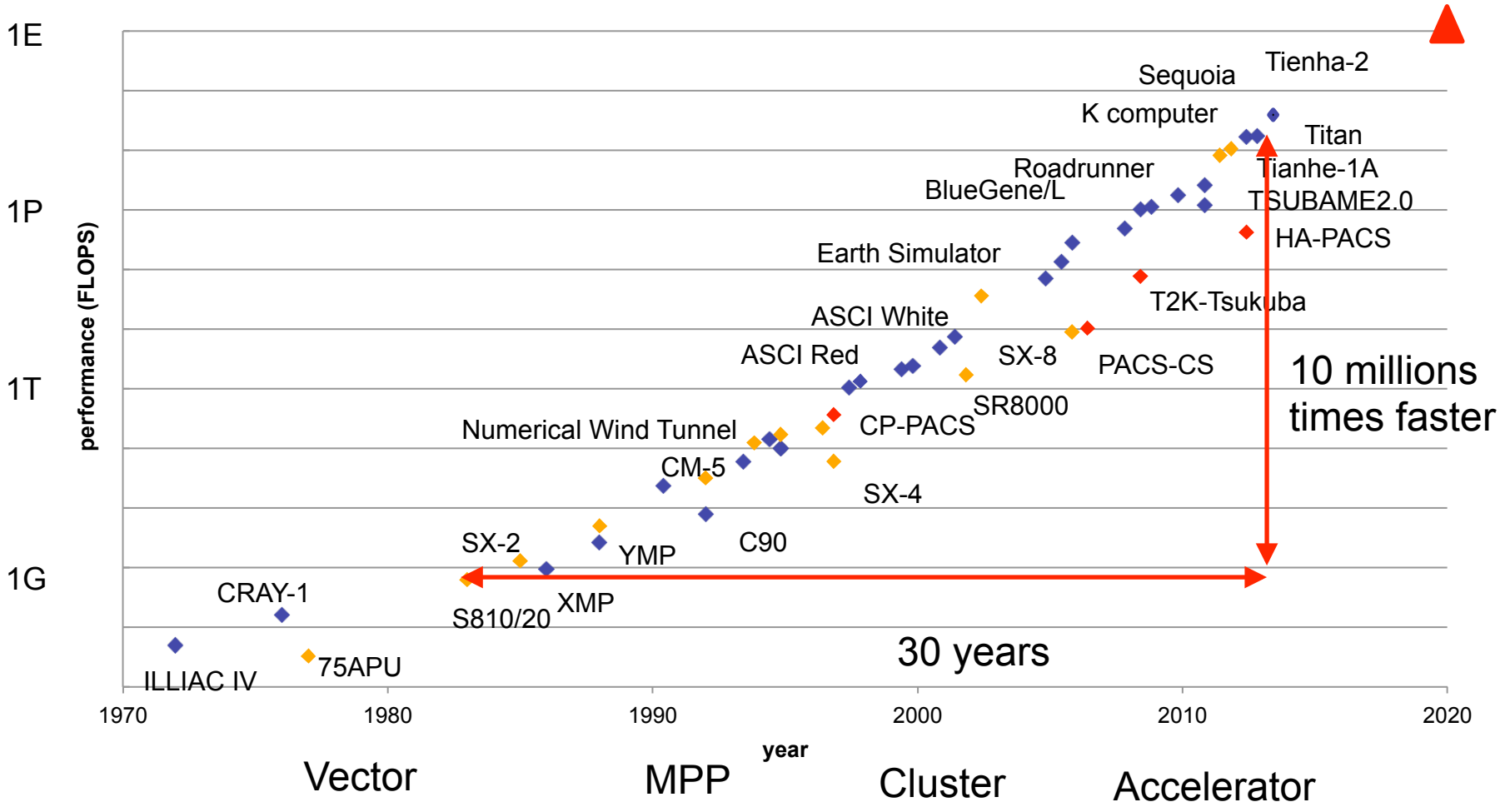
Korea-Japan HPC Winter School 2016
“Parallel Systems”

Taisuke BOKU
taisuke@cs.tsukuba.ac.jp
Center for Computational Sciences
University of Tsukuba

Outline

- History of parallel systems
- Architecture of parallel systems
- Interconnection Network of parallel systems
- Overview of Supercomputers

History of supercomputer

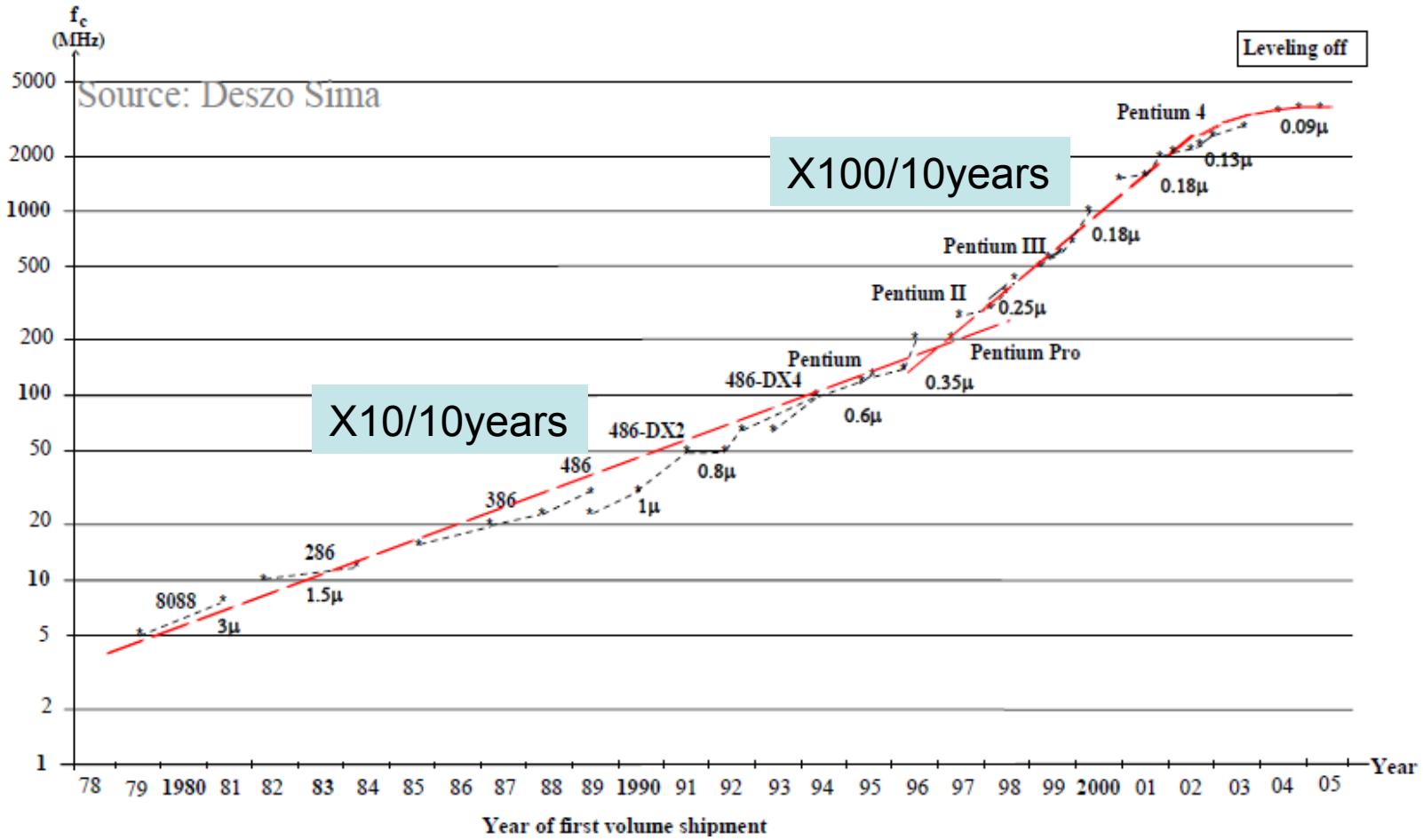


Advances in parallel computer
-- processor level --

- Vector processor ⇒ many supercomputer used vector processors in 20 years ago
  - Single processor can calculate array processing in high speed
- Scalar processor
  x86 (IA32), Power, Itanium (IA64), Sparc

- Recent trend in processor
  - multi-core processor becomes popular
    - Intel & AMD ⇒ 8 ～12 core IA-32
  - many-core（8～512 cores）processor is available for accelerator
    - IBM Cell Broadband Engine (8 core)
    - ClearSpeed (96 core×2)
    - GPU (nVIDIA K20X 896 DP unit)
    - Intel Xeon Phi (60 core)

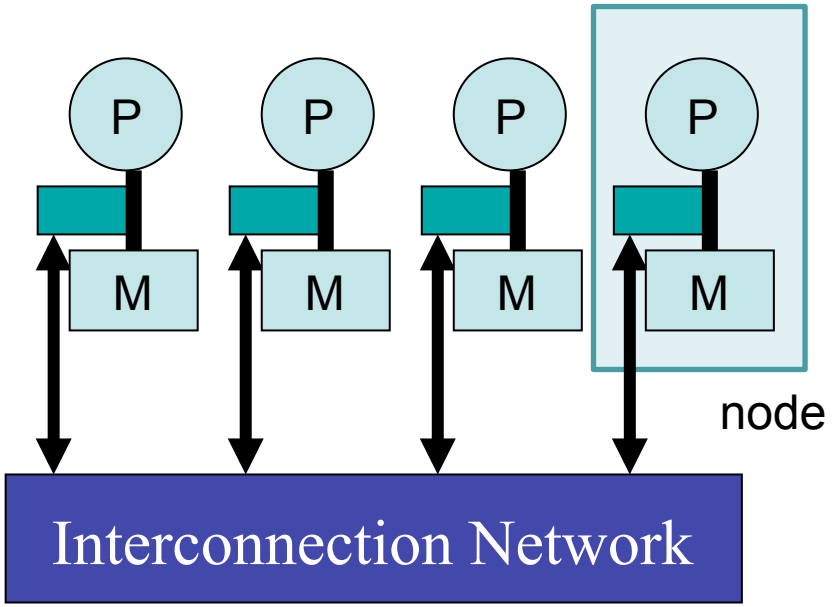
Clock speed of scalar processors



Memory architecture of parallel systems

- Distributed memory system
  Each processor has own memory that cannot be directly accessed by other processors, and accesses the remote data by message passing using interconnection network.
- Shared memory system
  Each processor has physically shared memory, and accesses the memory by normal load/store instructions.
- Hybrid memory system
  Combination of shared-memory system and distributed-memory system. In a node, it is shared-memory system using multi-core CPU, and between nodes, it is distributed-memory system using interconnection network.

Distributed-memory system (1)



Message passing
between any processors

P ... Processor
M ... Memory
NIC (network interface controller)

- Nodes are the unit of parallel systems, and each node is a complete computer system with CPU and memory. Nodes are connected by interconnection network via NIC.

