FUJITSU

shaping tomorrow with you

Technologies beyond the K computer

September 5th, 2012

Takashi Aoki

Next Generation Technical Computing Unit

Fujitsu Limited

Agenda

FUJITSU

- Corporate profile
- Fujitsu supercomputer past and present
- Second generation Petascale supercomputer PRIMEHPC FX10
  - Hardware
  - Software
- Challenge to the future

Japan's largest IT services provider and
No. 3 in the world. *

We do everything in ICT. We use our experience and the power of ICT to shape the future of society with our customers.

Over 170,000 Fujitsu people support customers in more than 100 countries.

*2011 IT Services Vendor Revenue. Source: Gartner, "Market Share: IT Services, 2011" 9 April 2012

Our products and services

Technology Solutions

Services



Our datacenters in the world

Systems platform

PRIMERGY
TX120

ETERNUS
DX8000

Supercomputer
PRIMEHPC FX10

Ubiquitous Product Solutions

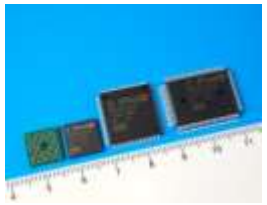
LIFEBOOK
E751C

Smart phone
F07D

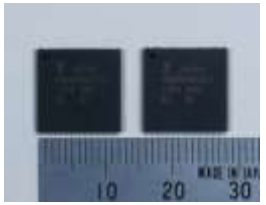
Tablet PC
ARROWS

Device solutions

High-end multi-core
processor
SPARC64 VII+



FM3 family
(32-bit RISC MCU)



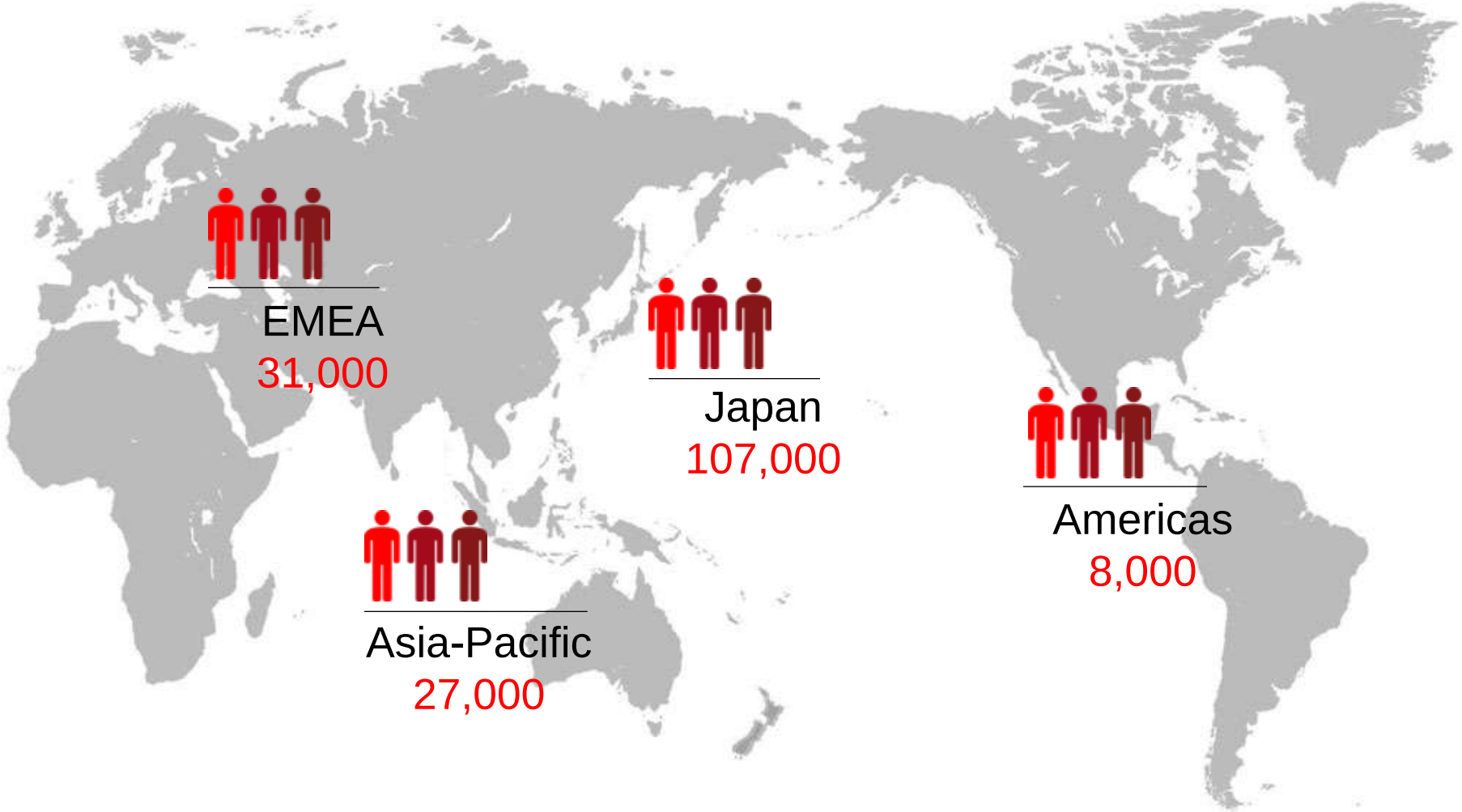
FRAM
(Ferroelectric
Random Access
Memory)

Where we work

FUJITSU

'shaping tomorrow with you' wherever you are.

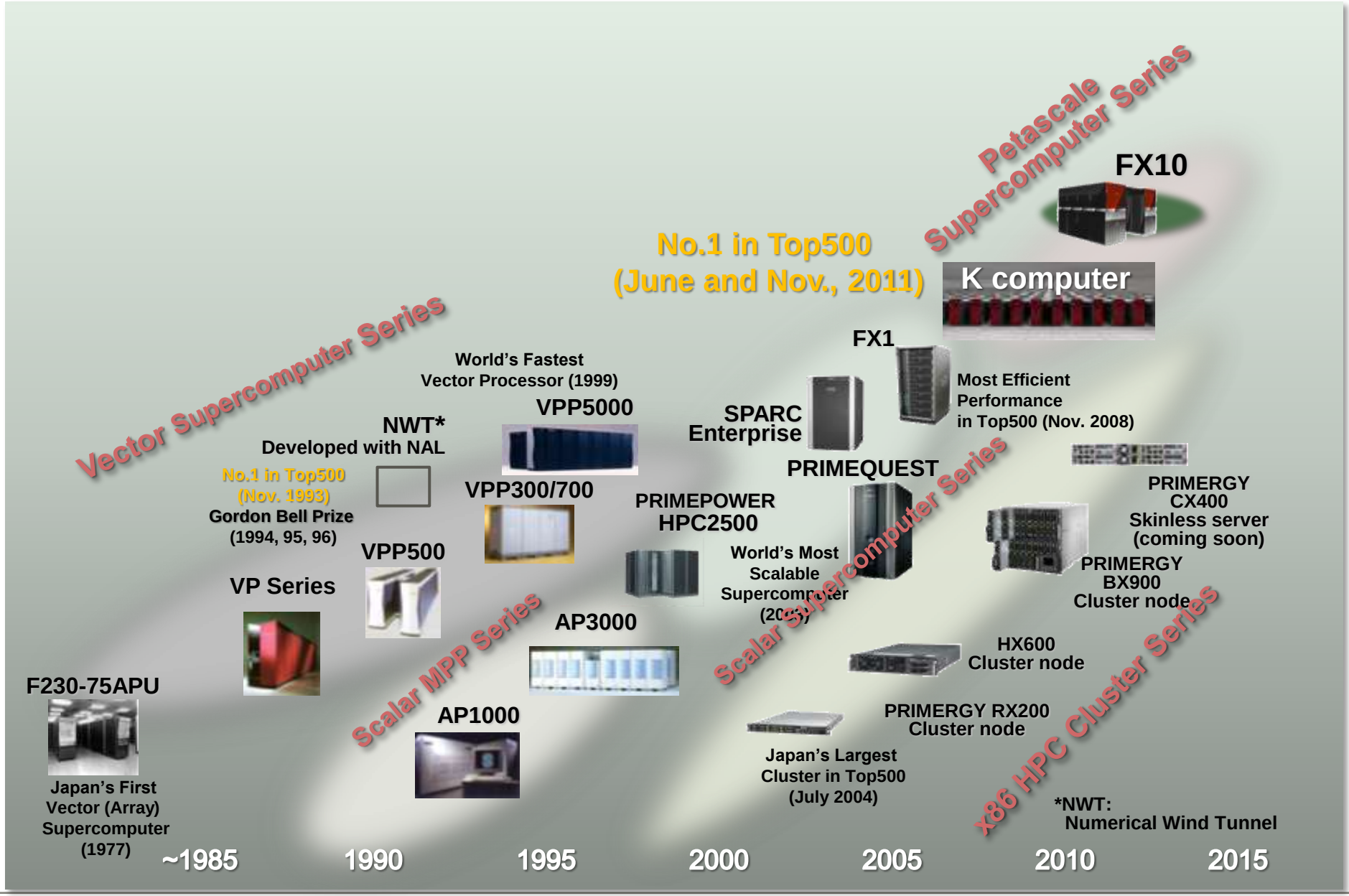
As of March 2012



Over 170,000 Fujitsu colleagues working with customers in over 100 countries

Fujitsu HPC Servers - past and present -

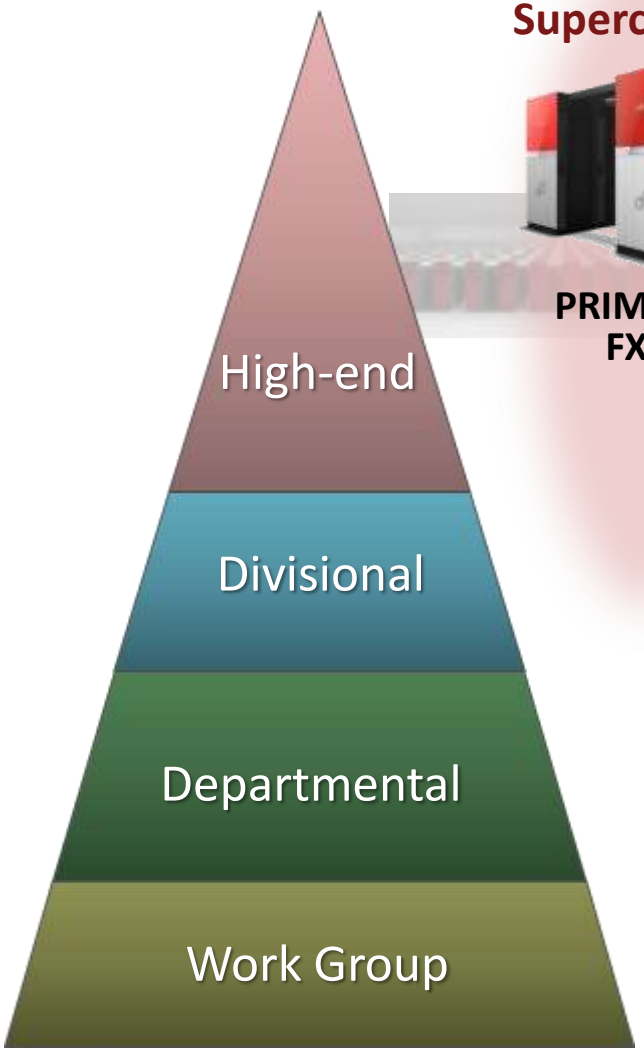
FUJITSU



HPC Platform Solutions - Hardware -

FUJITSU

- Full range coverage with choice of HPC hardware platform



Petascale Supercomputer

PRIMEHPC FX10

High Performance scaling over several PFlops

➔ Fujitsu propriety CPU and interconnect technologies for high performance, high reliability and high operability

x86 HPC Cluster

CX400
Skinless server

High Performance de facto HPC cluster

➔ Following Intel CPU and MIC roadmap and adopt Fujitsu latest packaging technologies for high performance and high operability

BX Series
BX900
BX400

RX Series
RX200

Large-Scale SMP System

RX900

PRIMERGY series

Design targets and features of FX10

FUJITSU

- High Performance
  - High peak performance and high application performance
- High parallel application productivity
  - Easy to achieve high performance running highly paralleled programs without inordinate effort of programming

Customer 's requirement and FX10 design targets

- High operability
  - Low power consumption
  - High reliability and ease of operation
- K computer compatibility
  - Binary compatibility
  - Same programing environment

