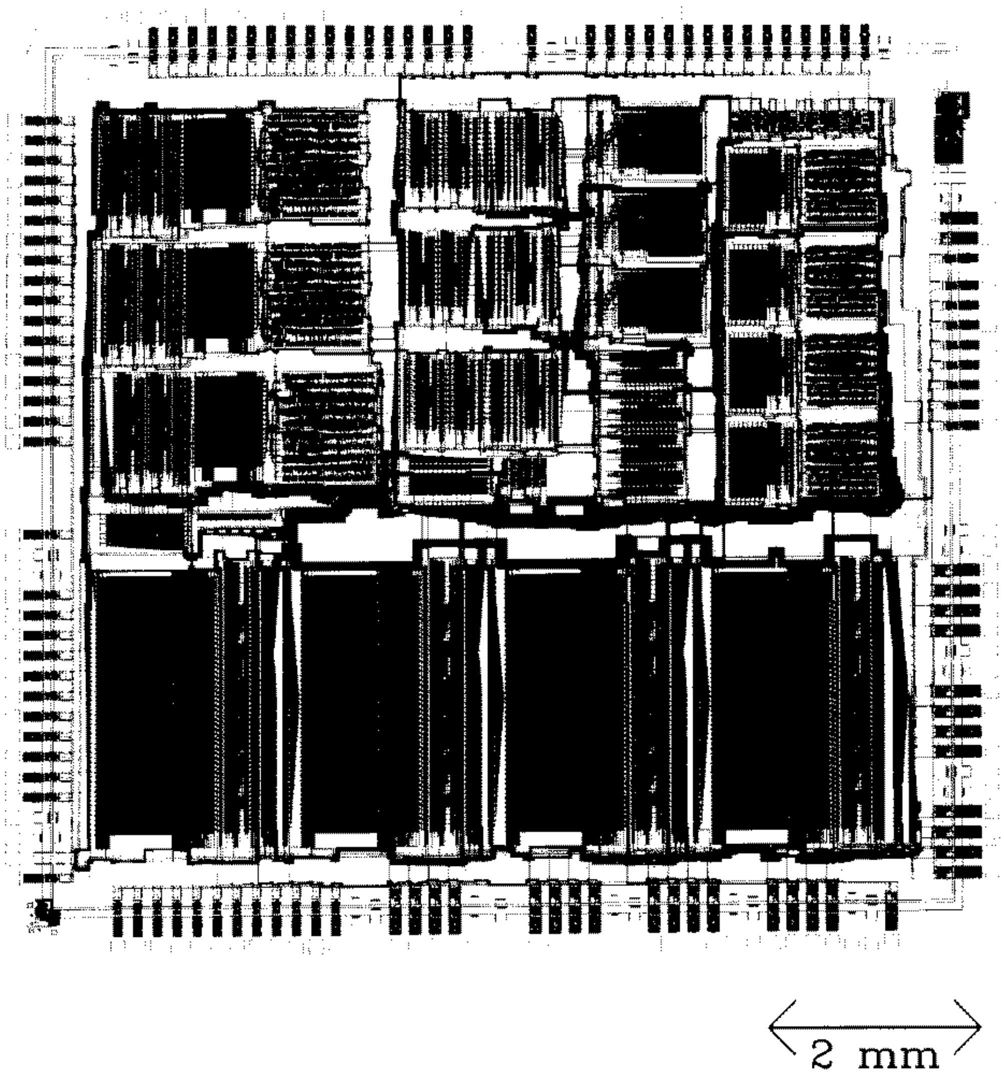
GRAPE-2 Summary

- **Real floating-point arithmetic**
- **VME-bus for host communication**
- **40 Mflops peak speed (sounds slow, but 15 years ago it was fast)**

GRAPE-3 — 1991

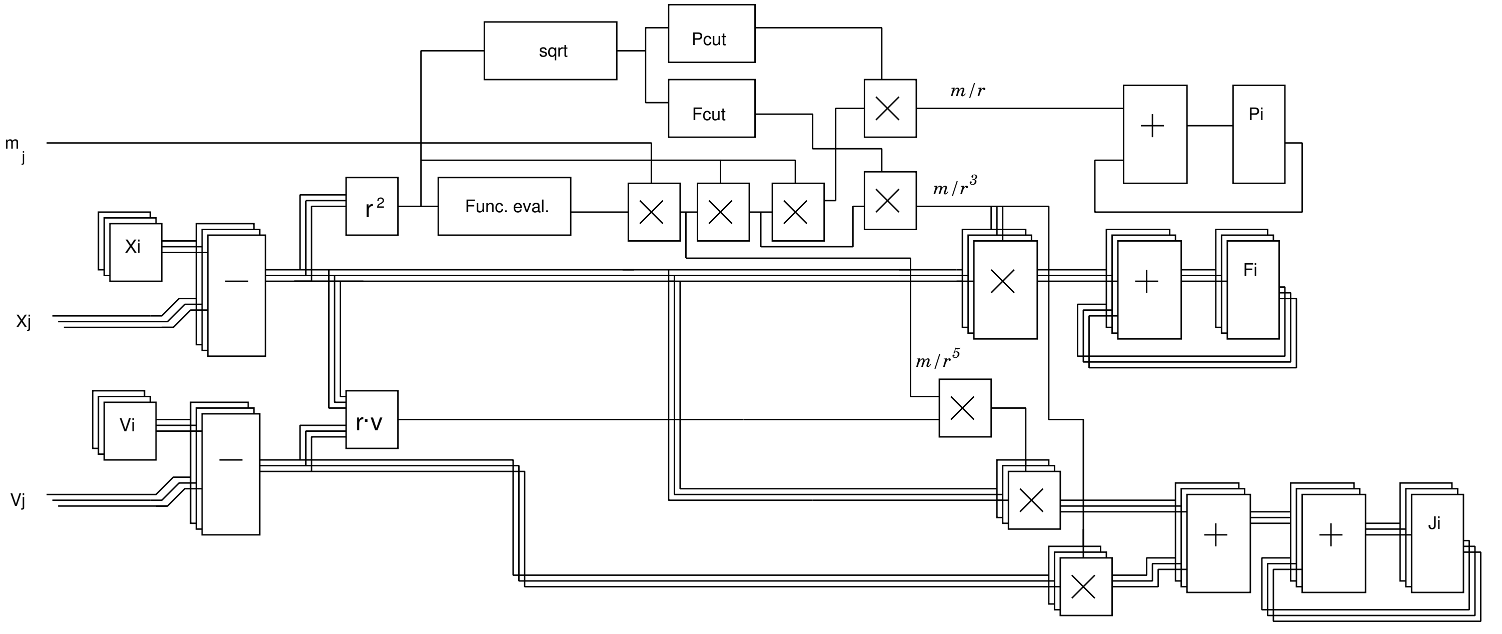


GRAPE-3 chip

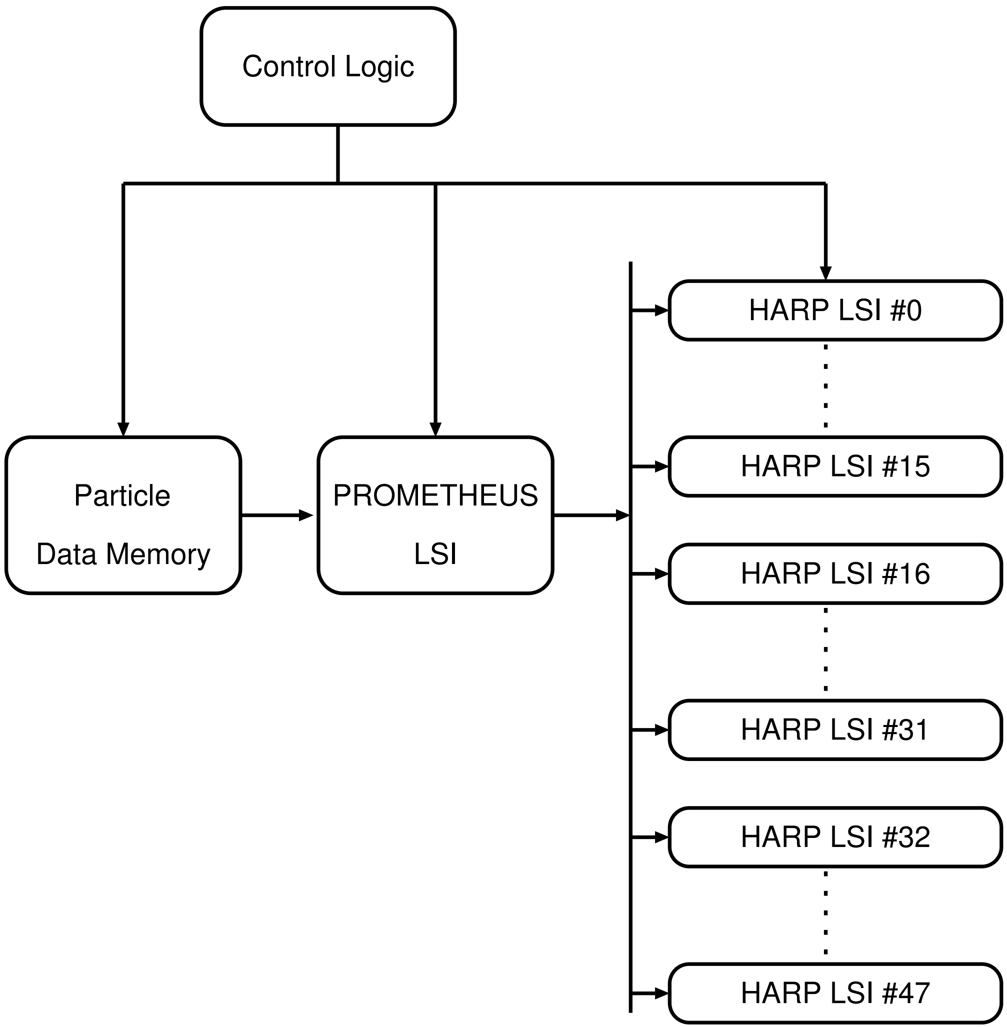


GRAPE-4 — 1995

GRAPE-4 pipeline



GRAPE-4 processor board



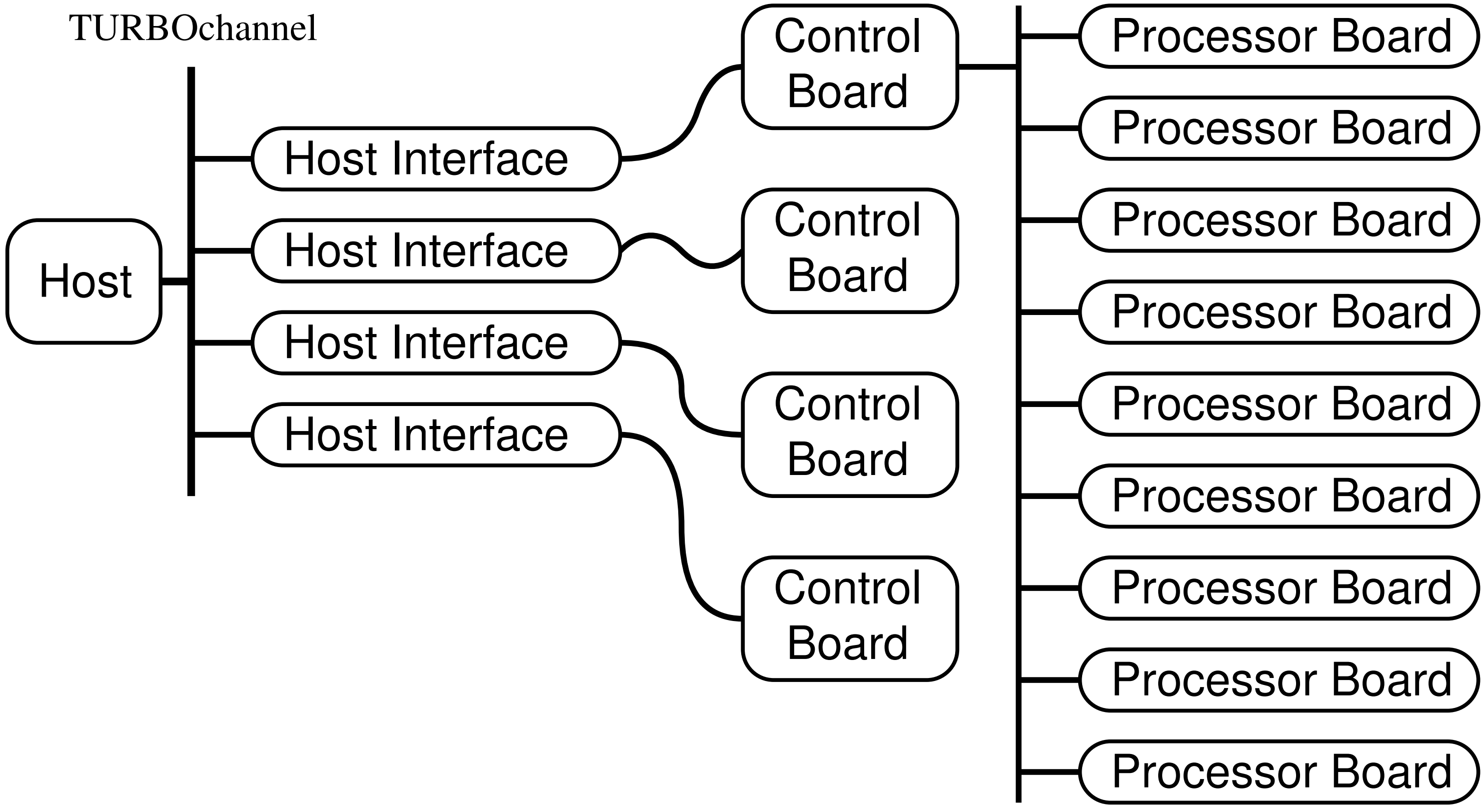
many pipeline LSI share one memory unit

↓

Less memory bandwidth

Support for individual (block) timestep in hardware

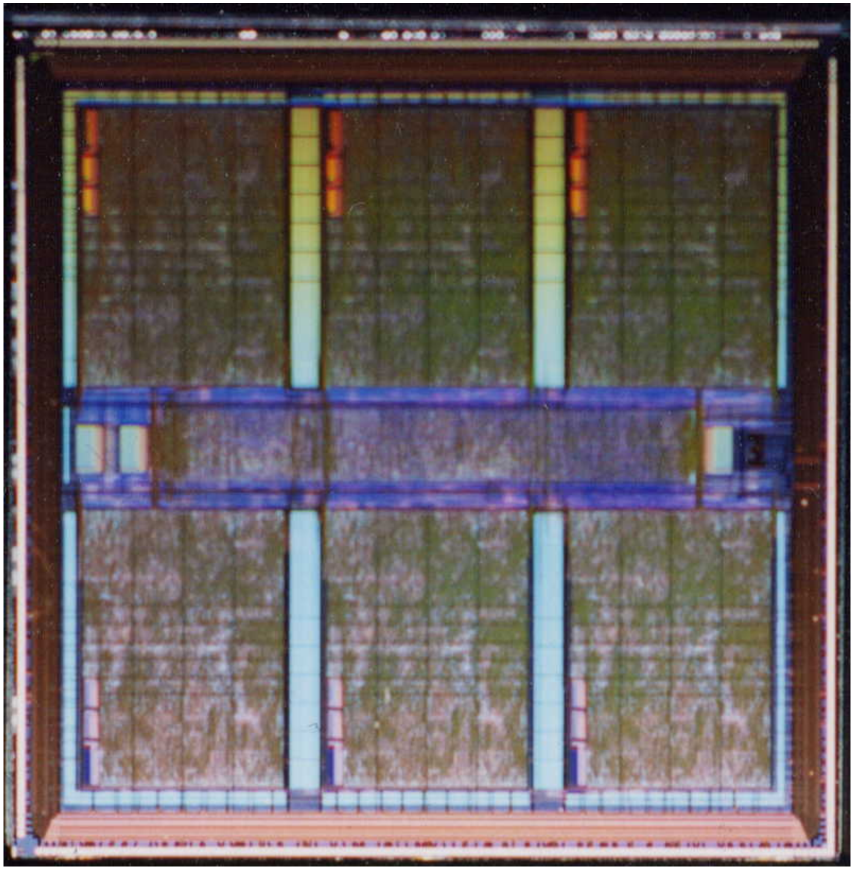
Structure of GRAPE-4



GRAPE-6 — 2001

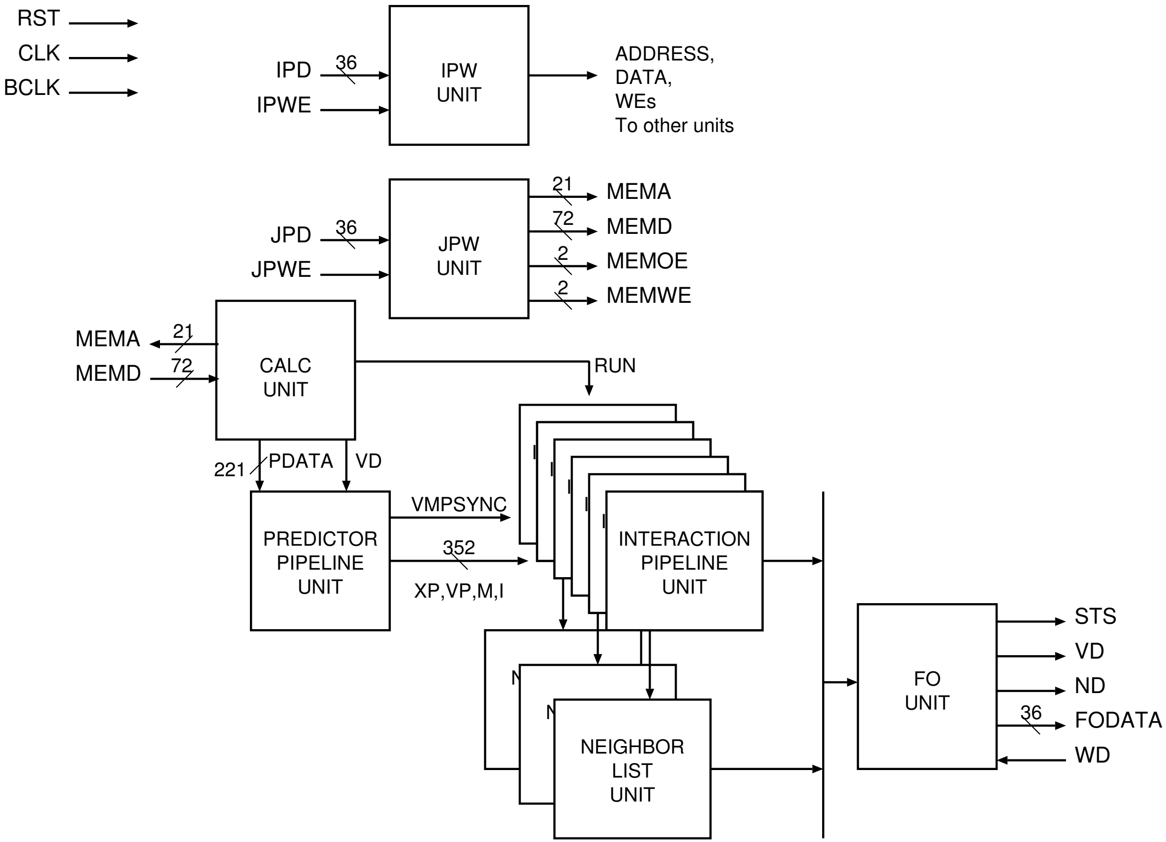
- **processor chip**
- **processor module**
- **processor board**
- **total system**

