FUJITSU

shaping tomorrow with you

PRIMEHPC FX10: Advanced Software

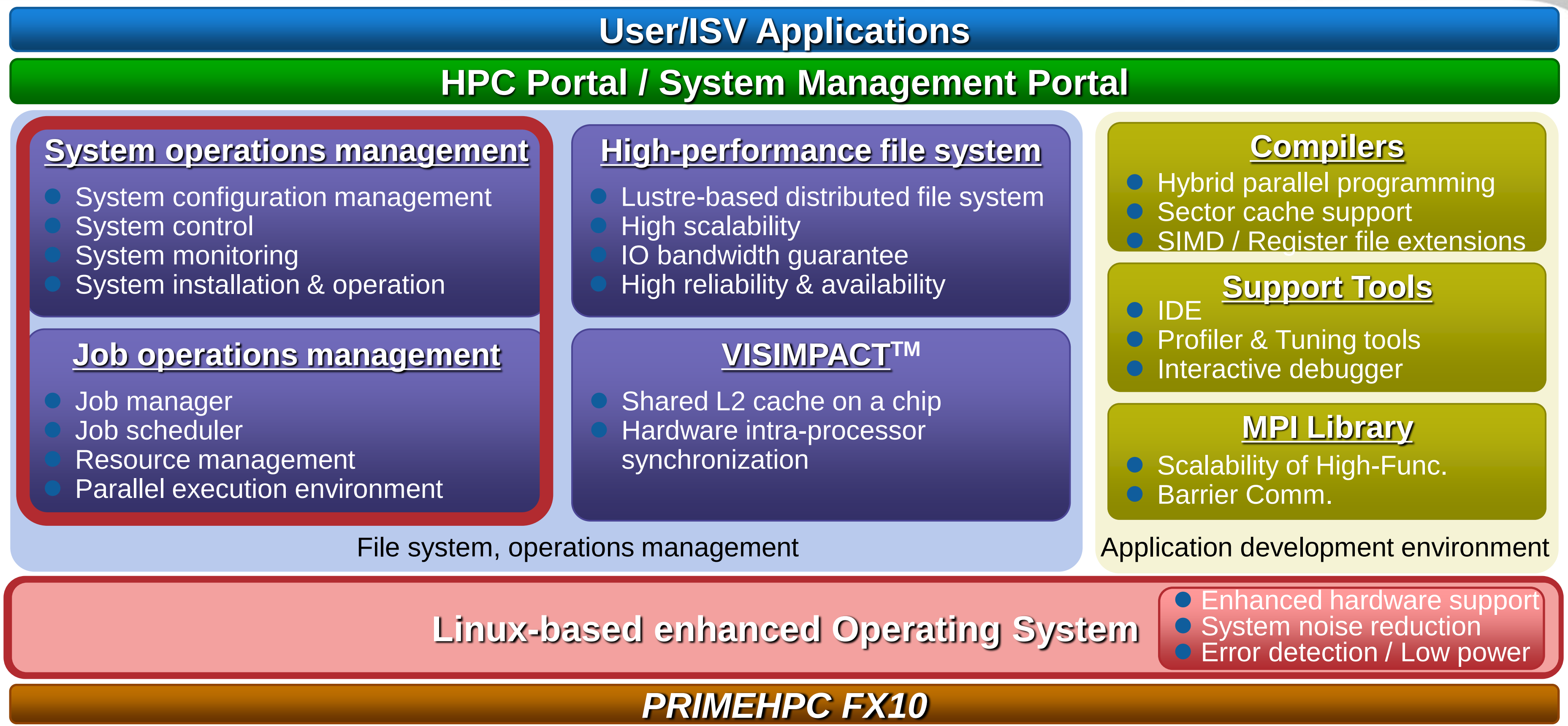
Koh Hotta

Fujitsu Limited

System Software supports ---

- Stable/Robust & Low Overhead Execution of Large Scale Programs
  - Operating System
  - File System
- Program Development for High Speed Execution
    - Just Compile and Enjoy High Performance
  - Compiler
  - MPI
  - Tuning Support Environment

System Software Stack



User/ISV Applications

HPC Portal / System Management Portal

System operations management

- System configuration management
- System control
- System monitoring
- System installation & operation

Job operations management

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

High-performance file system

- Lustre-based distributed file system
- High scalability
- IO bandwidth guarantee
- High reliability & availability

VISIMPACT™

- Shared L2 cache on a chip
- Hardware intra-processor synchronization

File system, operations management

Compilers

- Hybrid parallel programming
- Sector cache support
- SIMD / Register file extensions

Support Tools

- IDE
- Profiler & Tuning tools
- Interactive debugger

MPI Library

- Scalability of High-Func.
- Barrier Comm.