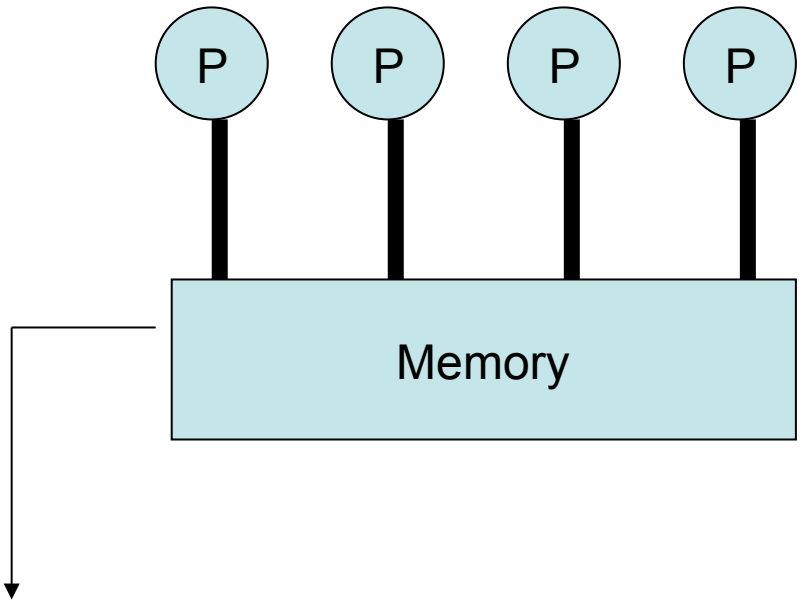
- A process runs on each node and communicates data (message) between nodes through network.

- Building system is easy and scalability is high.
  - ◆MPP：Massively Parallel Processor
  - ◆Cluster computer

Distributed-memory system (2)

- Program on each node should communicate to other nodes explicitly by message passing, user programming is complicated.
  - MPI (Message Passing Interface) is a standard tool to communication
  - Typical style of parallel application is relatively easy to write a program such as data parallel of domain decomposition or master/worker processing
- System performance depends on performance of interconnection network as well as performance of processor and memory.
- Typical implementation of MPP in late 1980, and also basic architecture of current PC cluster

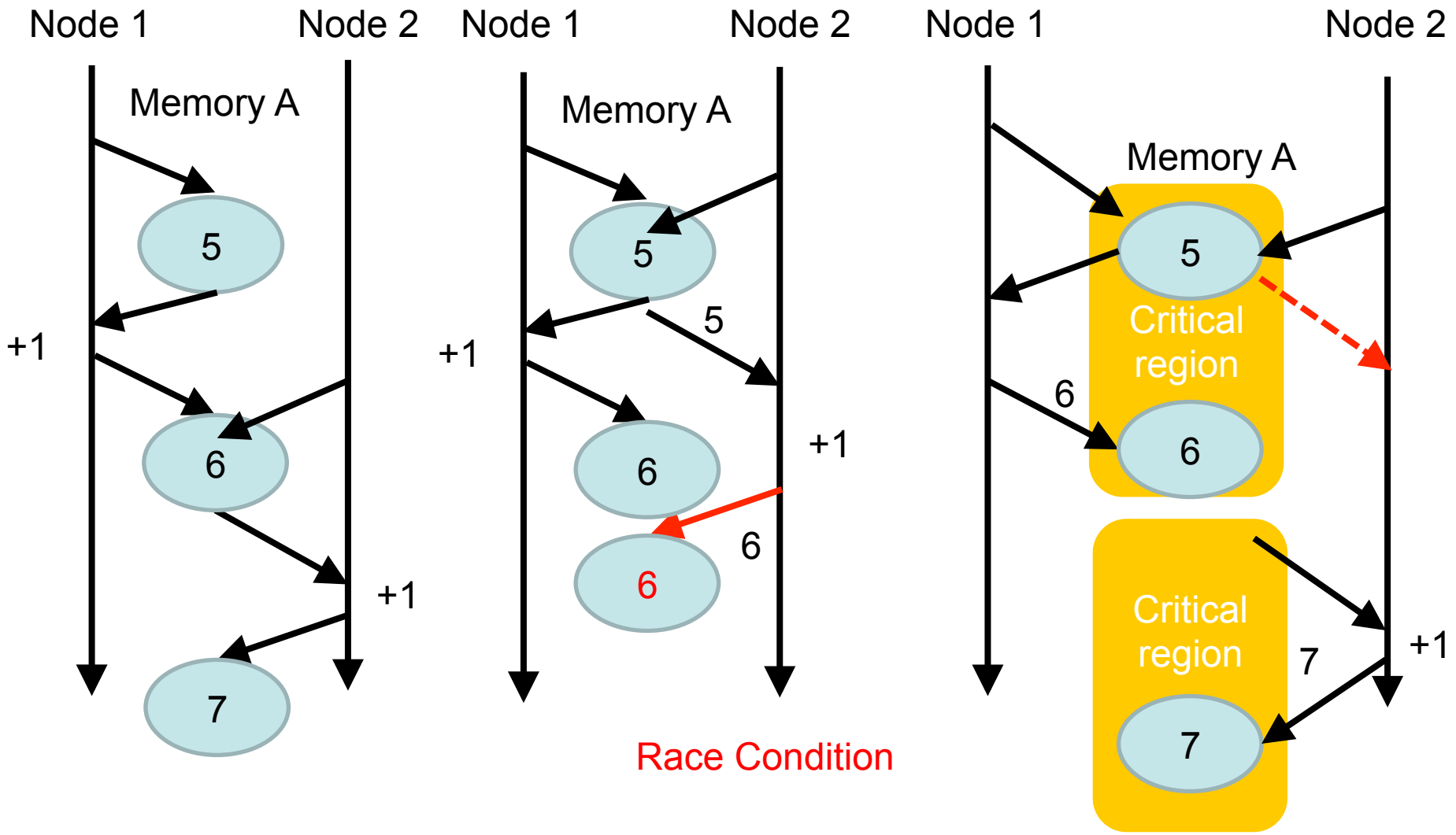
Shared-memory system (1)



Memory should arbitrate simultaneous requests from multiple processors.

- Multiple processor access the same memory
- Each program (thread) on processor accesses data on memory freely. In other words, it should be care of race condition for multiple updates.
- Small to medium scale server
- Recent multi-core processor is shared memory in a processor.
- Architecture is classified to SMP and NUMA.

Access conflict in shared memory

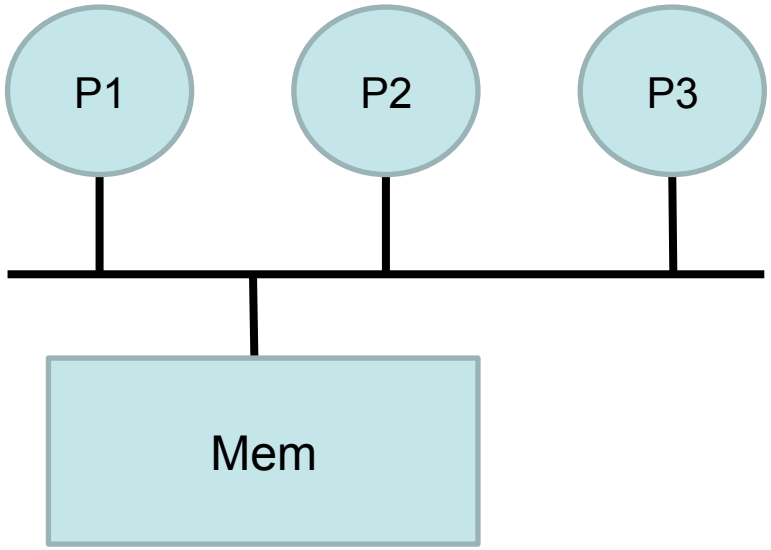


Shared-memory system (2)

- Programming on shared memory is easy for users.
  - Multithread programing model (POSIX thread, etc)
  - Programming tools based on shared memory (OpenMP: directive base programming)
- Shared-memory is simple as model, but hardware will be complicated to realize the model with high performance because “memory” is a quite primitive element of computer.
- Memory access becomes bottleneck when many processor access a location of memory
  - It is difficult to achieve scalability of system (about 100 processors will be a limit)

Architecture of shared memory

- Shared memory bus is the simplest shared memory system, but it cannot achieve scalability.
  - Shared bus was popular in PC cluster.
  - The Bus becomes bottleneck (bus is occupied by a transaction in a time)
  - To reduce the overhead of the bus conflict, each node has coherent cache.