Pipeline LSI



- 0.25 μm design rule (Toshiba TC-240, 1.8M gates)
- 90 MHz clock
- 6 pipelines
- one predictor pipeline
- 31 Gflops /chip

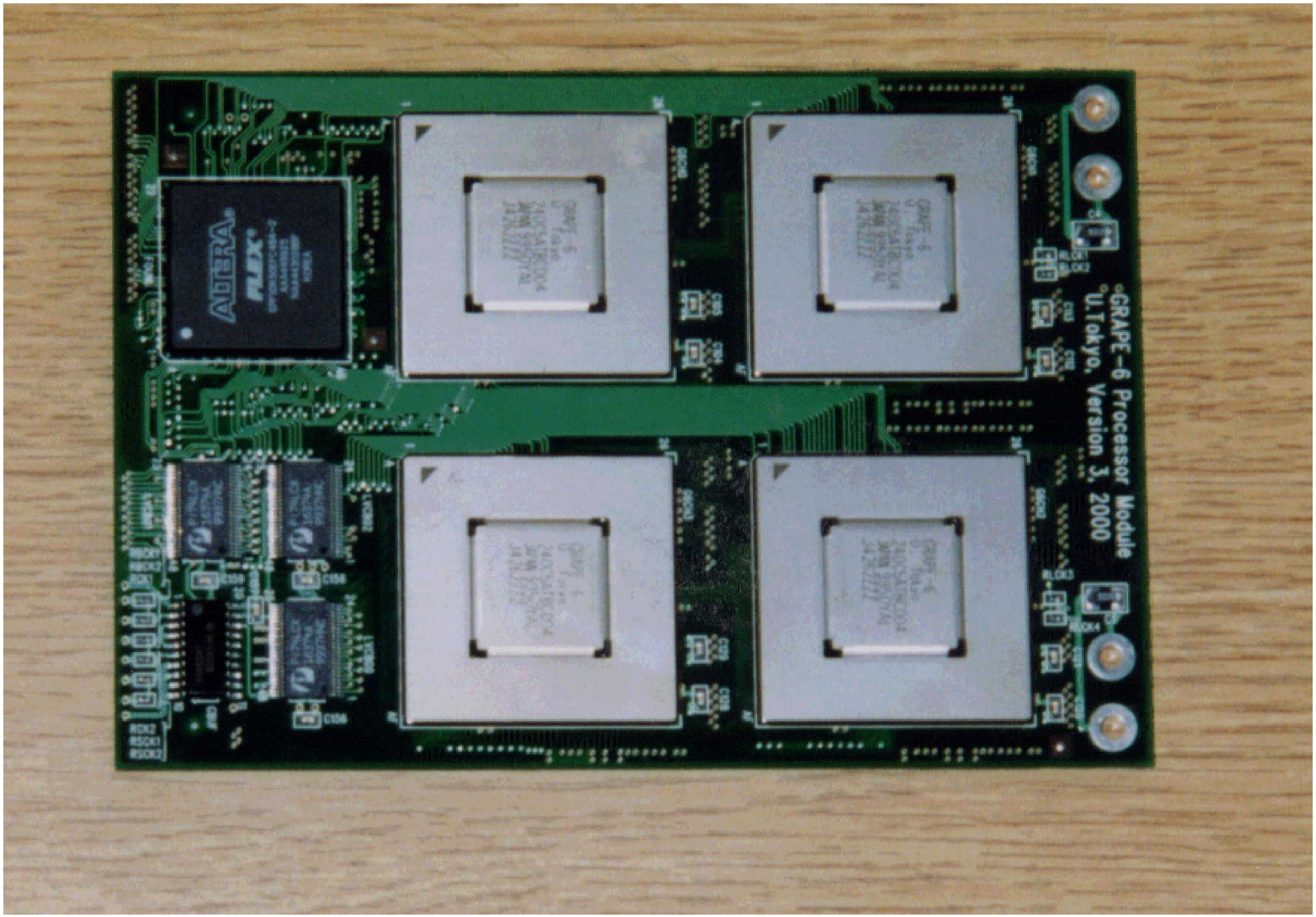
Pipeline LSI



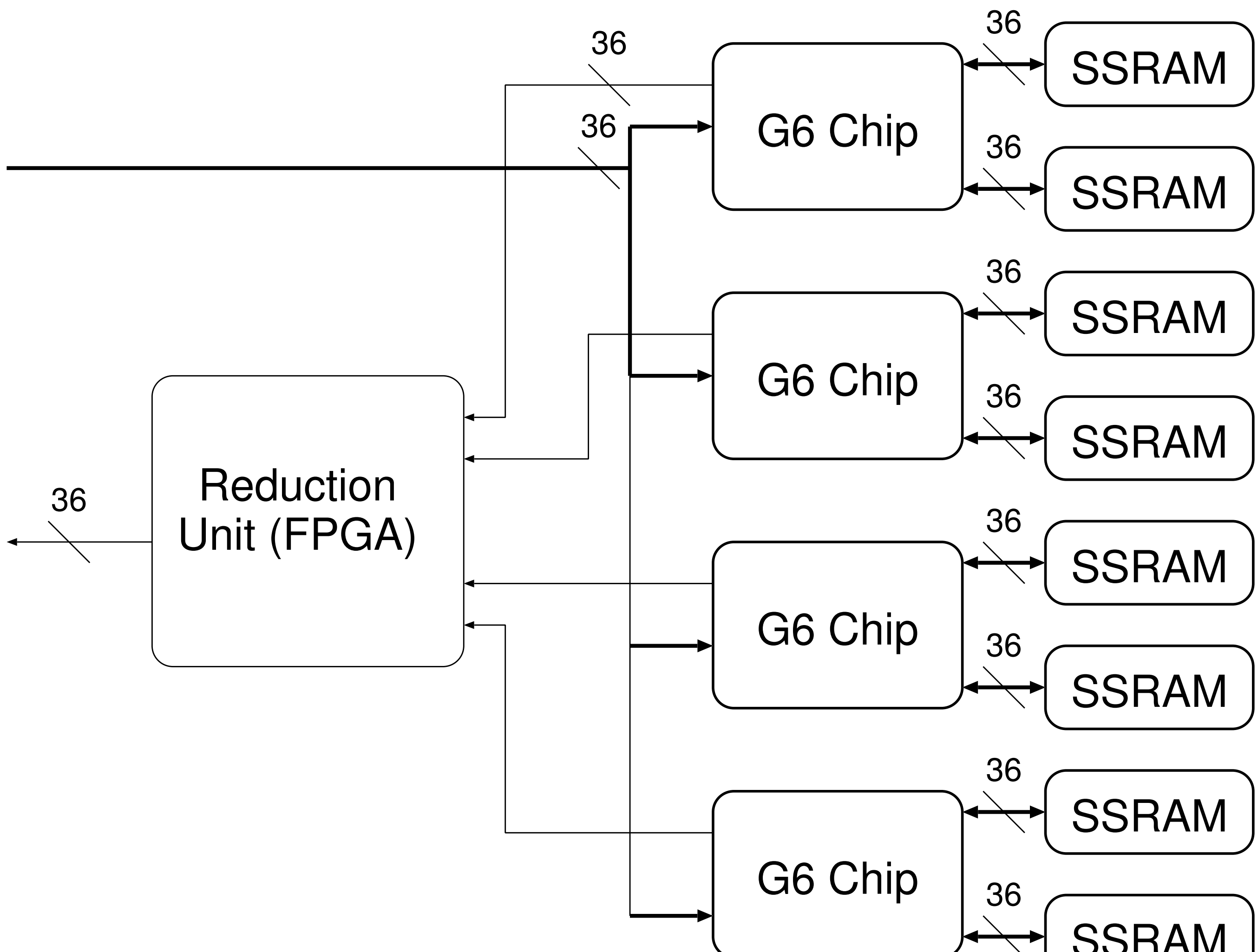
Essentially GRAPE-4 processor board on a chip

- Host Interface
- Memory Interface
- Force calculation pipeline
- Control logic

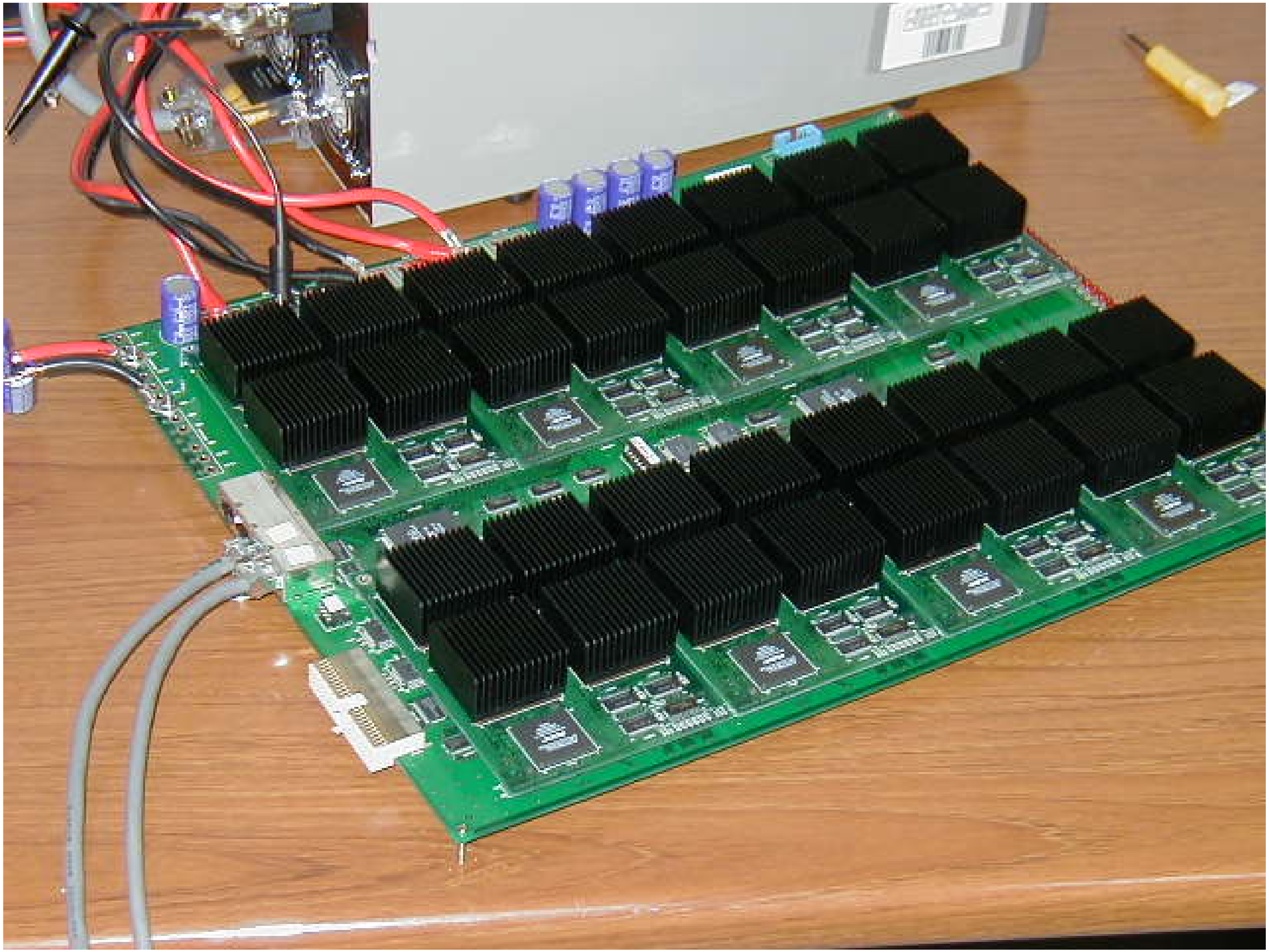
GRAPE-6 processor module



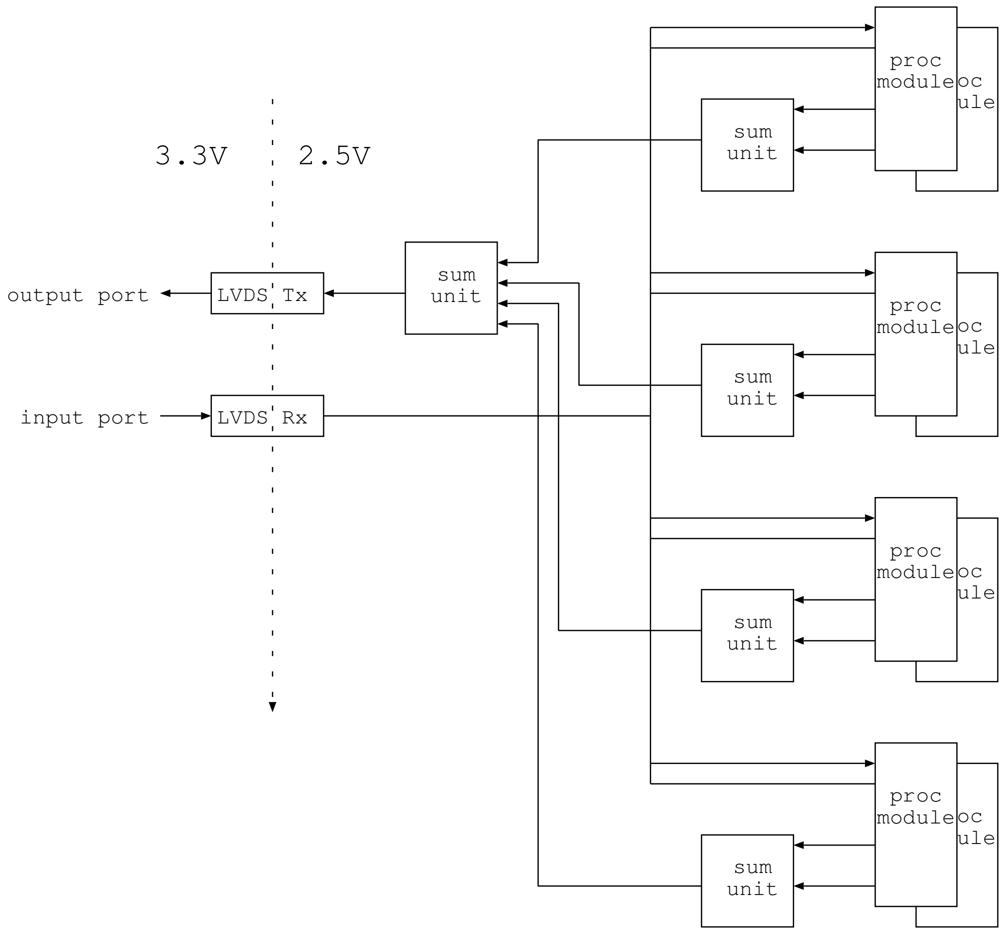
GRAPE-6 processor module



GRAPE-6 processor board

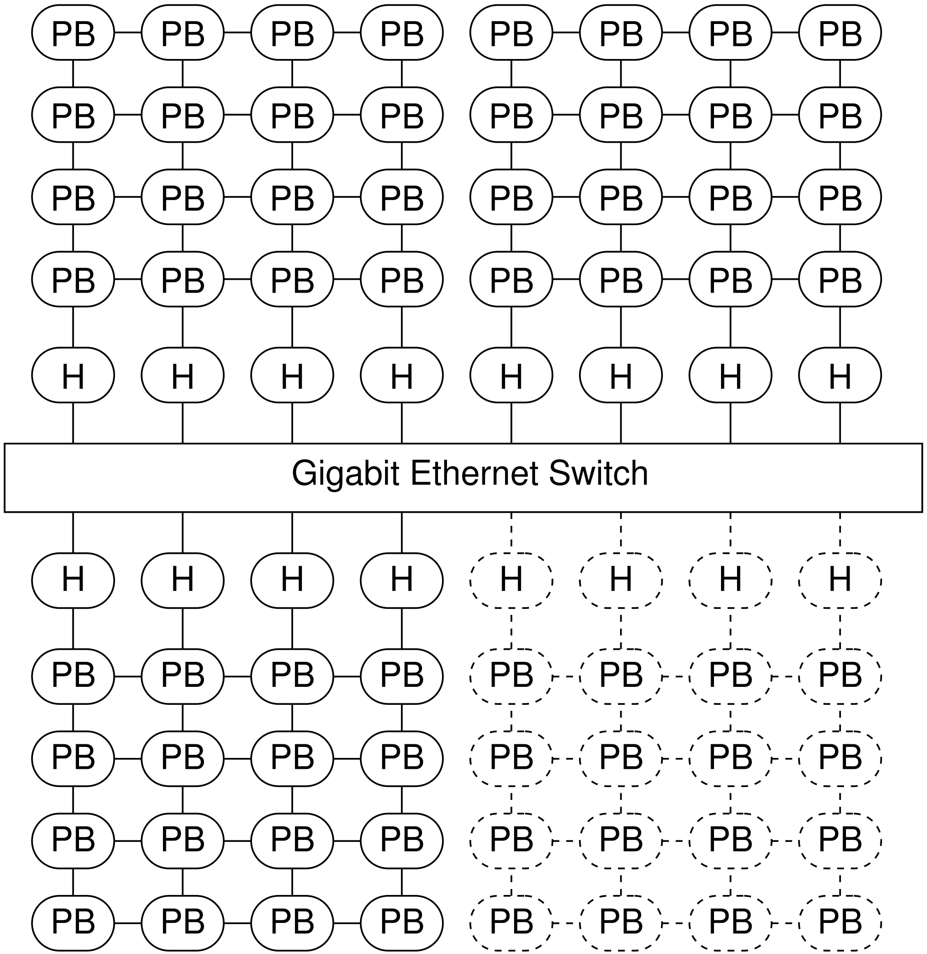


GRAPE-6 Processor board



- **32 chips/board**
- **Semi-serial (LVDS) interface(350MHz clock, 4 wires)**

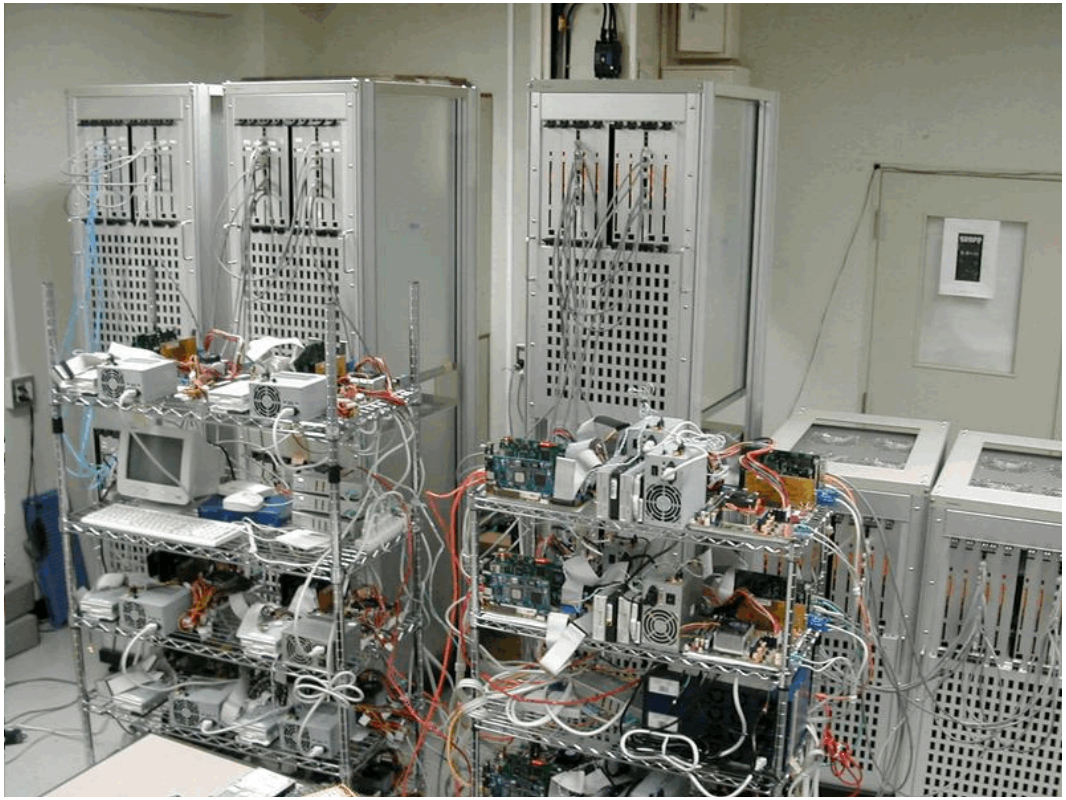
The full 64 Tflops GRAPE-6 system



- 4-host, 16-board "block" with dedicated network
- 4 (currently 3) "blocks" connected through GbE network

Combination of host network solution and dedicated network solution.