Design targets and features of FX10

FUJITSU

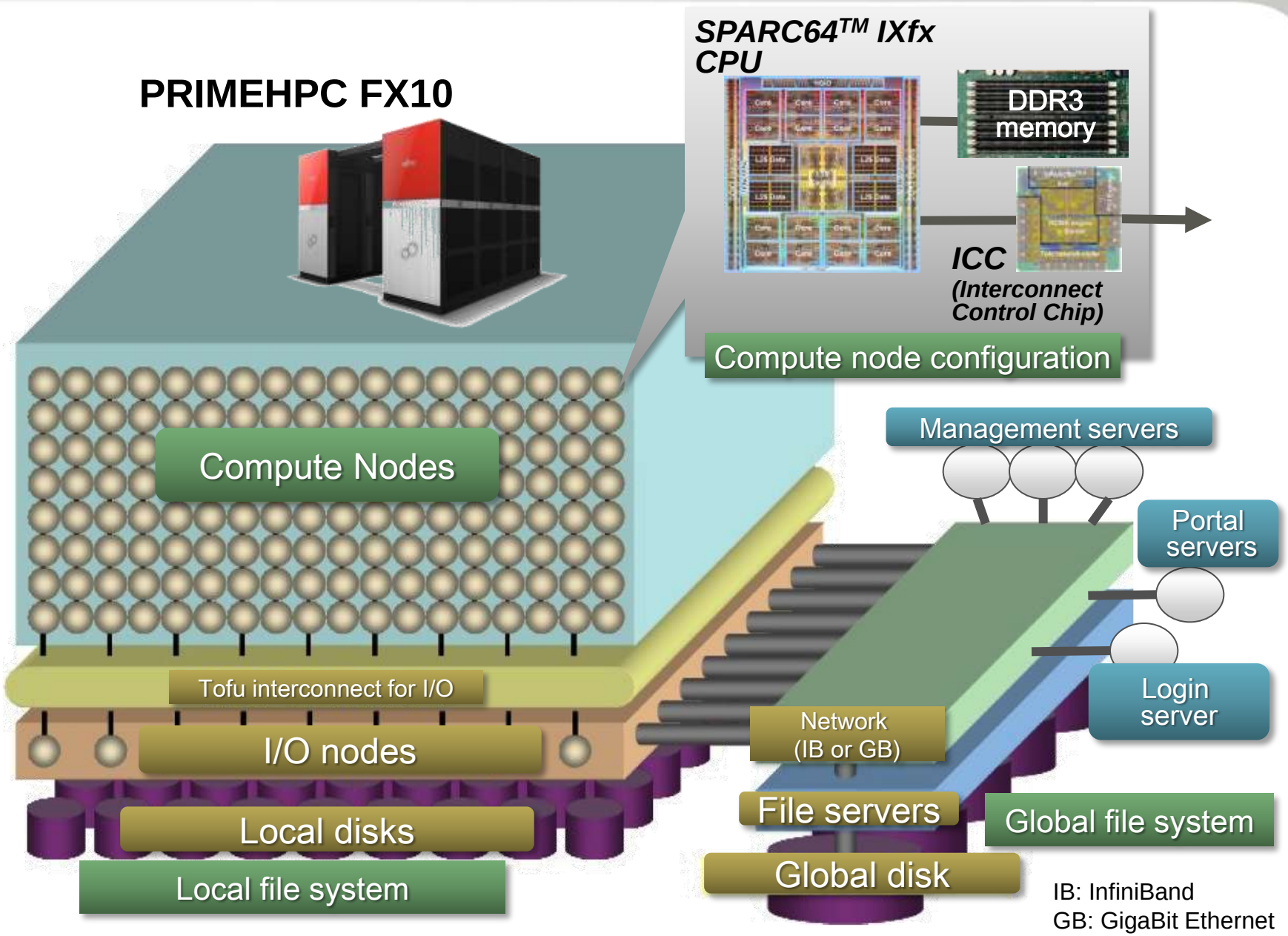
- High Performance
  - High-performance CPU "*SPARC64 IXfx*" with SPARC V9 + HPC-ACE architecture
  - High performance, highly reliable and fault tolerant 6D mesh/torus interconnect "Tofu[*1]"
- High parallel application productivity
  - Easy to achieve high
  - "*VISIMPACT*[*2]" supports efficient hybrid parallel execution
  - ordinate effort of
  - mming

- sign targets
  - Parallel Language, programing tools and Petascale HPC middleware for high reliability and operability
- High operability
  - Low power consumption
  - High reliability and ease of
  - Water cooling system
- K computer compatibility
  - Binary compatibility
  - Same programing environment
- High reliability components & functions based on mainframe development experience

*1) Tofu: Torus Fusion
*2) VISIMPACT: Virtual Single Processor by Integrated Multicore Parallel Architecture

PRIMEHPC FX10 System Configuration

FUJITSU

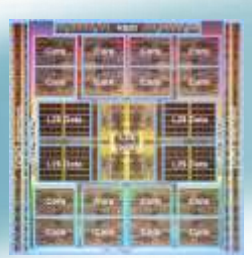


FX10 System H/W Specifications

FUJITSU

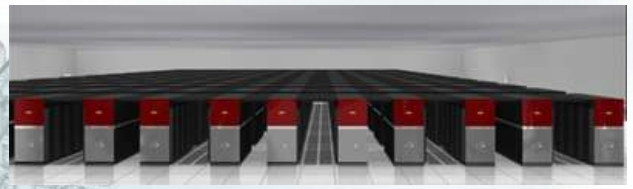
PRIMEHPC FX10 H/W Specifications		
CPU	Name	*SPARC64™ IXfx*
	Performance	236.5GFlops@1.848GHz
Node	Configuration	1 CPU / Node
	Memory capacity	32, 64 GB
Rack	Performance/rack	22.7 TFlops
System (4 ~1024 racks)	No. of compute node	384 to 98,304
	Performance	90.8TFlops to 23.2PFlops
	Memory	12 TB to 6 PB

- ***SPARC64™ IXfx CPU***
  - 16 cores/socket
  - 236.5 GFlops



- ***System rack***
  - 96 compute nodes
  - 6 I/O nodes
  - With optional water cooling exhaust unit

- ***System board***
  - 4 nodes (4 CPUs)



- ***System***
  - Max. 23.2 PFlops
  - Max. 1,024 racks
  - Max. 98,304 CPUs

Comparison of System H/W Specifications

FUJITSU

		K computer	FX10
CPU	Name	*SPARC64™ VIIIfx*	*SPARC64™ IXfx*
	Performance	128GFlops@2GHz	236.5GFlops@1.848GHz
	Architecture	SPARC V9 + HPC-ACE extension	←
	Cache configuration	L1(I) Cache:32KB/core, L1(D) Cache:32KB/core	←
		L2 Cache: 6MB(shared)	L2 Cache: 12MB(shared)
	No. of cores/socket	8	16
	Memory band width	64 GB/s.	85 GB/s.
Node	Configuration	1 CPU / Node	←
	Memory capacity	16 GB	32, 64 GB
System board	Node/system board	4 Nodes	←
Rack	System board/rack	24 System boards	←
	Performance/rack	12.3 TFlops	22.7 TFlops

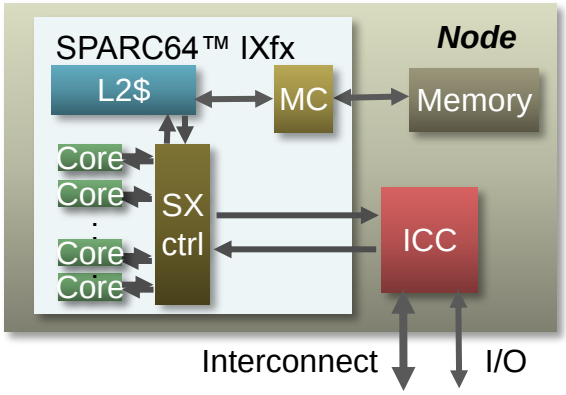
Comparison of System H/W Specifications (cont.)

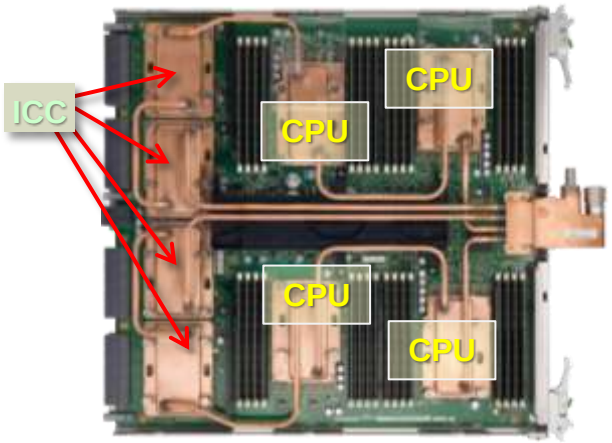
FUJITSU

		K computer	FX10
Interconnect	Topology	6D Mesh/Torus	←
	Performance	5GB/s x2 (bi-directional)	←
	No. of link per node	10	←
	Additional features	H/W barrier, reduction	←
		no external switch box	←
Cooling	CPU, ICC(interconnect chip), DDCON	Direct water cooling	←
	Other parts	Air cooling	Air cooling + Exhaust air water cooling unit (Optional)

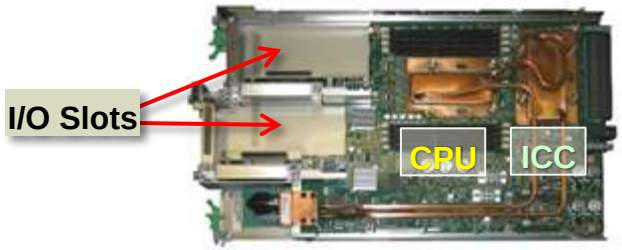
Node configuration

- Single CPU as a node
  - SPARC64™ IXfx based
  - 32/64GB memory capacity
  - Single CPU per node to maximize memory BW
  - High memory bandwidth of 85 GB/s
- On board InterConnect Controller (ICC)
  - Direct RDMA and global synchronization operations
  - No external switch
- Node type
  - Compute node
    - Consist of CPU, ICC and memory
    - No I/O capability except interconnect
    - Four nodes are mounted on a system board
  - I/O node
    - Same CPU as compute node
    - Includes four PCI Express Gen2 x8 slots
    - 8 GB/s I/O bandwidth per I/O node
    - One node is mounted on an I/O system board





System Board



I/O SB

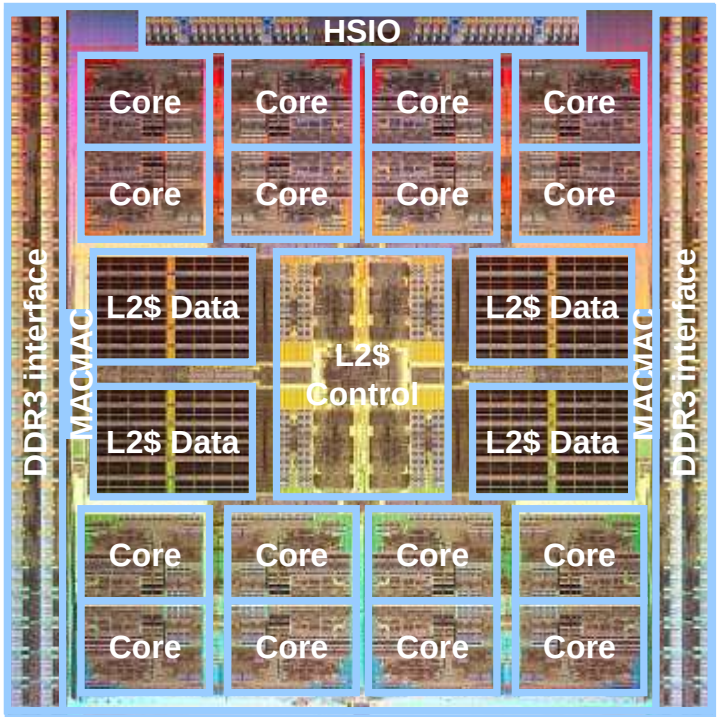
SPARC64™ IXfx

FUJITSU

- High-performance and low-power multi-core CPU
  - High performance core by HPC-ACE
    - Multiply number of register, SIMD operation, software controllable cache, etc.
  - VISIMPACT : Support highly efficient hybrid execution model (thread + process)
    - Shared second cache, hardware barrier among cores and compiler support

SPARC64™ IXfx specifications	
Architecture	SPARC V9 + HPC-ACE
# of FP operations /clock/core	8 (= 4 Multiply and Add)
No. of cores	16
Peak performance and clock	236.5 Gflops@1.848GHz
Memory bandwidth	85 GB/s
Power consumption	110 W (typical)

  - High performance-per-power ratio and High reliability
    - Water cooling system has lowered the CPU temperature and leak current
    - Wide-ranging error detection/self-recovery functions, instruction retry function



Overview of HPC-ACE

FUJITSU

“**H**igh **P**erformance **C**omputing - **A**rithmetic **C**omputational **E**xtensions”

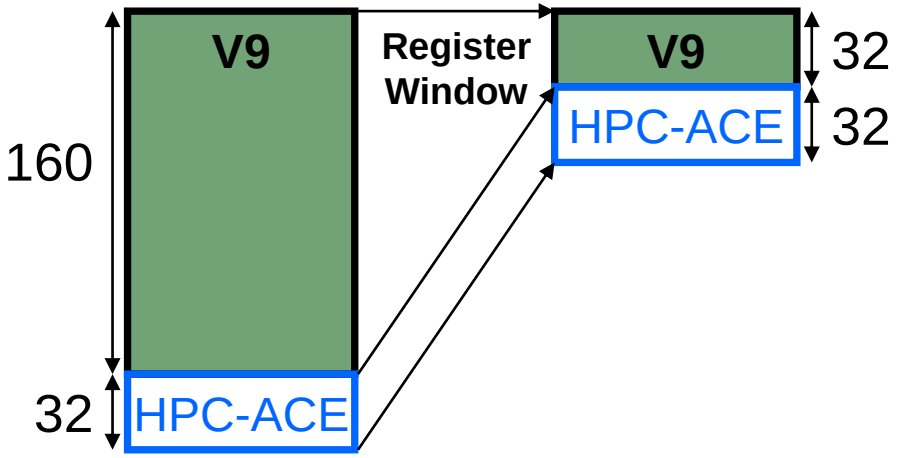
- Extended number of integer registers and floating point registers
- Software-controllable “Sector Cache”
- Flexible Single Instruction Multiple Data (SIMD) operation
- Hardware barrier synchronization for VISIMPACT
  - VISIMPACT: automatic thread-parallelization compiler technology
- Other special features
  - XFILL instruction
  - Reciprocal approximation instruction
  - Reciprocal square root approximation instruction
  - Trigonometric function acceleration instructions

HPC-ACE:Extended Number of Registers

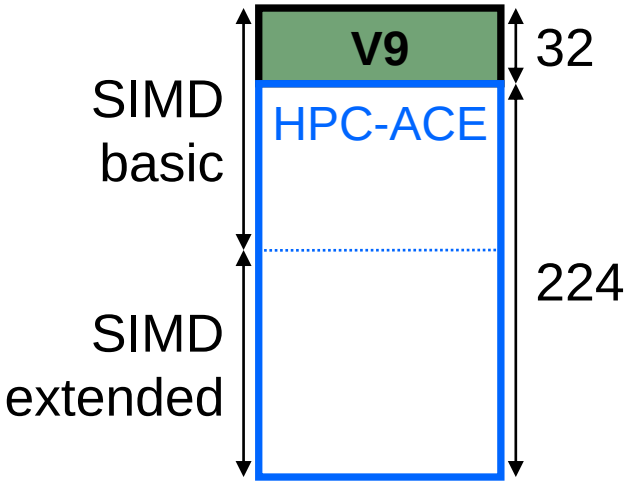
FUJITSU

- Enables larger loop unrolling and eliminates register spills

- Integer registers
  - SPARC-V9 160 / 32
  - HPC-ACE 192 / 64



- Double precision floating-point registers
  - SPARC-V9 32
  - HPC-ACE 256 (Scalar) / 128 (SIMD)