Application development environment

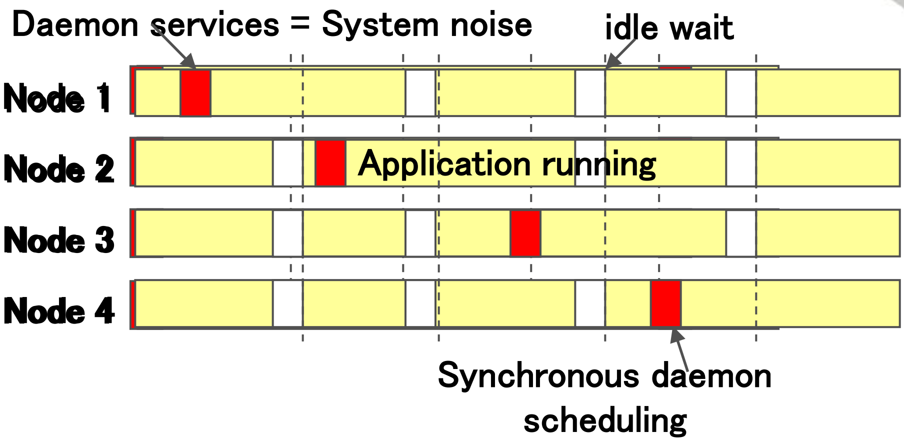
Linux-based enhanced Operating System

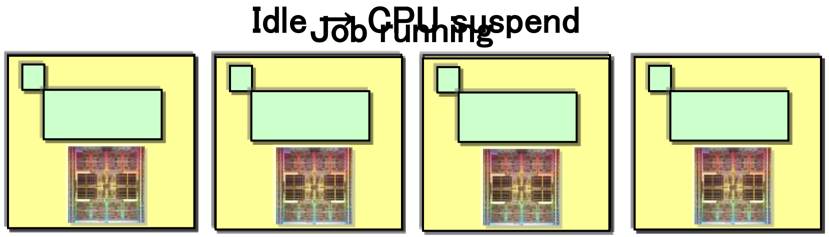
- Enhanced hardware support
- System noise reduction
- Error detection / Low power

PRIMEHPC FX10

OS: Linux ported on SPARC64

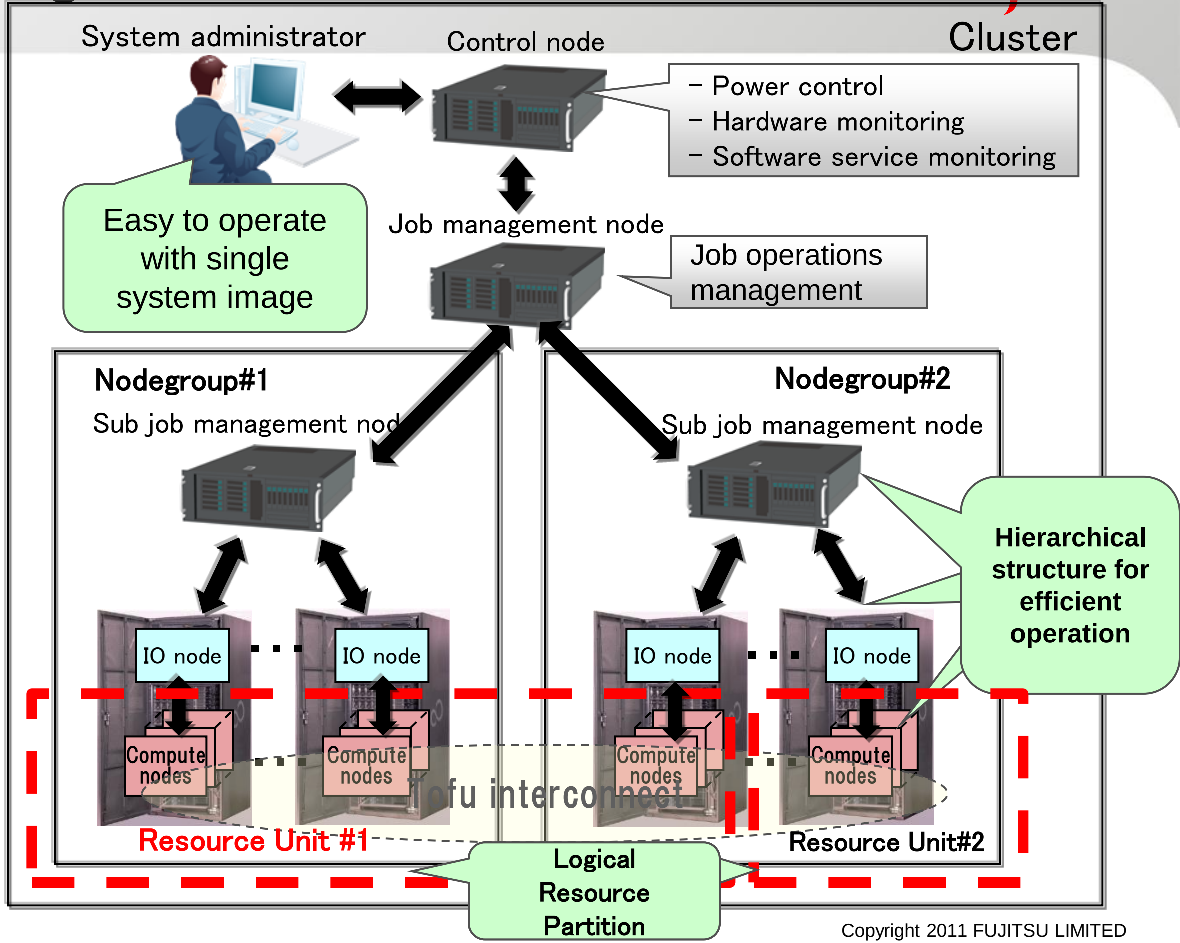
- **You can port your applications on PC clusters with little effort**
- **Additional feature for large scale system**
  - **Daemons are scheduled to reduce waiting**
  - **CPU suspension facility**





