Quantum Spin S=1/2 or Qubit

$$|\phi\rangle = c_\uparrow|1\rangle + c_\downarrow|0\rangle$$

$$\hat{S}^\alpha_j|\phi\rangle = c'_\uparrow|1\rangle + c'_\downarrow|0\rangle$$

$$\begin{pmatrix} c'_\uparrow \\ c'_\downarrow \end{pmatrix} = \frac{1}{2}\hat{\sigma}^\alpha \begin{pmatrix} c_\uparrow \\ c_\downarrow \end{pmatrix}$$

$$\hat{\sigma}^x = \begin{pmatrix} 0 & +1 \\ +1 & 0 \end{pmatrix}$$

$$\hat{\sigma}^y = \begin{pmatrix} 0 & -i \\ +i & 0 \end{pmatrix}$$

$$\hat{\sigma}^z = \begin{pmatrix} +1 & 0 \\ 0 & -1 \end{pmatrix}$$

$$\hat{S}^\alpha_j \doteq \frac{1}{2}\hat{\sigma}^\alpha$$

Quantum Spins: Two Site TFIM

Decimal representation of orthonormalized basis

		0 th site		1 st site
$\|0\rangle_{\mathrm{d}}$	$=$	$\|\downarrow\rangle$	$\otimes$	$\|\downarrow\rangle$
$\|1\rangle_{\mathrm{d}}$	$=$	$\|\uparrow\rangle$	$\otimes$	$\|\downarrow\rangle$
$\|2\rangle_{\mathrm{d}}$	$=$	$\|\downarrow\rangle$	$\otimes$	$\|\uparrow\rangle$
$\|3\rangle_{\mathrm{d}}$	$=$	$\|\uparrow\rangle$	$\otimes$	$\|\uparrow\rangle$

$$L = 2 \qquad \hat{H} = J \sum_{i=0}^{L-1} \hat{S}_i^z \hat{S}_{i+1}^z - \Gamma \sum_{i=0}^{L-1} \hat{S}_i^x$$

$$\hat{H} \doteq \begin{pmatrix} +J/2 & -\Gamma/2 & -\Gamma/2 & 0 \\ -\Gamma/2 & -J/2 & 0 & -\Gamma/2 \\ -\Gamma/2 & 0 & -J/2 & -\Gamma/2 \\ 0 & -\Gamma/2 & -\Gamma/2 & +J/2 \end{pmatrix} \quad {}_{\mathrm{d}}\langle i|\hat{H}|j\rangle_{\mathrm{d}}$$

Larger TFIM

$$\hat{H} = J \sum_{i=0}^{L-1} \hat{S}_i^z \hat{S}_{i+1}^z - \Gamma \sum_{i=0}^{L-1} \hat{S}_i^x$$