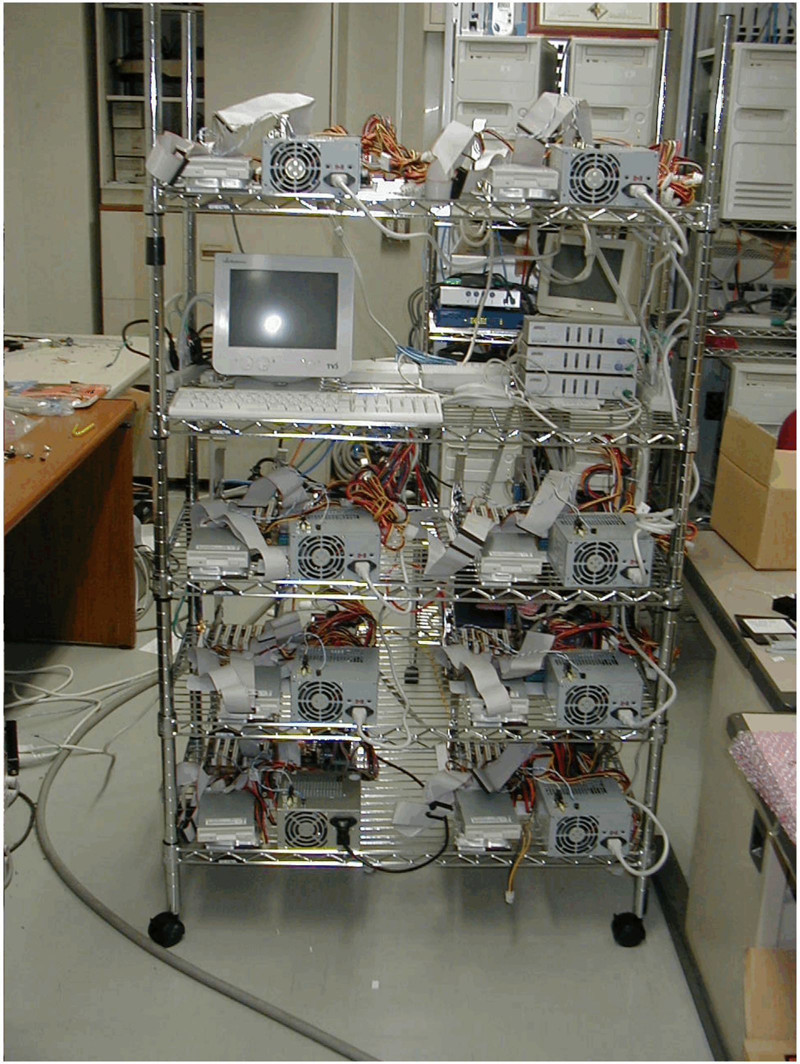
The 64-Tflops GRAPE-6 system

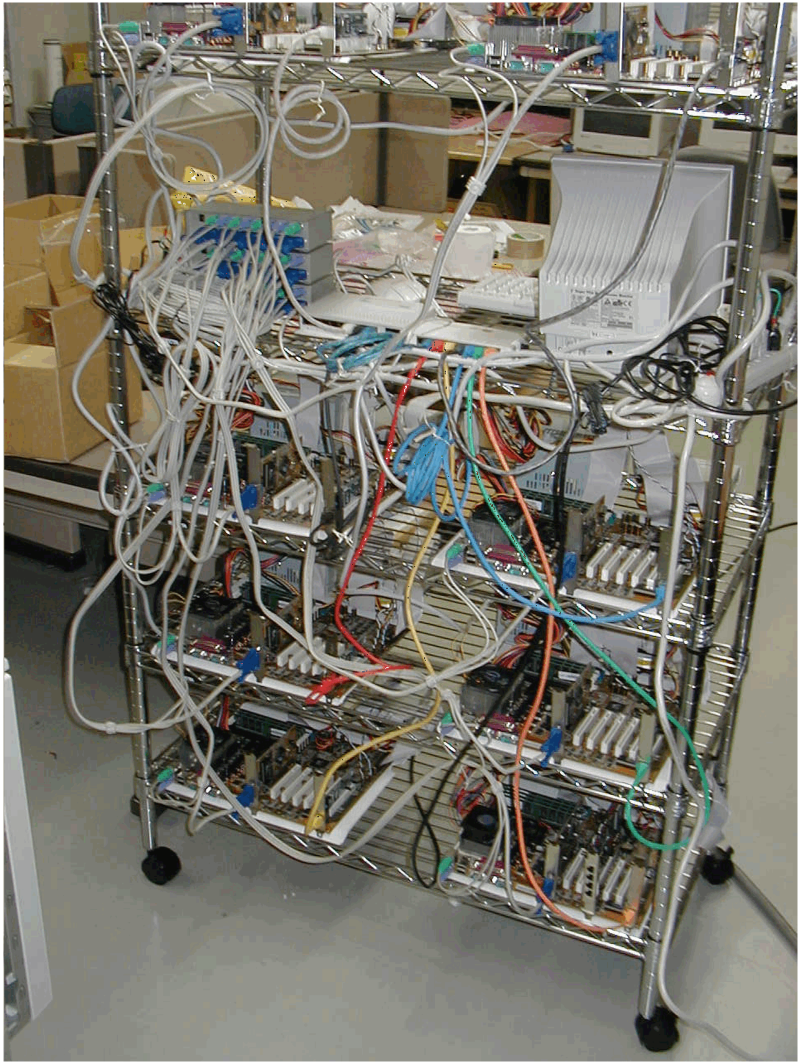


Present 64-Tflops system.

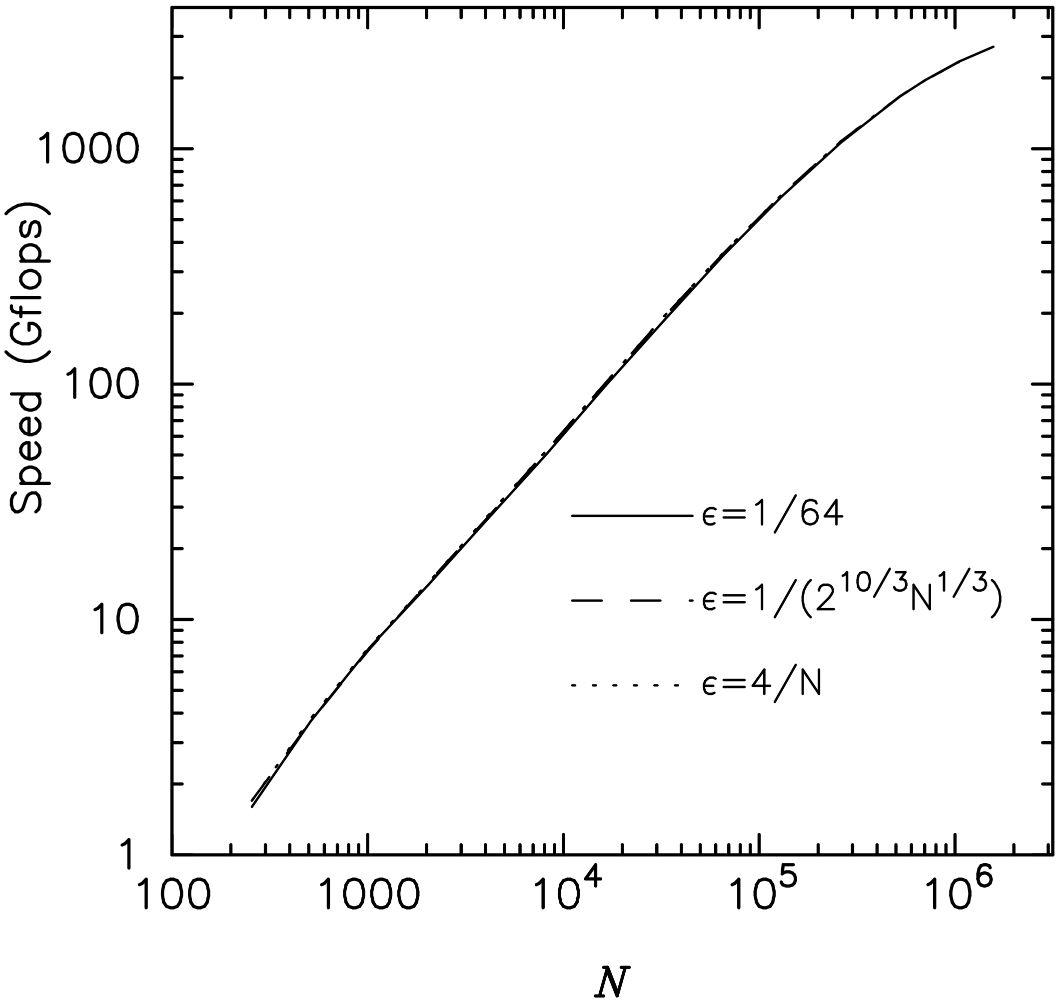
4 blocks with 16 host computers.

The host "PC Cluster"





Some performance numbers

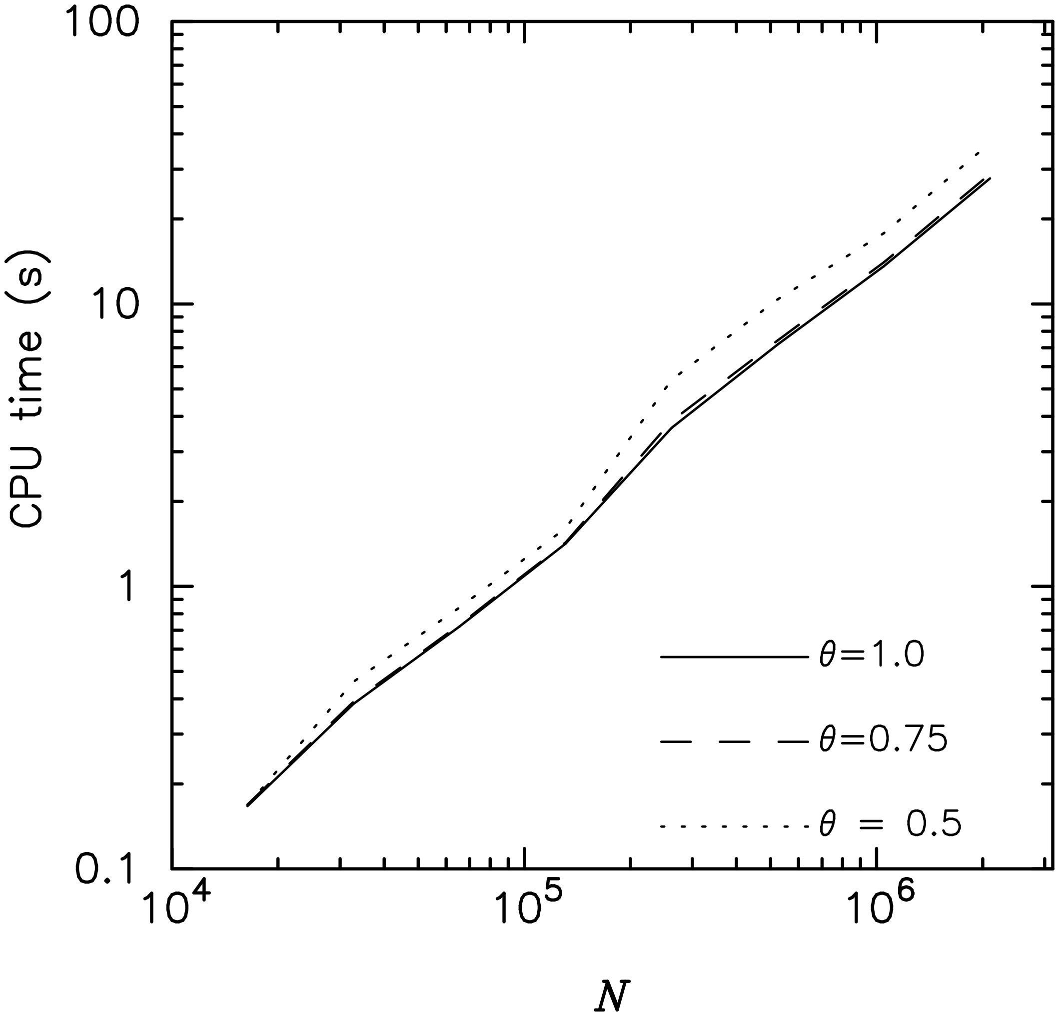


Direct summation (individual timestep)

Peak 4 Tflops

Half of the peak: 400K particles

Some performance numbers (2)

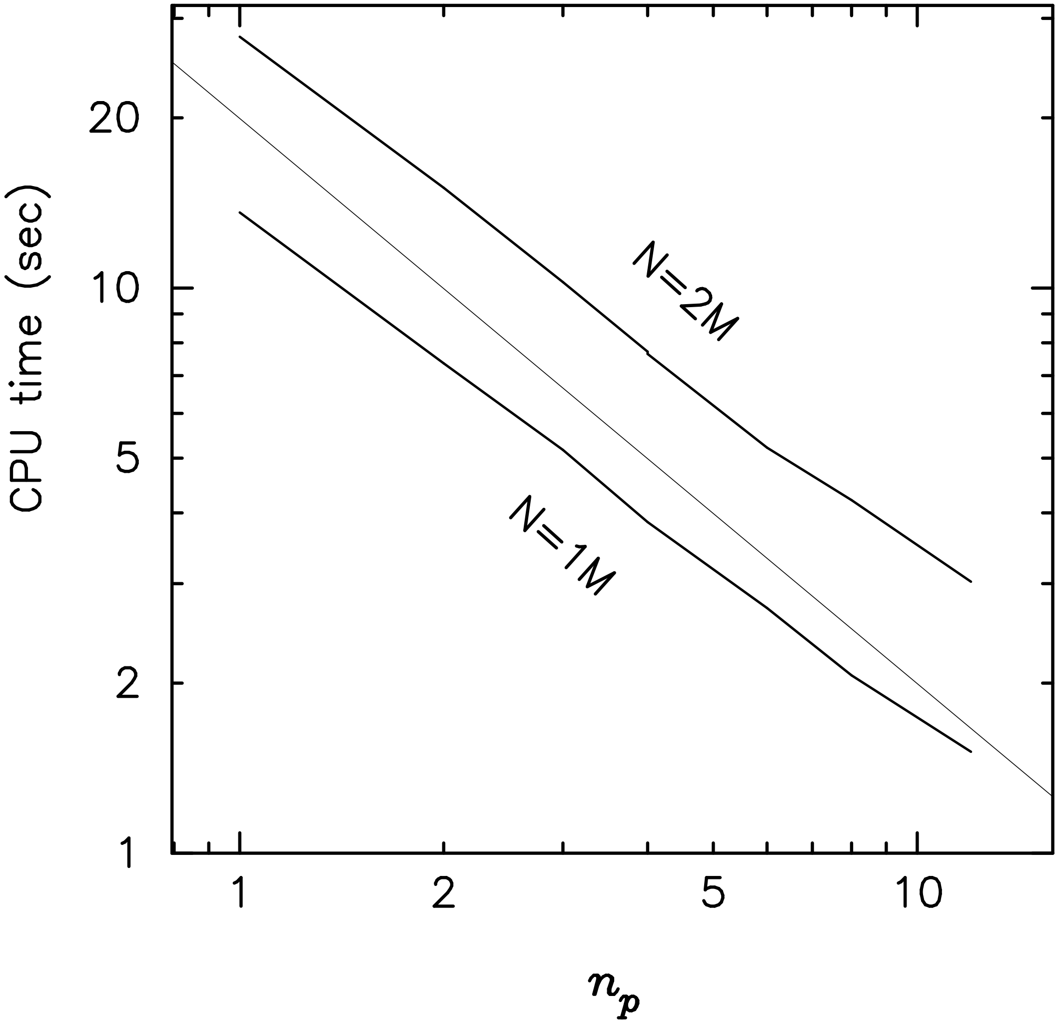


Treecode (Barnes' "modified tree" algorithm)

Plummer model with $r_{cut} = 22.8$ (Heggie unit)

GRAPE-6 is suboptimal for tree... GRAPE too fast for the host.