Flexible System Management

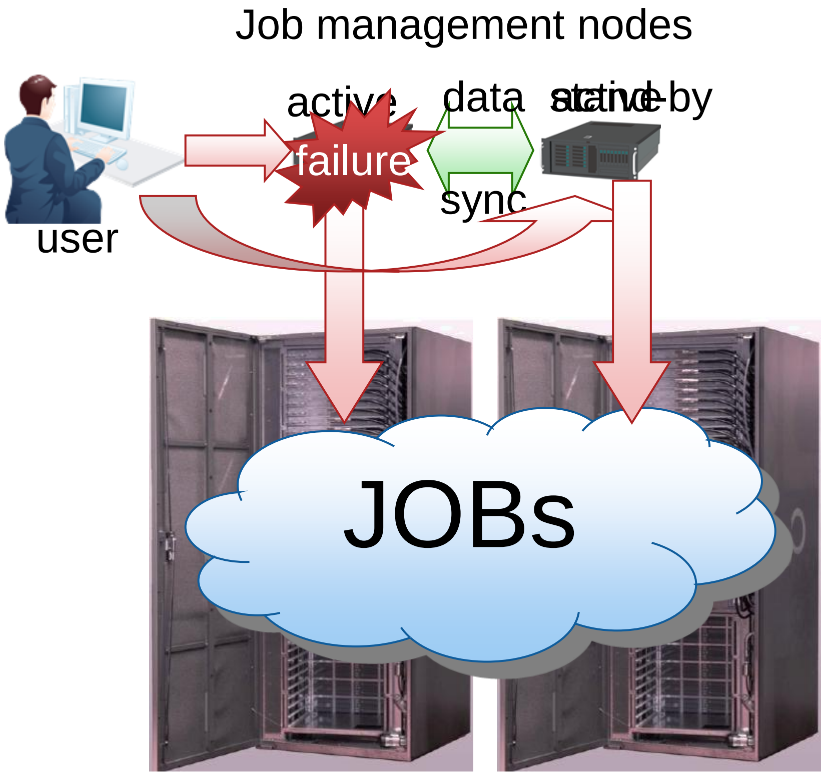
- Hierarchical structure in large scale systems
  - A job is distributed to each node thorough the sub job management node.
- A single system image through the control node
- A logical resource partition, named "Resource Unit" allocated flexibly