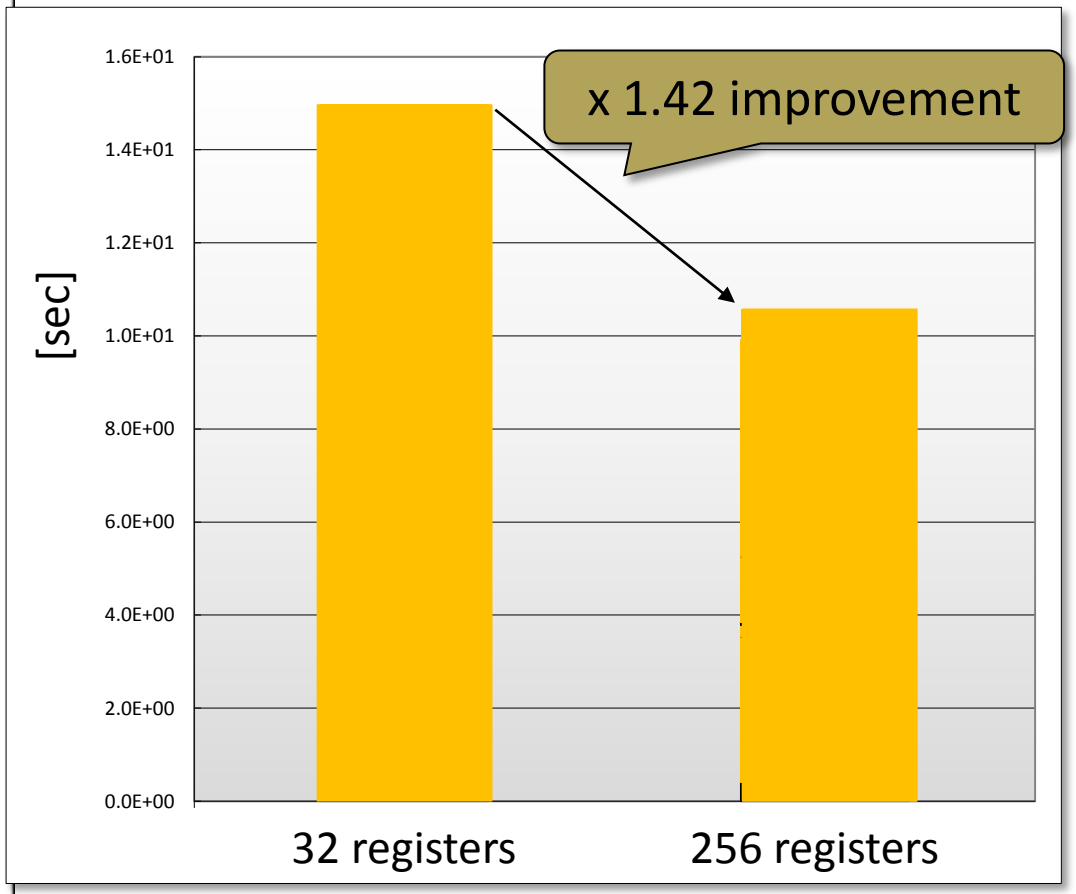


HPC-ACE:Number of FP registers extension (1)

FUJITSU

- ***NPB3.3-LU*** high cost loop
  - By using extended number of registers, compiler can generate more efficient scheduling and also eliminate unnecessary memory operations

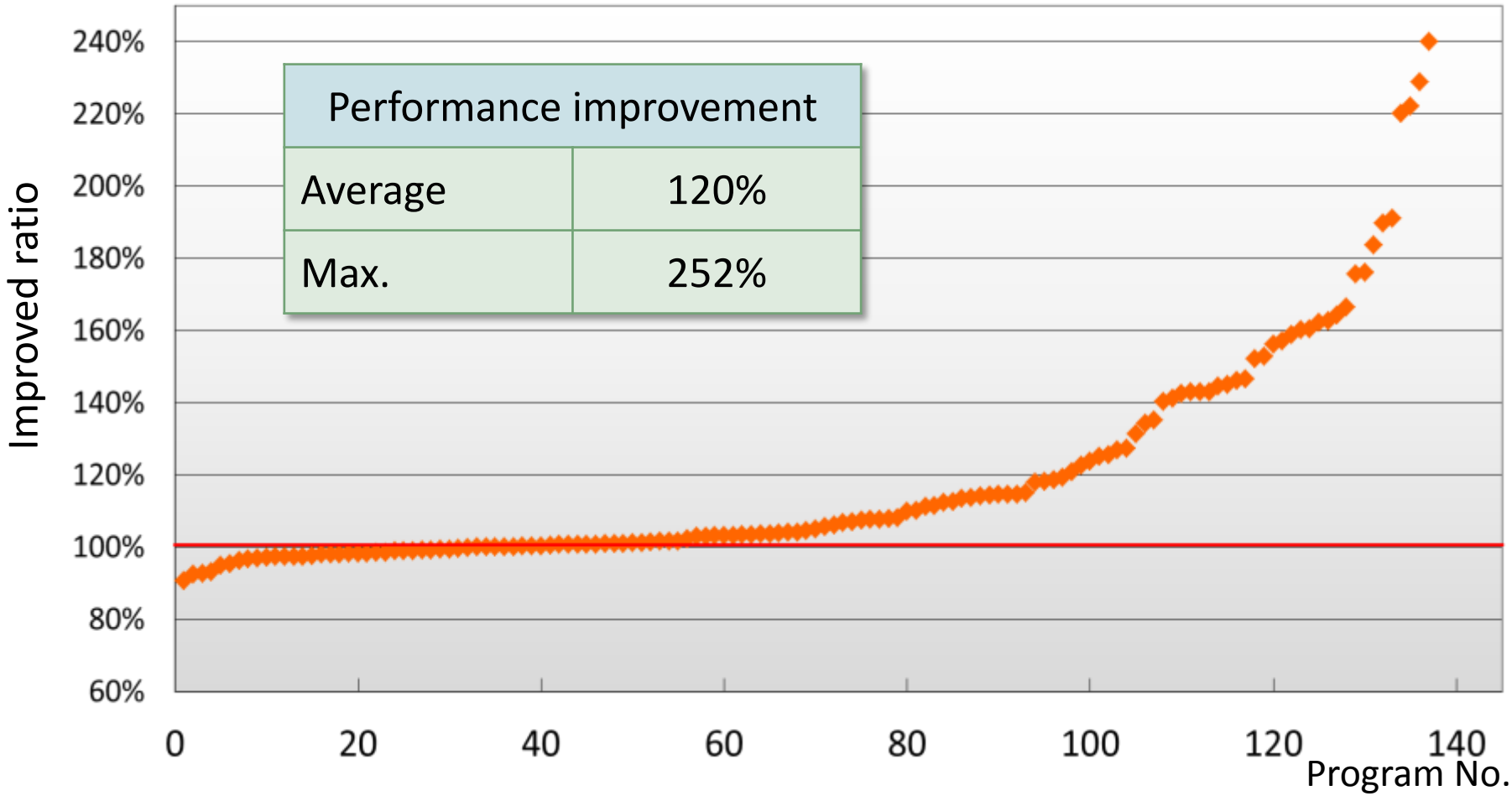
```
39   1              do j = jst, jend
                  <<< Loop-information  Start >>>
                  <<<  [OPTIMIZATION]
                  <<<    PREFETCH      : 8
                  <<<      c: 8
                  <<< Loop-information  End >>>
40   2    2            do i = ist, iend
41   2
42   2         c-------------------------------------
43   2         c  form the block daigonal
44   2         c-------------------------------------
45   2    2              tmp1 = 1.0d+00 / u(1,i,j,k)
46   2    2              tmp2 = tmp1 * tmp1
47   2    2              tmp3 = tmp1 * tmp2
48   2
49   2    2              d(1,1,i,j) =  1.0d+00
50   2              >                 + dt * 2.0d+00 * (   tx1 * dx1
51   2              >                                    + ty1 * dy1
52   2              >                                    + tz1 * dz1 )
53   2    2              d(1,2,i,j) =  0.0d+00
54   2    2              d(1,3,i,j) =  0.0d+00
55   2    2              d(1,4,i,j) =  0.0d+00
56   2    2              d(1,5,i,j) =  0.0d+00
 :
 :
367  2    2              c(5,3,i,j) = - dt * tx2
368  2              >       * ( - c2 * ( u(3,i-1,j,k)*u(2,i-1,j,k) ) * tmp2 )
369  2              >       - dt * tx1
370  2              >       * (  c34 - c1345 ) * tmp2 * u(3,i-1,j,k)
371  2    2              c(5,4,i,j) = - dt * tx2
372  2              >       * ( - c2 * ( u(4,i-1,j,k)*u(2,i-1,j,k) ) * tmp2 )
373  2              >       - dt * tx1
374  2              >       * (  c34 - c1345 ) * tmp2 * u(4,i-1,j,k)
375  2    2              c(5,5,i,j) = - dt * tx2
376  2              >       * ( c1 * ( u(2,i-1,j,k) * tmp1 ) )
377  2              >       - dt * tx1 * c1345 * tmp1
378  2              >       - dt * tx1 * dx5
379  2
380  2    2            end do
381  1              end do
```



HPC-ACE:Number of FP registers extension (2)

FUJITSU

- Performance boost by 256 FP registers w/ 138 application program kernels

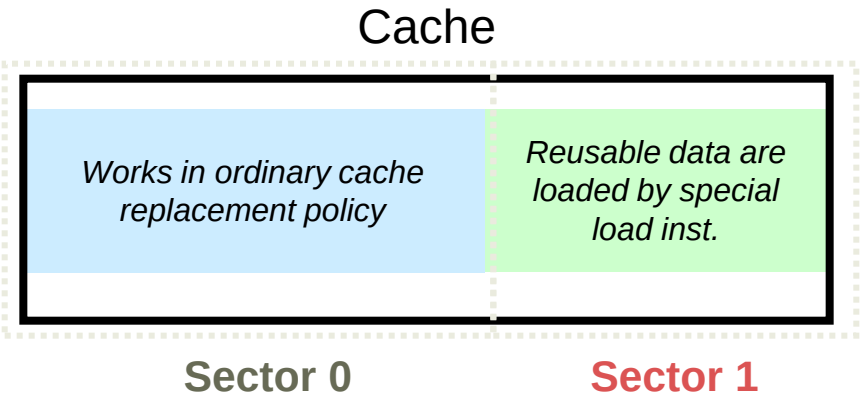


Performance improvement by # of FP registers extension(from 32 to 256)

HPC-ACE:Sector Cache(1)

FUJITSU

- Increasing the cache hit rate by selectively leave a reused data in the cache
  - The cache is divided into two sectors (Sectors 0 and 1).
  - Sector 1 is used for data that will be reused.
  - Sector 0 is used for other data.
  - ➔ Data in Sector 1, which will be used again soon, is no longer removed from cache, by the access of data that uses Sector 0.
  - The user can specify the data to be retained in Sector 1 by specifying it on the compiler directive line.




```
!ocl CACHE_SECTOR_SIZE(N1,N2)
!ocl CACHE_SUBSECTOR_ASSIGN(a)
do j=1,m
   do i=1,n
      a(i) = a(i) + b(i,j) * c(i,j)
   enddo
Enddo
```

Dividing N ways of the L2 cache as follows:
N1: Sector 0
N2: Sector 1

- Array **a** is held in Sector 1.
- All others are held in Sector 0.

Array **a** is no longer removed from the cache by references to array b or c.

HPC-ACE:Sector Cache (2)

FUJITSU

- ***NPB3.3-CG*** case
  - By putting array P on sector 1, floating point data cache access wait is reduced

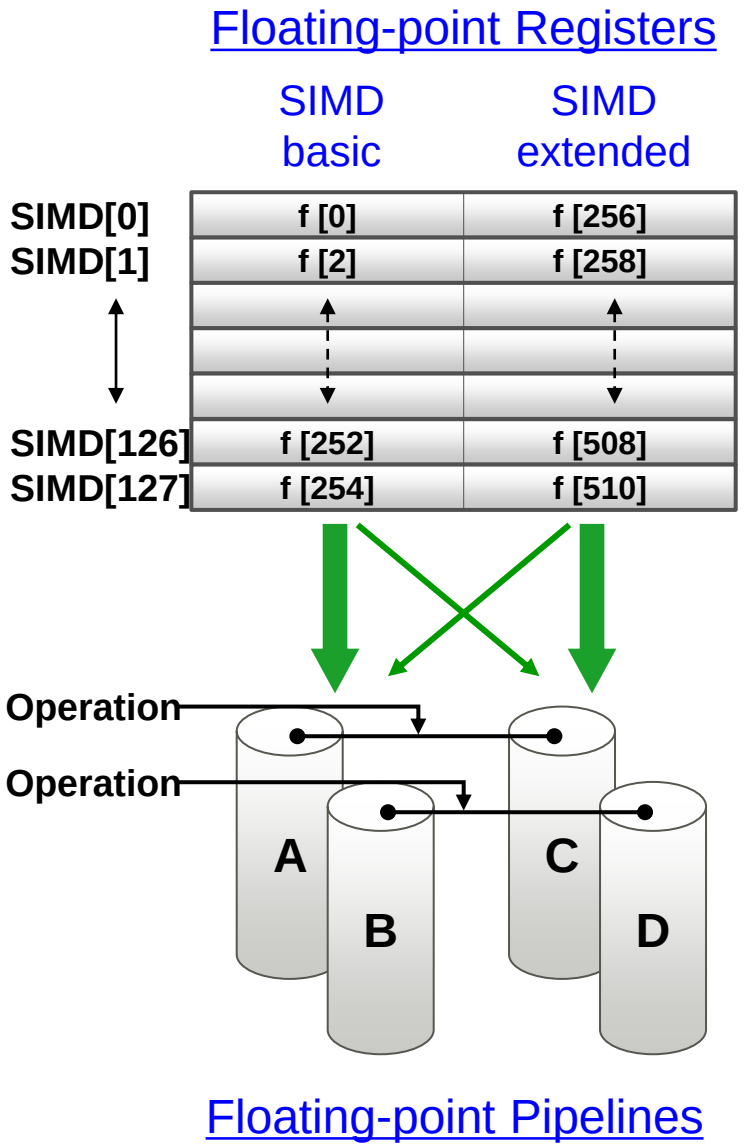
Optimized code

```
111              !ocl CACHE_SECTOR_SIZE(4,8)
112              !ocl CACHE_SUBSECTOR_ASSIGN(p)
113
120   1          !----- npb_cg kernel loop -----
                 <<< Loop-information Start >>>
                 <<< [PARALLELIZATION]
                 <<<   Standard iteration count: 4
                 <<< Loop-information  End >>>
121   2 pp             do j=1,n
122   2  p               sum = 0.d0
                 <<< Loop-information Start >>>
                 <<< [OPTIMIZATION]
                 <<<   SIMD
                 <<<   SOFTWARE PIPELINING
                 <<< Loop-information  End >>>
123   3  p  4v           do k=rowstr(j), rowend(j) ! 64LOOP
124   3  p  4v             sum = sum + a(k) * p(colidx(k))
125   3  p  4v           enddo
126   2  p              q(j) = sum
127   2  p            enddo
128   1          !-----------------------------
133
134              !ocl END_CACHE_SUBSECTOR
135              !ocl END_CACHE_SECTOR_SIZE
```