Some performance numbers (3)

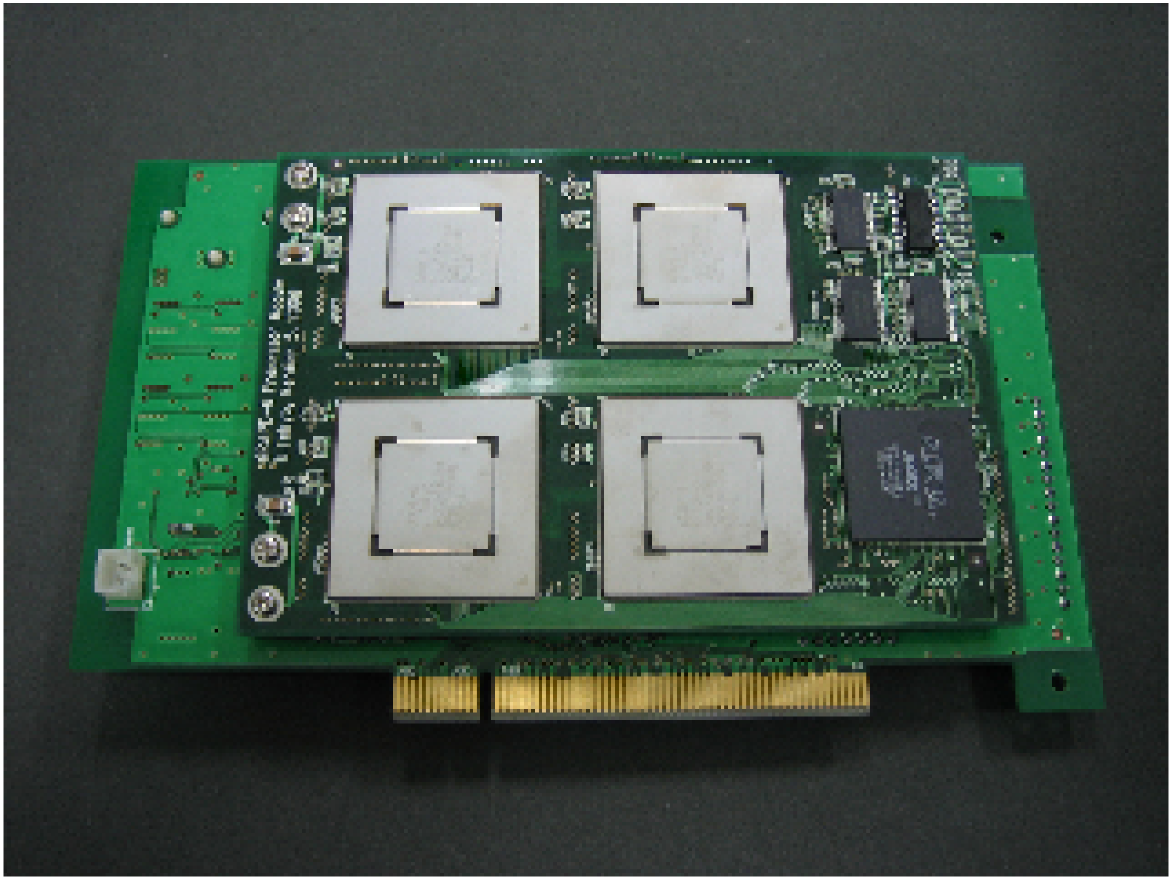


Parallel treecode (JM 2004 PASJ)

Orthogonal Recursive "Multisection"

http://jun.artcompsci.org/softwares/pC++tree/index.htm

BabyGRAPE (aka microGRAPE)



Fukushige et al 2005
Single PCI card with peak speed of 123 Gflops

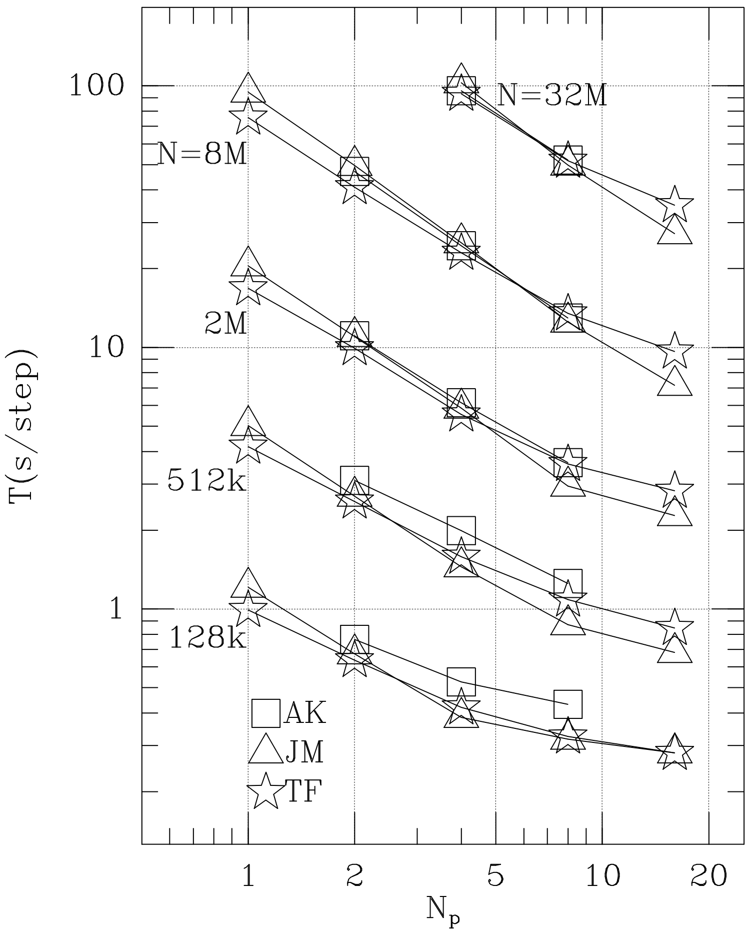
Commercial version: `http://www.metrix.co.jp/micro_grape_eng.html`

24-nodes BabyGRAPE Cluster

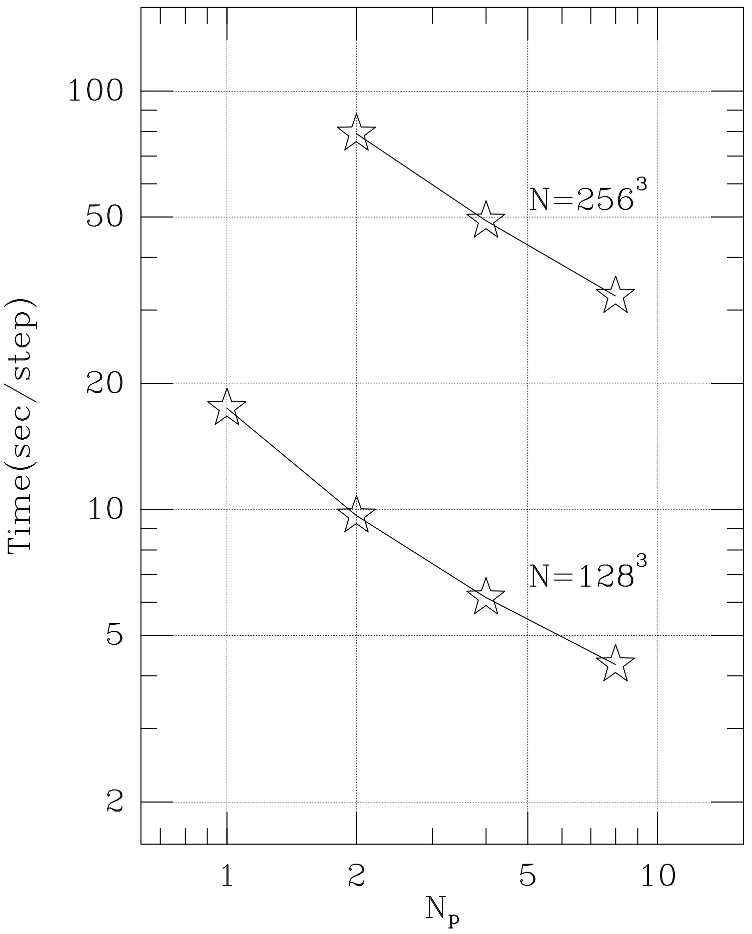
Pentium 4 hosts, GbE connection.

Parallel BabyG Performance

Parallel tree



TreePM



astro-ph/0504095, 0504407

GRAPE6 worldwide

incomplete **list of GRAPE-6s**

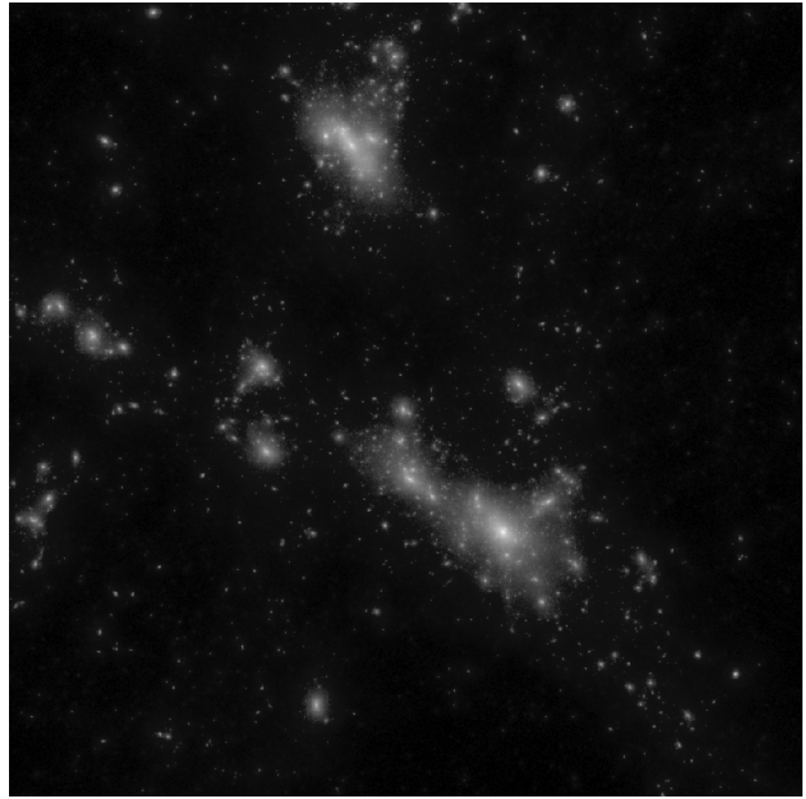
AMNH 4 G6s
Amsterdam
ARI Heidelberg 32 BGs
Bonn
Cambridge
Drexel 2 G6s?
Indiana
Marseilles
McMaster
Michigan
MPIA
Munich
NAOJ 12 G6s
Rochester 32 BGs
TIT
Tsukuba 256 BGs (06?)

Science with GRAPE

- Cosmology (CDM halo)
- Globular clusters
- Galactic nuclei (black hole binaries)
- Planet formation
- Star formation
- Young star cluster (Portegies Zwart)
- Galactic dynamics
- galaxy formation
- ...
- ...