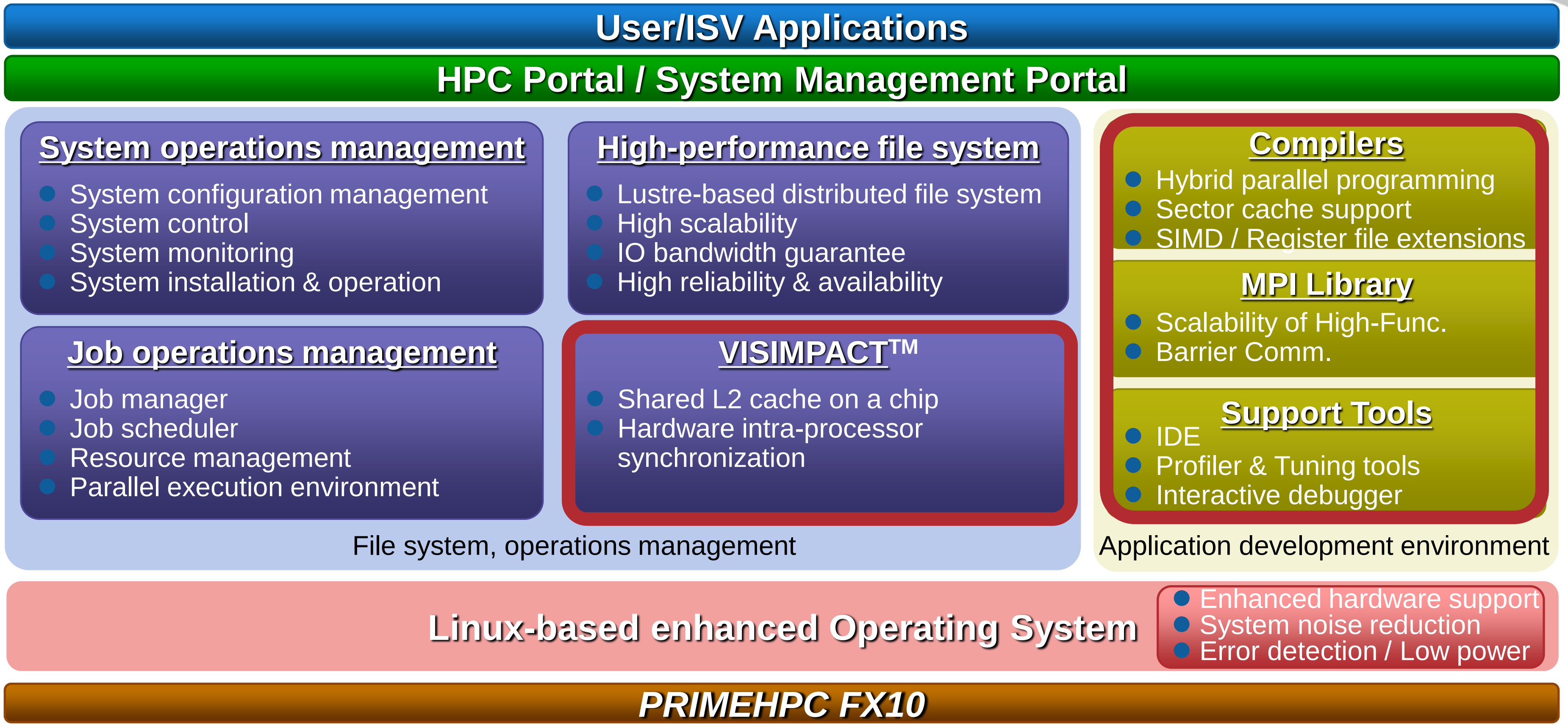
Robust system operation

- The important nodes have redundancy.
  - Control node
  - Job management node
  - Sub job management node
  - File servers

- In case of job management node failure
  - A stand-by node succeeds
    - Job data synchronization between active nodes and stand-by nodes.
  - Executing jobs can continue to run



System Software Stack



User/ISV Applications

HPC Portal / System Management Portal

System operations management
- System configuration management
- System control
- System monitoring
- System installation & operation

High-performance file system
- Lustre-based distributed file system
- High scalability
- IO bandwidth guarantee
- High reliability & availability

Job operations management
- Job manager
- Job scheduler
- Resource management
- Parallel execution environment

VISIMPACT™
- Shared L2 cache on a chip
- Hardware intra-processor synchronization

File system, operations management

Compilers
- Hybrid parallel programming
- Sector cache support
- SIMD / Register file extensions

MPI Library
- Scalability of High-Func.
- Barrier Comm.

Support Tools
- IDE
- Profiler & Tuning tools
- Interactive debugger

Application development environment

Linux-based enhanced Operating System
- Enhanced hardware support
- System noise reduction
- Error detection / Low power

PRIMEHPC FX10

I have a dream

that one day you just compile your programs and enjoy high performance on your high-end supercomputer.

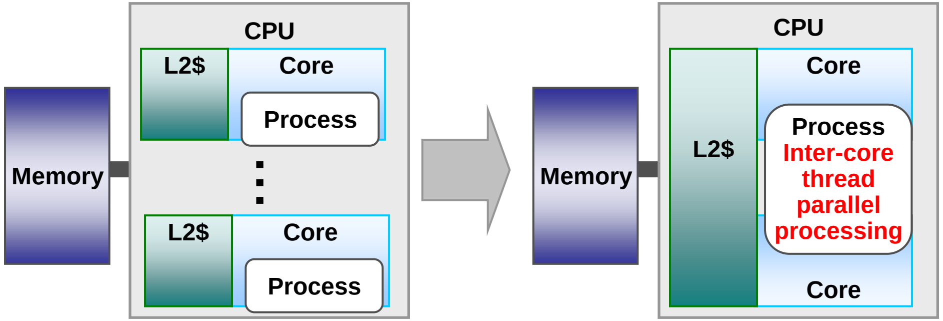
- So, we must provide easy hybrid parallel programming method including compiler and run-time system support.

Hybrid Parallelism on huge # of cores

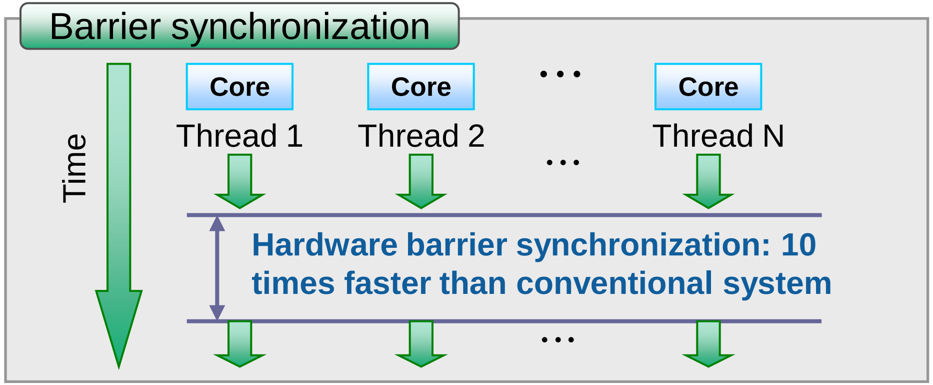
- Too large # of processes to manipulate
  - To reduce number of processes,
    hybrid thread-process programming is required
  - But
    Hybrid parallel programming is annoying for programmers
- Even for multi-threading, procedure level or outer loop parallelism was desired
  - Little opportunity for such coarse grain parallelism
  - *System support for "fine grain" parallelism is required*
- **VISIMPACT solves these problems**