HPC-ACE: SIMD (Single Instruction Multiple Data)

FUJITSU

- Eight floating-point ops can be executed simultaneously per core
  - Two SIMD instructions can be executed simultaneously per core
  - SIMD instruction executes two floating-point ops (single or double precision)
  - FMA is supported
- Software can flexibly perform SIMD optimization
  - It is possible to execute operations in SIMD by obtaining pieces of data one by one from noncontiguous memory spaces
  - It is possible to selectively store floating register into memory (mask operation)

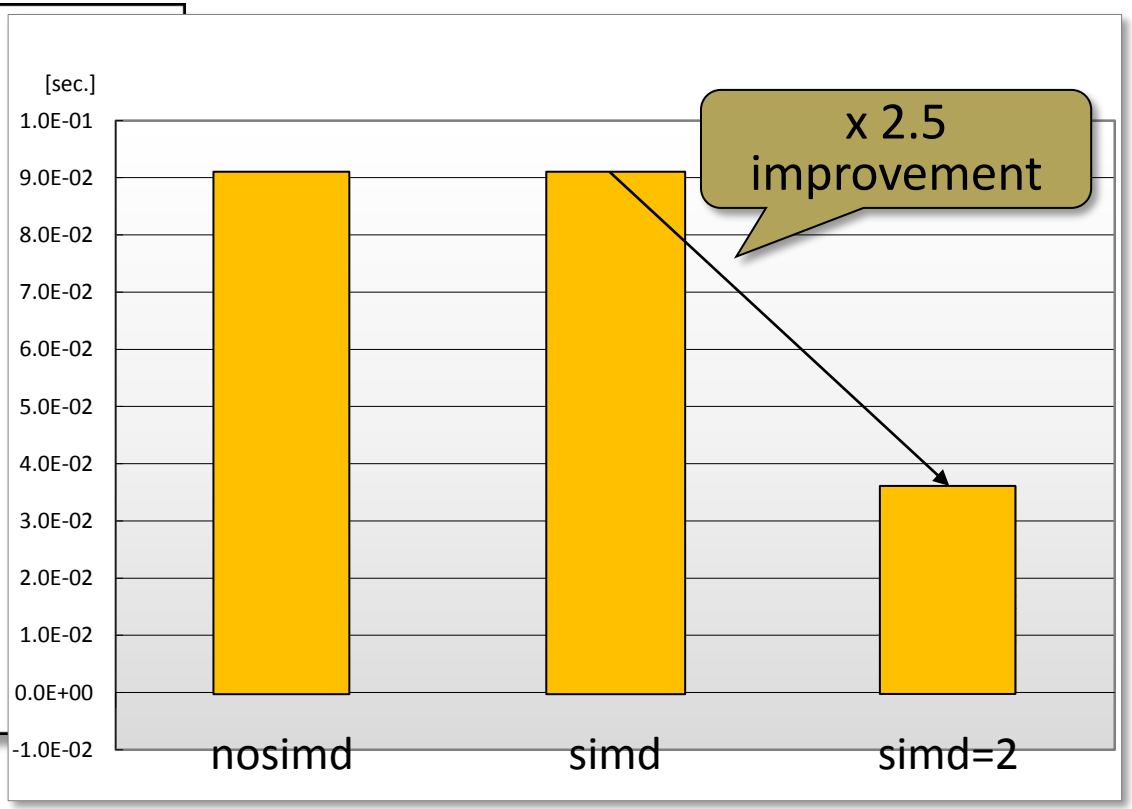


HPC-ACE:SIMD extension (mask operation effect)

FUJITSU

- Example of ***Computational chemistry program***
  - Due to the branch operation, “if” in the loop, SIMD option shows NO effect
  - By using mask operation, compiler can SIMDize the loop and utilize software pipelining. Results 2.5x performance improvement

```
40   1                    do iv=1, natv
41   1                !ocl unroll(4)
42   1                !$omp parallel do default(none)
43   1                !$omp+private(earg,work)
44   1                !$omp+shared(ngr,iv,uuv,tuv,tuvres)
                    <<< Loop-information Start >>>
                    <<<  [OPTIMIZATION]
                    <<<    SIMD
                    <<<    SOFTWARE PIPELINING
                    <<< Loop-information  End >>>
45   2  p  4v             do ig=1, ngr
46   2  p  4v               earg = - uuv(ig,iv) + tuv(ig,iv,1)
47   3  p  4v               if (earg >= 0) then
48   3  p  4v                 work = 1.0d0 + earg
49   3  p  4v               else
50   3  p  4v                 work = exp(earg)
51   3  p  4v               endif
52   2  p  4v               tuvres(ig,iv,1) = work
53   2  p  4v             enddo
54   1                  enddo
55                  !$omp parallel
```

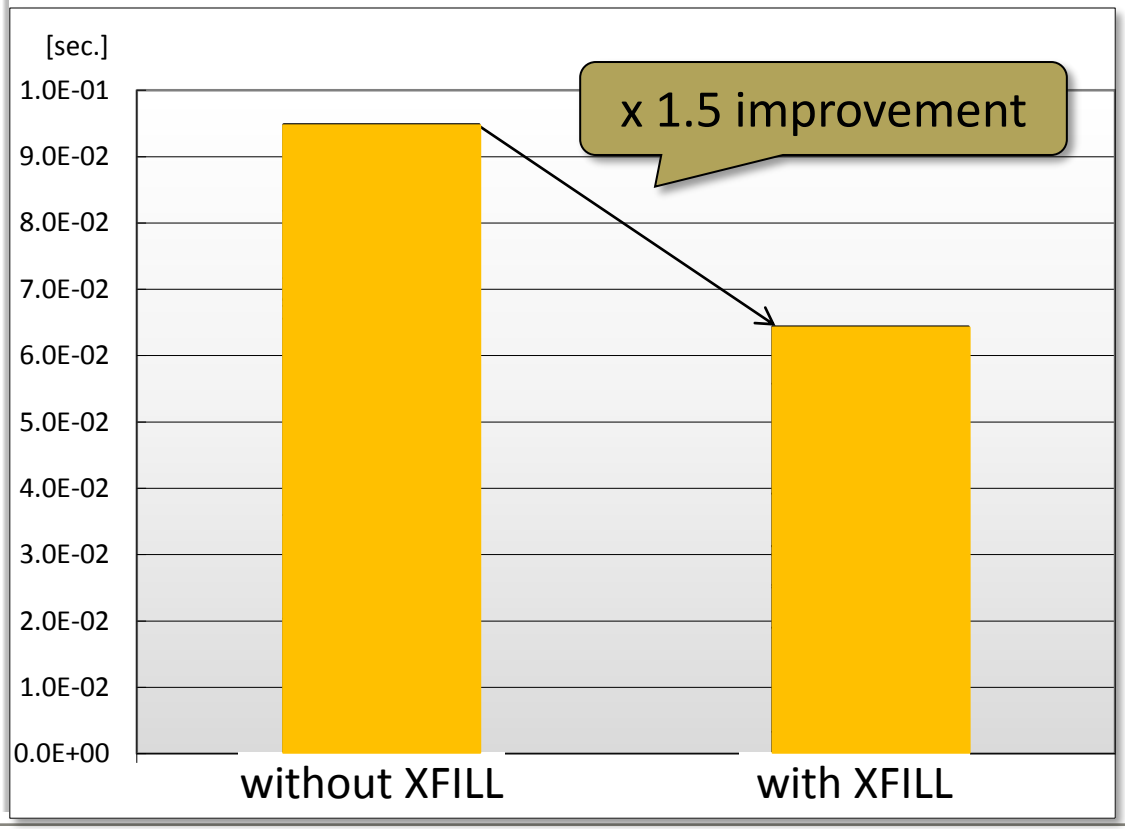


HPC-ACE:XFILL capability

FUJITSU

- XFILL capability works in ***Earthquake simulation program***
  - XFILL fills L2 cache line with undetermined data(allocate cache line without data load)
  - So, with XFILL in advance, following FP reg store instructions should hit and would not cause data load from memory
  - XFILL can reduce memory read accesses and improve performance when a memory throughput is the bottleneck

```
  7     integer, parameter :: NXP    = 400   ! X-size per 1 pro
  8     integer, parameter :: NYP    = 200   ! Y-size per 1 pro
  9     integer, parameter :: NZ     = 4300  ! Z-size per 1 pro
184     1  pp    do J = 1, NY
185     2  p       do I = 1, NX
        <<< Loop-information Start >>>
        <<< [OPTIMIZATION]
        <<<   SIMD
        <<<   SOFTWARE PIPELINING
        <<<   PREFETCH      : 2
        <<<     DZV: 2
        <<<   XFILL         : 2
        <<<     DZV: 2
        <<< Loop-information  End >>>
186     3  p   v    do K = 3, NZ-1
187     3  p   v      DZV (k,I,J) = (V(k,I,J) -V(k-1,I,J))*R40 &
188     3                          - (V(k+1,I,J)-V(k-2,I,J))*R41
189     3  p   v    end do
```



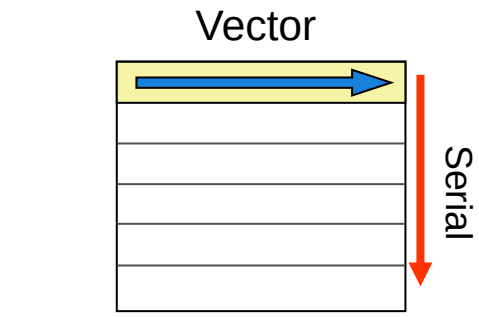
VISIMPACT technology

FUJITSU

- Fine-grain thread-parallelization
  - Low-overhead barrier synchronization with HPC-ACE ASI registers
  - Coalesced memory access exploits shared L2 cache
  - "**Vi**rtual **S**ingle Processor by **I**ntegrated **M**ulti-core **P**arallel **Arc**hi**t**ecture"

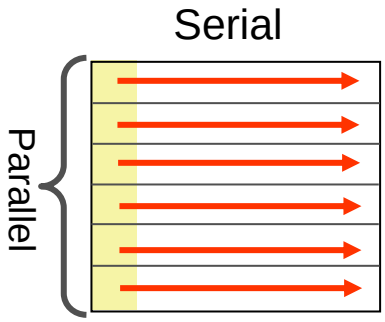
Vectorization

```
V  DO J=1,N
V    DO I=1,M
V      A(I,J)=...
V    END
   END
```



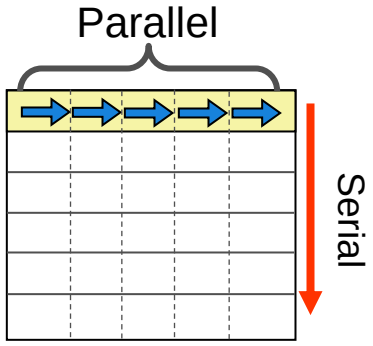
Conventional Threading

```
P  DO J=1,N
P    DO I=1,M
P      A(I,J)=...
P    END
P  END
```