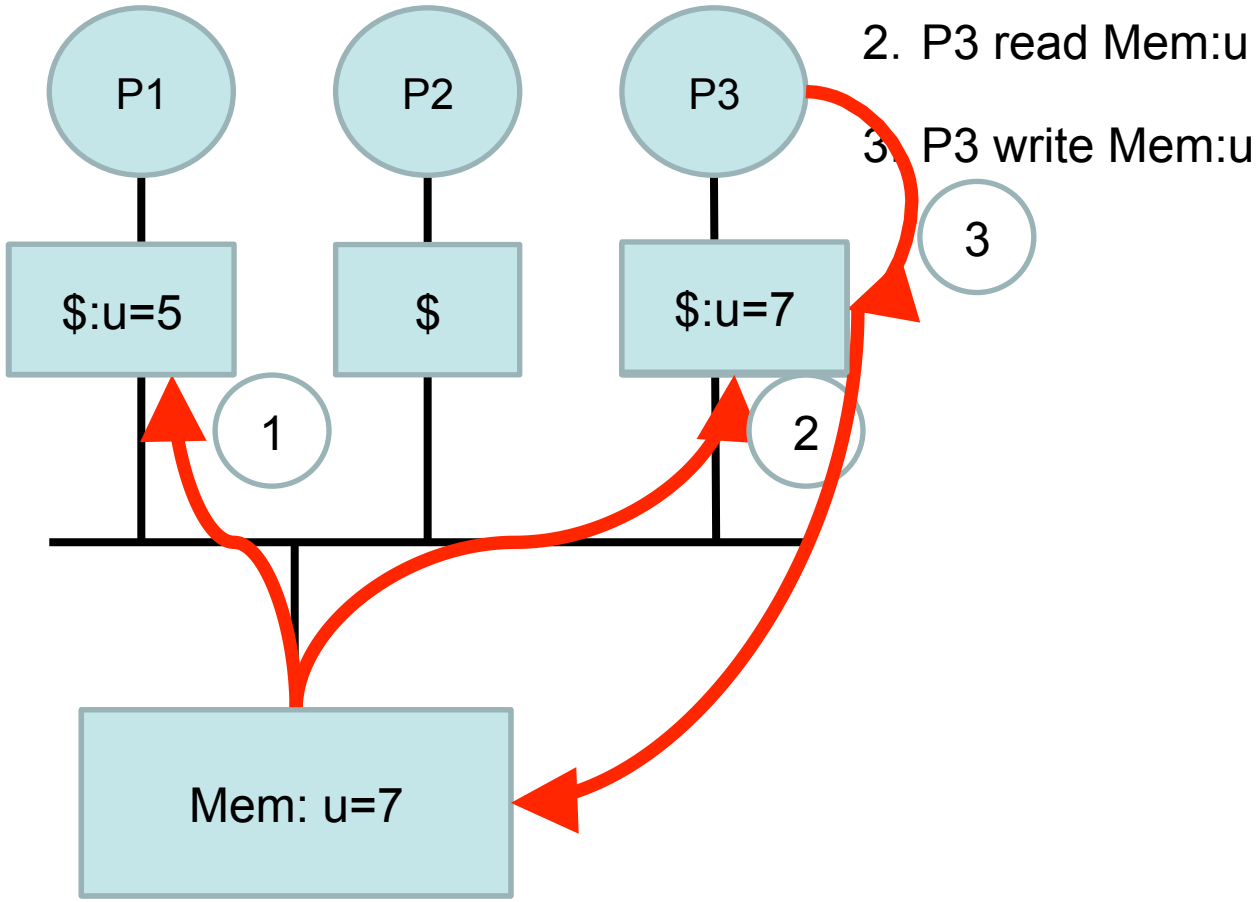


Non-coherent cache

1. P1 read Mem:u

4. P1 re-read Mem:u

Read old value u=5 from cache



2. P3 read Mem:u

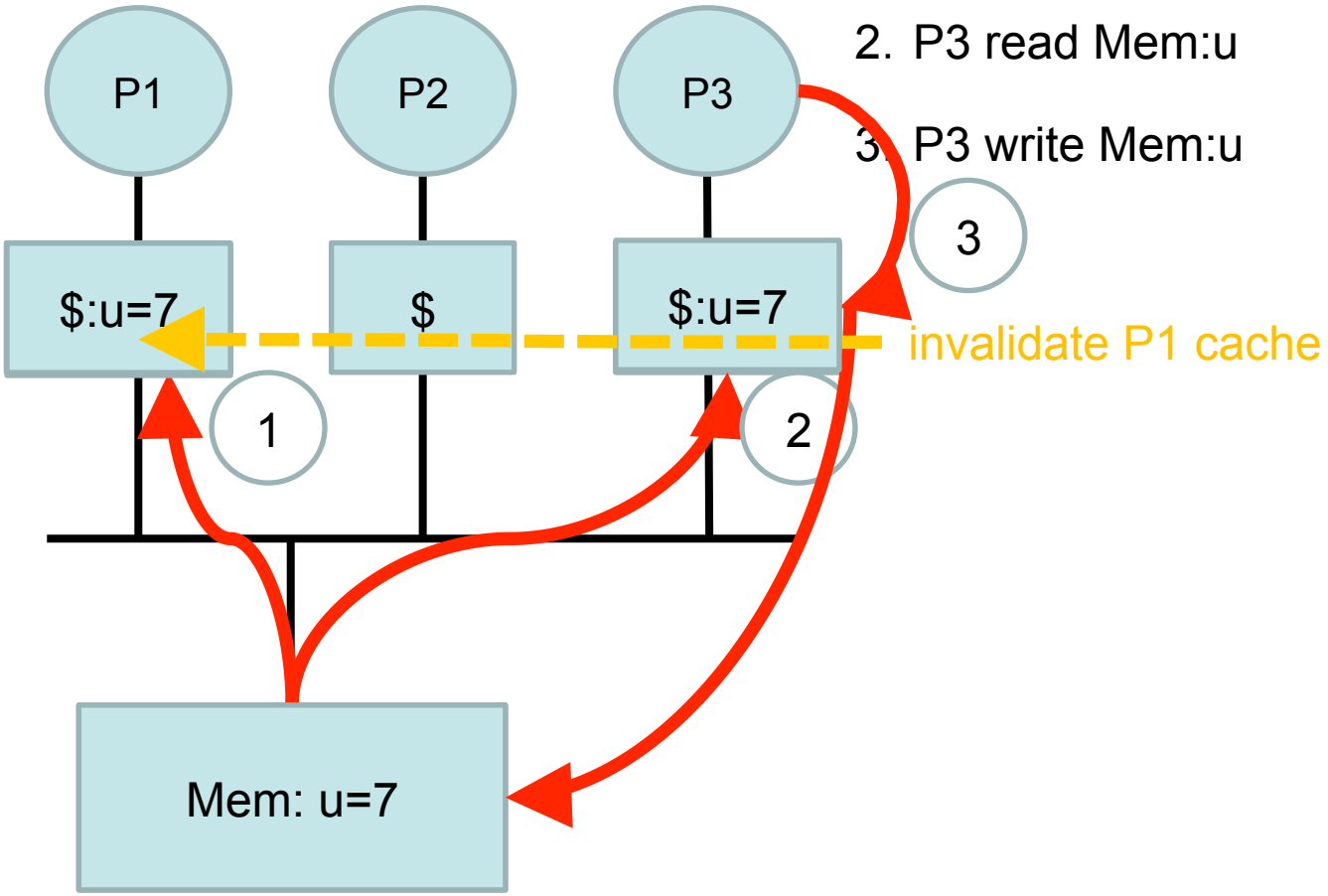
3. P3 write Mem:u

Coherent cache

1. P1 read Mem:u

4. P1 reread Mem:u

Read correct value from memory

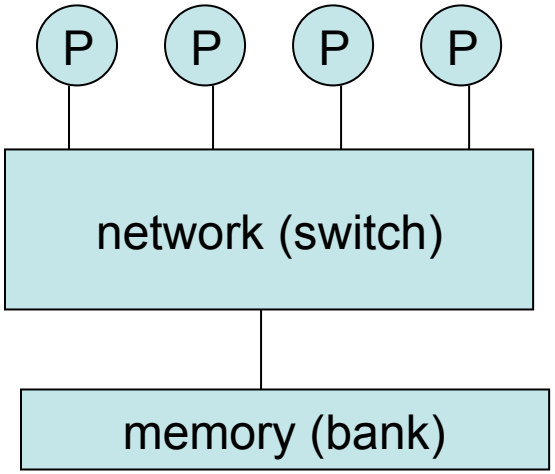


Architecture of shared memory

- How to avoid access conflict to shared memory
  - memory bank: address blocks are distributed to memory modules
  - split transaction: request and reply for memory are separated.
  - crossbar network : processors and memories are connected by switch not a bus.
  - coherent cache : each processor has own cache. If other processor update the memory, cache will be updated automatically.
  - NUMA (Non-Uniformed Memory Access) : memory modules are distributed, and the distance to each memory is different.

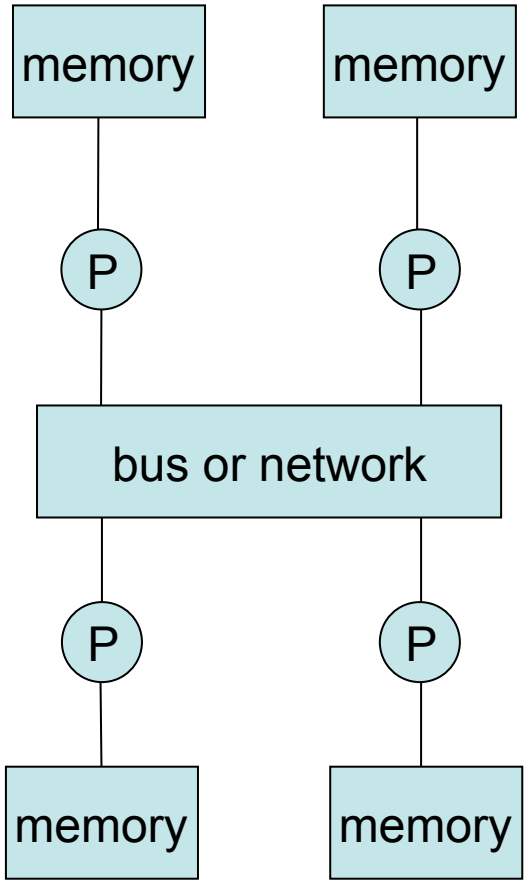
Symmetric Multi-Processor (SMP)

- The distance to any memory from a processor is same
- Shared bus or switches connect multiple processors and memory modules evenly.
- Multiprocessor node with previous Intel processor was SMP
- Large scale SMP system: Fujitsu HPC2500 and Hitachi SR16000
- Coherent cache is usually used
- Processor don’t have to consider the data location
- When memory access is concentrated, the performance is reduced

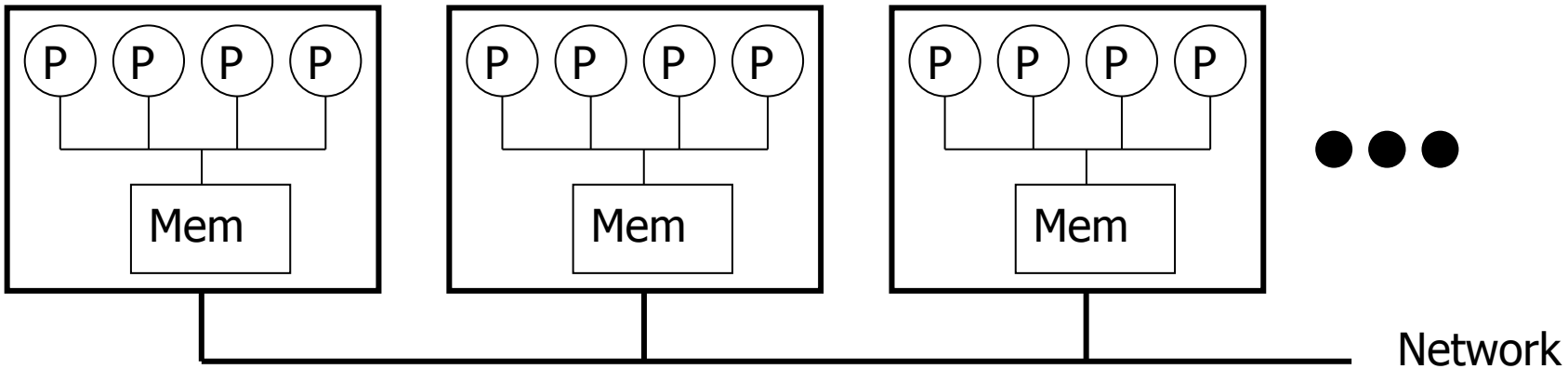


Non-Uniformed Memory Access (NUMA)

- A processor has own memory (local memory)
- Processor can access the memory of other processor (remote memory) via shared bus or network between processors
- It needs excessive time to access the remote memory（non-symmetric）
- AMD used NUMA from Opteron in 2003. Intel also uses NUMA from Nehalem in 2008.
- Large scale NUMA system: SGI Origin, Altix series.
- If data is distributed in each memory, and processor accesses to local memory, the memory performance will be increased (memory affinity)



Hybrid memory system



- Combination of shared memory and distributed memory
- Node itself is a shared memory multiprocessor system (SMP or NUMA).
- Each node is connected to other nodes with network, and access the remote memory with distributed memory.
- Hybrid system becomes popular because CPU becomes multi-core processor where each core is a shared memory architecture.