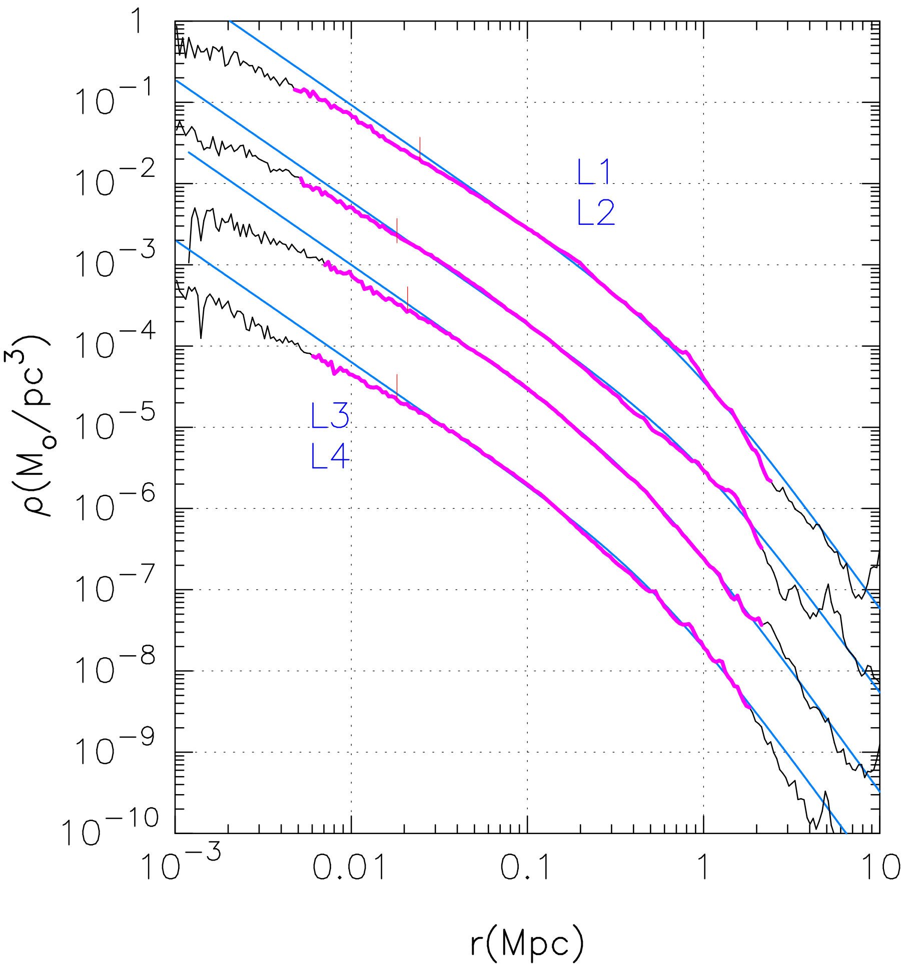
CDM halo simulation

GRAPE-5 Cluster



Simulated Cluster

Density profiles

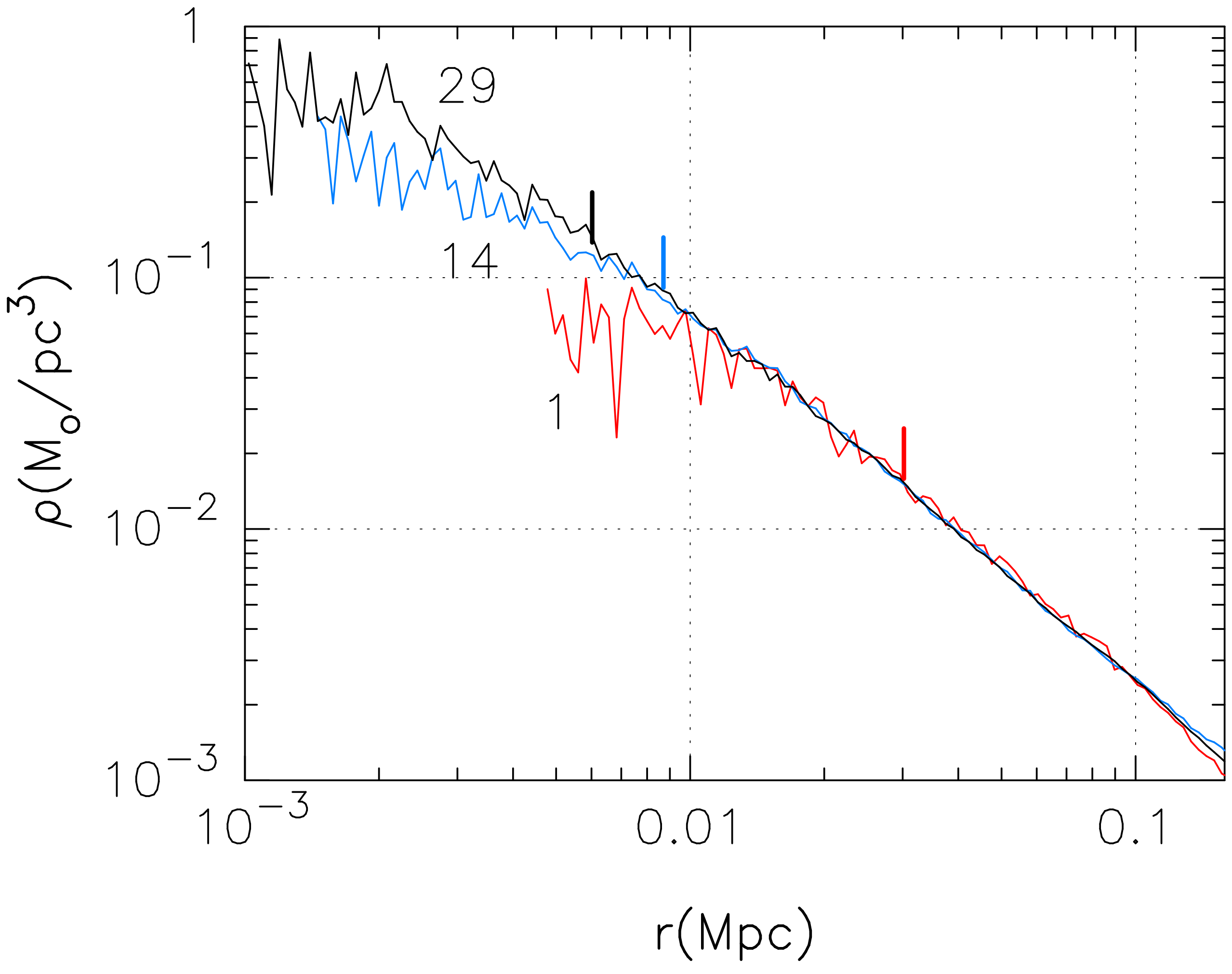


LCDM model

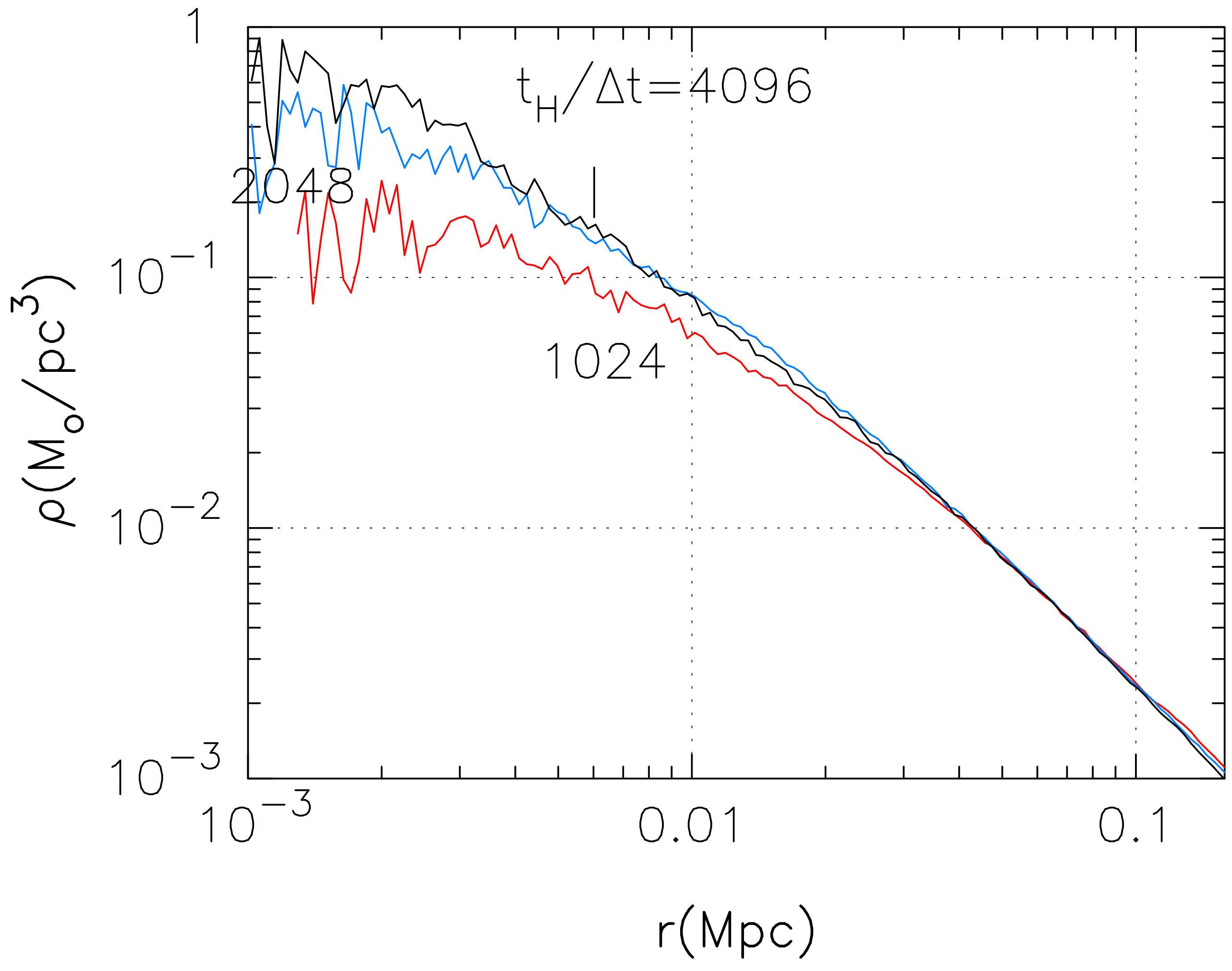
$\boldsymbol{N_v \sim 30M}$

Dependence on N

1M, 14M and 29 M

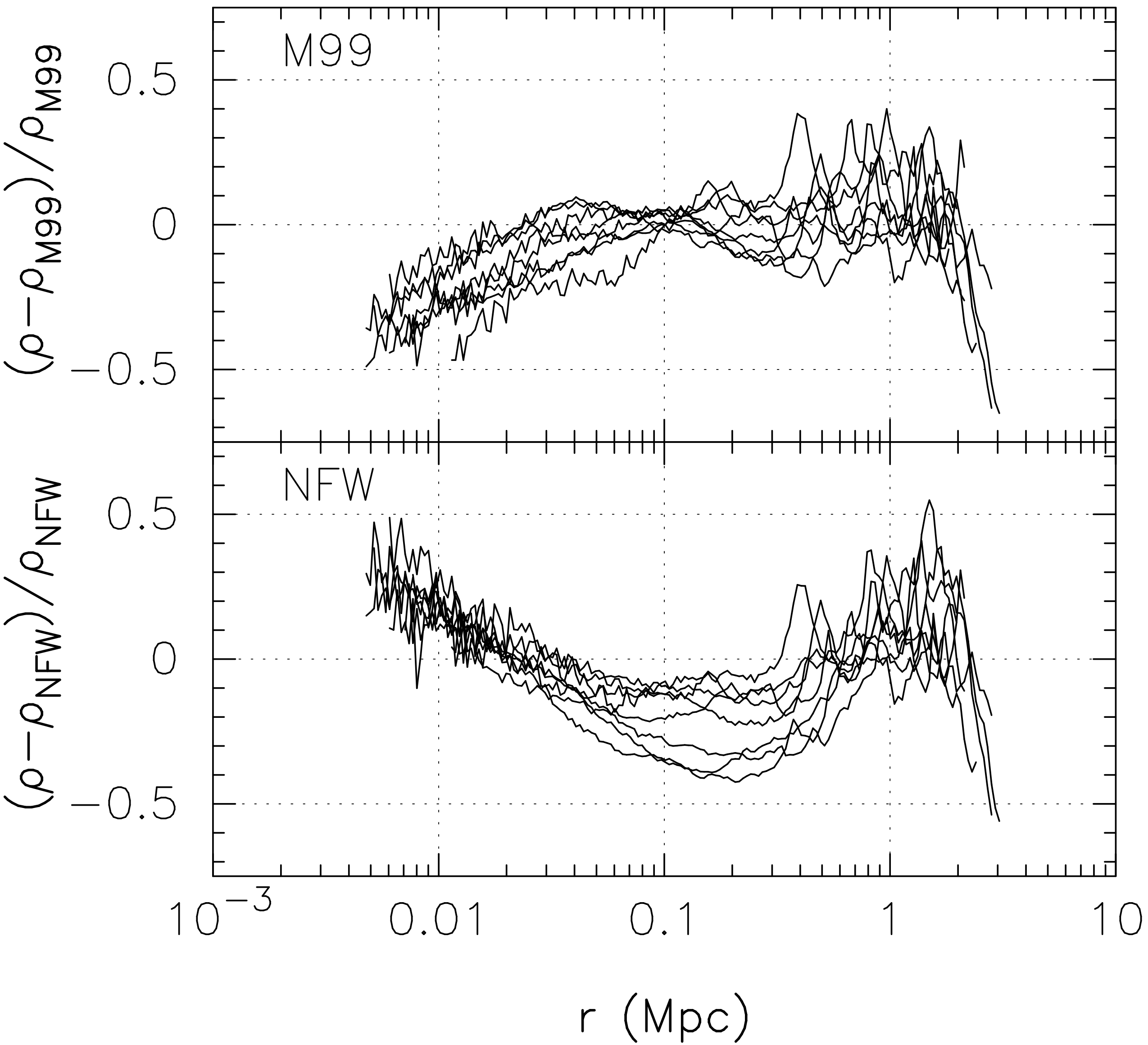


Effect of timestep

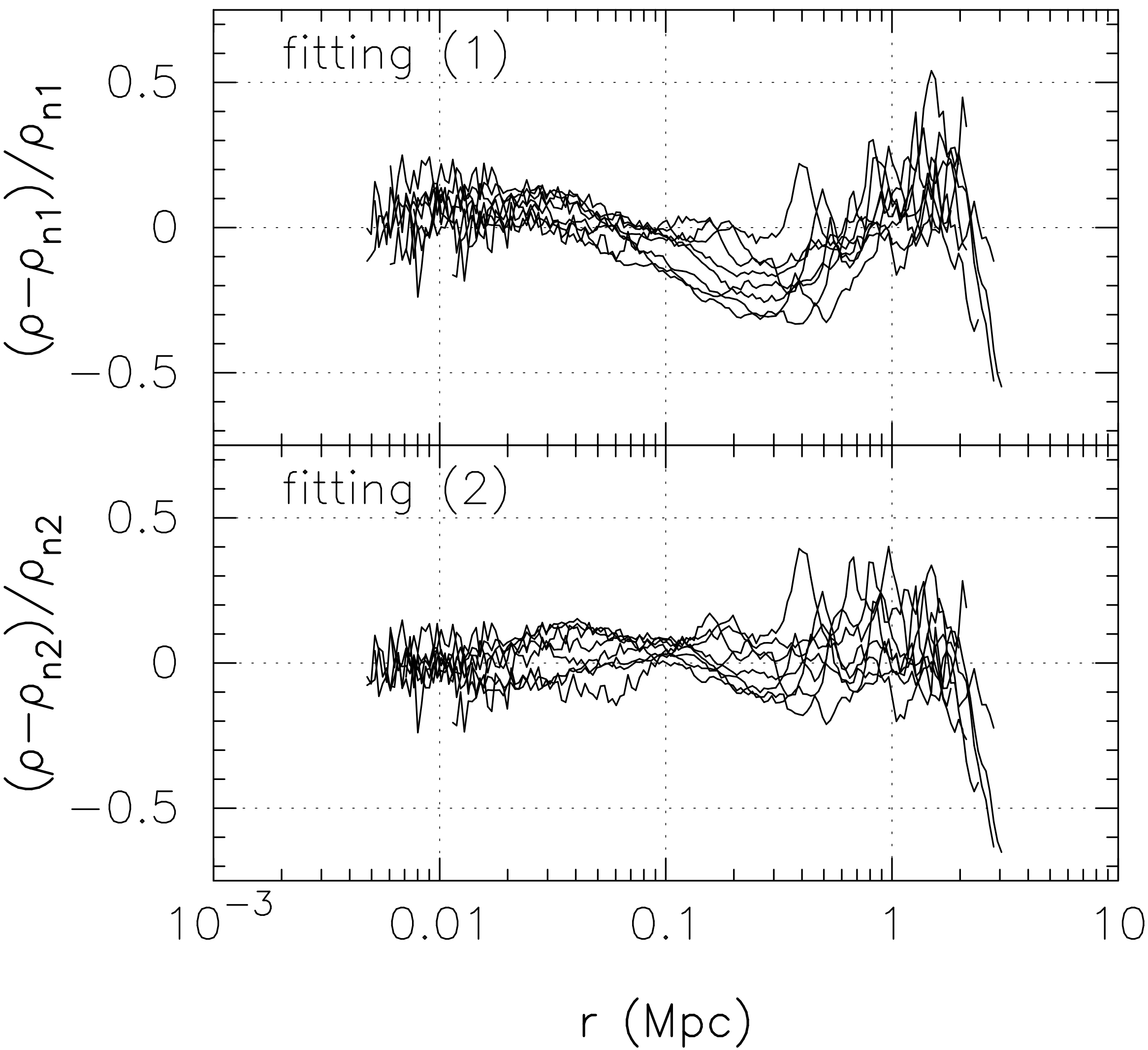


Note: softening = 1kpc.

NFW or Moore?


M99
NFW
0.5
0
−0.5
$(\rho-\rho_{M99})/\rho_{M99}$
$(\rho-\rho_{NFW})/\rho_{NFW}$
10^{-3}
0.01
0.1
1
10
r (Mpc)

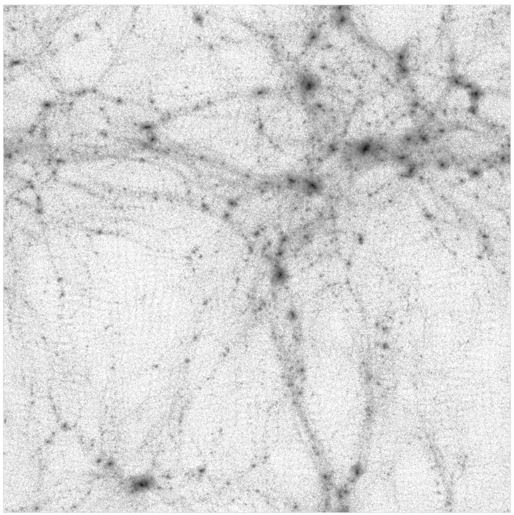
Or something in between?



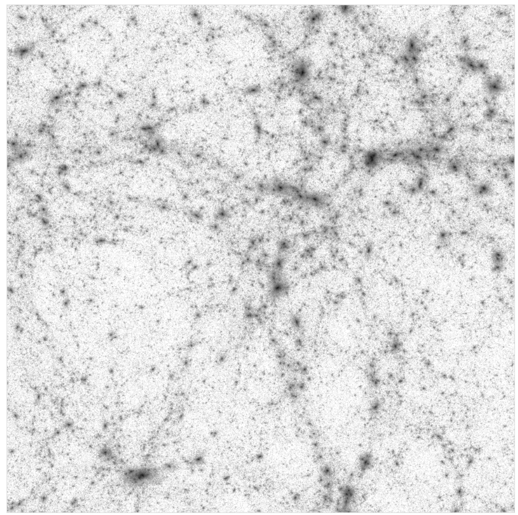
Work in progress

Power-Law Cosmology $P(k) \propto k^n$

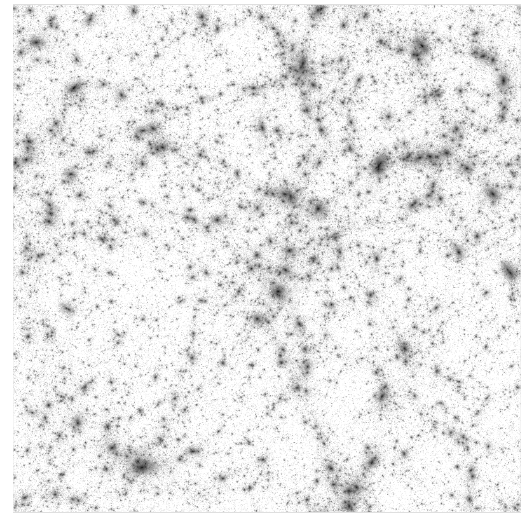
Understanding the origin of the cusp



$n = -2.8$



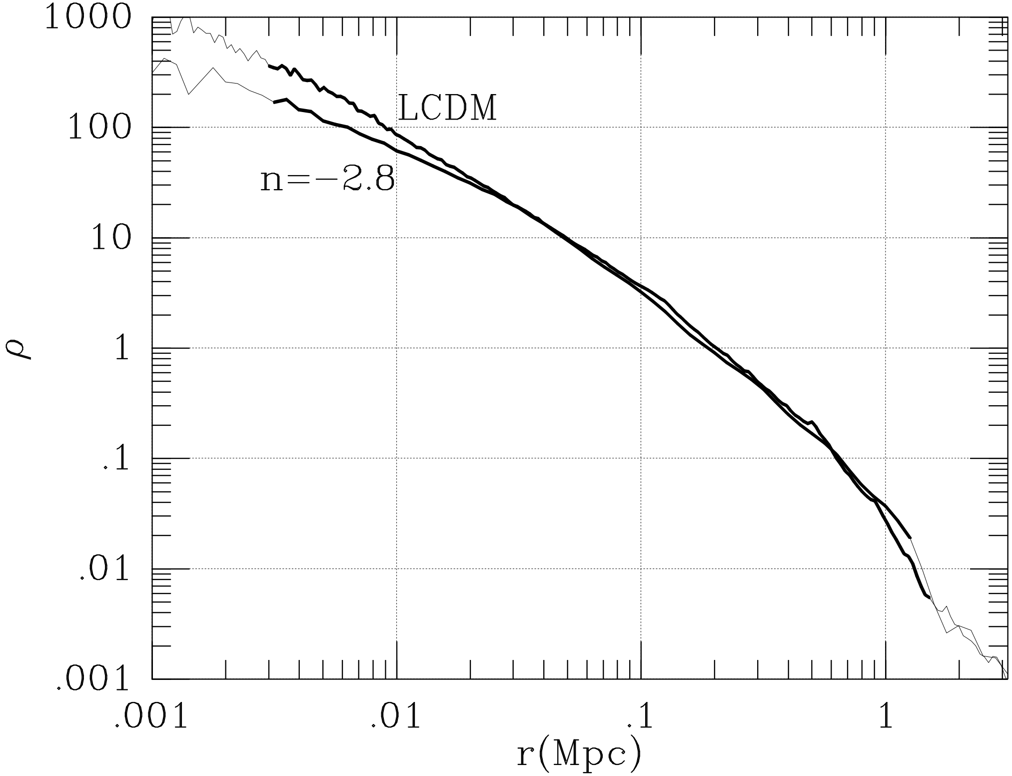
$n = -2$



$n = -1$

CDM is in between -2.8 and -2

Power-law cosmology



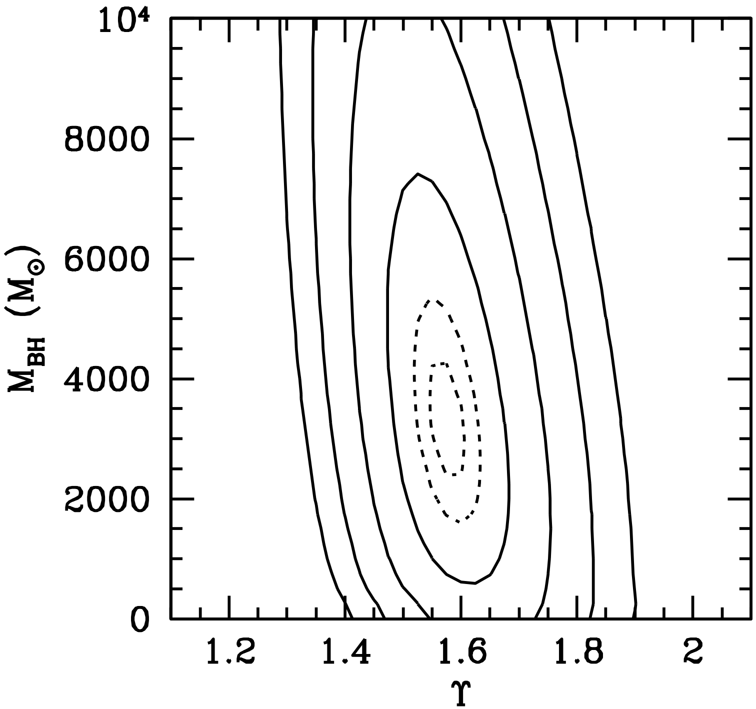
$n = -2.8$ resulted in shallower cusp. Cusp slope dependent on the initial spectrum?