VISIMPACT

```
P  DO J=1,N
P    DO I=1,M
P      A(I,J)=...
P    END
   END
```

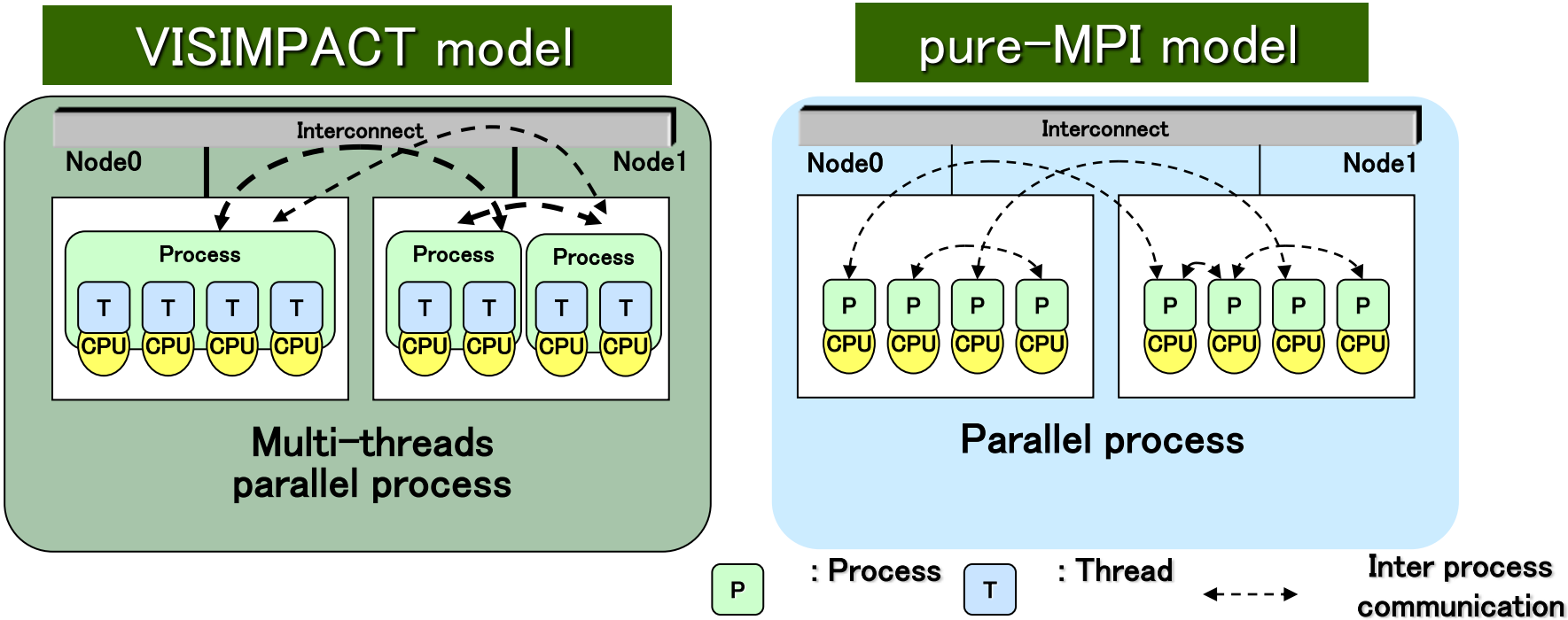


requires separate or large L2 cache

- Fujitsu compilers support VISIMPACT automatic parallelization

VISIMPACT technology

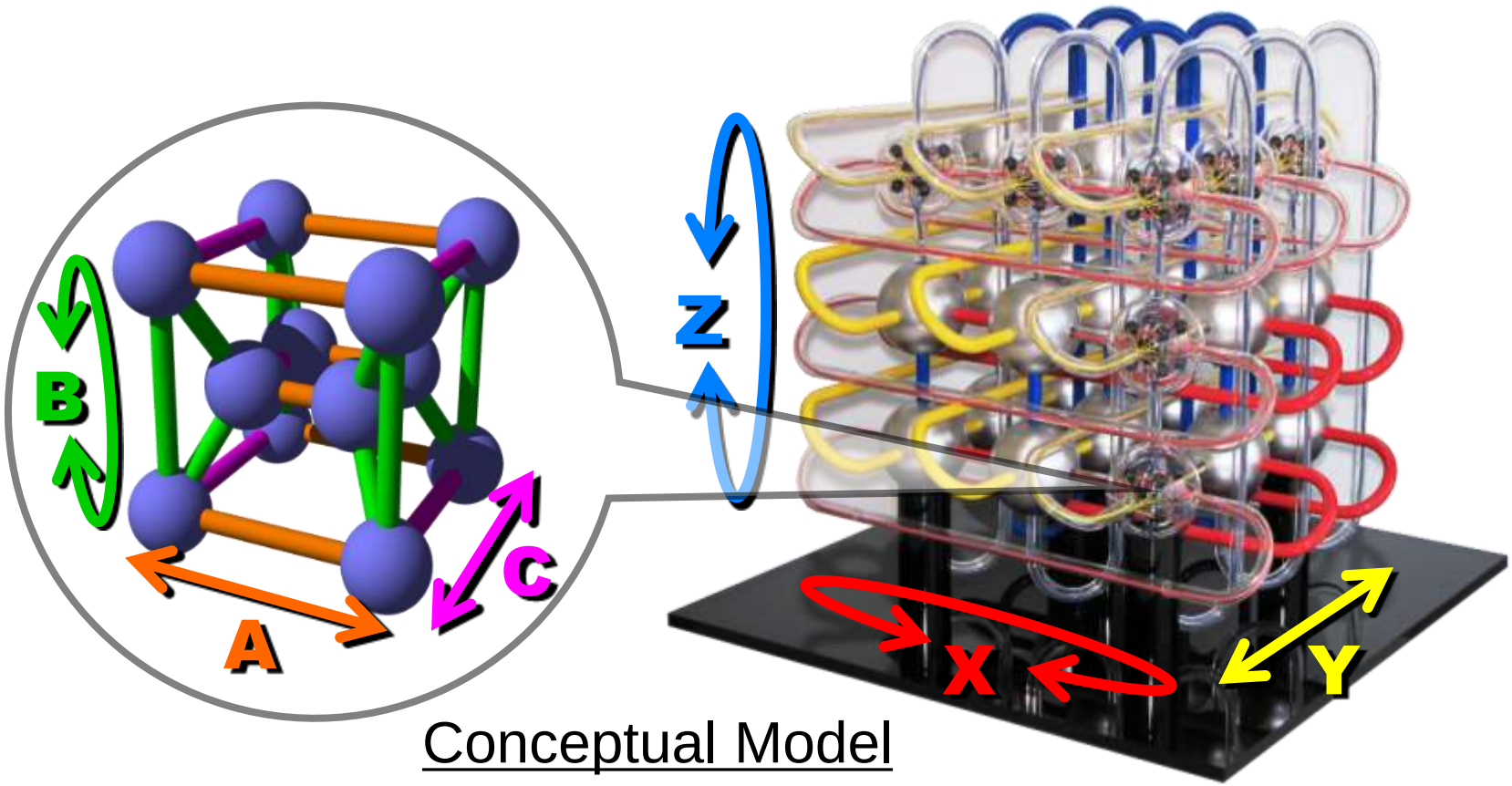
- Fujitsu compiler transforms MPI programs to hybrid parallel executions automatically, by parallelizing a process on a CPU into multi-threads to cores
- By reducing the number of ranks, communication efficiency would be improved
- Inter-core hardware barrier and shared L2 cache help efficient execution



6D-Mesh/Torus Network Topology

FUJITSU

- Higher bisection bandwidth and smaller hops than 3D-Torus
- **To**rus **fu**sion
  - Every XYZ Cartesian grid point has another ABC 3D-Torus
  - X, Z and B are torus (ring) axes
  - A, C and Y are mesh (linear) axes

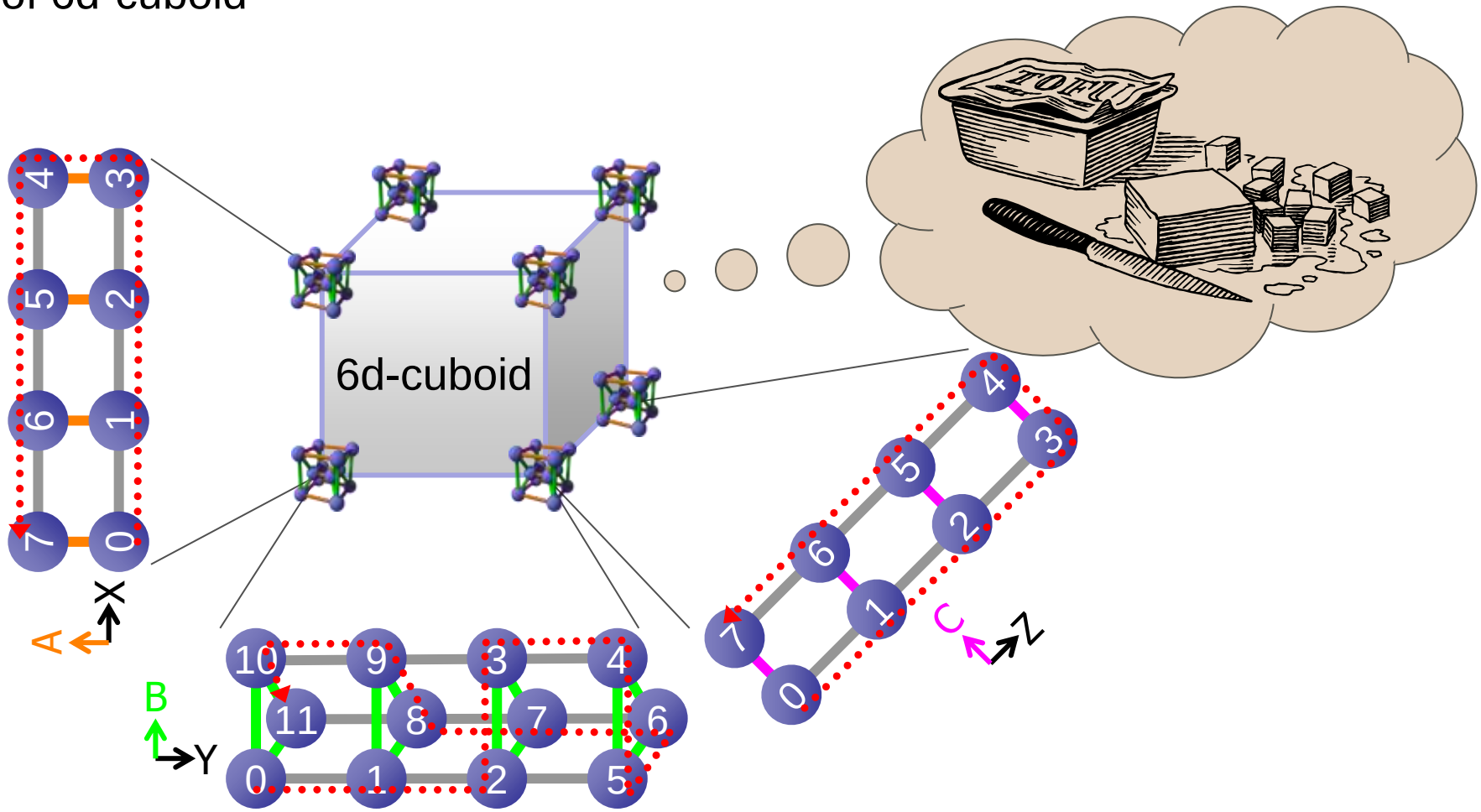


Conceptual Model

Virtual Topology

FUJITSU

- System software generates virtual 1d-, 2d- or 3d-torus for an arbitrary size of 6d-cuboid



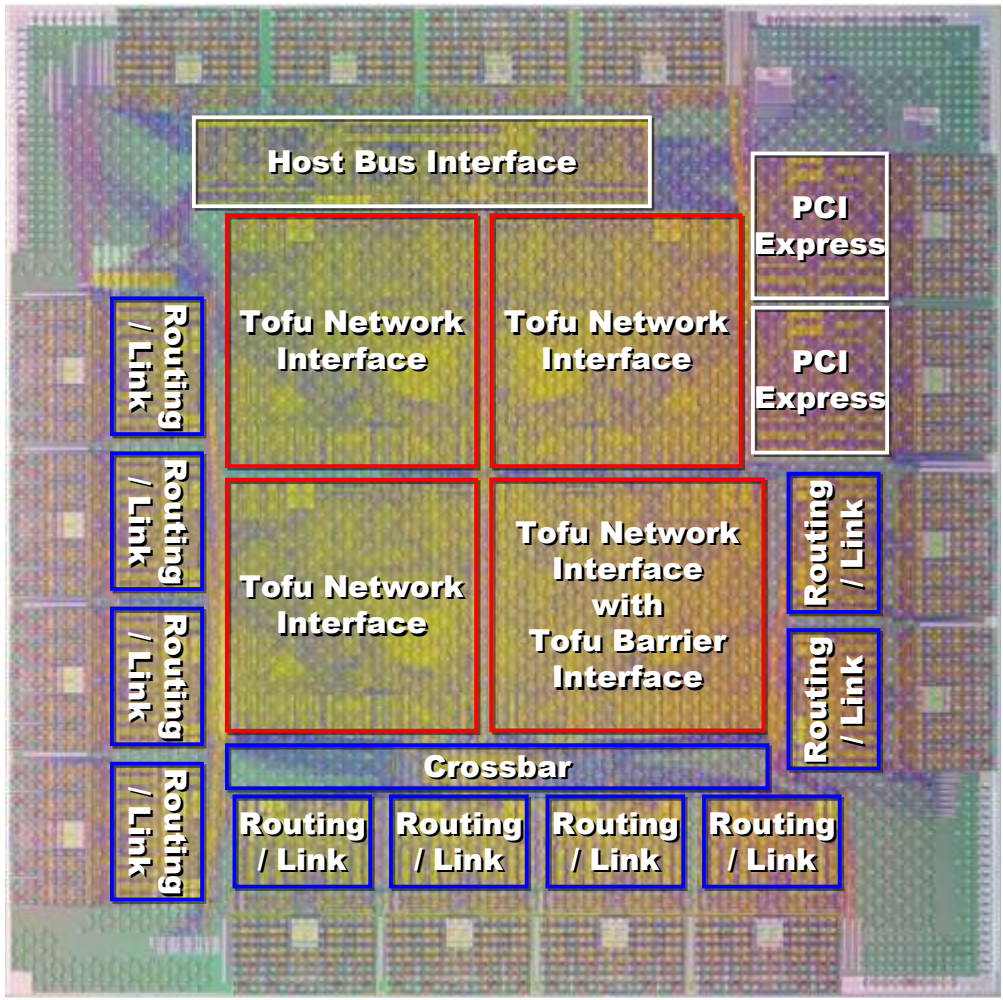
- Virtual topology expands the range of applicable algorithms

ICC : Tofu Interconnect Controller

FUJITSU

- Companion chip for SPARC64™ VIIIfx / IXfx processors
- Tofu Interconnect
  - 4 Tofu Network Interfaces
  - Tofu Network Router
- PCI Express Gen2
  - 2 ports for I/O nodes
- Water-cooled