VISIMPACT™
(Virtual Single Processor by Integrated Multi-core Parallel Architecture)

- Mechanism that treats multiple cores as one high-speed CPU
  - Practical automatic parallelization
  - Program and compile
    And enjoy high-speed
- You need not think about hybrid



- CPU technologies
  - Shared L2 cache memory
    to avoid false sharing
  - Inter-core hardware barrier facilities
    to reduce overhead of
    thread synchronization

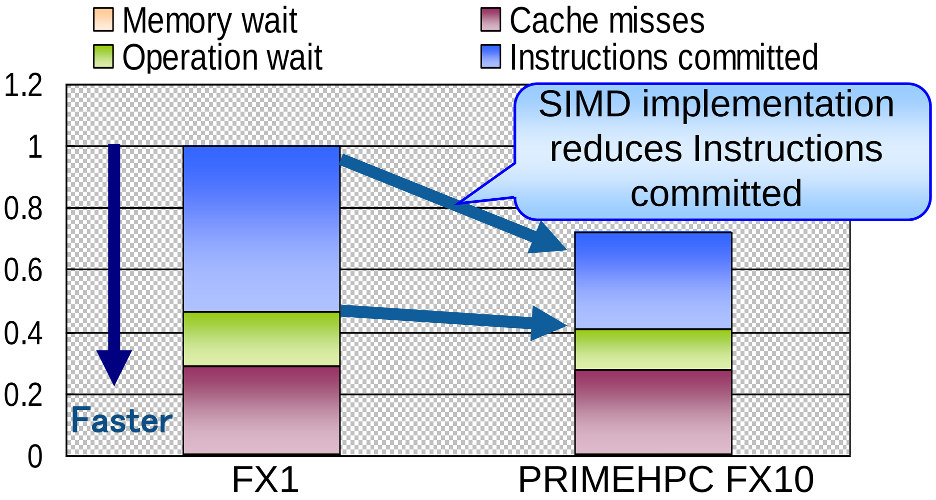


Hardware barrier synchronization: 10 times faster than conventional system

Compiler uses HPC-ACE architecture

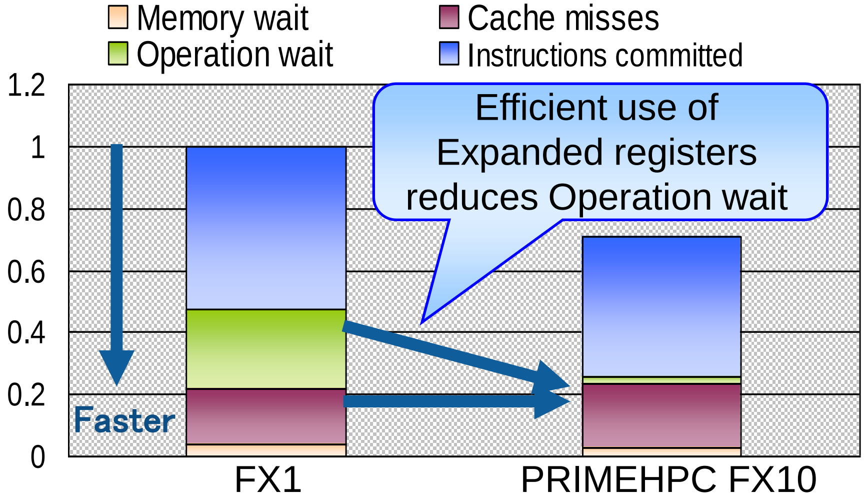
- Instruction-level parallelism with SIMD instructions
- Improvement of computing efficiency used 256 floating point registers
- Improvement of cache efficiency used “sector cache”

NPB3.3 MG
Execution time comparison (relative values)



SIMD implementation reduces Instructions committed

NPB3.3 LU
Execution time comparison (relative values)