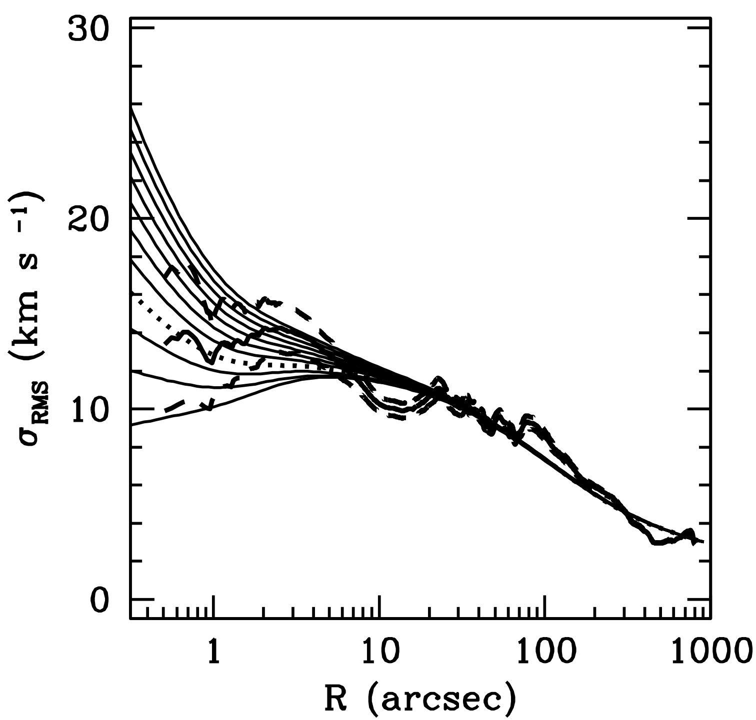
Globular clusters with and without IMBH

- **M15 — without BH**
- **GCs with BH**

Central Black Hole in Globular Clusters?

Observation + Interpretation

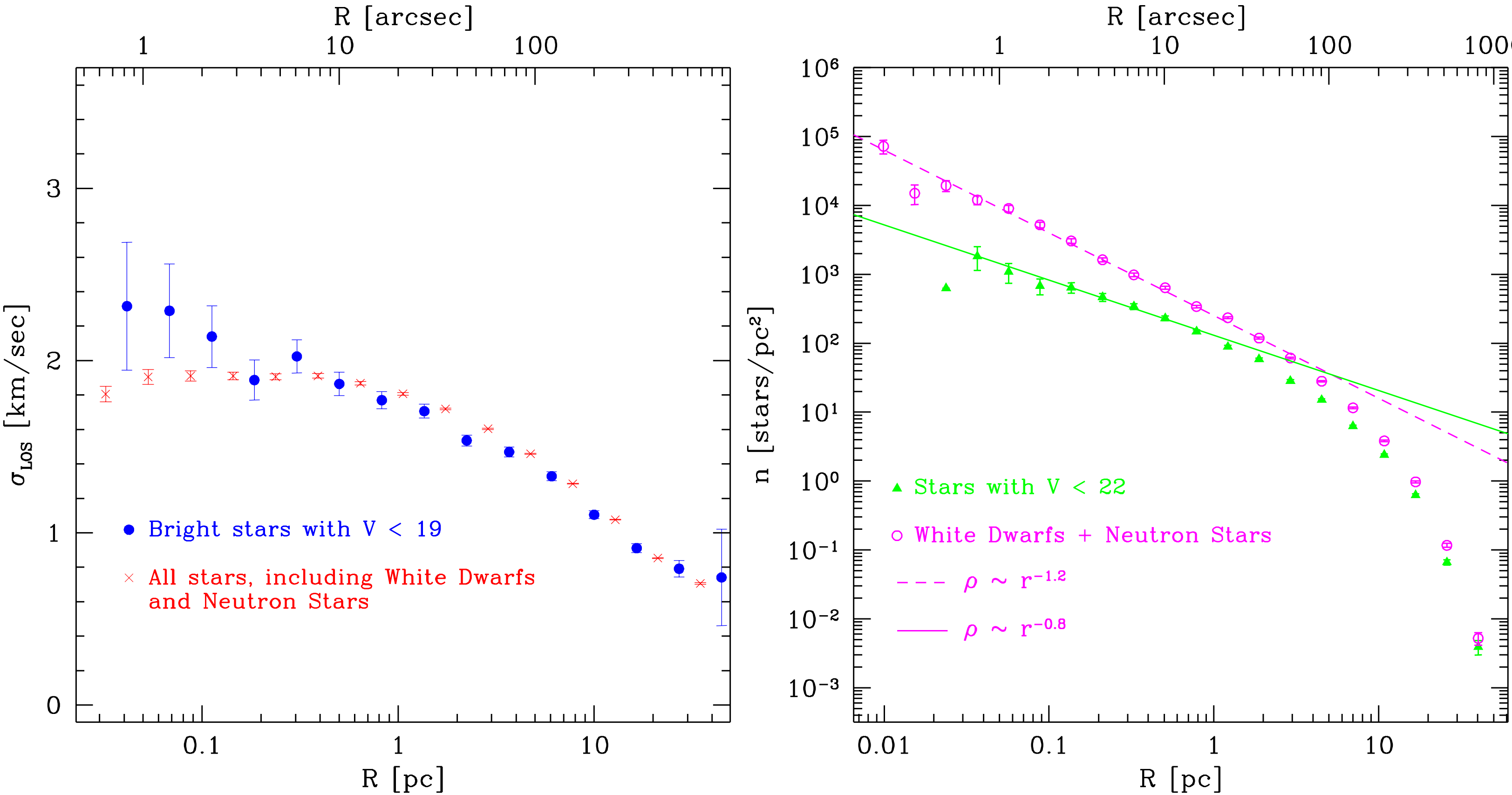




3000 $M_\odot$ black hole? (Gerssen et al 2002)

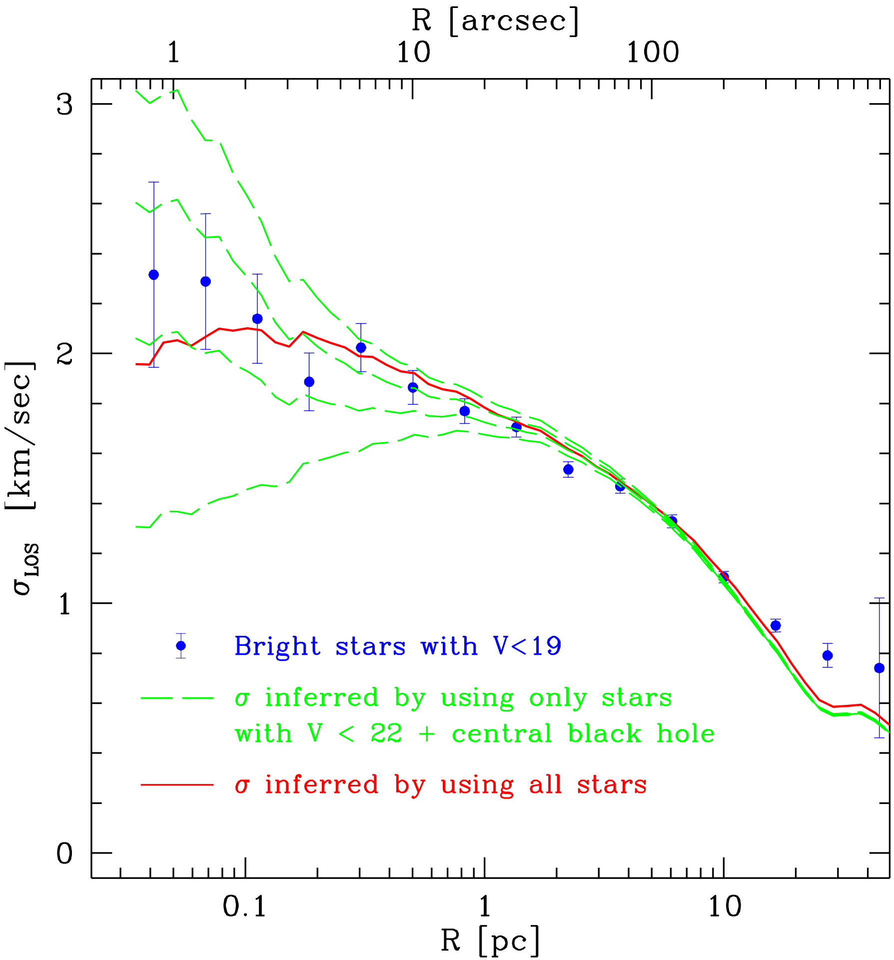
N-body simulation without BH

Baumgardt et al., ApJ 2003, 582, L21.



Left: velocity dispersion; Right: Surface density.

We "found" BH, though there wasn't



Inversion of surface number density of bright stars gives too small central velocity dispersion without central BH.

Estimated BH mass = $80M_\odot$. If scaled to M15, $\sim 3 \times 10^3 M_\odot$ (Gerssen *et al.*: $\sim 3 \times 10^3 M_\odot$)

M 15 does not need black hole.

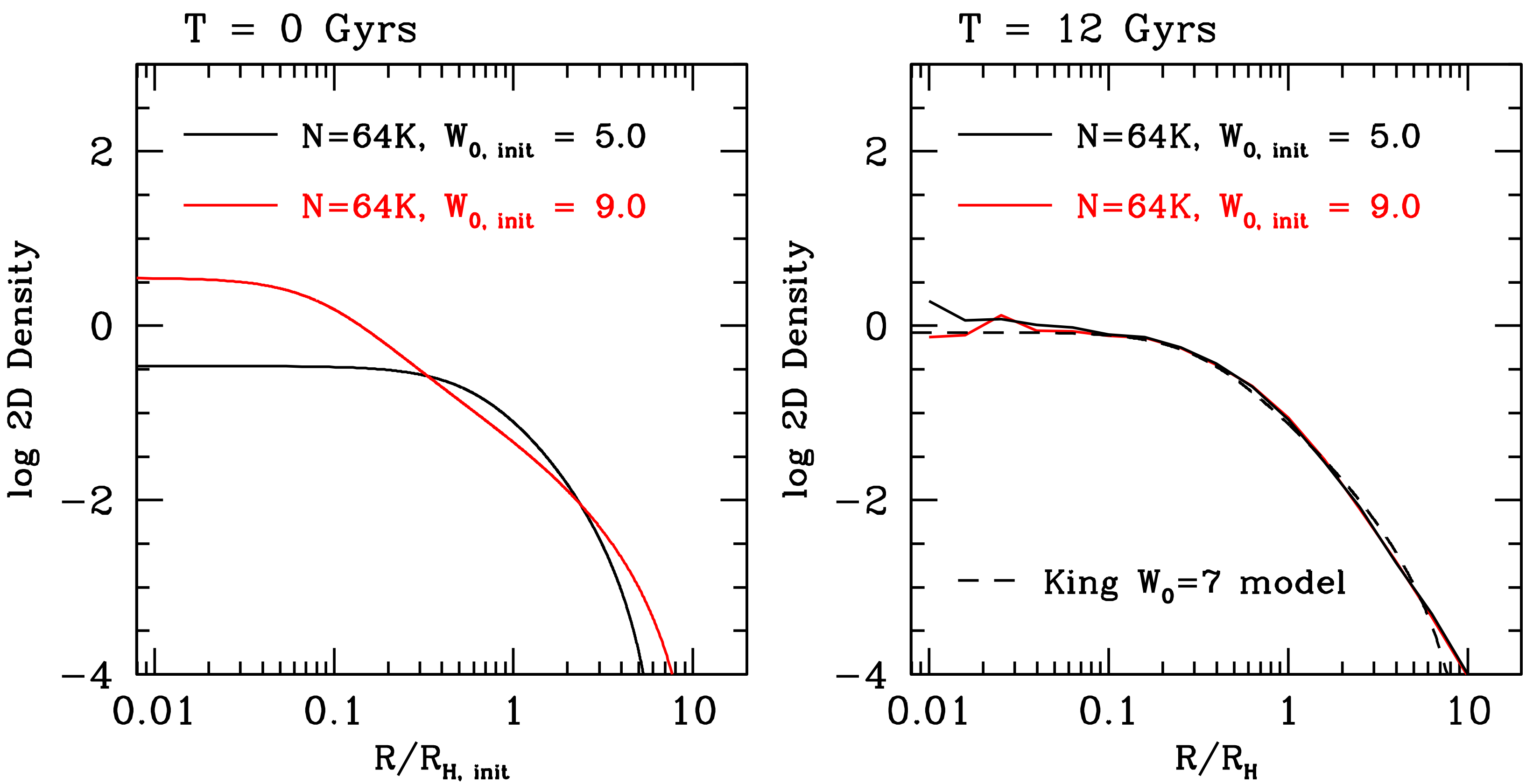
Is there any globular cluster with central BH?

Baumgardt, J.M. and Hut (ApJL 620, 238, 2005)

How would it look like?

Evolution of globular clusters with central BH for Hubble time.

Profile evolution



Surface brightness profile becomes King7-like, almost independent of initial profile and BH mass (in the range of 0.1% to 1%)

Globular cluster summary

- **Globular clusters with central luminosity cusp do not contain massive central BH. They are really clusters in deep core collapse, with NS and WD dominating the central cusp.**
- **Most likely place to find massive central BH is some of normal-looking clusters with relatively large cores.**

Galactic nuclei with SMBH

What will happen to SMBH binary after a galaxy merger?
(talks by Moore, Stadel)

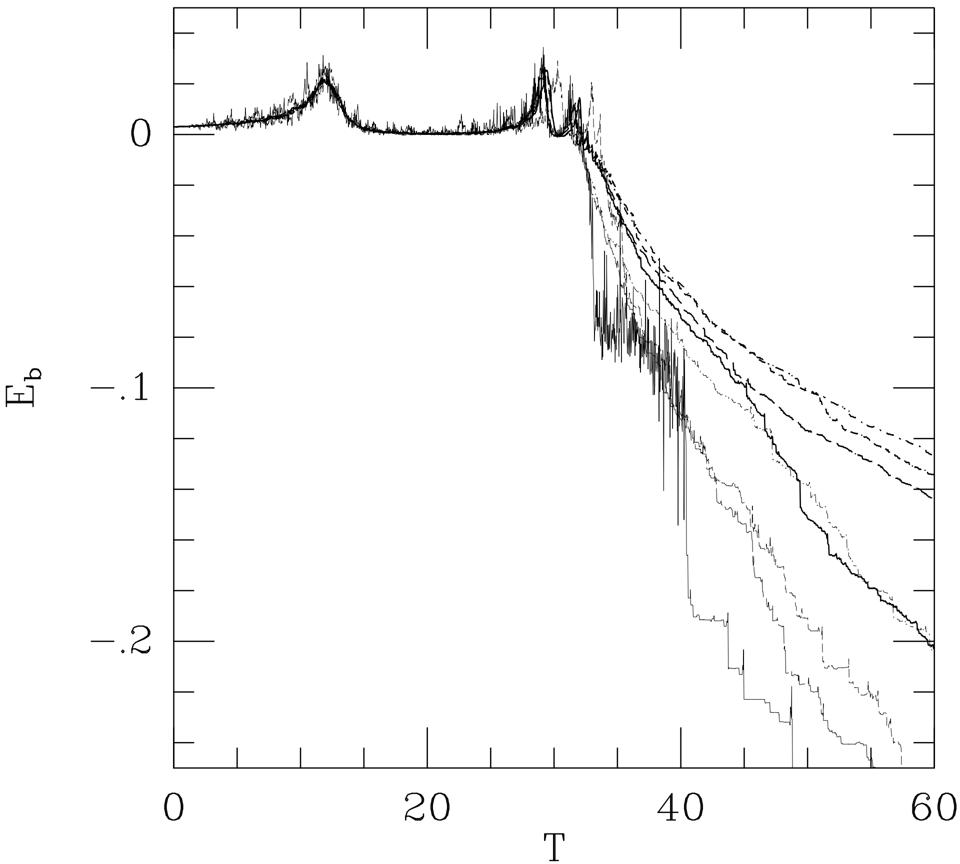
Begelman, Blandford and Rees (1980)
Theoretical argument:

Evolution will stop when BH binary cleaned out its neighbourhood (loss cone depretion)

JM 1997

- King model ($W_o = 7$) merger
- N 2K — 256K
- $M_{BH} = M_{Gal}/32$
- GRAPE-4 direct calculation (NBODY1)
- potential between field particles is softened
- No GW

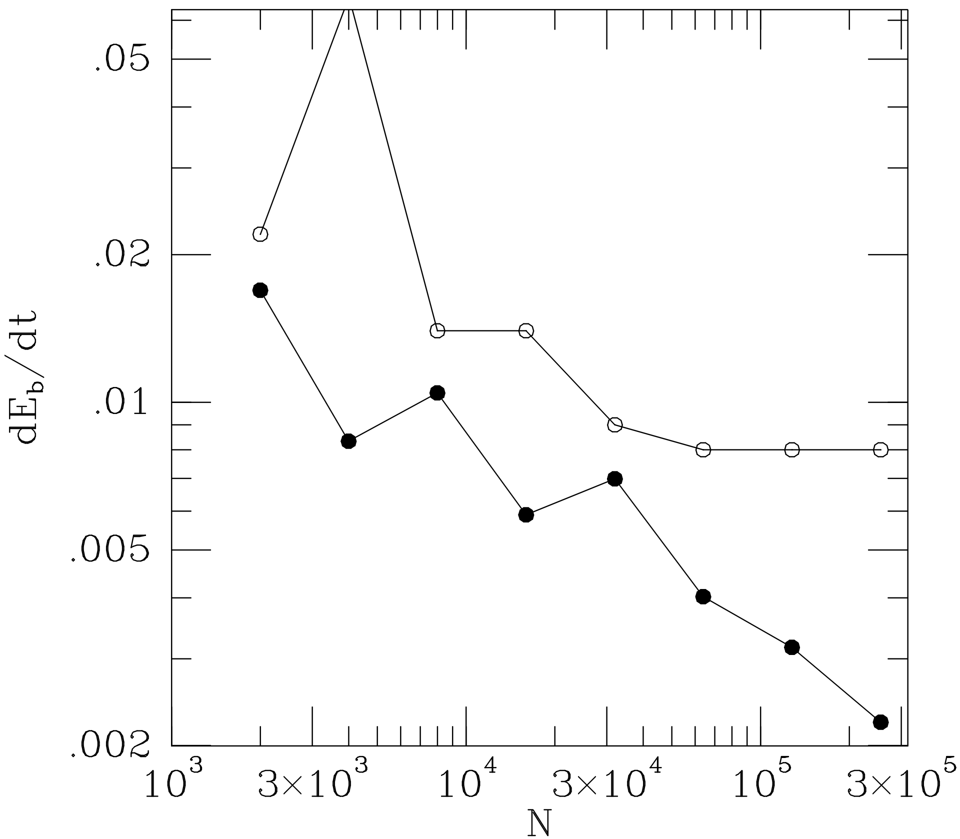
Binding energy



Hardening rate depends on *N*

Dependence?