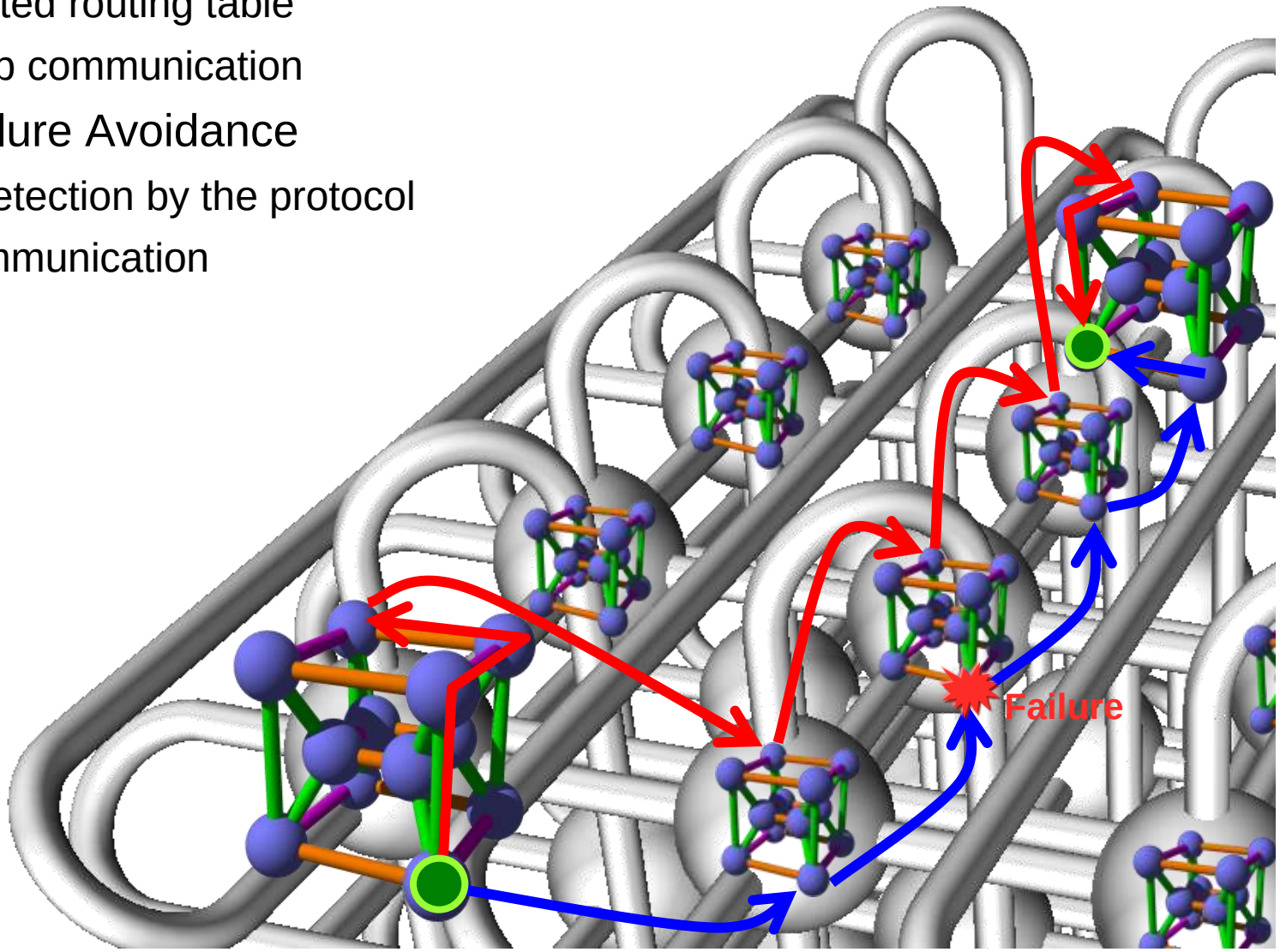
Process technology	65 nm
Die size	18.2 mm x 18.1 mm
Frequency	312.5 MHz
No. of Tofu link	10 ports
Tofu link throughput	in 5 GB/s + out 5 GB/s
PCI Express Gen2	8 lane × 2 ports
Host Bus Interface	in 20 GB/s + out 20 GB/s
Power consumption	28 W (typical)
No. of transistors	200 million
Signal Transfer Speed	6.25 Gbps
Differential signals	128 lanes



Static and Dynamic Failure Avoidance

FUJITSU

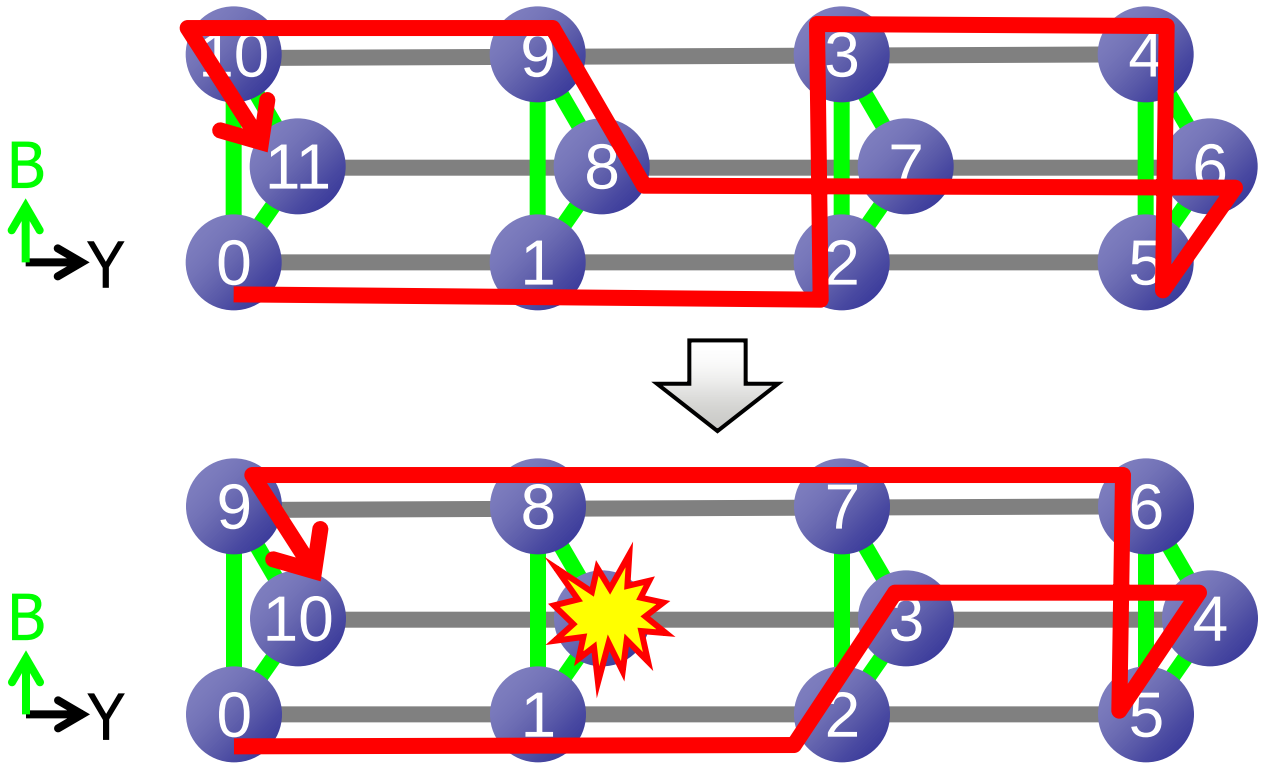
- Static Failure Avoidance
  - Pre-calculated routing table
  - For intra-job communication
- Dynamic Failure Avoidance
  - Time-out detection by the protocol
  - For I/O communication



Fault Isolation by Virtual Topology

FUJITSU

- Jobs using virtual topology can use rectangle region including failed node

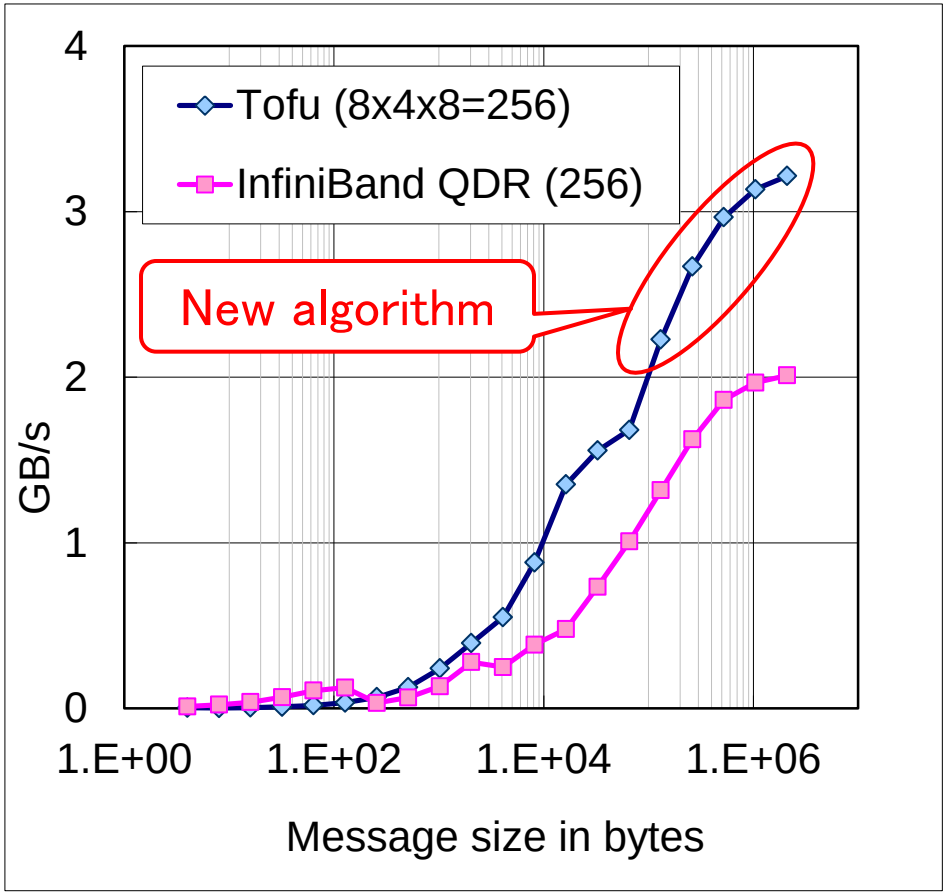


- Decreases in executable job size and in system availability are minimized

All-to-all communication performance

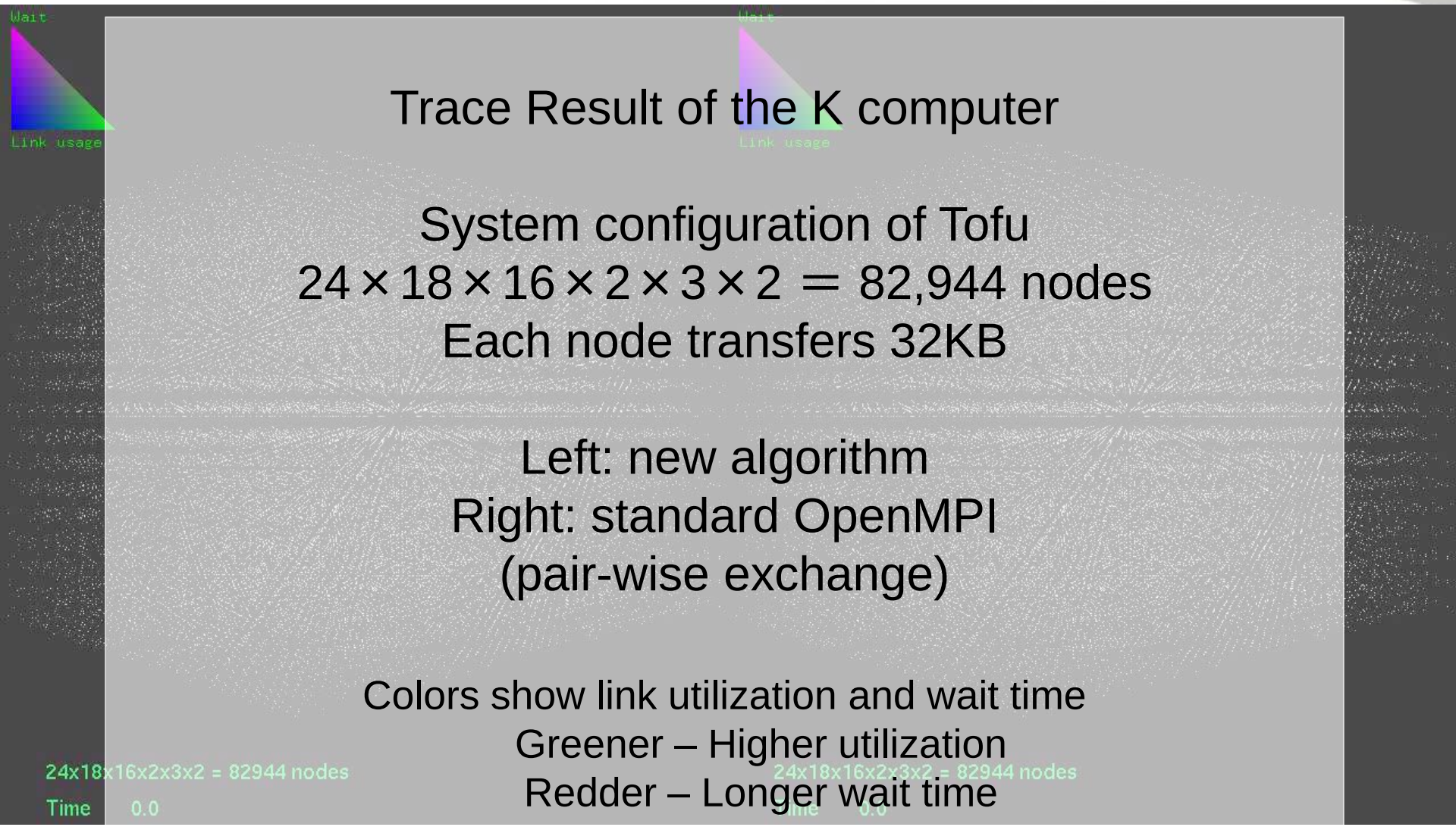
FUJITSU

- Link utilization is important for actual communications
- New optimized algorithm
  - Uses all links uniformly to maximize All-to-All communication performance
  - Four RDMA engines execute 4 sends and 4 receives simultaneously
- Using Tofu features
  - Virtual 3D-Torus
  - Flow-control features
    - for congestion prevention
- Many applications use All-to-All type of communication and enjoy this acceleration



All-to-all communication trace on Tofu

FUJITSU



Trace Result of the K computer

System configuration of Tofu
$24 \times 18 \times 16 \times 2 \times 3 \times 2 = 82{,}944$ nodes
Each node transfers 32KB

Left: new algorithm
Right: standard OpenMPI
(pair-wise exchange)

Colors show link utilization and wait time
Greener – Higher utilization
Redder – Longer wait time

New Algorithm
Elapsed Time: 2.77sec

Standard OpenMPI
(pair-wise exchange)
Elapsed Time: 24.08sec

FX10 Software Stack

FUJITSU

Applications

HPC Portal / System Management Portal

Technical Computing Suite

System Management

- System management
- System control
- System monitoring
- System operation support

Job Management

- Job manager
- Job scheduler
- Resource management
- Parallel job execution

High Performance Parallel File System *FEFS*

- Lustre based high performance distributed file system
- High scalability, high reliability and availability

Automatic parallelization compiler

- Fortran
- C
- C++

Tools and math. libraries

- Programming support tools
- Mathematical libraries (SSL II/BLAS etc.)