-Non-commutative

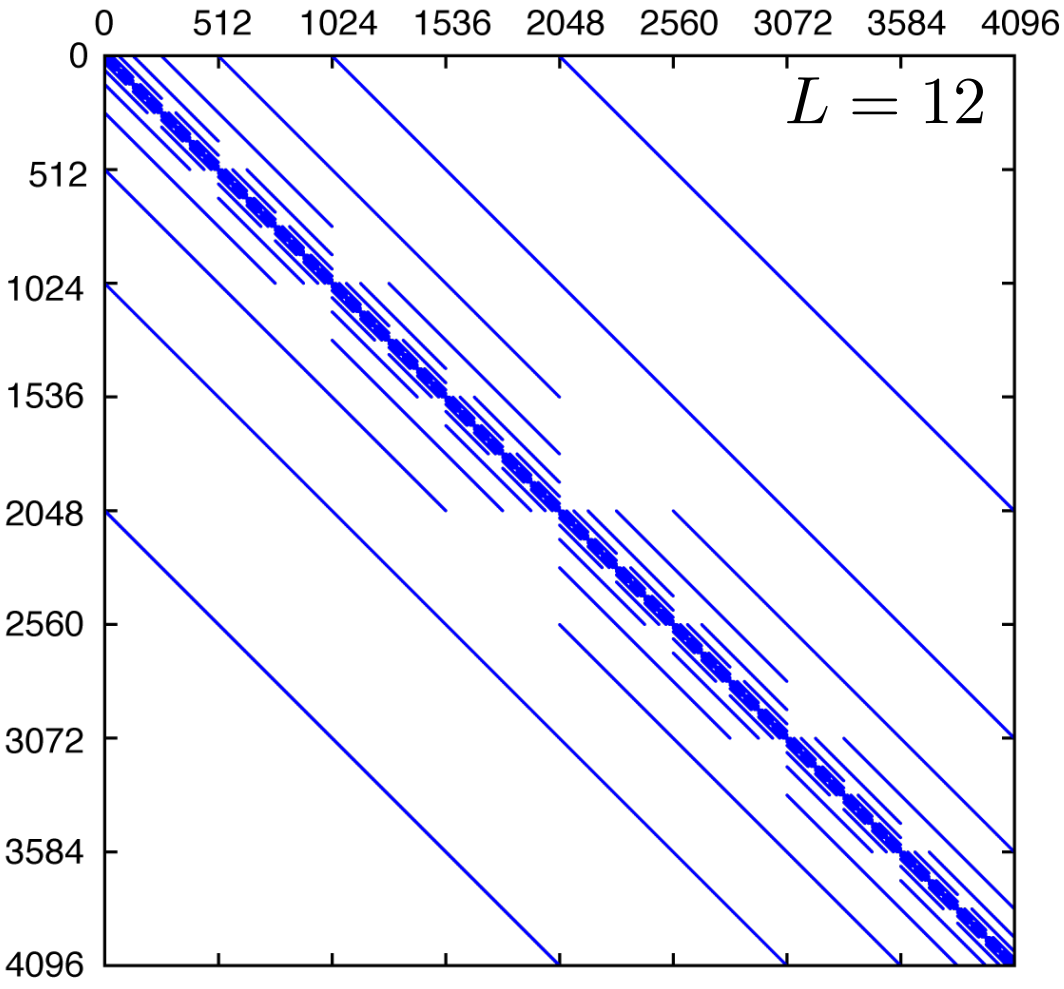
$$\left[\sum_{i=0}^{L-1} \hat{S}_i^z \hat{S}_{i+1}^z, \sum_{i=0}^{L-1} \hat{S}_i^x \right] \neq 0$$

→Quantum fluctuations
or Zero point motion

-Sparse
# of elements ∝ $O(2^L)$

-Solvable

-Hierarchical matrix?



Eigenvalue Problems

Diagonalizing Hermitian matrices

Standard approach:
Call LAPACK (Linear Algebra PACKage)

subroutine zheev:
Householder reflection + QR algorithm
$\rightarrow O(N^3)$ numerical cost

Difficulties in Many-Body Problems

-Longer simulation

-Summation over exponetially large
# of configurations

←Monte Carlo (2nd to 9th lecture), …

-$O(N^3)$ numerical cost and $O(N^2)$ memory

←Krylov subspace method (11th to 12th), …

Computer

Supercomputers in UTokyo

Oakbridge-CX

Intel Xeon Platinum 8280
1,368 nodes

Wisteria/BDEC-01

Odyssey: A64FX 7,680 nodes
Aquarius: Intel Xeon + NVIDIA A100
45 nodes

A single CascadeLake: 4.8384 TFLOPS
A single A64FX: 3.3792 TFLOPS
A single A100: 19.5 TFLOPS