Parallel system with Accelerator

- Each node includes not only CPU but also accelerator that is a hardware to accelerate arithmetic operations.
  - GPU (Graphic Processing Unit)
    recently called GPGPU (General Purpose GPU), available general programming
  - FPGA (Field Programmable Gate Array)
    Reconfigurable hardware for specific purpose
  - General accelerator
    ClearSpeed, etc.
  - Processor itself is hybrid architecture with fat core and thin core
    CBE (Cell Broadband Engine) ⇒ LANL Roadrunner

MultiCore, ManyCore, GPU

	Multi Core	Many Core	GPU
Ex.	Intel Xeon E5-2670v2	Intel Xeon Phi 7110P	Nvidia K20X
# of cores	10	61	192x14=2688
Frequency	2.5 GHz	1.1GHz	732 MHz
Performance	200 GFLOPS	1.1 TFLOPS	1.3 TFLOPS
Memory Capacity	64 GB	8 GB	6 GB
Memory Bandwidth	60 GB/s	350 GB/s	250 GB/s
Power	115 W	300 W	235 W
Program	C, OpenMP, MPI	C, OpenMP, MPI	Cuda (C, Fortran)
Execution Model	MIMD	MIMD	STMD

Interconnection Network

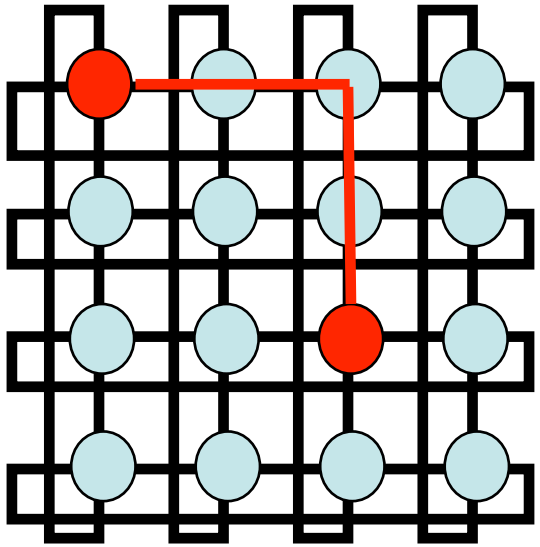
- Aim
  - Explicit data exchange on distributed memory architecture
  - Transfer data and control message on shared memory architecture
- Classification
  - static (direct) / dynamic (indirect)
  - diameter (distance)
  - degree (number of links)
- Performance metric
  - throughput
  - latency

Direct network

- All network nodes have processor (or memory) with multiple links connected to other nodes.
- In other words, direct connection between nodes, and no switches.
- Messages are routed on nodes.
- Typical topology of direct network
  - 2-D/3-D Mesh/Torus
  - Hypercube
  - Direct Tree

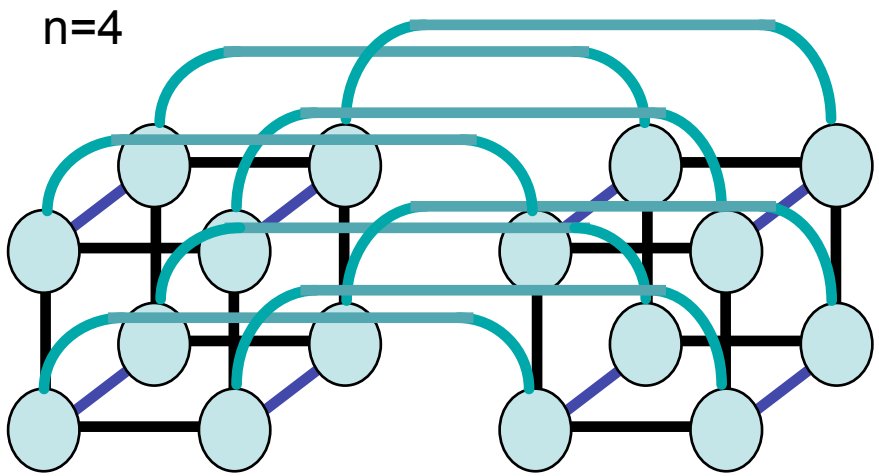
Examples of Direct network

Mesh/Torus (k-ary n-cube)



Cost: N (=k^n)
Diameter: n(k-1) in mesh
nk/2 in torus

Hypercube (n-cube)

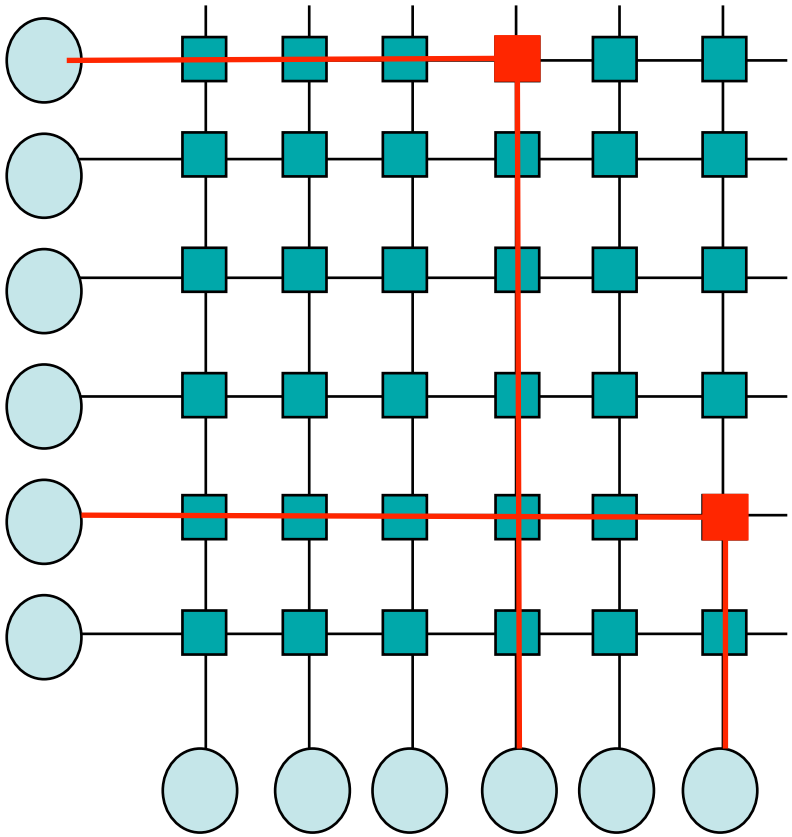


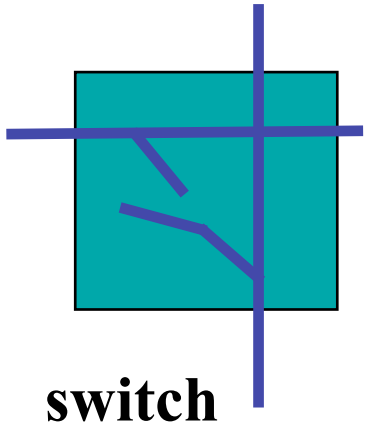
Cost: N (=2^n)
Diameter: n

Indirect network

- Each node (processor) has a link, or multiple links.
- The link connects to switch that connects to other switches.
- Messages are routed on switches.
- No direct connection between processors
- Typical topology of indirect network
  - Crossbar
  - MIN (Multistage Interconnection Network)
  - HXB (Hyper-Crossbar)
  - Tree (Indirect)
  - Fat Tree