Efficient use of Expanded registers reduces Operation wait

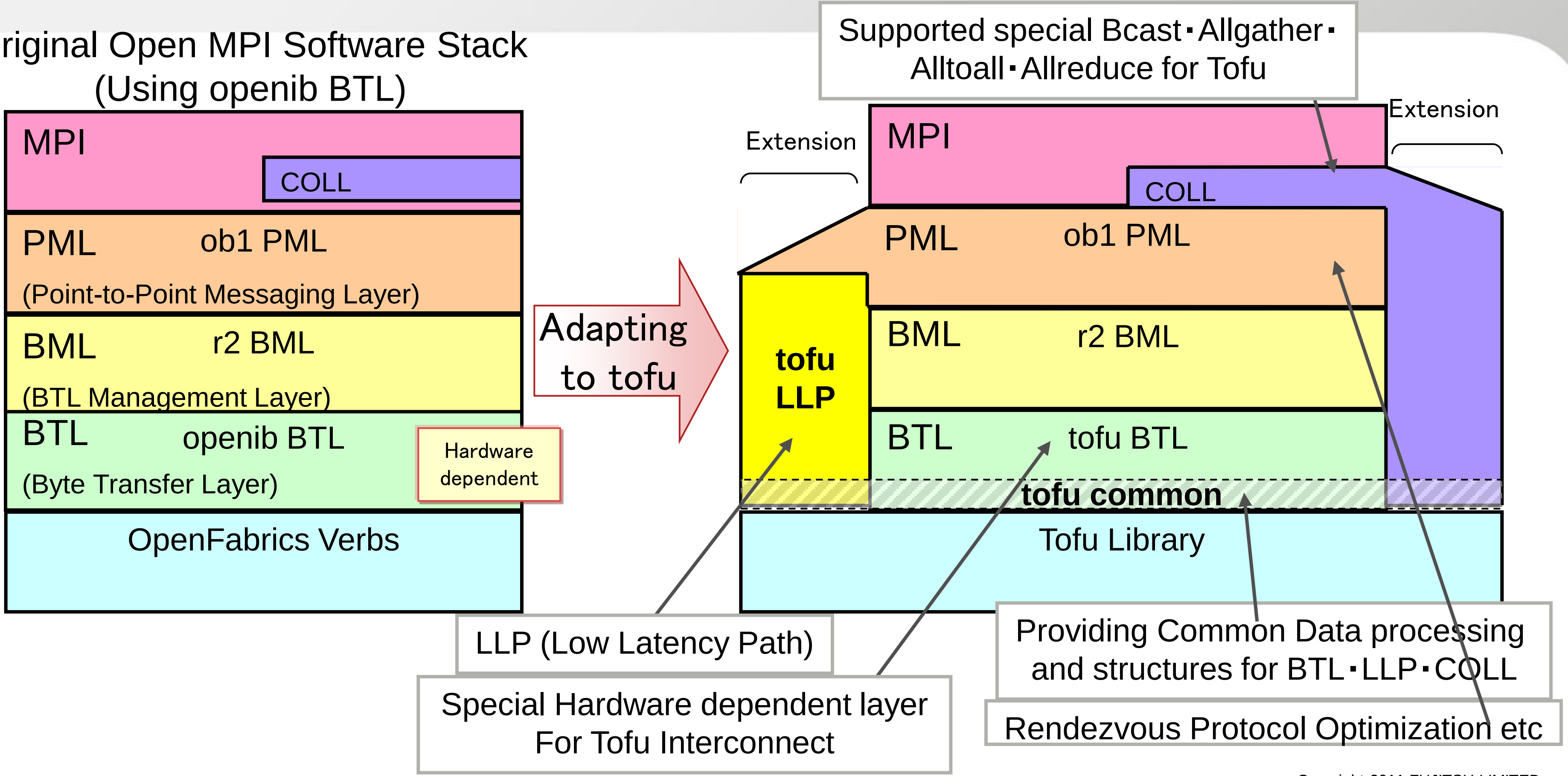
MPI Approach for FX10

- Open MPI based
  - Open Standard, Open Source, Multi-Platform including PC Cluster
  - Adding extension to Open MPI for “*Tofu*” interconnect
- High Performance
  - Short-cut message path for low latency communication
  - Torus oriented protocol: Message Size, Location, Hop Sensitive
  - Trunking Communication utilizing multi-dimensional network links by Tofu selective routing.

Goal for MPI on FX10

- High Performance
  - Low Latency & High Bandwidth
- Highly Scalability
  - Collective Performance Optimized for Tofu interconnect
- High Availability, Flexibility and Easy to Use
  - Providing Logical 3D−Torus for each JOB with eliminating failure nodes.
  - Providing New up version of MPI Standard functions as soon as possible

MPI Software stack



Original Open MPI Software Stack
(Using openib BTL)

MPI

COLL

PML ob1 PML

(Point-to-Point Messaging Layer)

BML r2 BML

(BTL Management Layer)

BTL openib BTL

(Byte Transfer Layer)

Hardware dependent

OpenFabrics Verbs

Adapting to tofu

Supported special Bcast･Allgather･Alltoall･Allreduce for Tofu

Extension

Extension

MPI

COLL

PML ob1 PML

tofu LLP

BML r2 BML

BTL tofu BTL

tofu common

Tofu Library

LLP (Low Latency Path)

Special Hardware dependent layer
For Tofu Interconnect

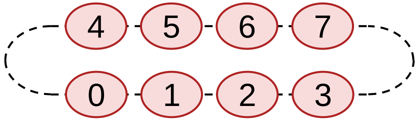
Providing Common Data processing and structures for BTL･LLP･COLL

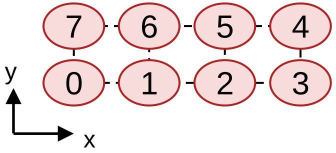
Rendezvous Protocol Optimization etc

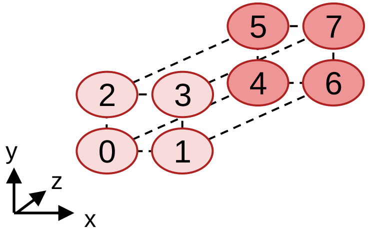
Flexible Process Mapping to Tofu environment

- You can allocate your processes as you like.
- Dimension Specification for each rank
  - 1D :(x)
  - 2D :(x,y)
  - 3D :(x,y,z)