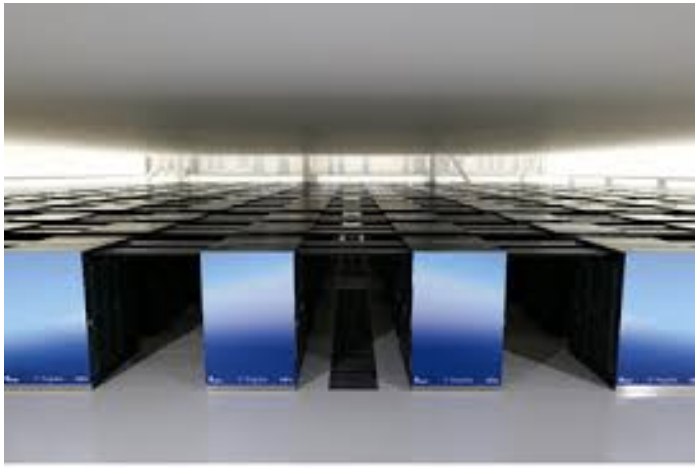
Ohtaka (CPU server)

ISSP, UTokyo@Kashiwa
Theoretical Peak: 6,881TFlop/s
AMD EPYC 7720 2.0GHz, 64 cores x 2
Memory: 258GiB
1,680 nodes

東京大学 物性研究所
THE INSTITUTE FOR SOLID STATE PHYSICS
THE UNIVERSITY OF TOKYO

“Fugaku” Supercomputer

Fugaku 富岳 @RIKEN, Kobe
-158,976 nodes
A64FX
48 cores/node
32 GiB/node



- 1st in Top500 & 25th in Green500 on November 2021

How Computer Handles Numbers

Numbers in Computers

Integer

Integer(4): 31bit+1bit

$$-2^{31} \leq j \leq 2^{31} - 1 \ (j \in \mathbb{Z})$$

Real number

IEEE Std 754-2008 binary64
(double-precision floating-point number)

Double precision/real(8):
Sign (1 bit) + Exponent (11 bit) + Significand (52 bit)

$$(-1)^s \times 2^e \times m$$

$$-1022 \leq e \leq 1023 \ (e \in \mathbb{Z}) \qquad m = \frac{\sum_{\ell=0}^{p-1} d_\ell \cdot 2^\ell}{2^p} \qquad p = 53$$

Performance of computer

How many times does the computer multiply/add per second?

How much data does the computer memorize?

How much data does the computer read/write per second?

FLOP/s

Floating-point Operations Per Second

An example: Intel Xeon Phi Nights Landing
Intel Xeon Phi 7250 (1.4GHz, 68 cores, 112GB)

-1.4×10^9 instructions per second
(instruction to perform double precision add or multiply)

-Intel AVX-512 instruction*
double precision floating point number (8byte=64bit)
8 double-precision multiply-add** operations
$\rightarrow 1.4 \times 10^9 \times 16 \times 68$ FLOP/s per processor

*Other instruction set:
SVE (for example, A64FX of Fugaku)
**Multiply-add: $a \leftarrow a + (b \times c)$