Hardening rate



Top: E 1/160 to 1/80

Bottom: 1/10 to 1/5

Initially no N dependence

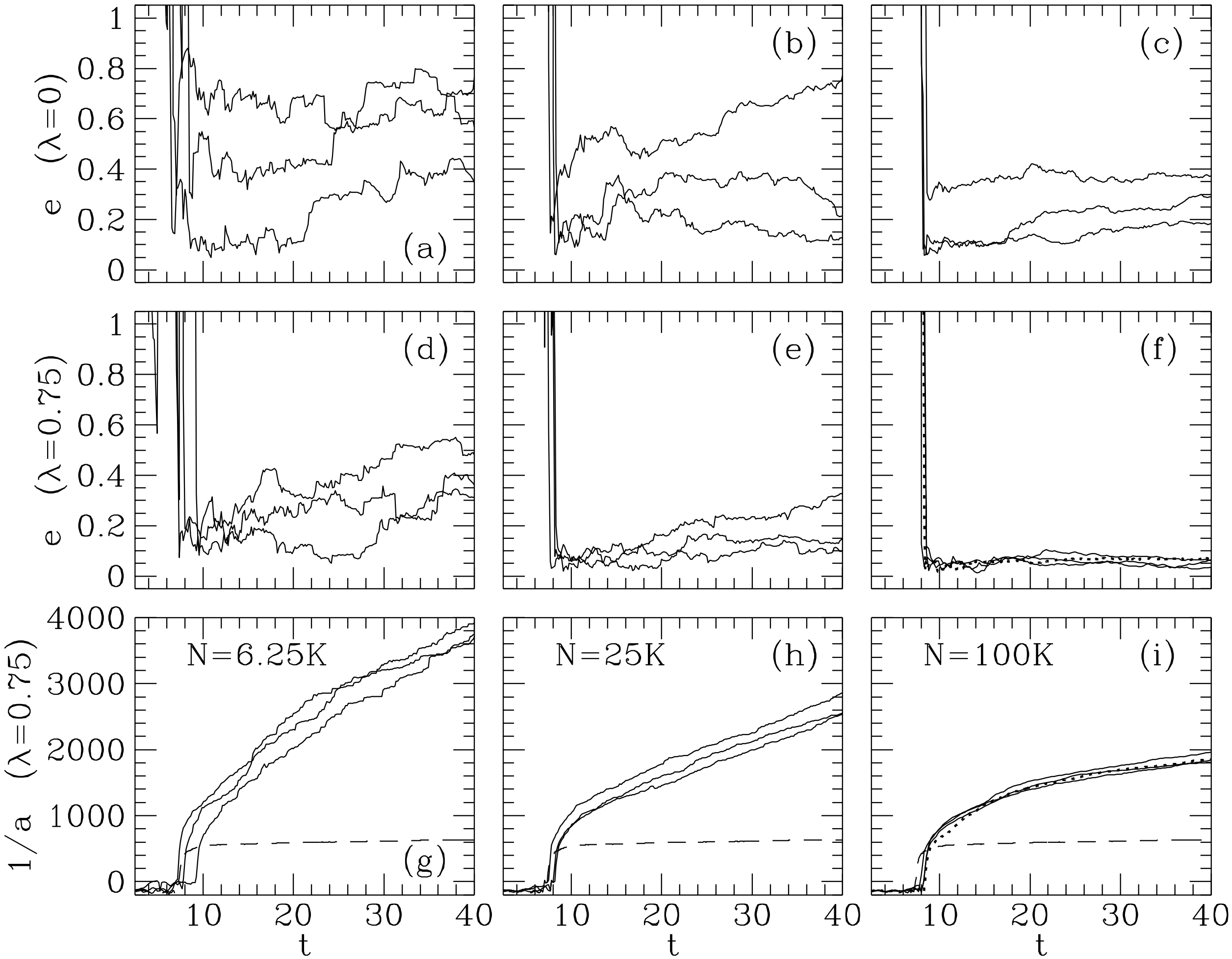
Later stage: $N^{-1/3}$???

SHOULD be $1/N$ (relaxation time)

Quinlan 1997

- **Plummer model, 2 BHs**
- **N 6.25K — 200K**
- $M_{BH} = M_{Gal}/100$
- **SCF + direct**

Result

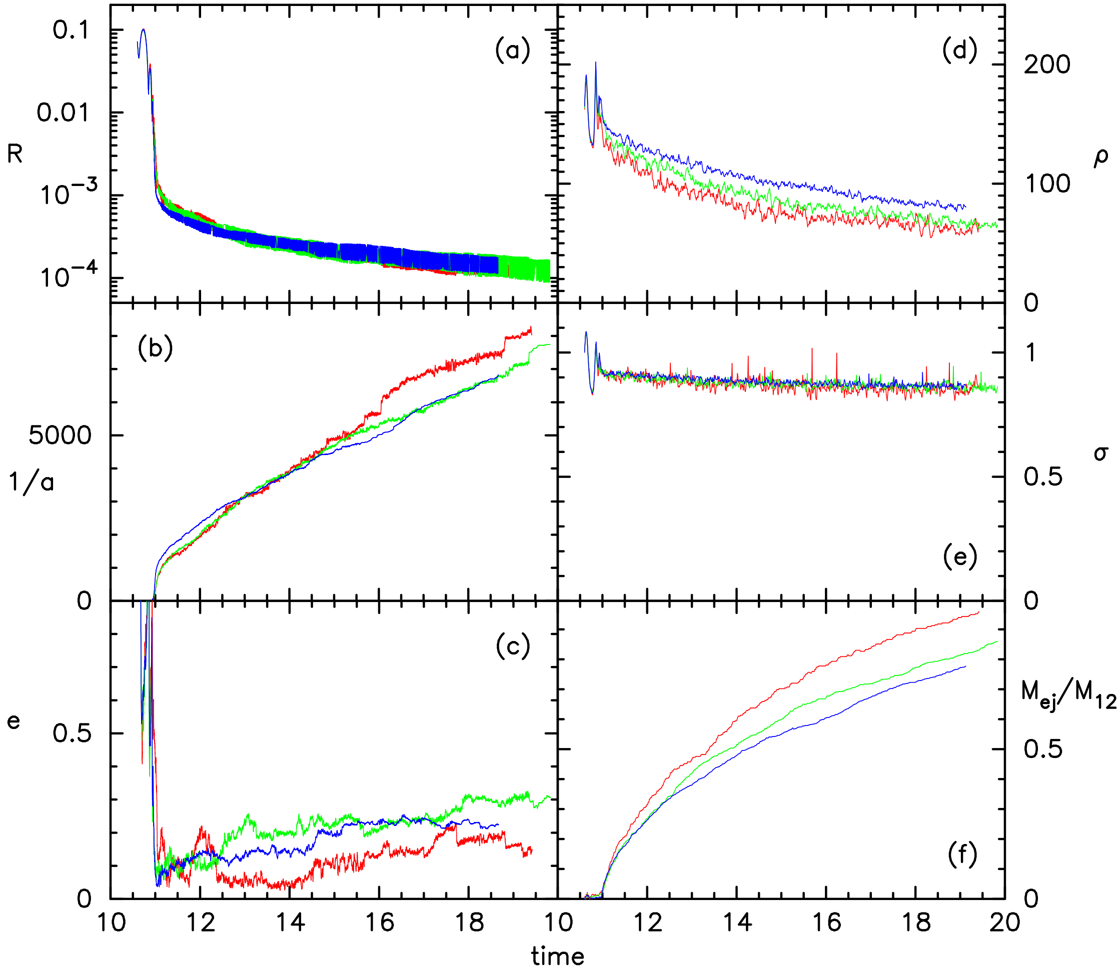


Independent of N for $N > 100\text{K}$???

Milosavljević & Merritt 2001

- $\rho \propto r^{-2}$ cusp model with BH
- N 8K — 32K
- $M_{BH} = M_{Gal}/32$
- Tree+direct
- Tree before BH binary formed (N=256K) Direct after BH formation (Sun Starfire)

Result



No N dependence

Chatterjee, Hernquist & Loeb 2003

- **Same method as Quinlan 1997**
- **N up to 400K**
- **Various M_{BH}**

Claim:
No N dependence for $N > 200K$.

Summary of previous results

Mess

Summary of previous results

Mess

- **Numerical results contradict with each other**
- **All numerical results contradict with the theoretical prediction of loss cone depletion**

What's wrong?

If we knew, we could have done better!

- Too small N?
- Something wrong with codes?
- Initial condition?
- All of above combined?

New calculations

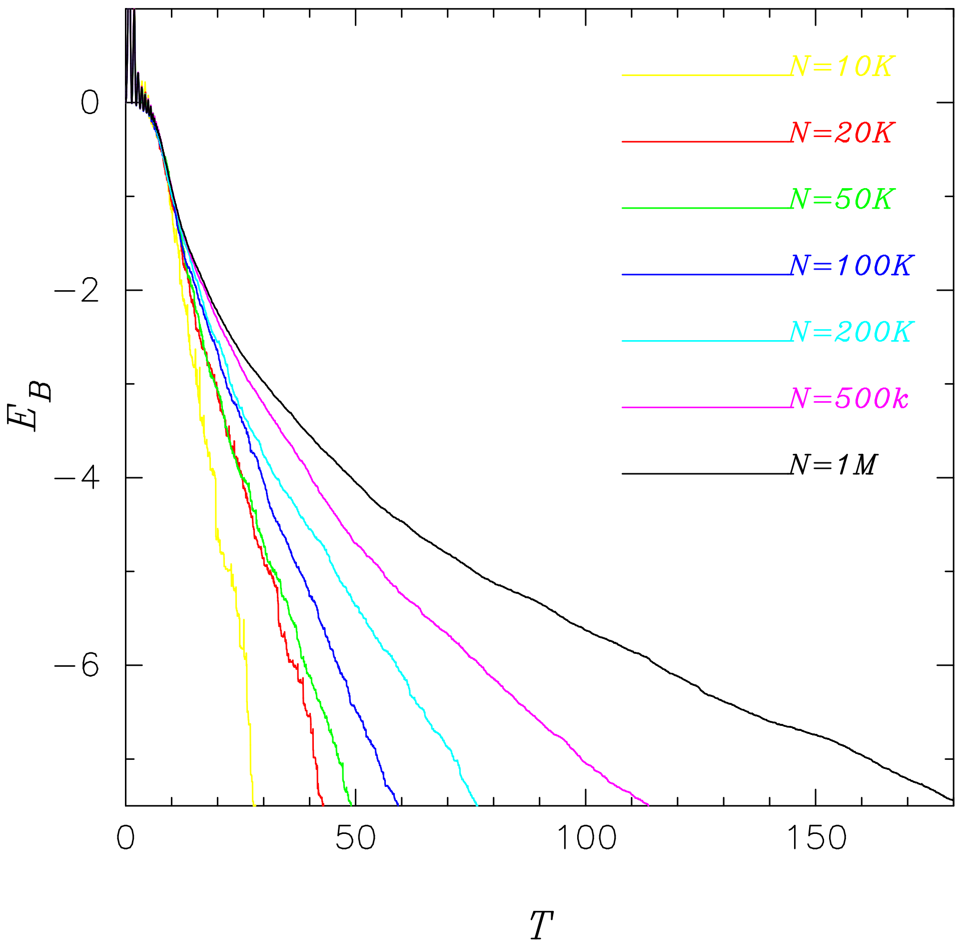
JM and Funato 2004
Goal:

- **For simple model**
- **in which loss cone "should" form**
- **using simple numerical method**
- **perform large-N, long calculations**

Simulation setup

- **Single King model ($W_o = 7$), two BH**
- **N 2K — 1M**
- $M_{BH} = M_{Gal}/100$
- **Direct method on GRAPE-6**
- **Force from BH unsoftened, handled on the host computer**

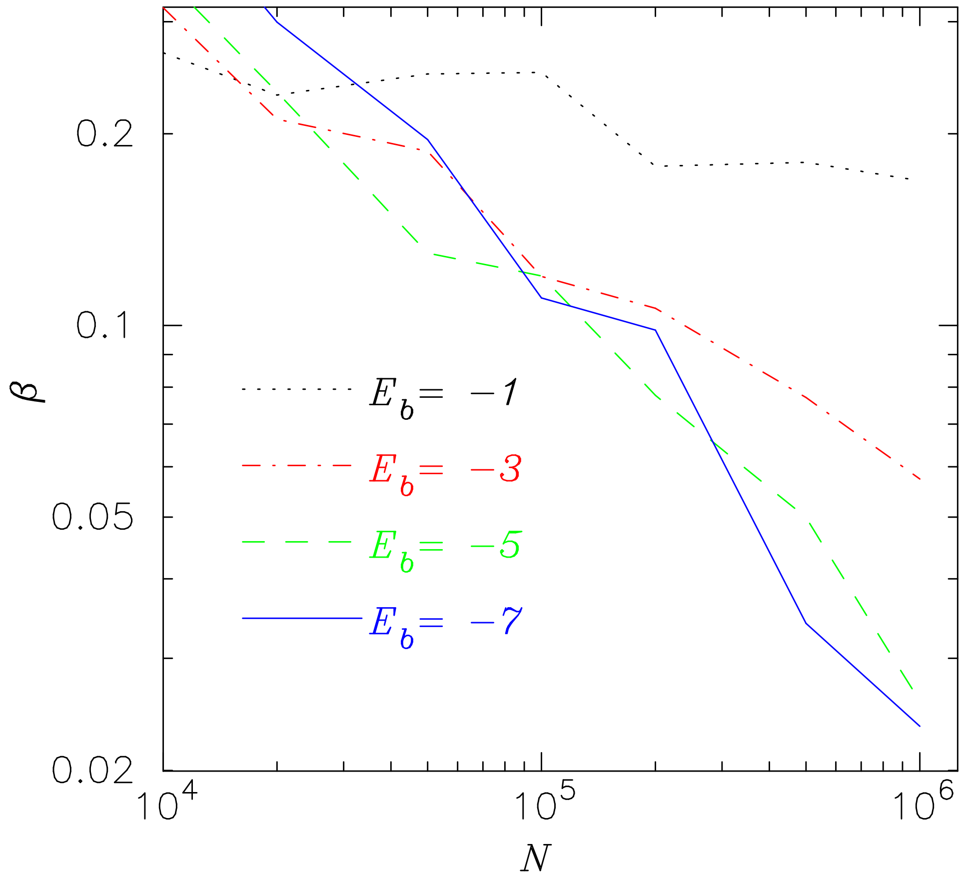
Binding energy



Large N $\longrightarrow$ slow evolution

How slow?

Hardening rate



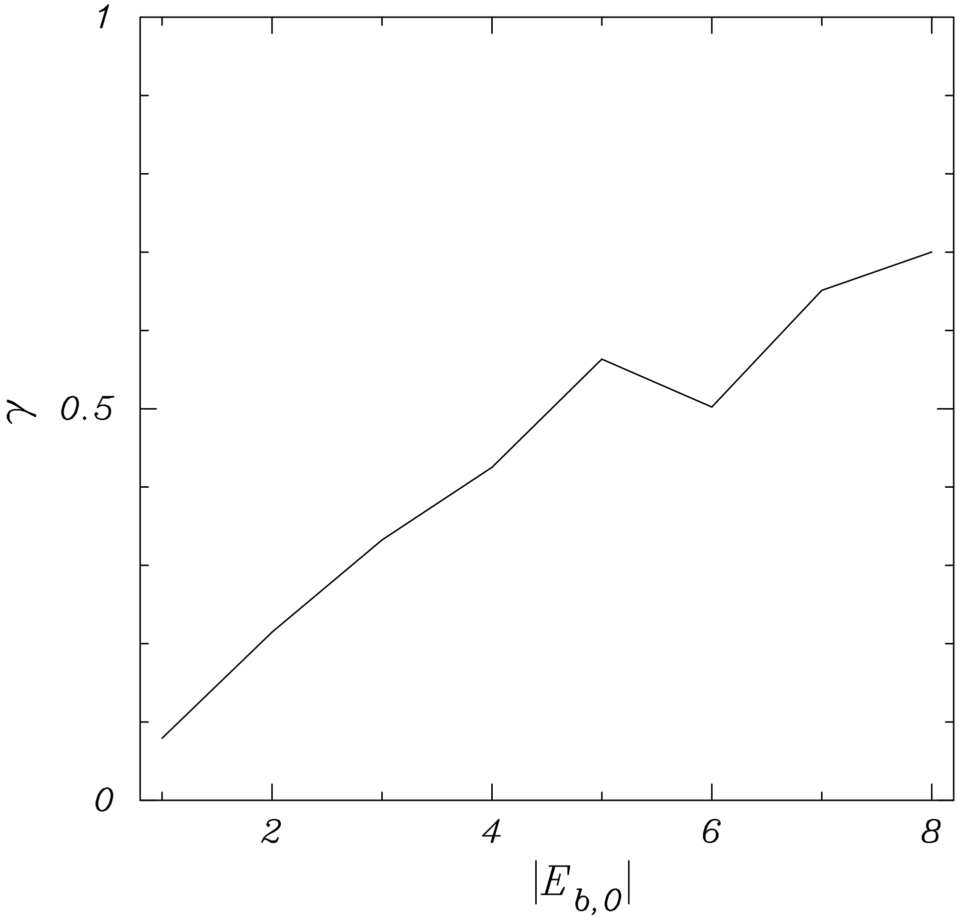
$$\beta = -\frac{dE_b}{dt}$$

N-independent in early phase

Later phase

Dependence becomes stronger as the BH binary evolves?

Dependence on binding energy



Slope becomes steeper as binaries becomes harder.

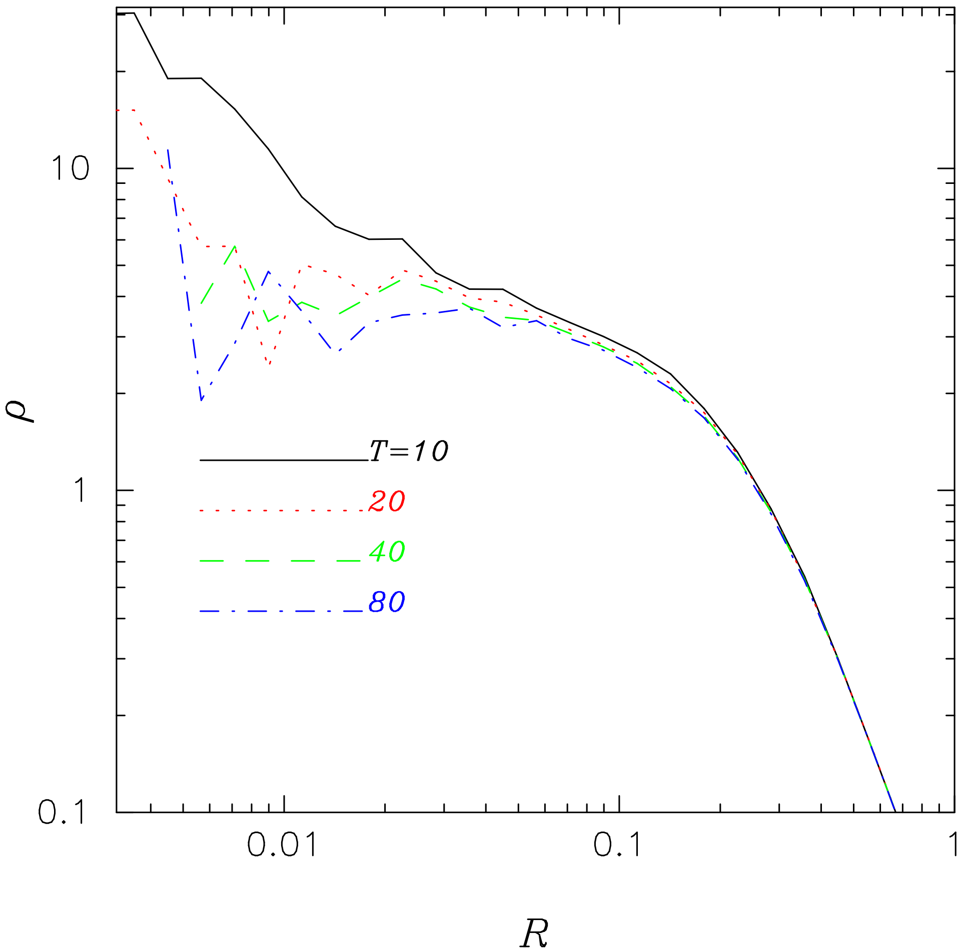
Has not converged in the range we could calculate.