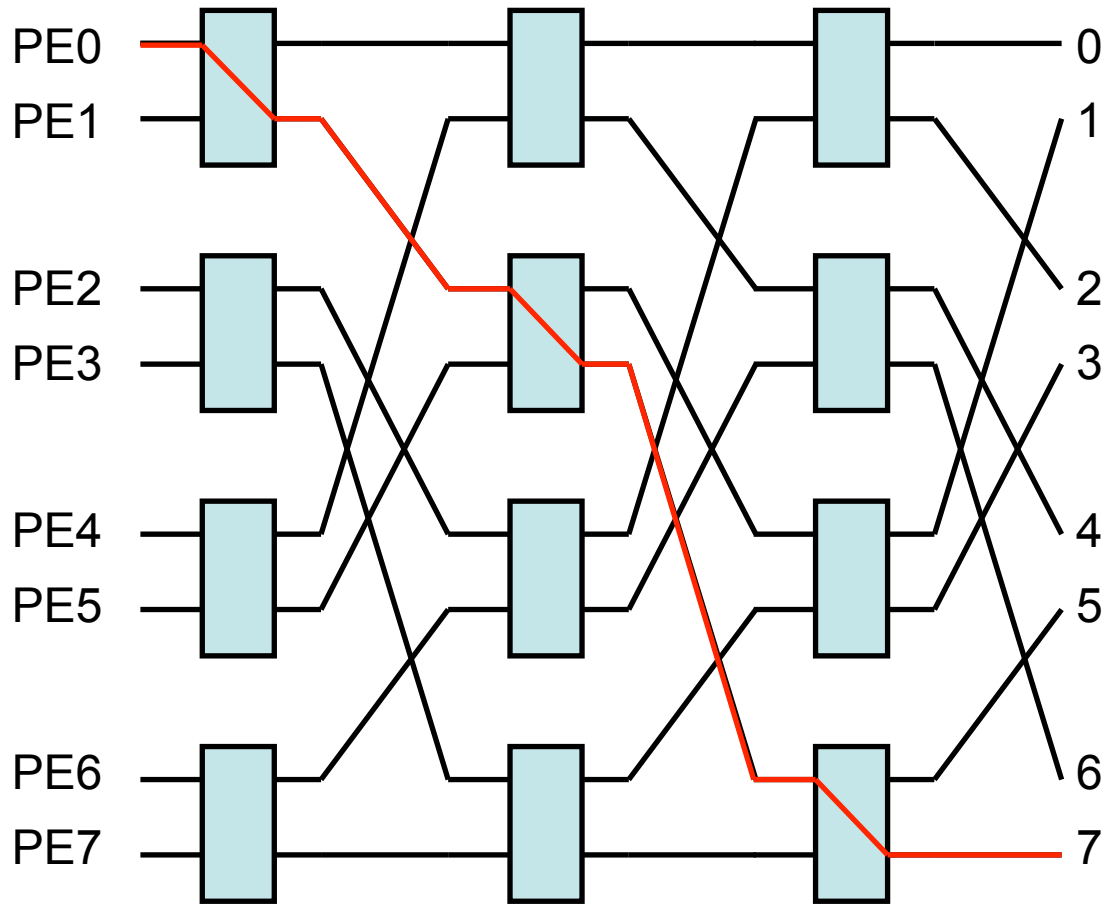
Crossbar



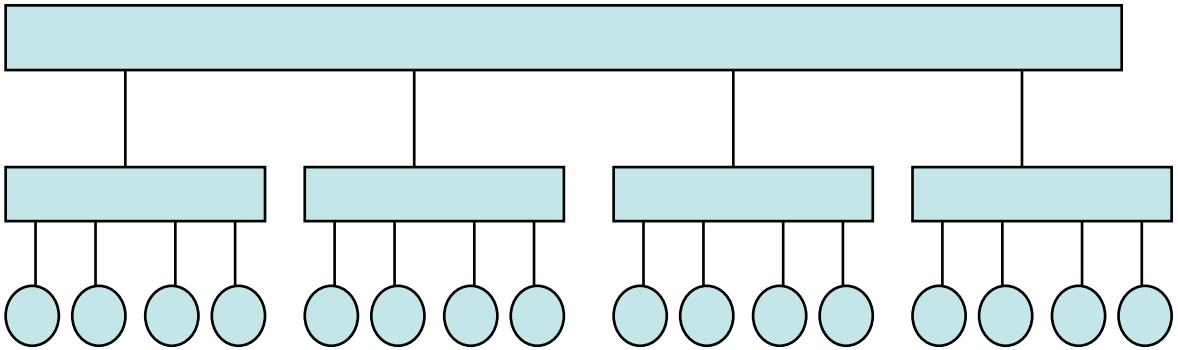


Cost: N^2
Diameter: 1

MIN (Multi-stage Interconnection Network)



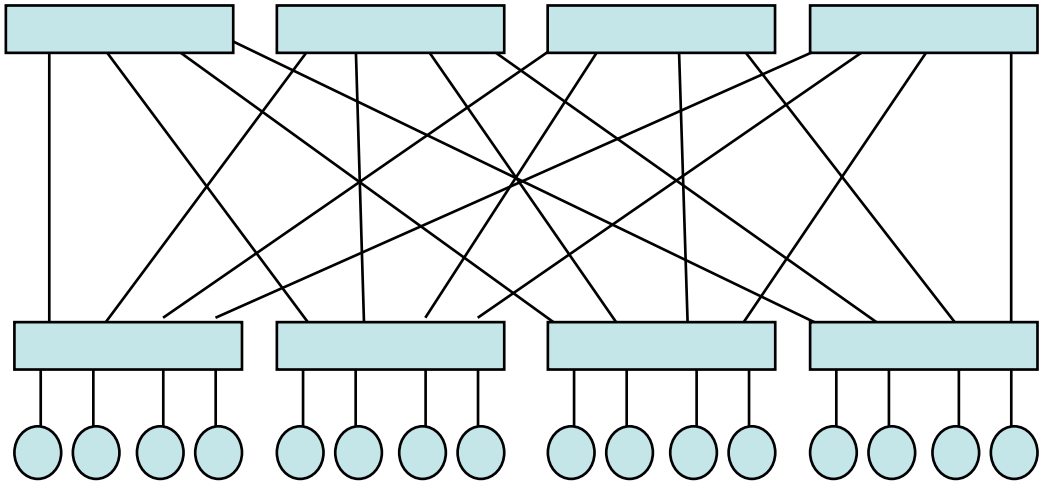
Cost: $N \log N$
Diameter: $\log N$



Tree

Diameter: $2\log_k N$

Fat Tree



Diameter: $2\log_k N$

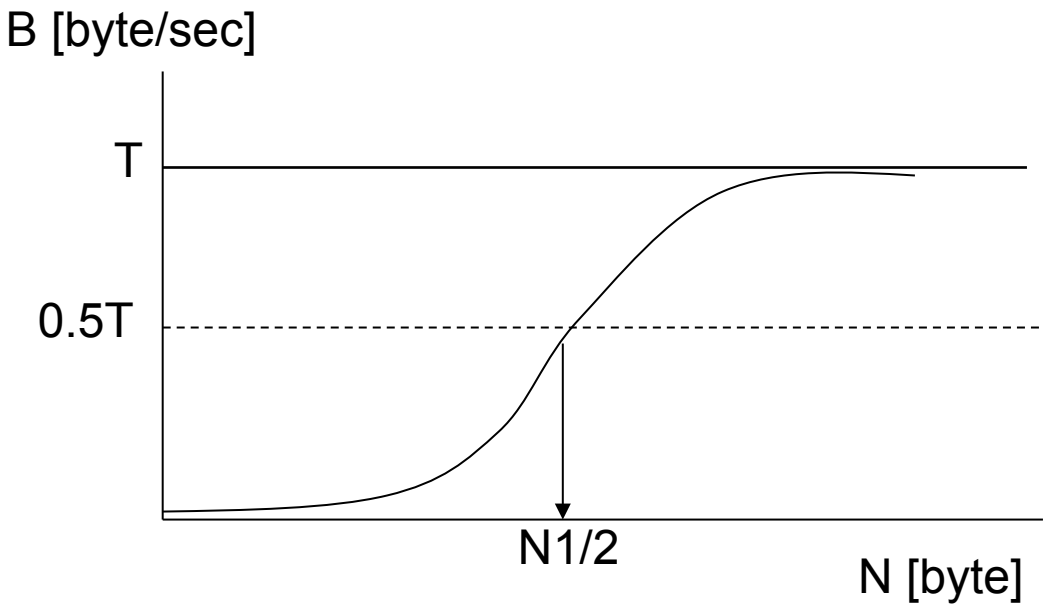
Performance metric of interconnection

- Throughput
  - The size of transferred message in a time unit
  - Unit:[Byte/sec]
    (or bit/sec, where 8bit = 1byte is not always true by encoding such as 8b/10b)
- Latency
  - Narrow: the time from the source sending the beginning of a packet to the destination receiving it. (here this definition used)
  - Wide: the time from the source sending a packet to the destination receiving its whole data. It depends on the size of packet.
  - Unit:[sec]

Performance and message size

- The relation between the size of message N[byte] and the effective bandwidth B[byte/sec] is the following equations and graph where there is no conflict on interconnect network, the network throughput T[byte/sec], latency L[sec], and the total transfer time t [sec] .

  $t = L + N/T$ $B = N / t$



$N_{1/2}$: the message length to achieve the half of theoretical peak performance, and calculated in theory.

$N_{1/2}[byte] = L \times T$

$N_{1/2}$ means that L is dominant if N is less than $N_{1/2}$, and T is dominant if N is more than $N_{1/2}$. Smaller $N_{1/2}$ shows the network has good performance for shorter message.

Overview of Supercomputers

- System classification
  - MPP
    - Univ. of Tsukuba/Hitachi CP-PACS (SR2201)
    - RIKEN/Fujitsu K computer
    - LLNL /IBM BlueGene/Q Sequia
    - ORNL/Cray XK7 Titan
  - Large scale parallel vector computer
    - NEC Earth simulator
  - Scalar parallel computer (including cluster)
    - Univ. of Tsukuba/Hitachi/Fujitsu PACS-CS
    - Univ. of Tsukuba・Tokyo・Kyoto/Appro・Hitachi・Fujitsu T2K
  - Hybrid parallel computer with accelerator
    - LANL/IBM Roadrunner
    - TITECH/NEC・HP TSUBAME2.0
    - SYU/NUDT Tianhe-2
    - Univ. of Tsukuba/Appro HA-PACS

TOP500 List

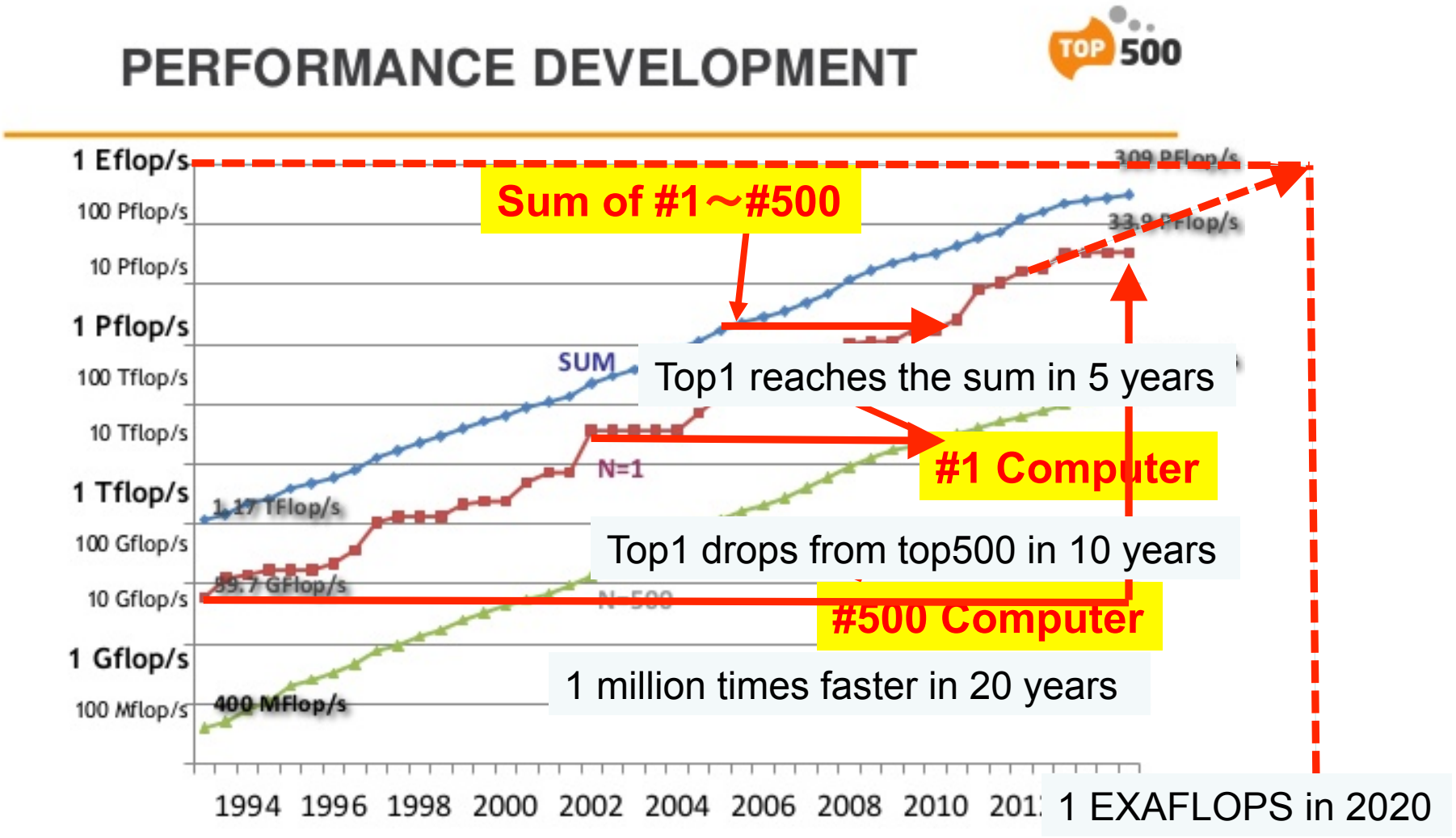
- The list ranked supercomputers in the world by their performance on one index based on user submission.
- Index = performance (FLOPS) of LINPACK (solver for a dense system of linear equations by Gaussian elimination)
- update the list half-yearly, every June in ISC and November in SC http://www.top500.org
- Easy to understand because of one index
- Benchmark characteristic
  - The kernel part of Gaussian elimination is implemented by small scale matrix multiplication, and cache architecture can reuse the highly part of data. Good estimation of peak performance.
  - The performance is not highly depend on the network performance
- Since LINPACK does not require the high memory bandwidth and high network performance, “Is LINPACK suitable for HPC benchmark ?” is discussed, but LINPACK is one of well-known performance index. (HPCC(HPC Challenge Benchmark) is another benchmark for HPC)