Increasing FLOP/s

Clock rate saturated

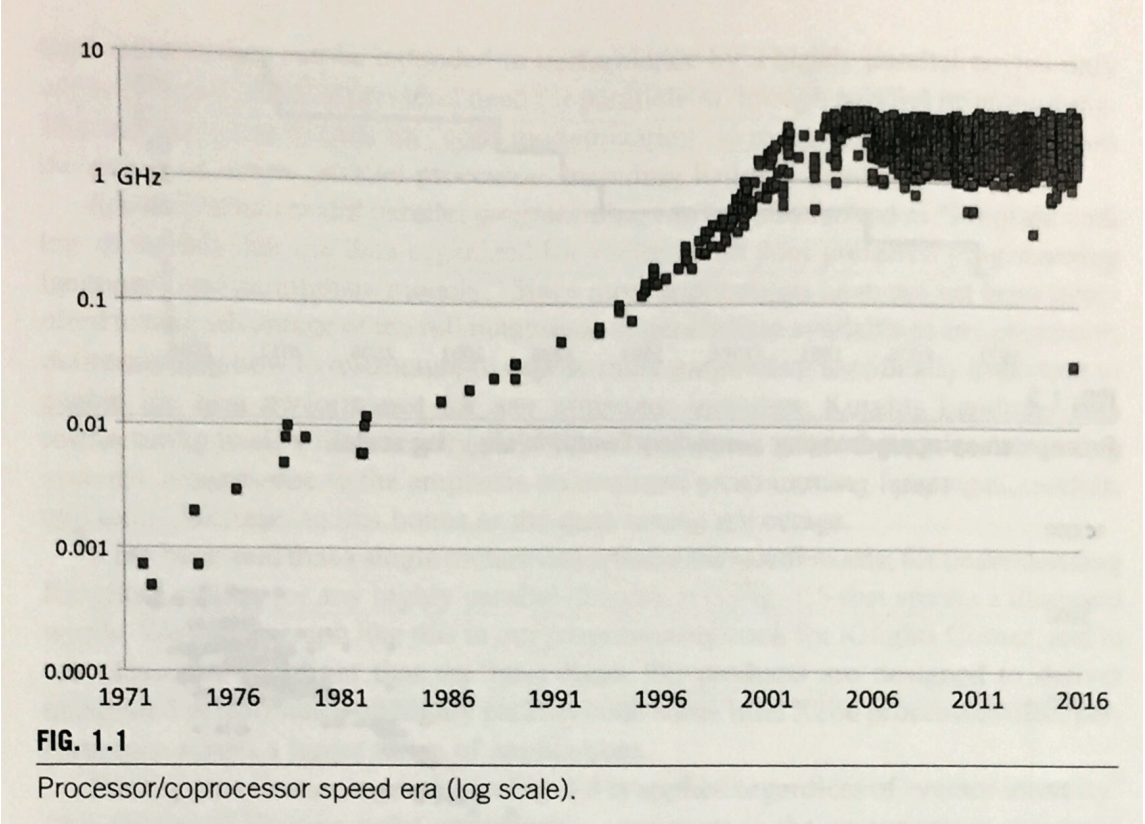


FIG. 1.1

Processor/coprocessor speed era (log scale).