Parallel languages and libraries

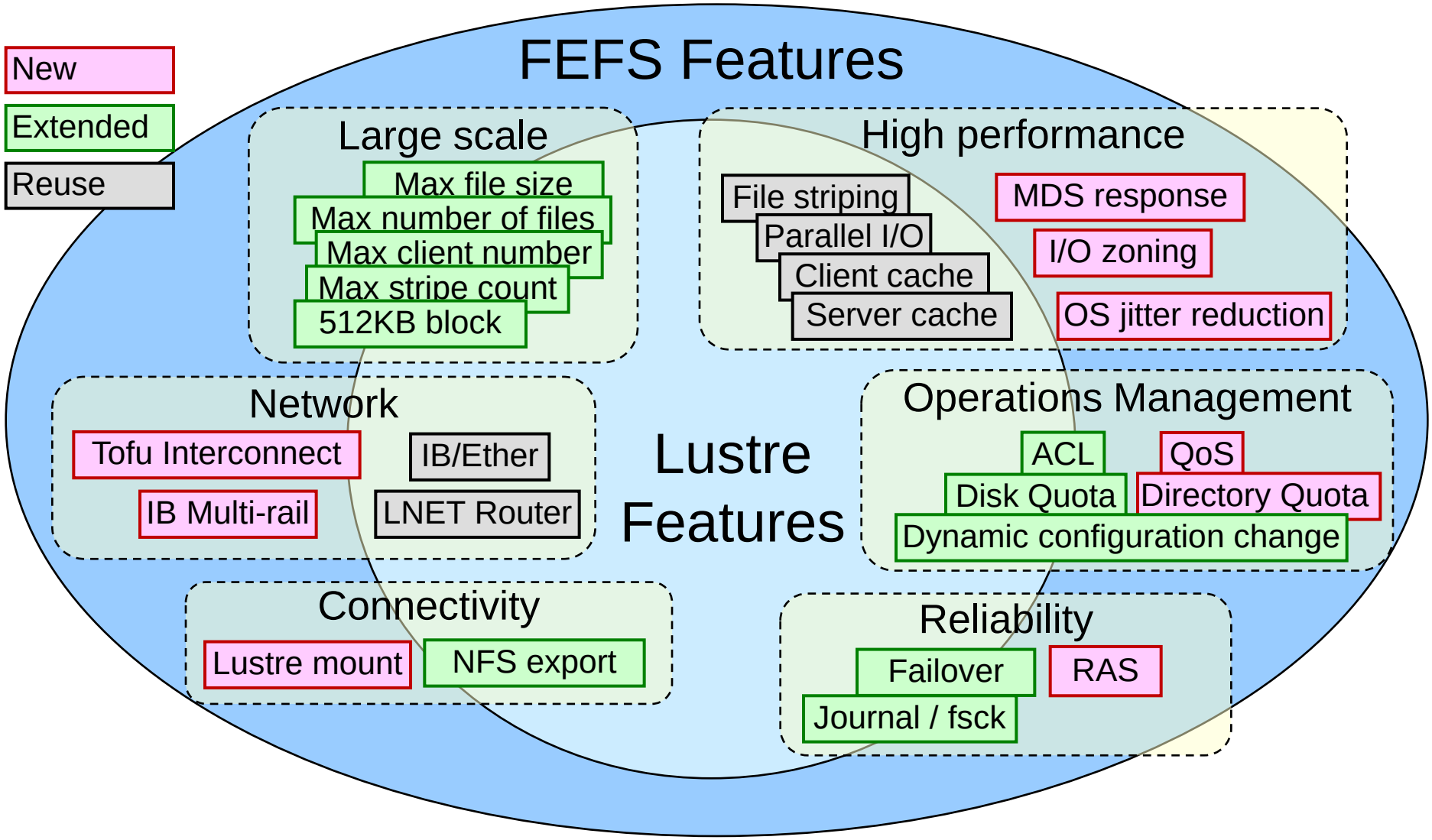
- OpenMP
- MPI
- XPFortran

Linux based OS enhanced for FX10

PRIMEHPC FX10

Lustre Extension of FEFS: Features

FUJITSU

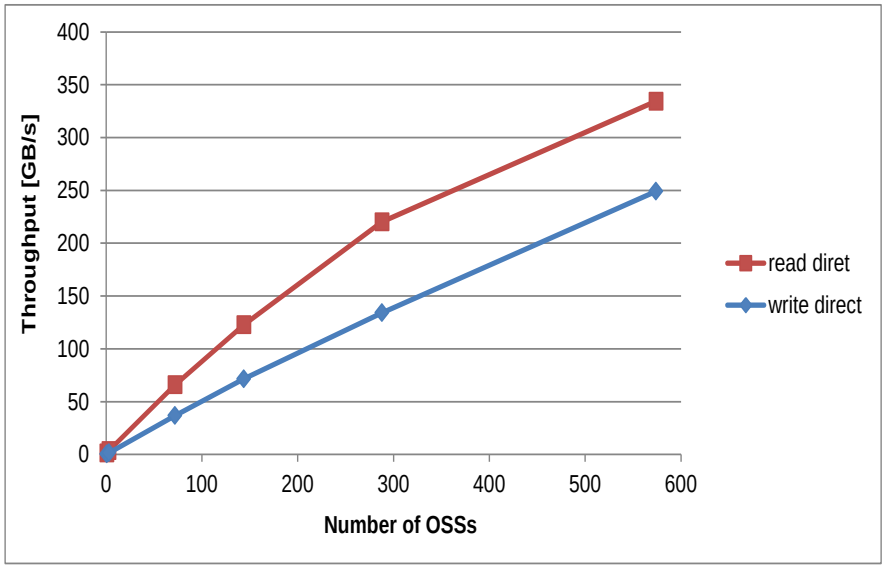


FEFS performance

FUJITSU

* : Collaborative work with RIKEN on the K computer

- Achieved the world's top-level throughput*
  - Read 334GB/s, Write 249GB/s (574 OSSs, 18432 Clients, 192 racks)



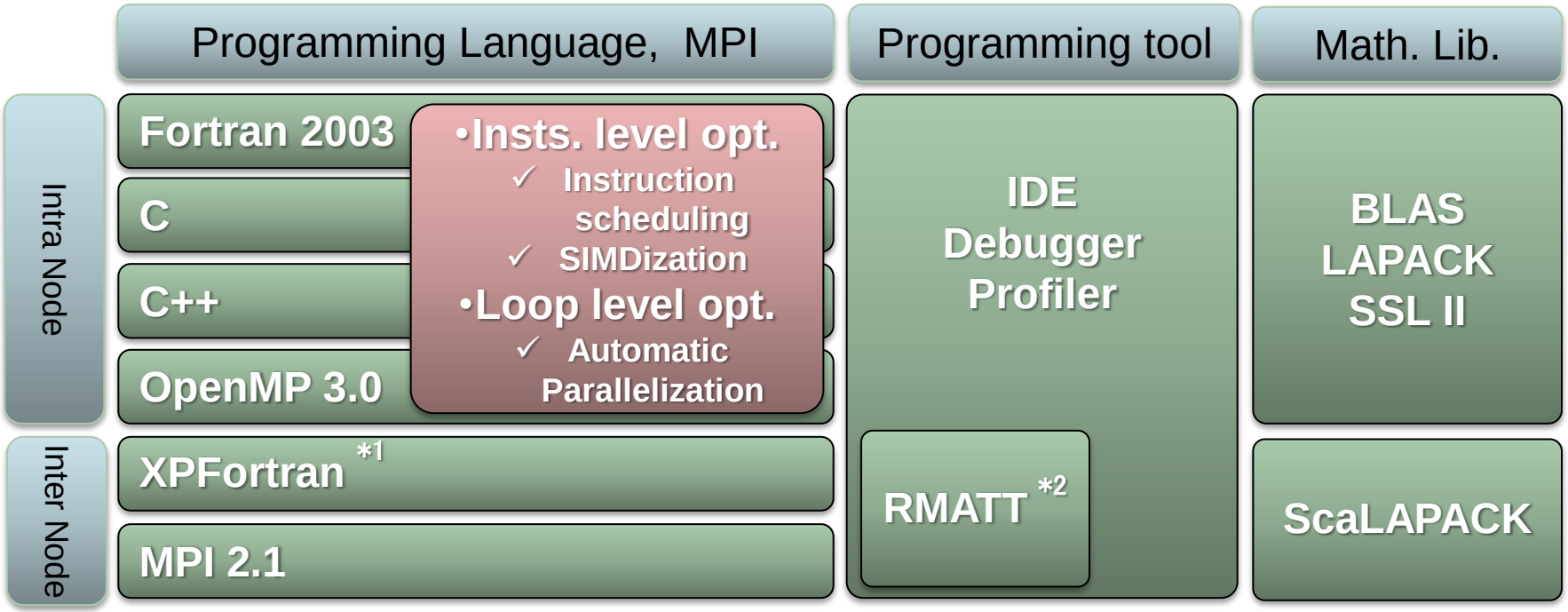
- Metadata performance of mdtest* (distributed directory)

	FEFS		Lustre	
IOPS	K computer**	IA***	IA***	
			1.8.5	2.0.0.1
create	34697.6	31803.9	24628.1	17672.2
unlink	39660.5	26049.5	26419.5	20231.5
mkdir	87741.6	77931.3	38015.5	22846.8
rmdir	28153.8	24671.4	17565.1	13973.4

** : MDS:RX300S6 (X5680 3.33 GHz 6core x2, 48GB, IB(QDR)x2)
*** : MDS:RX200S5 (E5520 2.27GHz 4core x2, 48GB, IB(QDR)x1)

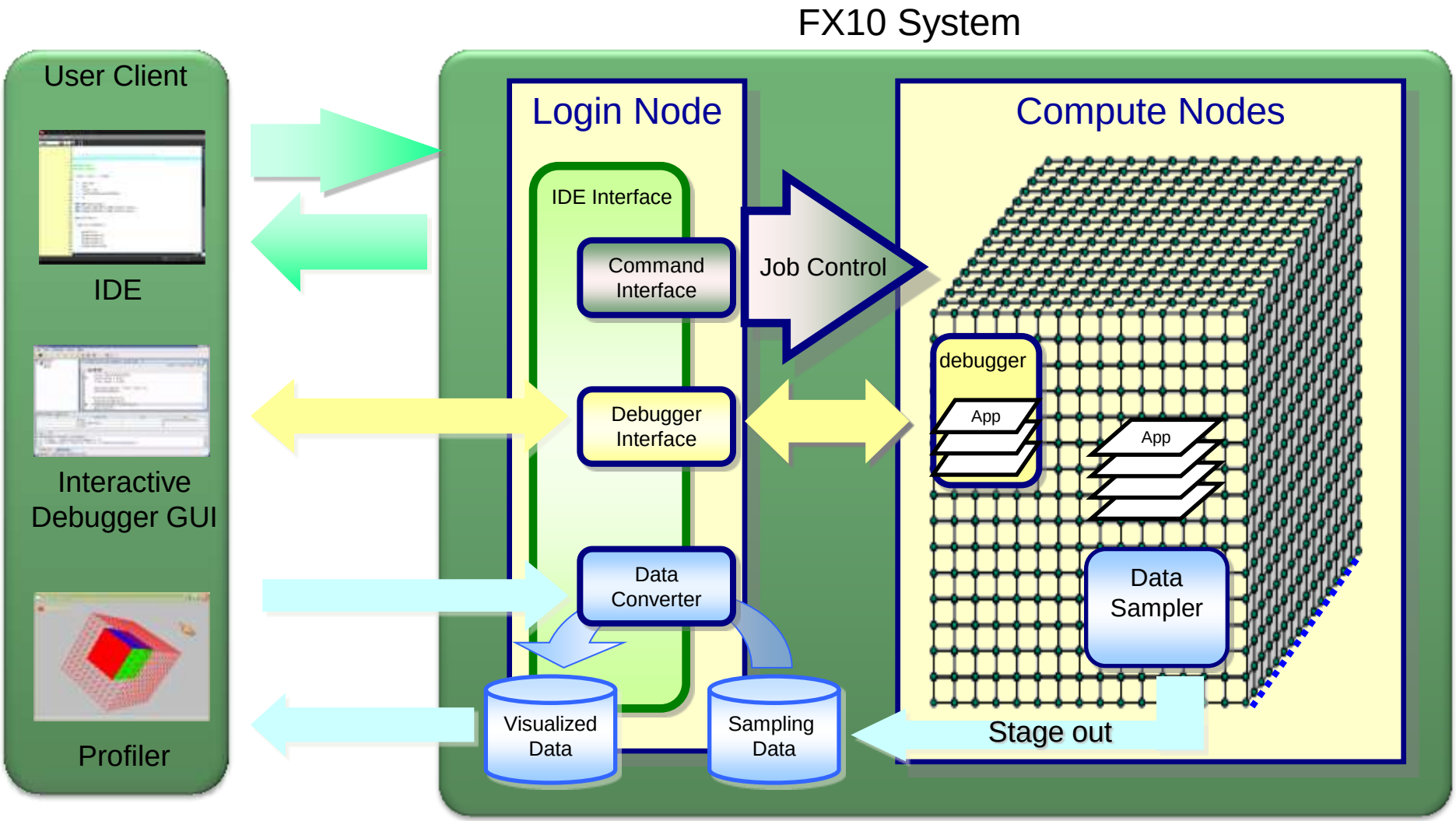
Language System overview

- Fortran C/C++/Fortran Compiler
- Programming model (OpenMP, MPI, XPFortran)
- Instruction level /Loop level optimization using HPC-ACE
- Debugging and Tuning tools for highly parallel computer