(0)	(0,0)	(0,0,0)
(1)	(1,0)	(1,0,0)
(2)	(2,0)	(0,1,0)
(3)	(3,0)	(1,1,0)
(7)	(3,1)	(0,0,1)
(6)	(2,1)	(0,1,1)
(5)	(1,1)	(1,0,1)
(4)	(0,1)	(1,1,1)







Customized MPI Library for High Scalability

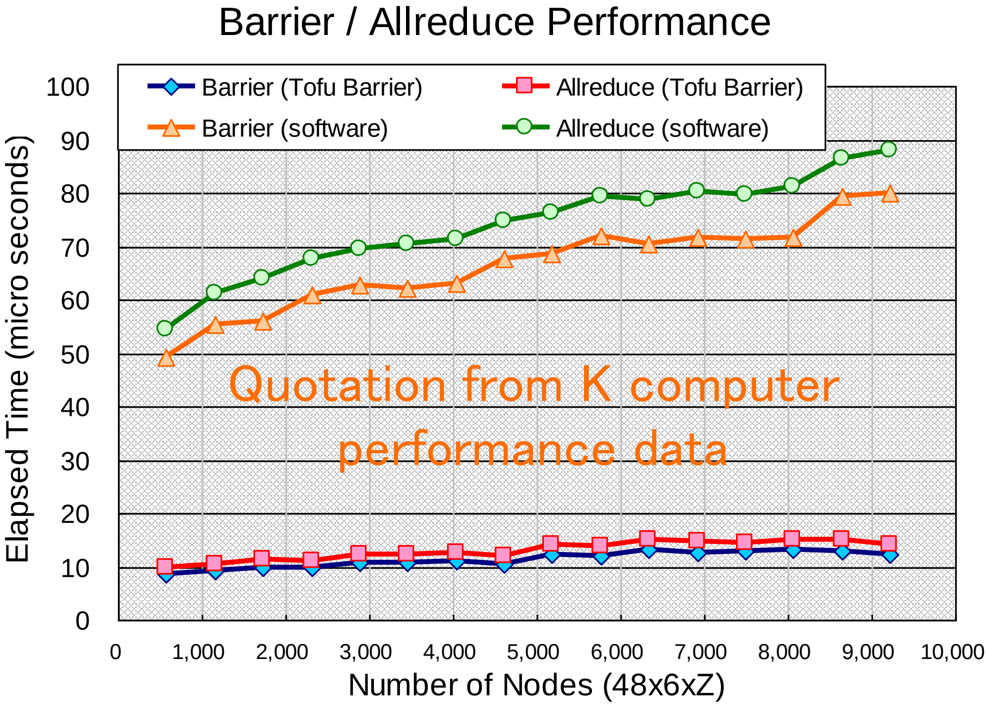
FUJITSU

Point-to-Point communication

- Use a special type of low-latency path that bypasses the software layer
- The transfer method optimization according to the data length, process location and number of hops

Collective communication

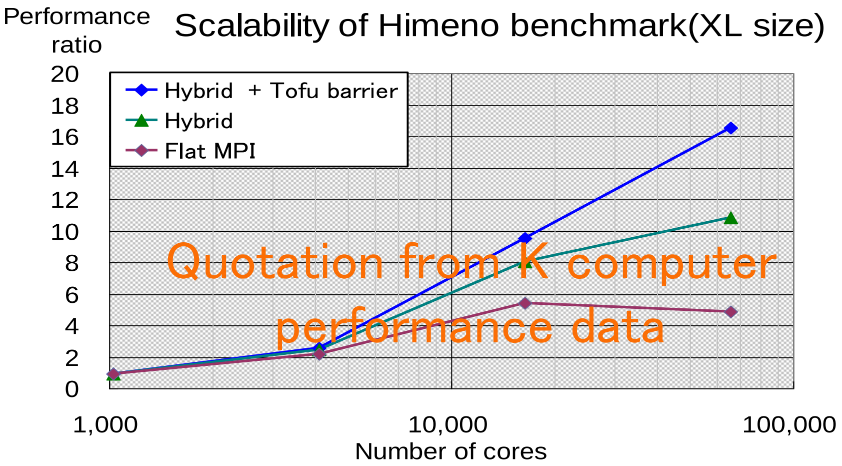
- High performance Barrier, Allreduce, Bcast and Reduce used Tofu barrier facility
- Scalable Bcast, Allgather, Allgatherv, Allreduce and Alltoall algorithm optimized for Tofu network

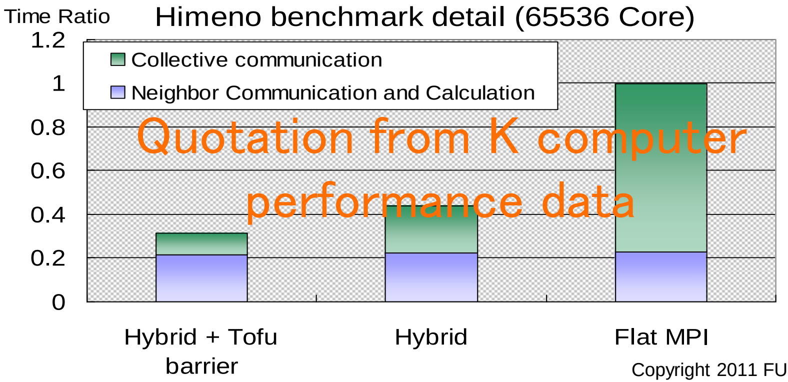


Programming Model for High Scalability

Hybrid parallelism by VISIMPACT and MPI library

- VISIMPACT
  - Multi-thread parallelization
- MPI library
  - Collective communications using Tofu barrier facility