Green500

- Ranking by performance per watt (MFLOPS/W) from TOP500 list.
- Power supply becomes a bottleneck for large scale supercomputer, and the index is recently focused.
- update the list half-yearly same as TOP500 list http://www.green500.org/
- The value of TOP500 is used for the performance index, there is same problem as TOP500.
- Entry of Green500 should be in TOP500, but the amount of power supply is very different from 10MW to 30kW. In general, small system is better in the index of performance per watt. So large scale production level system got the special award, or small system not in TOP500 also listed as little Green500.

TOP500 List

http://www.top500.org 2014.11



TOP4 in Nov. 2014

TOP500

Rank	Name	Country	Rmax (PFLOPS)	Node Architecture	# of nodes	Power (MW)
1	Tianhe-2	China	33.9	2 Xeon(12C) + 3 Xeon Phi(57C)	16000	17.8
2	Titan	U.S.A	17.6	1 Opteron(16C)+ 1 K20X(14C)	18688	8.2
3	Sequoia	U.S.A	17.2	1 PowerBQC(16C)	98304	7.9
4	K computer	Japan	10.5	1 SPARC64(8C)	82944	12.7

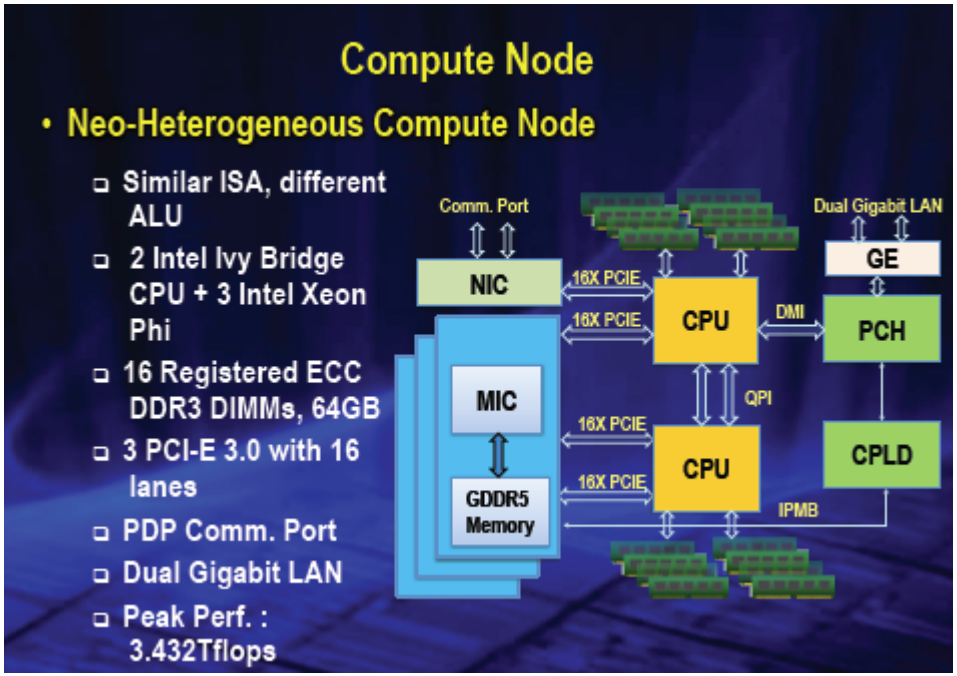
GREEN500

Rank	Name	Country	Mflops/ Watt	Node Architecture	# of node	Power (KW)	TOP 500
1	GSI Helmholtz Center	German	5271.8	Xeon(10C) + 2 AMD FirePro S9150(44C)	112	57.15	168
2	Suiren/KEK	Japan	4945.6	Xeon(10C) + 4 PEZY-SC(1024C)	64	37.83	369
3	TSUBAME-KFC	Japan	4447.6	2 Xeon(6C) + 4 K20X(14C)	44	34.6	392

Tianhe-2(天河-2)

- National University of Defense Technology, China
- Top500 2013/6 #1, 33.8PFLOPS (efficiency 62%), 16000node(2 Xeon(12core) + 3 Phi(57core), 17.8MW, 1.9GFLOPS/W
- TH Express-2 interconnection (same perf. of IB QDR (40Gbps) x2)
- CPU Intel IvyBridge 12core/chip, 2.2GHz
- ACC Intel Xeon Phi 57core/chip, 1.1GHz
- 162 racks (125 rack for comp. 128node/rack)

Compute Node of Tianhe-2



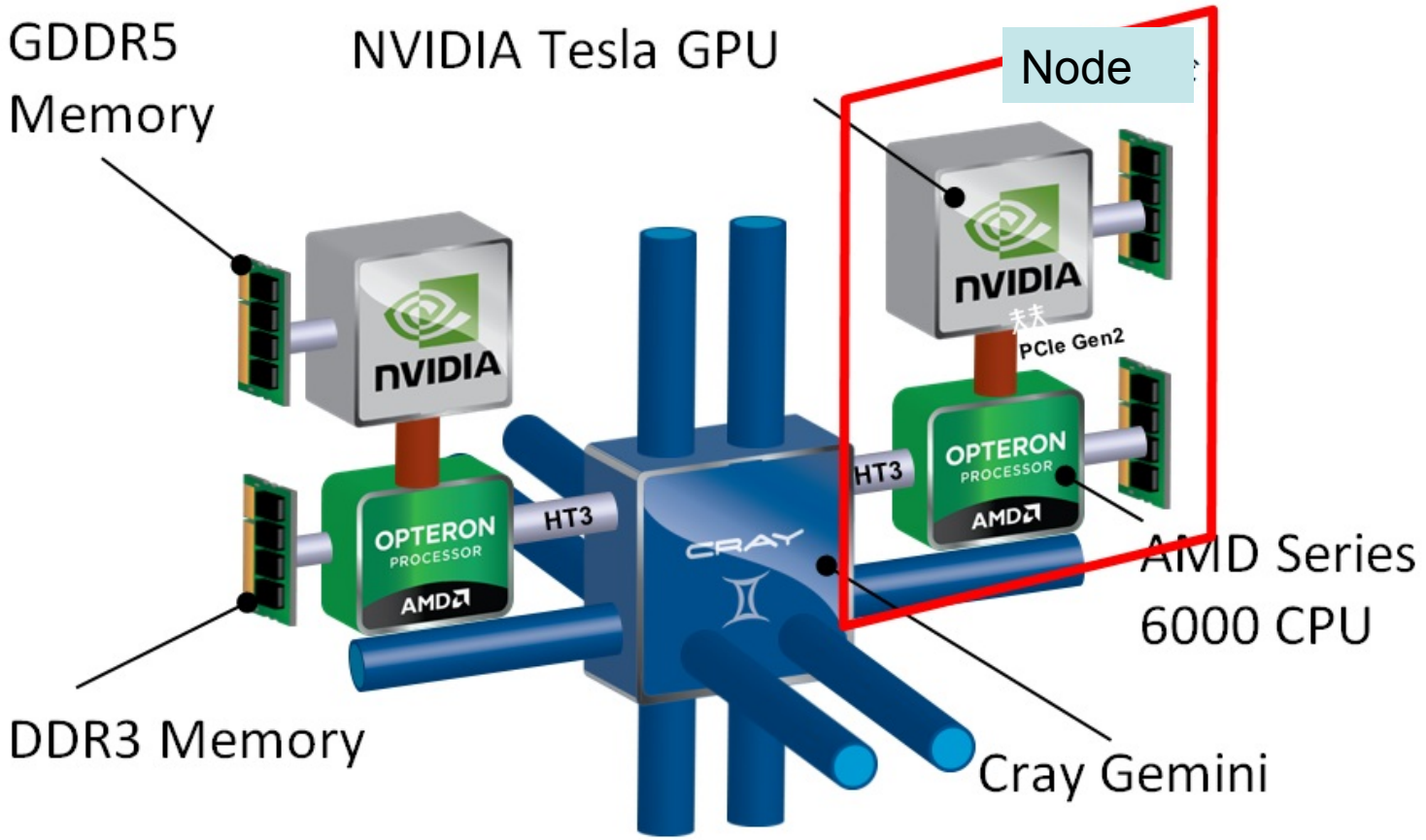
CPU x 2: 2.2GHz x 8flop/cycle x 12core x 2 = 422.4GFLOPS
MIC x 3: 1.1GHz x 16flop/cycle x 57core x 3 = 3.01TFLOPS

CPU: 422.4 x 16000 = 6.75PFLOPS
64GB x 16000 = 1.0 PB
MIC: 3010 x 16000 = 48.16PFLOPS
8GB x 3 x 16000 = 0.384 PB

ORNL Titan

- Oak Ridge National Laboratory
- Cray XK7
- Top500 2012/11 #1, 17.6PFLOPS (efficiency 65%), 18688 CPU + 18688 GPU, 8.2MW, 2.14GFLOPS/W
- Gemini Interconnect (3Dtorus)
- CPU AMD Opteron 6274 16core/ chip, 2.2GHz
- GPU nVIDIA K20X, 2688CUDA core (896 DP unit)