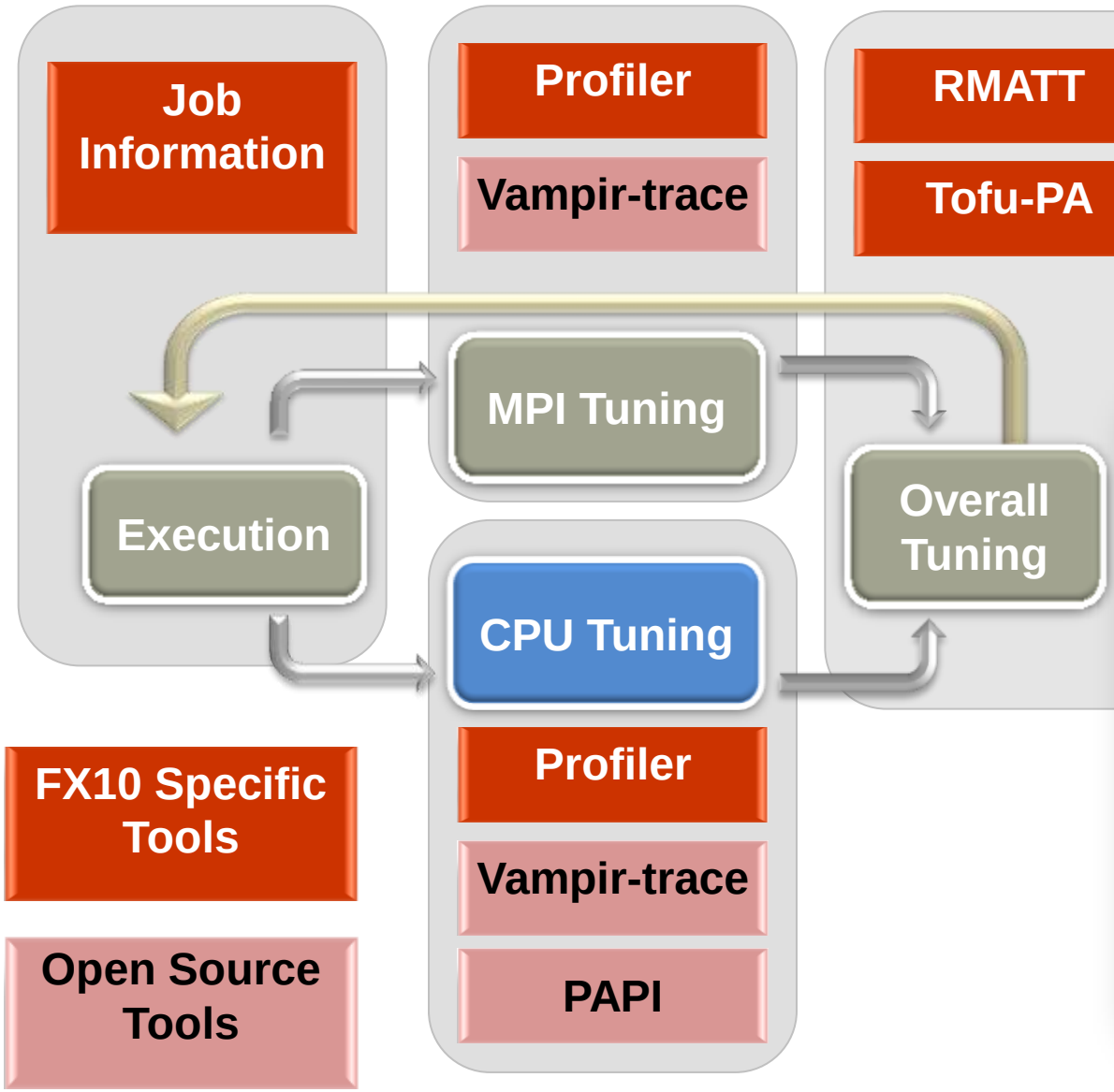
*1: eXtended Parallel Fortran (Distributed Parallel Fortran)
*2: Rank Map Automatic Tuning Tool

Programming Environment

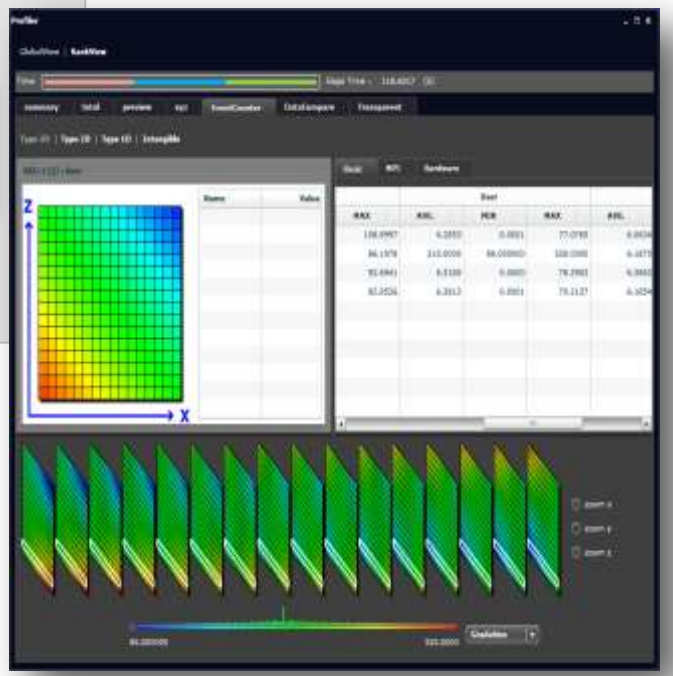


Application Tuning Cycle and Tools

FUJITSU



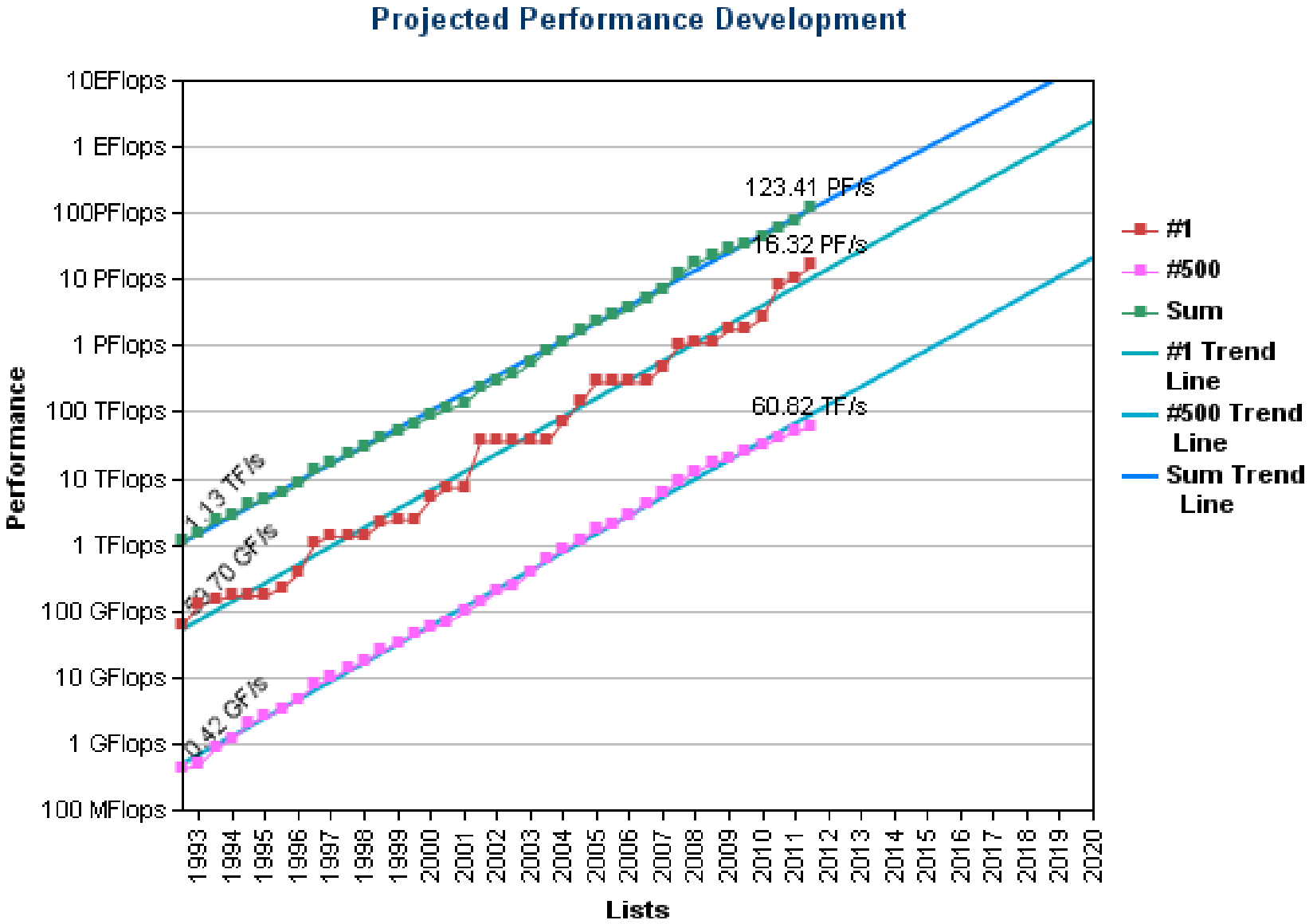
Profiler snapshot



On Course to Exascale

FUJITSU

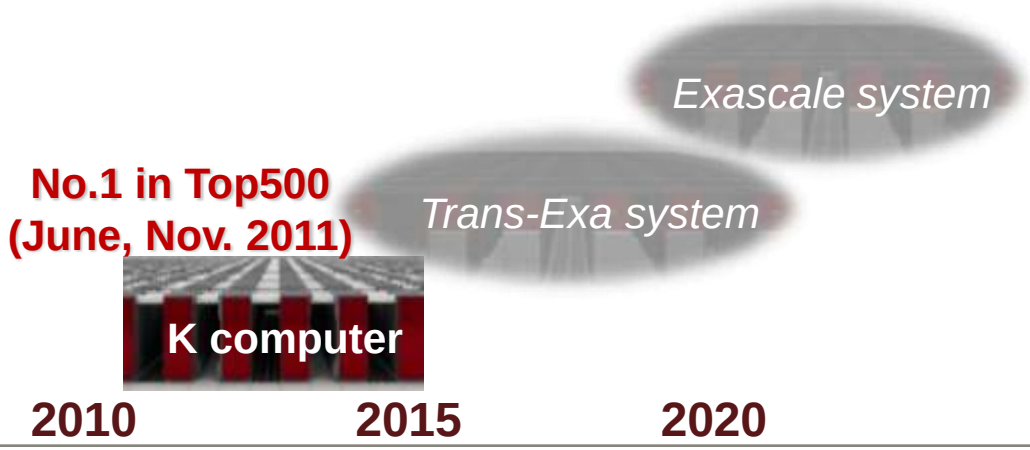
- World's first 1 Exa-Flops computer is expected to appear by 2020



Towards exascale

FUJITSU

- Realization of Exascale system is grand challenge
  - At least two-step development is necessary
  - The biggest challenge is high density and low power consumption
- Fujitsu is developing a Trans-Exa system as a midterm goal
  - The Trans-Exa system is expected to be scalable to 100 Petaflops
  - Employs
    - Wide SIMD and multicore CPU
    - High performance and lower power consumption interconnect
    - High performance and high density memory technologies
- Continues to invest effort in research for the exascale system
  - Higher performance and lower power consumption technologies
  - Technologies for higher reliability



Key technology developments on Trans-Exa

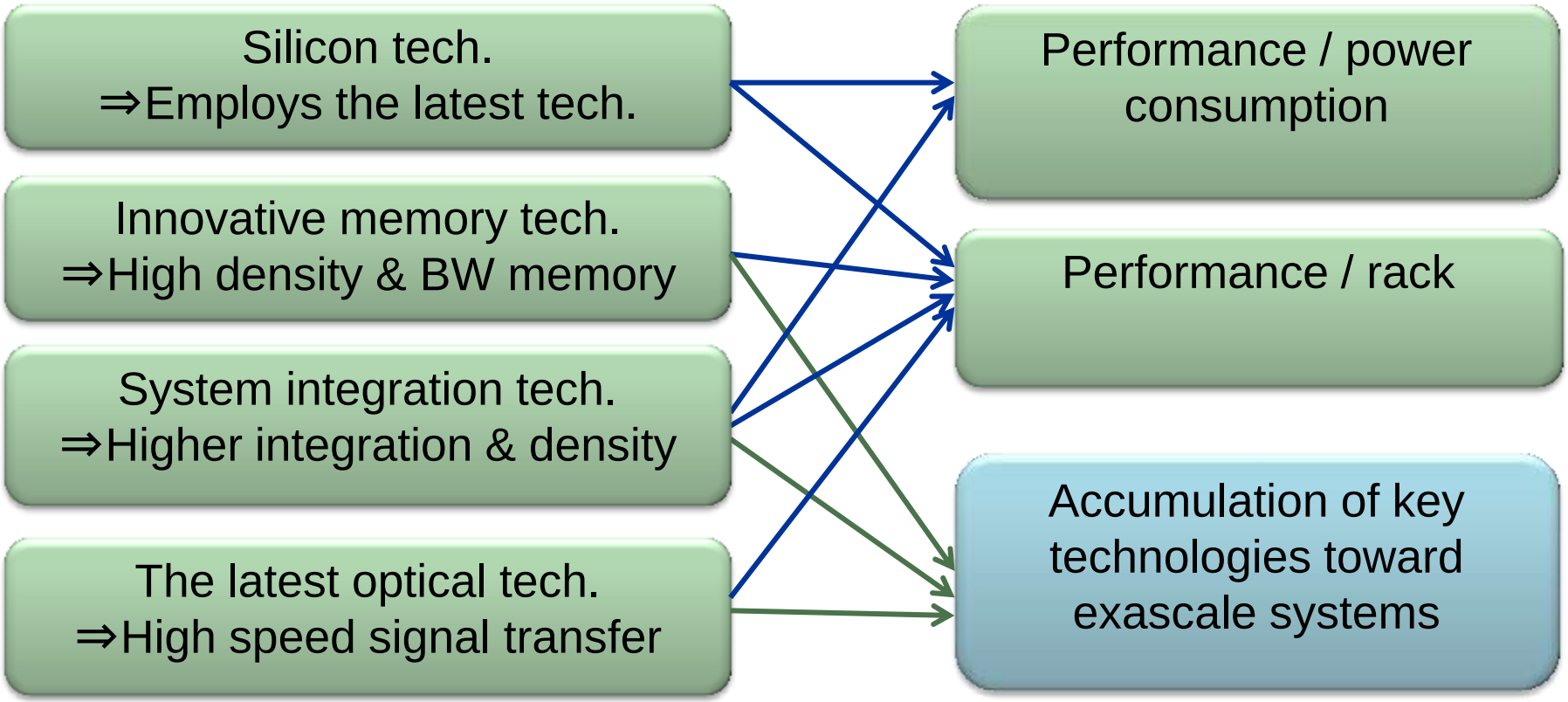
FUJITSU

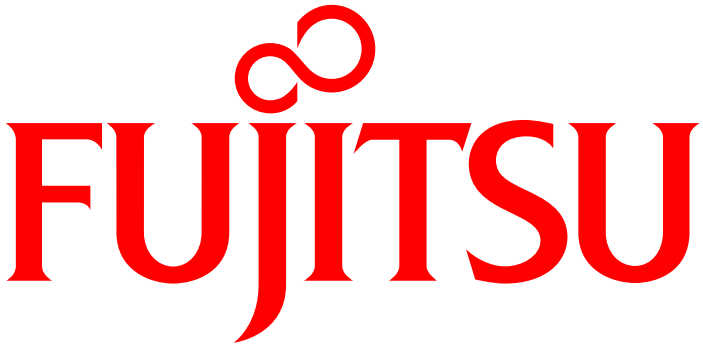
Goal

Significant improvement of power efficiency, high density

Technology

Gains





shaping tomorrow with you