J. Jeffers, J. Reinders, and A. Sodani,
Intel Xeon Phi Processor High Performance Programing

Increasing FLOP/s

Number of cores

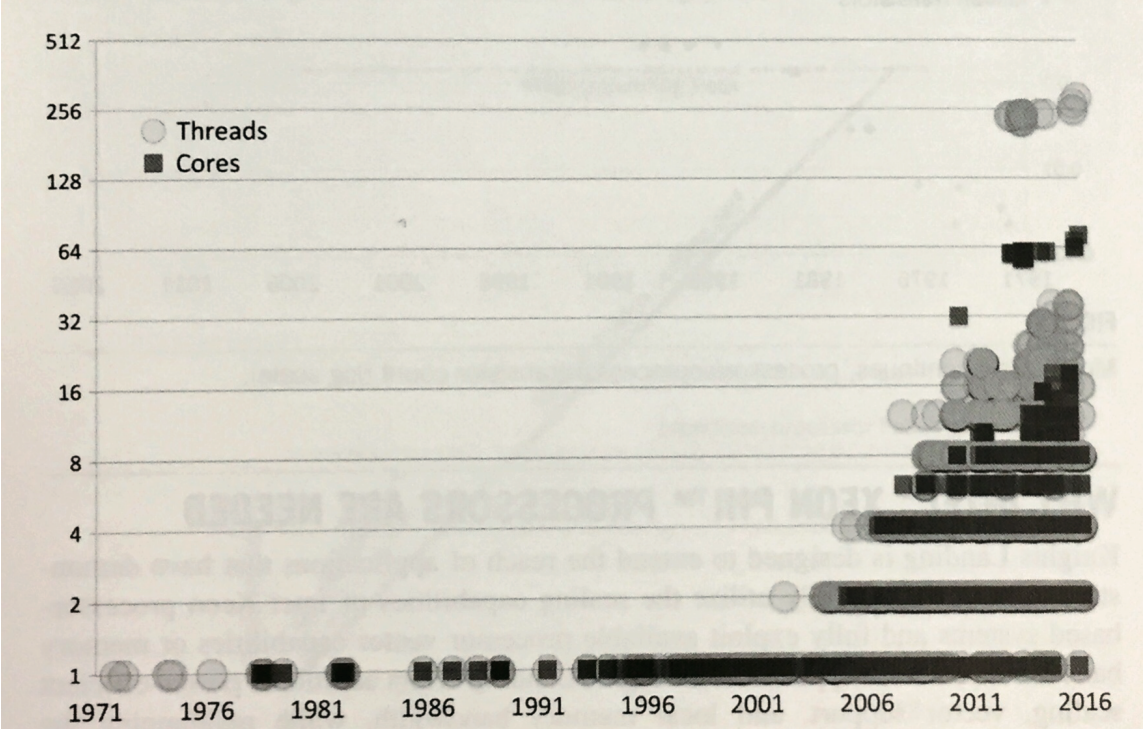


FIG. 1.2

Processor/coprocessor core/thread parallelism (log scale).

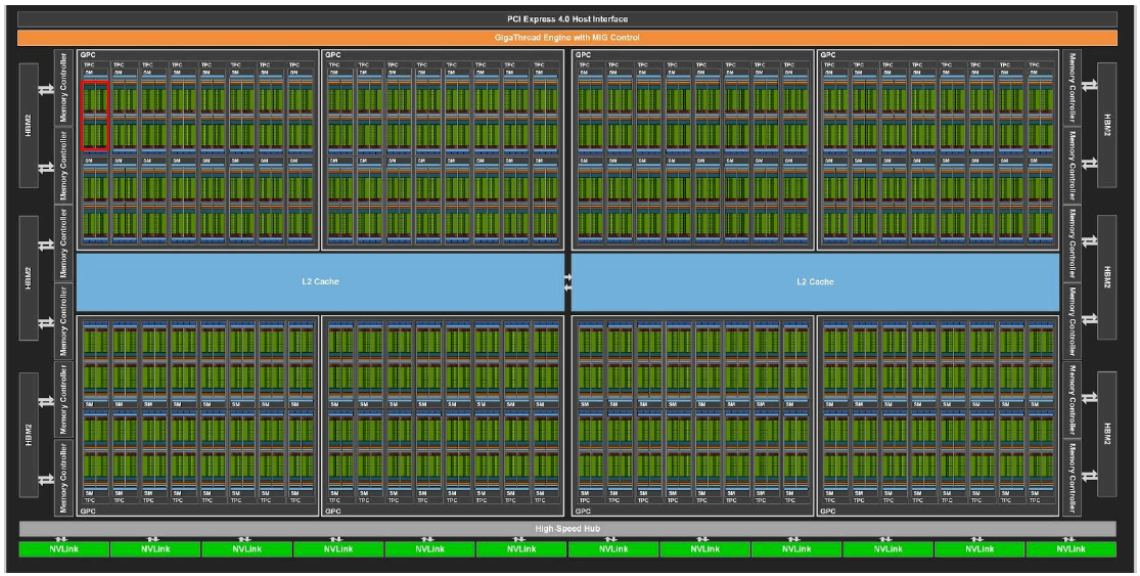
J. Jeffers, J. Reinders, and A. Sodani,
Intel Xeon Phi Processor High Performance Programing