Node of XK7



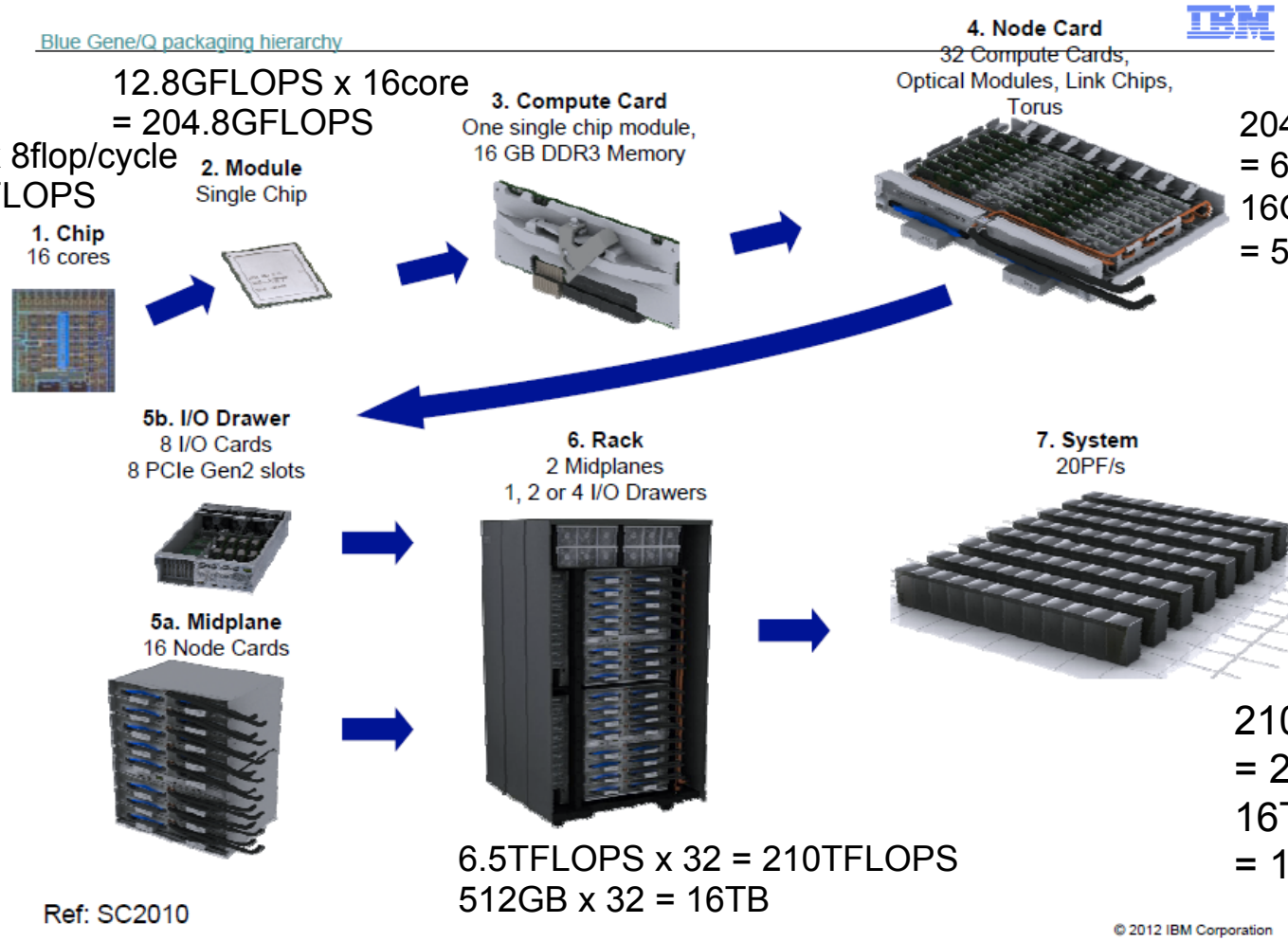
CPU: 2.2GHz x 4flop/cycle x 16core
= 140.8GFLOPS
GPU: 1.46GHz x 64flop/cycle x 14SMX
= 1.31TFLOPS

CPU: 140.8 * 18688 = 2.63 PFLOPS
32GB x 18688 = 0.598 PB
GPU: 1310 * 18688 = 24.48 PFLOPS
6GB x 18688 = 0.112 PB

LLNL Sequoia

- Lawrence Livermore National Laboratory (LLNL)
- IBM BlueGene/Q
- Top500 2012/6 #1, 16.3PFLOPS (efficiency 81%), 1.57Mcore, 7.89MW, 2.07GFLOPS/W
- 4 BlueGene/Q were listed in top10
- 18core/chip, 1.6GHz, 4way SMT, 204.8GFLOPS/55W, L2:32MB eDRAM, mem: 16GB, 42.5GB/s
- 32chip/node, 32node/rack, 96rack

Sequoia Organization



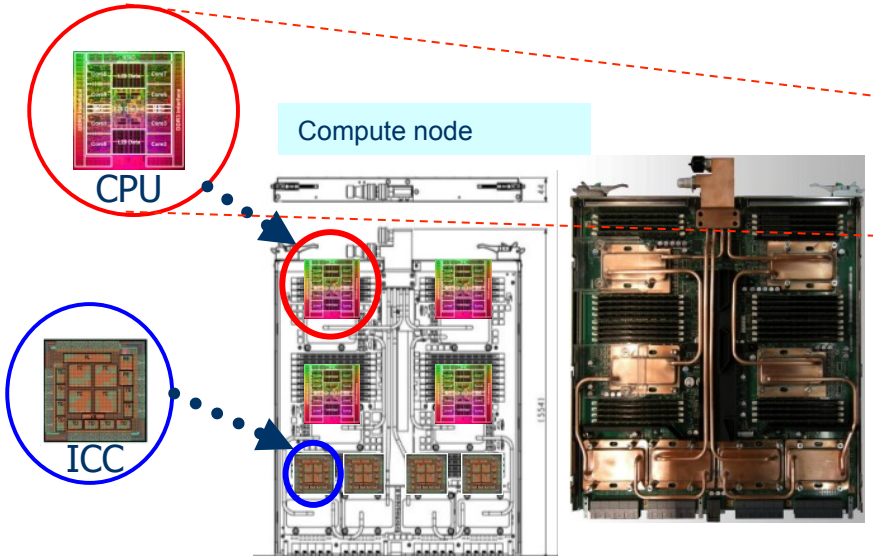
K computer

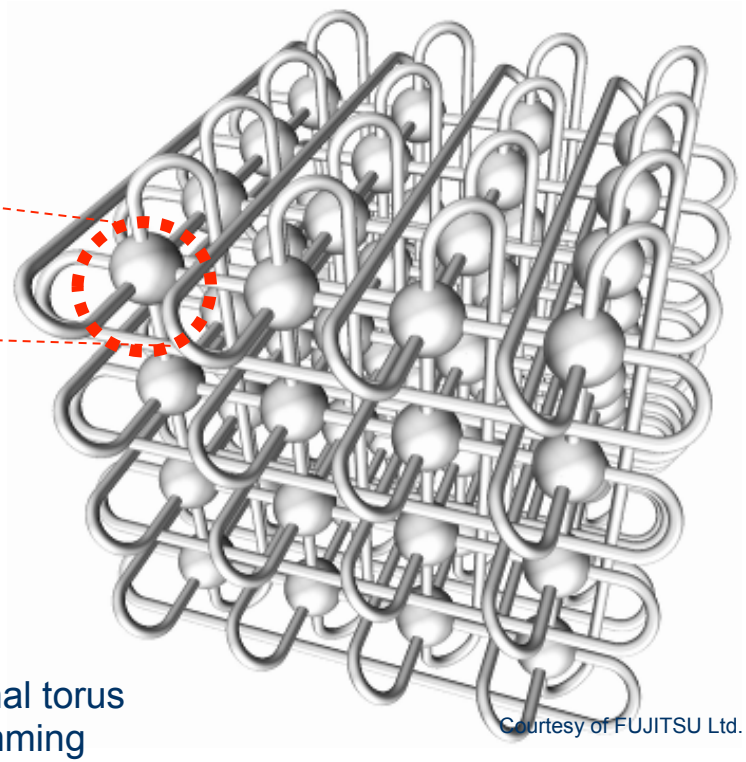
- RIKEN by Fujitsu in 2012
- Each nodes has 4 SPARC64 VIIIfx (8core) and network chip (Tofu Interconnect)
- TOP500#1 2011/6-11
- 705k core, 10.5 PFLOPS (efficiency 93%)
- LINPACK power consumption 12.7MW
- Green500#6 2011/6 (824MFLOPS/W)
- Green500#1 is BlueGene/Q Proto2 2GFLOPS/W (40kW)

Compute Nodes and network

- Compute nodes (CPUs): 88,128
  - SPARC64 VIIIfx
  - Number of cores: 705,024 (8core/CPU)
- Peak performance: 11.2PFLOPS
  - 2GHz x 8flop/cycle x 8core x 88128
- Memory: 1.3PB (16GB/node)

- Logical 3-dimensional torus network
- Peak bandwidth: 5GB/s x 2 for each direction of logical 3-dimensional torus network
- bi-section bandwidth: > 30TB/s

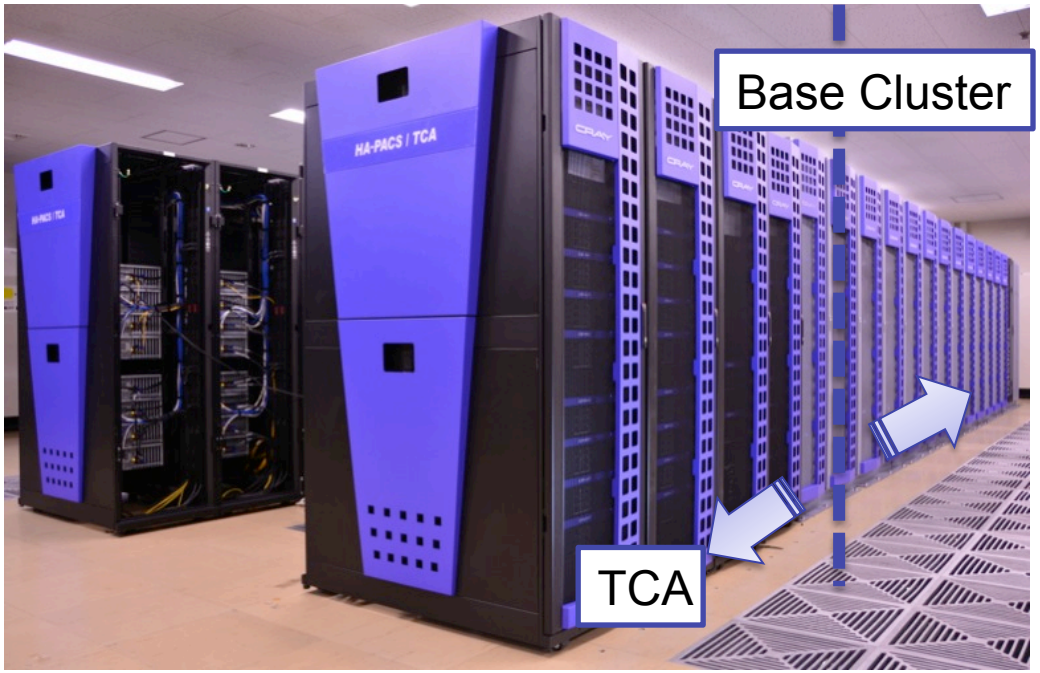




Logical 3-dimensional torus network for programming

Courtesy of FUJITSU Ltd.