Probably approaching to -1 slope.

Summary

- **Result is not inconsistent with the theory of loss cone depletion**

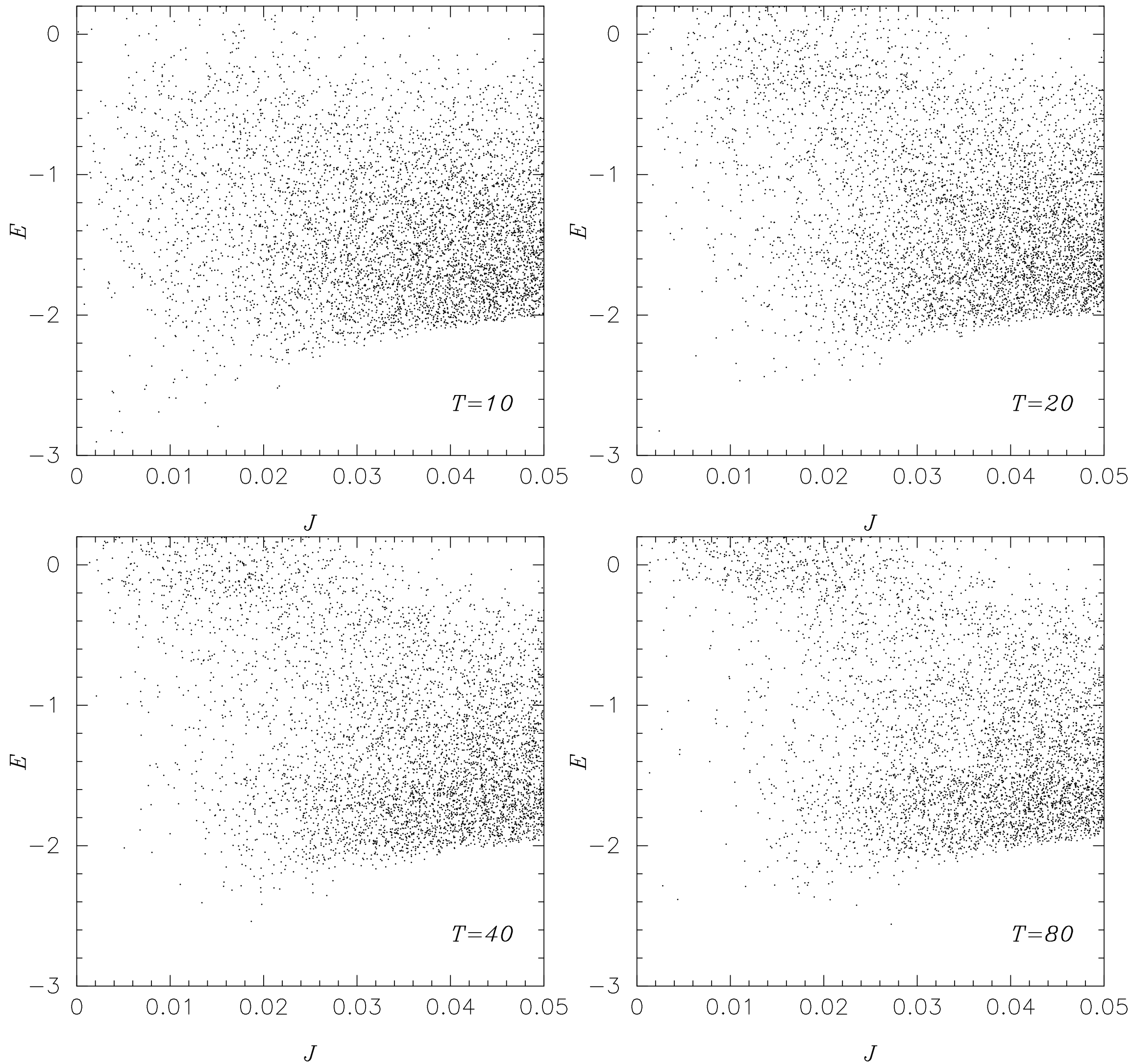
loss cone ?



Cusp vanished

No “density decrease” toward center

loss cone in phase space — (E, J)



particles with $J < 0.01$ depleted

particles accumulate in small J, almost-unbound orbit.

Loss cone is actually visible.

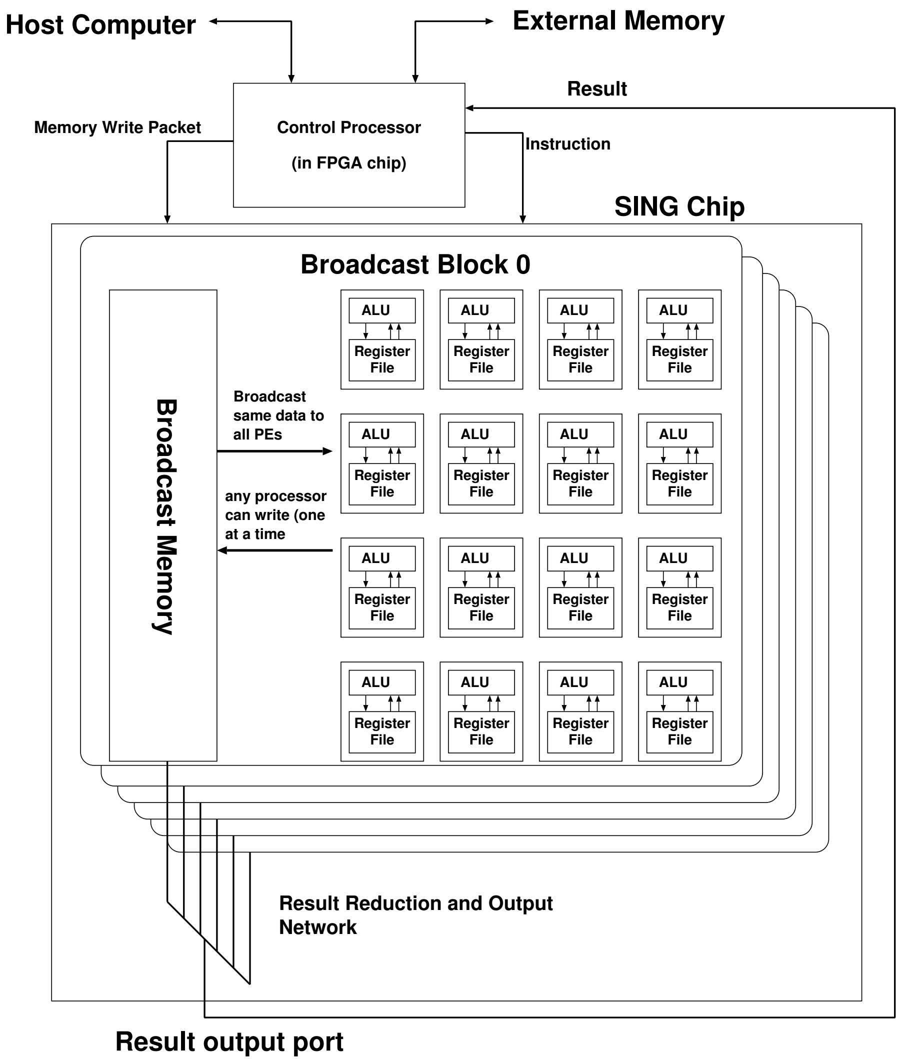
What was wrong with previous works?

- **JM 1997**
  - **Simulation time was too short**
- **Milosavljević & Merritt 2001**
  - **N was also too small**
- **SCF+BH**
  - **Not clear...**

Next-Generation GRAPE — GRAPE-DR

- **Budget approved. (1.5M\$ $\times$ 5 years)**
- **Planned peak speed: 2 Pflops**
- **New architecture — wider application range than previous GRAPEs**
- **Planned completion year: 2008**

GRAPE-DR processor structure

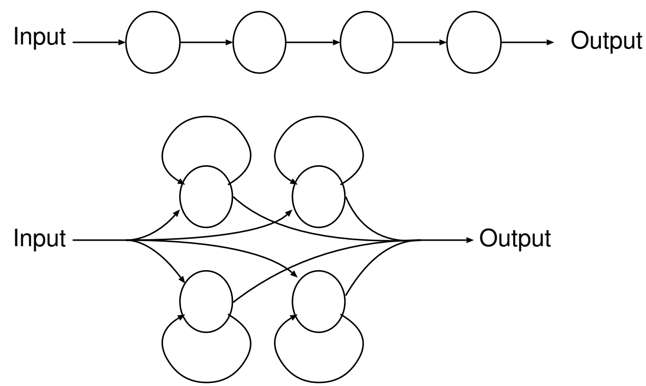


Collection of small processor, each with ALU, register file (local memory)

One chip will integrate (hopefully) 1024 processors
Single processor will run at 500MHz clock (2 operations/cycle).

Peak speed of one chip: 0.5 Tflops (20 times faster than GRAPE-6).

Difference from previous GRAPE architecture



- **No hardwired pipeline, simple SIMD parallel processor.**

  Development codename: SING (*S*ing *i*s *n*ot *G*RAPE) (Eiichiro Kokubo)

- **Much like the Connection Machine**

- **Performance hit: factor 3-10? (We'll see)**

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)**
- **major drawback: somebody (***which means me...***) need to develop the chip**

Why we changed the architecture?

- To get budget (N-body problem is too narrow...)
- To allow wider range of applications
  - Molecular Dynamics
  - Boundary Element method
  - Dense matrix computation
  - SPH
- To allow wider range of algorithm
  - FMM
  - Ahmad-Cohen
- To try something new.

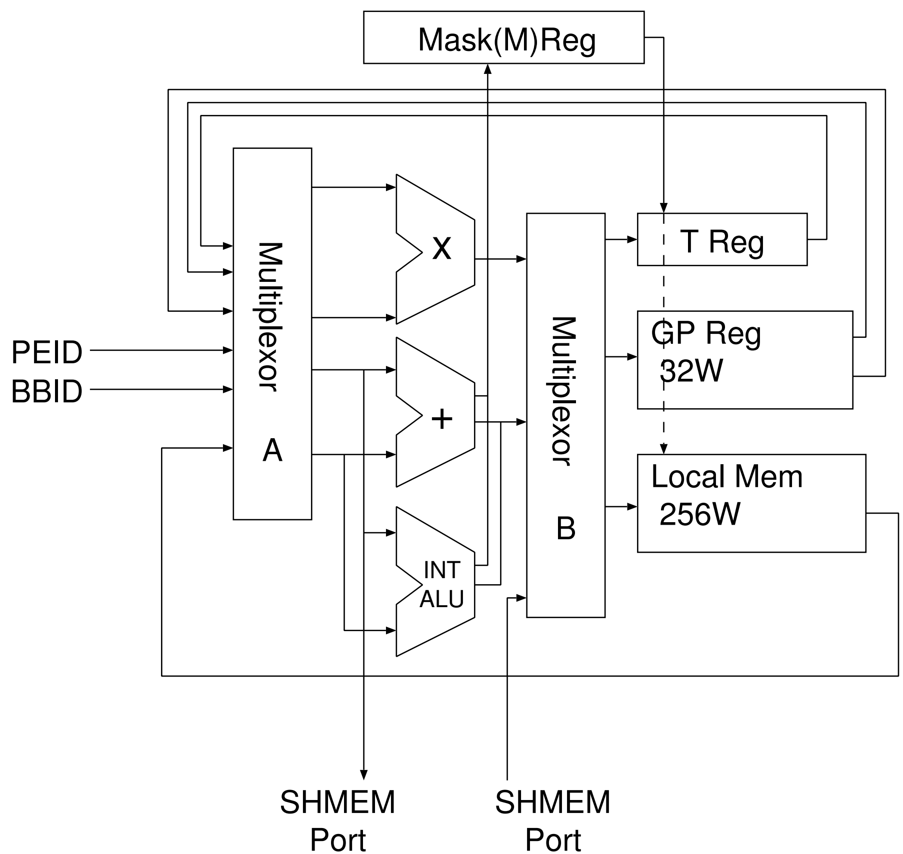
Why we changed the architecture?

- **To get budget (N-body problem is too narrow...)**
- **To allow wider range of applications**
  - **Molecular Dynamics**
  - **Boundary Element method**
  - **Dense matrix computation (Linpack, TOP500!)**
  - **SPH**
- **To allow wider range of algorithm**
  - **FMM**
  - **Ahmad-Cohen**
- **To try something new.**

How do you use it?

- **GRAPE: We'll write the necessary software. Move from GRAPE-6 will be less painful than move from GRAPE-4 to GRAPE-6.**
- **Matrix etc ... RIKEN/NAOJ will do something**
- **New applications:**
  - **Compiler will** *someday* **be provided**
  - **In the meantime, you need to write the kernel code in assembly language**

PE architecture



- **Float Mult (24 bit mantissa, with full 49 bit output)**
- **Float add/sub (60 bit mantissa)**
- **Integer ALU (72 bit)**
- **32-word (72 bit) general-purpose register file**
- **256-word (72 bit) memory**
- **ports to shared memory (shared by 32 processors)**

How do you really use it?

Machine language: 110 bits horizontal microcode

```
DUM
DUM ISP data test
DUM
DUM l m m m t t t t r r  r r r  r r r  r r r l l  l l l l f f f f f f f f f f f f f f f f f f f  i i f b  b b
DUM l _ _ _ _ _ _ _ _ _  _ _ _  _ _ _  _ _ _ _ _  _ _ _ _ m m m m m m m m m m m m a a a a a a a  a a s m  m m
DUM : i o i w l s i w i  w w w  r r r  r r r w i  a a t w u u u u u u u u u u u u d d d d d d d  l l e _  _ _
DUM : m m f r m h s r s  a a w  a a w  a a w r s  d d r l l l l l l l l l l l l l d d d d d d d  u u l w  a p
DUM : r r s i a o e i e  d d l  d d l  d d l i e  r r e : _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _  _ _ : r  d e
DUM : : : e t d r l t l  r r :  r r a  r r b t l  : i g : s s r n s s r n r n i i n n s r n i i  i u : i  r a
DUM : : : l e r t : e :  : i :  a i :  b i : e :  : : a : h h o o h h o o o o s s o o i o o s s  a n : t  : d
DUM : : : : : : s : : :  : : :  : a :  : b : : :  : : d : i i u r i i u r u r e e r r g u r e e  l s : e  : r
DUM : : : : : : t : : :  : : :  : : :  : : : : :  : : r : f f n m f f n m n m l l m m n n m l l  u i : :  : :
DUM : : : : : : o : : :  : : :  : : :  : : : : :  : : : : t t d a t t d a d a a b a a b d a a b  o g : :  : :
DUM : : : : : : p : : :  : : :  : : :  : : : : :  : : : : 2 5 a l 2 5 b l : l : : l l : : l : :  p n : :  : :
DUM : : : : : : : : : :  : : :  : : :  : : : : :  : : : : 5 0 : a 5 0 : b : o : : a b : : o : :  : e : :  : :
DUM : : : : : : : : : :  : : :  : : :  : : : : :  : : : : a a : : b b : : : : : : : : : : : : :  : d : :  : :
ISP 1 0 0 0 0 0 0 0 1 1  0 0 1  0 0 0  0 0 0 0 2  2 1 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 4 3  A 0 0 0  0 0
ISP 1 0 0 0 0 0 0 0 1 1  0 1 1  0 1 1  0 0 0 0 2  2 1 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0  2 0 0 0  0 0
ISP 1 0 0 0 0 0 0 0 1 1  2 0 1  0 0 0  0 0 0 0 2  2 1 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 5 3  A 0 0 0  0 0
ISP 1 0 0 0 0 0 0 0 1 1  4 1 1  0 1 1  2 1 1 0 2  0 1 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 3  0 0 0 0  0 0
ISP 1 0 0 0 0 0 0 0 0 0  0 0 1  4 1 1  0 0 0 0 0  0 1 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 3  A 0 0 1  0 0
DUM
DUM IDP header format: IDP len addr bbn bbnmask, all in hex
DUM RRN format
DUM ADDR N BBADR REDUC WL FSEL NA NB SB RND NO OP UN ODP SREGEN
IDP 1 1000  0 0
RRN 0    1  0     0    1   0   0  0   0  0  0  0   0  1    1
IDP 1 1000  0 0
RRN 0    1  0     1    1   0   0  0   0  0  0  0   0  1    1
```

Assembly language

```
 var vector long xi     hlt  flt64to72
 var vector long yi     hlt  flt64to72
 var vector long zi     hlt  flt64to72
 var vector short idxi  hlt  fix32to36ru
...
 bm vxj $lr0v
 vlen 1
 bm mj lmj
 bm eps2 leps2
 bm idxj lidxj
 nop
 upassa idxi  idxi  $t
 moi 1
 uxor  $ti lidxj $r8v
 moi 0
 upassa il"0" $t $t
 mi 1
 upassa il"1" $ti $t
 mi 0
 moi 2
 upassa $ti $ti $t
 moi 0
 nop
 fsub $lr0 xi $r6v  $t
 fsub $lr2 yi $r10v ;  fmul $ti $ti $t
 fsub $lr4 zi $r14v
 fmul $r10v $r10v $r18v ; fadd $t leps2 $t
 fmul $r14v $r14v ;  fadd $fb $ti  $t
 fadd $fb $ti  $r18v $t
...
```

High-level architecture

- **Single card: 4 chips, PCI-X/PCI-E/Hypertransport(? interface, 2 Tflops.**
- **Host network: 512 node, fast GbE or 10GbE switch**

Difference from GRAPE-6:

- **No custom network**
- **No large card**

Development schedule

2005 Spring	**Chip logical design**
2005 Fall	**Chip physical design**
2006 Fall	**First sample chip**
2007 Spring	**Prototype board**
2008 Spring	**Large parallel system**

Summary

- GRAPE project has successfully developed very high performance computers for astrophysical particle-based simulations.
- The next machine, GRAPE-DR, will have wider application range than traditional GRAPEs