Increasing FLOP/s: GPGPU

NVIDIA Ampere architecture whitepaper

64 FP32 CUDA cores
32 FP64 CUDA cores
in a unit (SM)

128 SM (108 SM in A100)

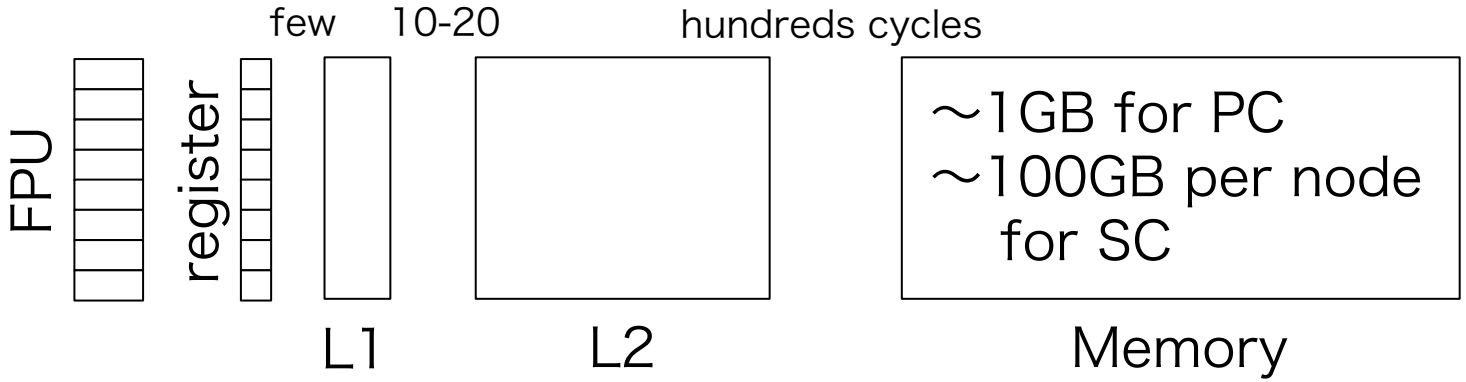


A single CascadeLake: 4.8384 TFLOPS
A single A64FX: 3.3792 TFLOPS
A single A100: 19.5 TFLOPS

Performance of computer

How many times does the computer multiply/add per second?
How much data does the computer memorize?
How much data does the computer read/write per second?

Cache, Memory, and Disk: CPU



Hierarchy:	Intel Xeon Phi 7250
Register	
L1 Cache	32KB per core
L2 Cache	512KB per core
Memory	~1.6GB per core (~10GB/s per core)
Disk	
Network	

Byte per flop (B/F)

Computational and Memory Costs

1. Vector-vector product $\sum_{j=0}^{N_{\mathrm{H}}-1} u_j^* v_j$

   Computational: $\mathcal{O}(N_{\mathrm{H}})$
   Memory: $\mathcal{O}(N_{\mathrm{H}})$

2. Matrix-vector product $v_i = \sum_{j=0}^{N_{\mathrm{H}}-1} A_{ij} u_j$

   Computational: $\mathcal{O}(N_{\mathrm{H}}^2)$
   Memory: $\mathcal{O}(N_{\mathrm{H}}^2)$

3. Matrix-matrix product $C_{ij} = \sum_{k=0}^{N_{\mathrm{H}}-1} A_{ik} B_{kj}$

   Computational: $\mathcal{O}(N_{\mathrm{H}}^3)$
   Memory: $\mathcal{O}(N_{\mathrm{H}}^2)$

Exercises (Not report problems)

1. Status of N qubits is represented as
a complex vector in 2^N-dimensional Hilbert space.
If you can use whole system of Ohtaka,
how many qubits can you store in the memory?

2. Usually, B/F of modern supercomputers
is less than 1.
Which kind of calculations is suitable for them?

Method of Evaluation

Based on 2 Reports:
-Exercise about algorithms
-Exercise of computer simulation

Closely Related Lectures

- 計算科学概論 (for undergraduate)
  Mon. 3rd period

- (archive/seminar)計算科学科学技術特論A,B,C

Lecture Schedule

#1 Many-body problems in physics and why they are hard to solve

Classical

#2 Classical statistical model and numerical simulation
#3 Classical Monte Carlo method
#4 Applications of classical Monte Carlo method
#5 Molecular dynamics and its application
#6 Extended ensemble method for Monte Carlo methods

Quantum

#7 Quantum lattice models and numerical approaches
#8 Quantum Monte Carlo methods
#9 Applications of quantum Monte Carlo methods
#10 Linear algebra of large and sparse matrices for quantum many-body problems
#11 Krylov subspace methods and their applications to quantum many-body problems
#12 Large sparse matrices and quantum statistical mechanics
#13 Parallelization for many-body problems