Exascale computers ?

Jun Makino
ELSI, Tokyo Institute of Technology
RIKEN Advanced Institute for Computational Science

Science as Method and Methodology for Problems on the Earth and Life,
Sept 15-17, Nagoya

Structure of the talk

- Advance of the Supercomputers: 1950-2010
- Problems we see today
  - Power consumption
  - Parallelization overhead
  - How you develop/maintain codes???
- Solutions?
- Japanese Exascale Project

Advance of the Supercomputers

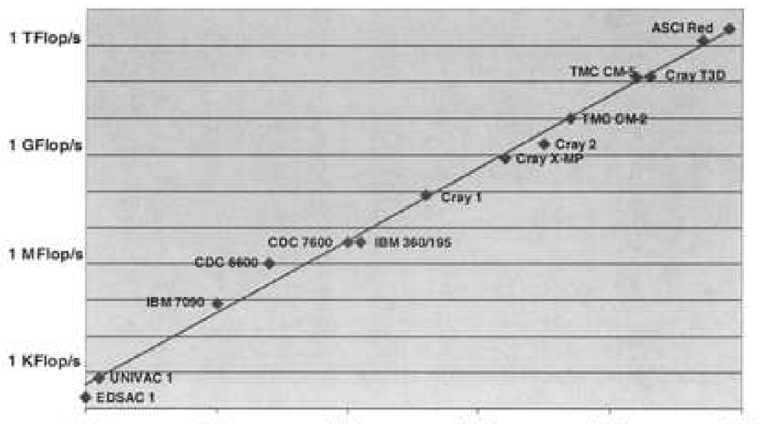
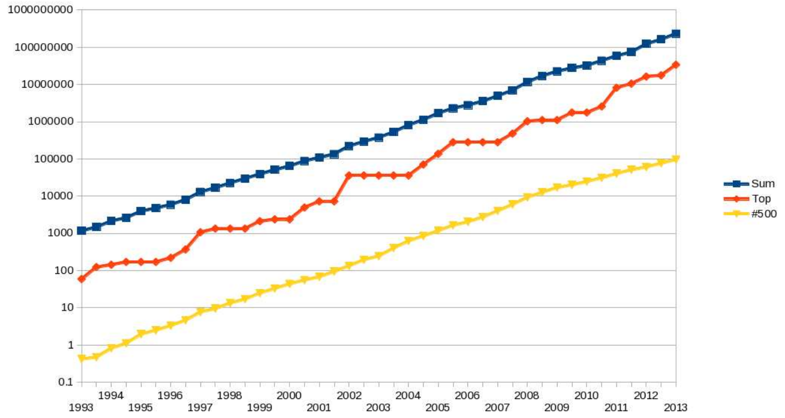


Figure 1. *Moore's law and peak performance of various "supercomputers" over time.*

1940-2000: 100 times per decade

Advance of the Supercomputers

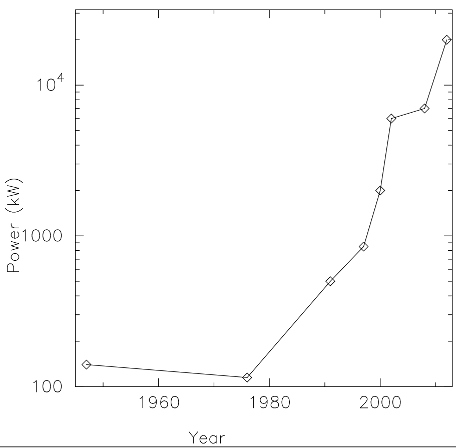


1993-2013: 500 times per decade(!?)

Problem 1: Power consumption

ENIAC	1947	140kW
Cray-1	1976	115kW
Cray C90	1991	500kW
ASCI Red	1997	850kW
ASCI White	2000	2MW
ES	2002	6MW
ORNL XT5	2008	7MW
K-computer	2012	20MW

If we plot data...



Little increase from ENIAC to Cray-1

Increase by a factor of 10 in 1975-95

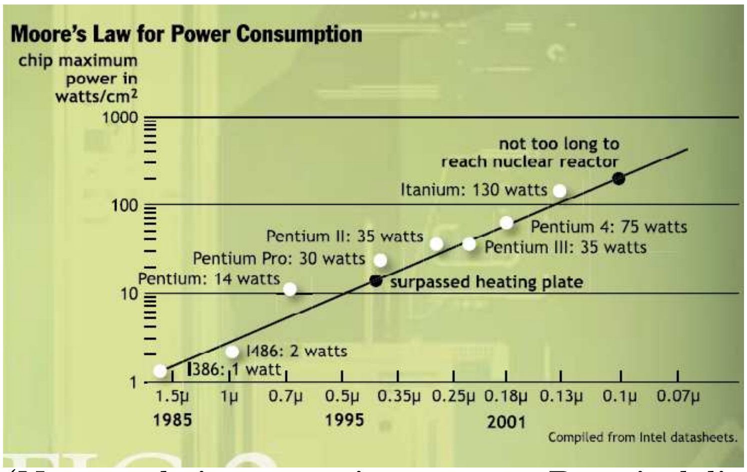
Factor of 30 in 1995-2012

Faster-than-exponential increase

Why?