HA-PACS



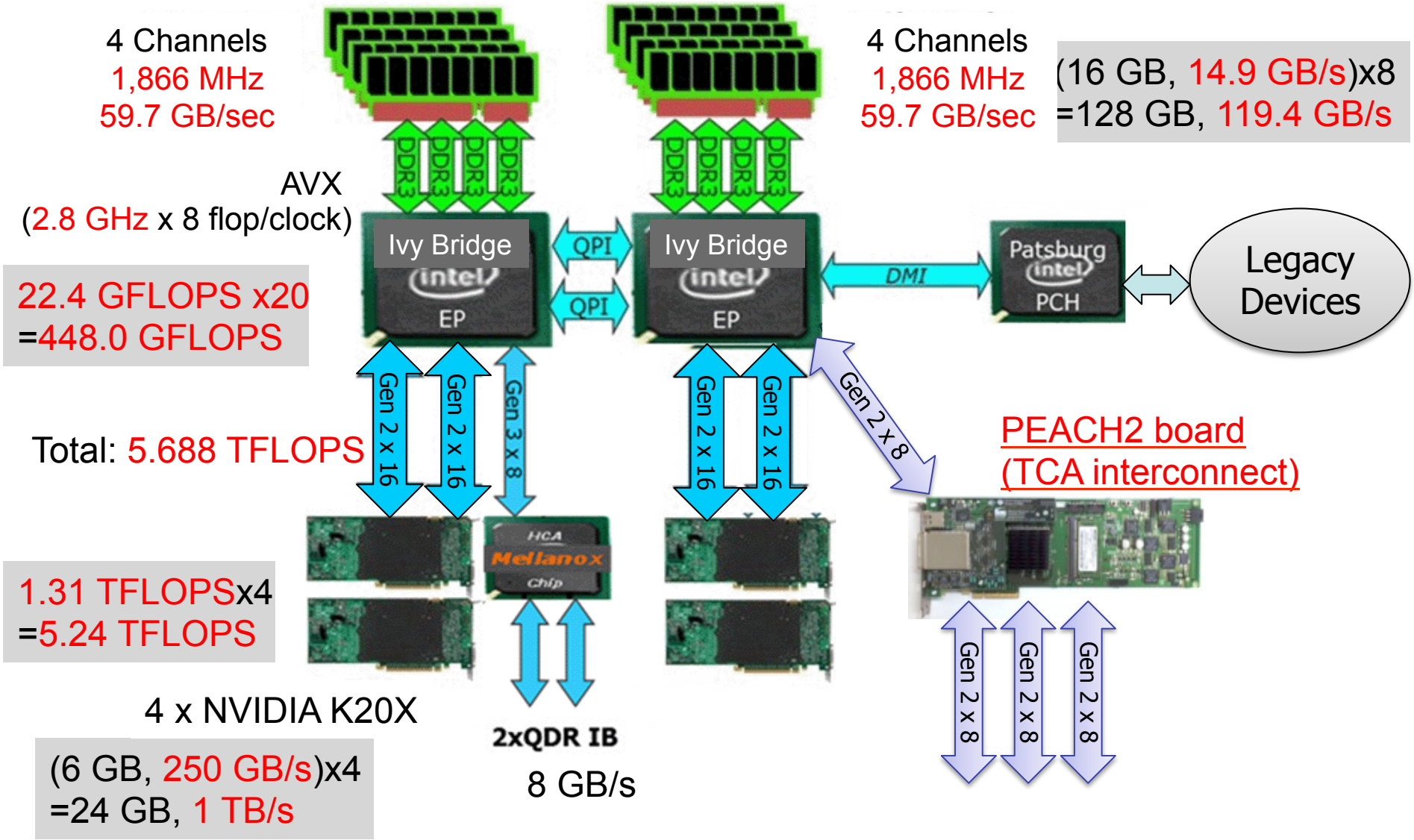
Base Cluster

- CCS in U-Tsukuba by Appro in 2012
- Commodity based PC cluster with multiple GPU accelerators
- For computational sciences
- 268 node = 4288 CPU core and + 1072 GPU 802 (= 89 + 713) TFLOPS
- 421 TFLOPS by LINPACK
- 1.15 GFLOPS/W
- 40 TByte memory

TCA

- by Cray in Oct. 2013
- latest CPU(Intel IvyBridge) and latest GPU(NVIDIA K20X)
- proprietary network for direct connecting GPU (PEACH2)
- 64 node : 277 TFLOPS by LINPACK (76% efficiency)
- 3.52 GFLOPS/W 3rd Green500 Nov. 2013

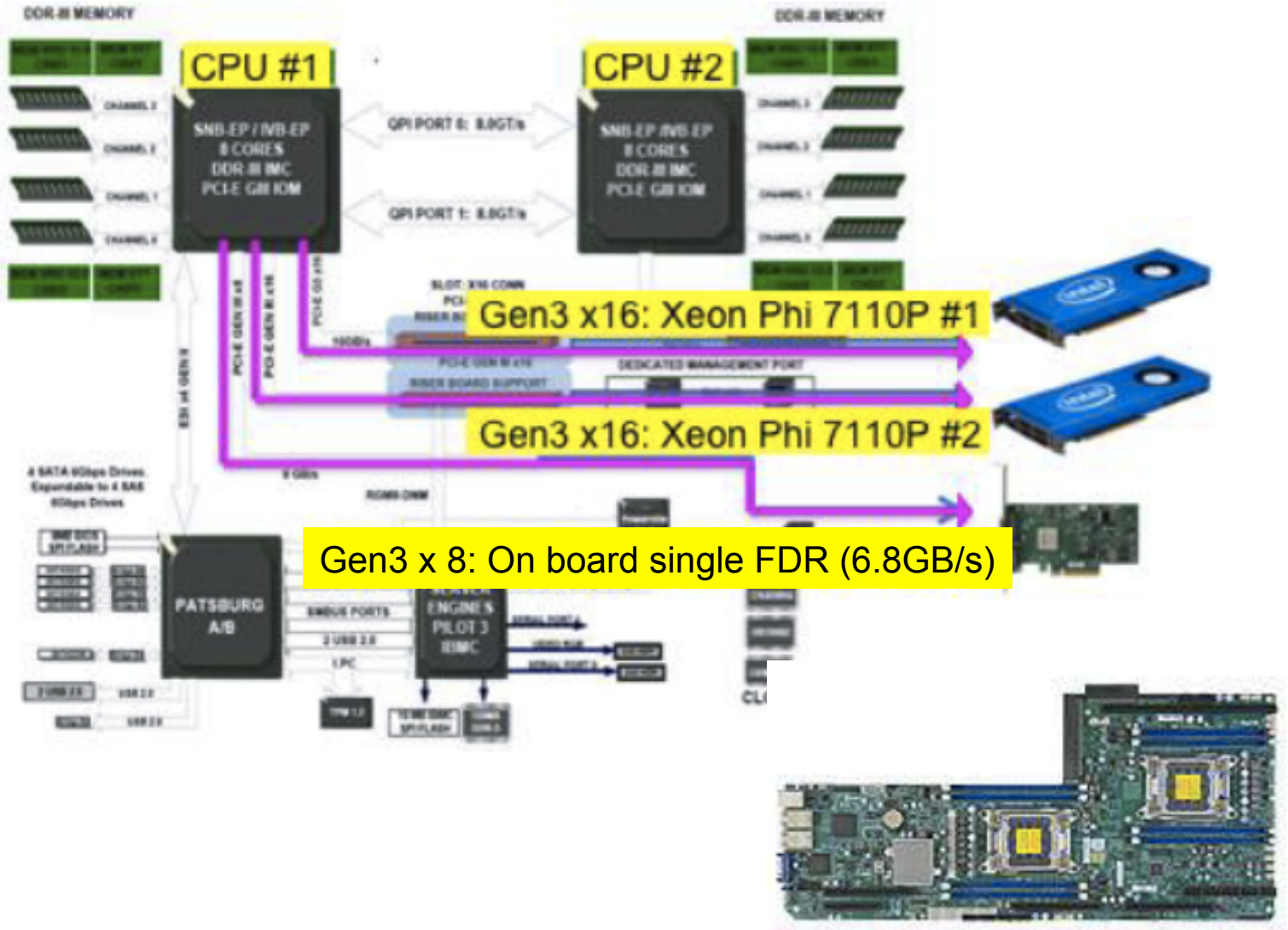
HA-PACS/TCA (node)



COMA

- CCS in U-Tsukuba by Cray in 2014/4
- Each node consists of 2 Xeon and 2 Xeon Phi (Many core)
- For computational sciences
- 393 node = 7860 (10x2x393) CPU core + 47160 (60x2x393) Xeon Phi core
- 1001 (= 157 + 844) TFLOPS
- 40 TByte memory
- 17rack
- #51 TOP500 2014/7 746TFLOPS

COMA (node)


CPU #1
CPU #2
Gen3 x16: Xeon Phi 7110P #1
Gen3 x16: Xeon Phi 7110P #2
Gen3 x 8: On board single FDR (6.8GB/s)

Trend of parallel system

- Commodity based cluster increase
  - Commodity scalar processor (IA32=x86)
  - Commodity network I/F and switch
    - Ethernet (1Gbps ⇒ 10Gbps)
    - Infiniband (2GB/s ⇒ 8GB/s, the price gradually reduced)
- The balance between processor, memory and communication performance becomes worse.
  - Arithmetic performance increases smoothly with multi-core processor
  - Memory performance (bandwidth) will not increase, and relatively reduced for a core (pin-bottleneck, 3D or wide-I/O memory ?)
  - Communication performance will increase step-wise, but relatively reduce for a core (Ethernet, etc)
  - Processor cost is *O(N),* but network cost is *O(N log N),* so the network cost relatively large
  - It becomes difficult to improve the parallel efficiency in large scale parallel system. Algorithm level improvements are required.
- Cluster with accelerator will increase
  - high performance per cost, performance per watt, …
- Feasible study for Exa FLOPS machine was started

Summary

- Parallel system / architecture
  - The performance of sequential processor (core) has been limited, so total performance will be increased by parallel processing.
  - Scalability with keeping performance is important
  - Distributed memory vs. shared memory
- Interconnection network
  - Scalability is most important
  - MPP had wide variety of implementations (custom network)
  - Current cluster network has scalability with fat-tree topology using commodity network.
  - Two performance metrics: throughput & latency
- Trend and problems for parallel computer
  - The number of core will be increased, to 1 million cores with multicore processor.
  - The balance between processor, memory, network performance will be worse.
  - Node with accelerator is attracted