- Price increased: ASCI Red: \$ 50M, K-computer: \$ 1G
- Power consumption per chip (or per cm^2 of silicon) increased
- Price per chip (or per cm^2 of silicon) decreased

Power consumption per cm^2 of silicon



(Not much increase since 2003. Practical limit of cooling reached)

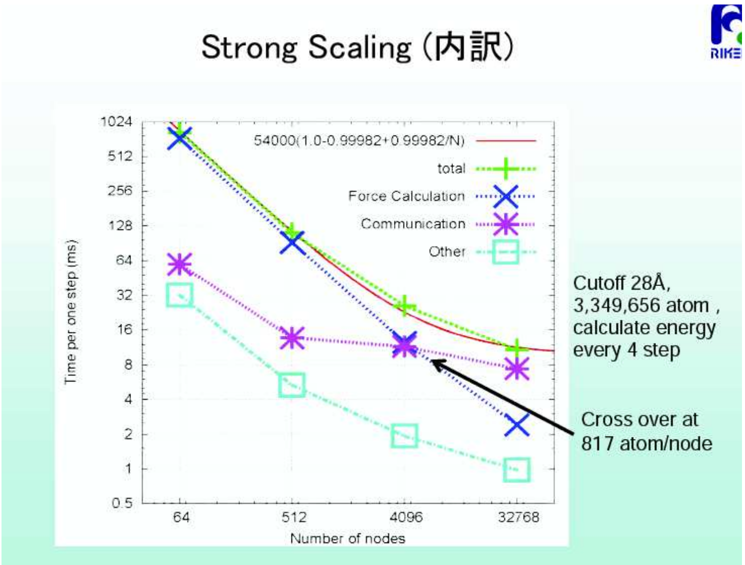
Problem 2: Parallelization overhead

Number of floating-point units (Multiply and add)

Cray-1	1976	1
Cray C90	1991	16
ASCI White	2000	16,384
ES	2002	40,960
K computer	2012	2,820,096

K computer is good for large problems (with small number of timesteps) but not so good for problems that require large number of timesteps.

Example of performance scaling



Molecular Dynamics on K computer

- One cannot go below 5ms/timestep
- Limitation: communication overhead

Is 5ms/step fast enough?

- Yes — for cosmology or other really large-N calculations with small number of timesteps
- No — for problems that require long simulation time (like planet formation...)

Very roughly speaking, integration of 10Myrs would take 1 year...

Problem 3: How you develop/maintain codes???

- MPI
- OpenMP
- SIMD extensions
- Cache-friendly code
- Accelerators
- ...
- ...

Problem 3: How you develop/maintain codes???

- MPI
- OpenMP
- SIMD extensions
- Cache-friendly code
- Accelerators
- ...
- ...

(I'll not discuss this aspect much...)

Solutions?

- We need to reduce power consumption AND communication overhead.
- We do not need much memory (1TB would be enough to keep 10^{10} particles)

Possible solution:

- Processors with "small" on-chip memory (small means 256MB or more)
- Large number of cores, but in SIMD mode to reduce communication overhead

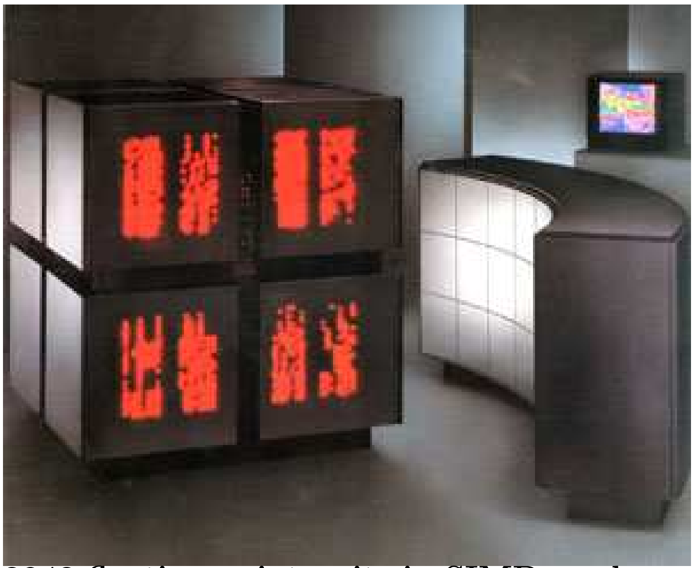
Massively-parallel SIMD machines

— A lost technology —

- Goodyear MPP (1970s)
- ICL DAP (Late 1970s)
- Thinking Machines Connection Machine-1/2 (Late 1980s)
- Maspar MP-1/2 (Early 1990s)

CM-2 was pretty successful

TMC CM-2



2048 floating point units in SIMD mode

TMC CM-2

- 64k 1-bit processors, each with 64k-bit memory
- 2048 floating-point units, each shared by 32 processors
- 12-dimensional hypercube network between processor chips (16 processors in one chip)

With the present-day technology, we can integrate 4-8 CM-2s into one chip, for the peak performance of 10-20 Tflops at < 100W

How we reduce power and communication overhead

- Power:
  - Minimize data movement: Remove external memory and cache
  - Minimize instruction fetch and decode: Massive SIMD
- Communication overhead:
  - Minimize data movement: Remove external memory and cache, reduce the number of chips
  - Reduce the handshake overhead: Cores in SIMD operation do not need handshake, since they are executing the same instruction

Japanese Exascale Project

NHK TV news reporting: Japan to develop new supercomputer with 100x power of K-computer

I was there as a member of a working group organized by the ministry of education

Current rough plan

- Follow-up of K-computer: would require 60-80 MW to reach exaflops in 2020
- Combine SIMD "accelerators" with MIMD general-purpose machine
- MIMD part: Fujitsu design
- SIMD part: Based on our design
  - reduce power consumption by 80%
  - reduce communication latency by at least a factor of 10

Summary

- Current big supercomputers are not ideal for long-term integration of “small” problems (“small” means 10^7 particles now and 10^9 particles in 2020)
- We need a new architecture (or revival of an old architecture...) to solve this problem: Massively-parallel SIMD
- If everything goes well, we will put this MP-SIMD system as part of Japanese Exascale project