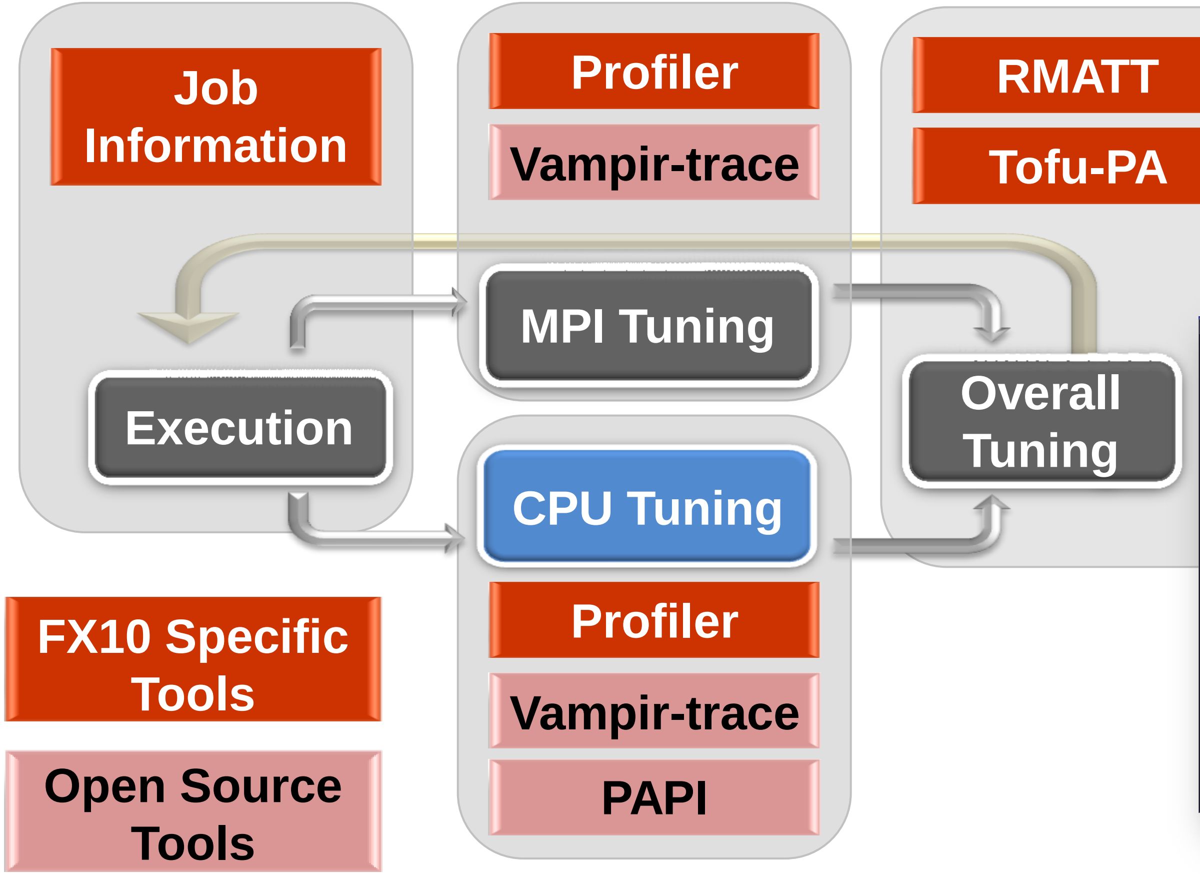


Performance Tuning

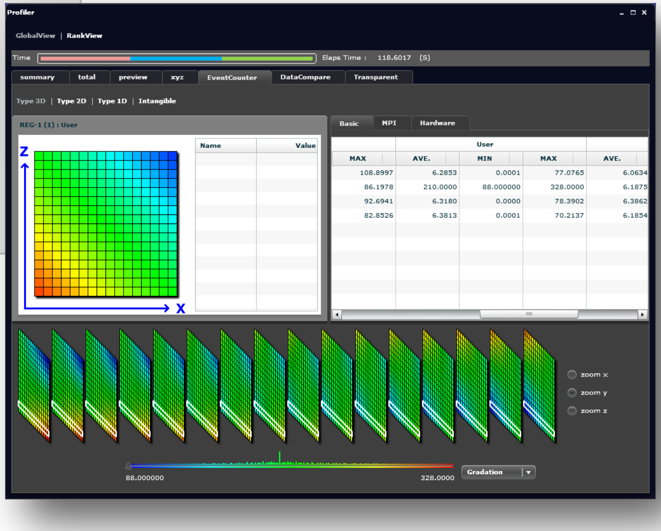
- Not only by compiler optimization,
  but also you can manipulate performance
  - Compiler directives to tune programs.
  - Tools to help your effort to tune your programs

Application Tuning Cycle and Tools

FUJITSU



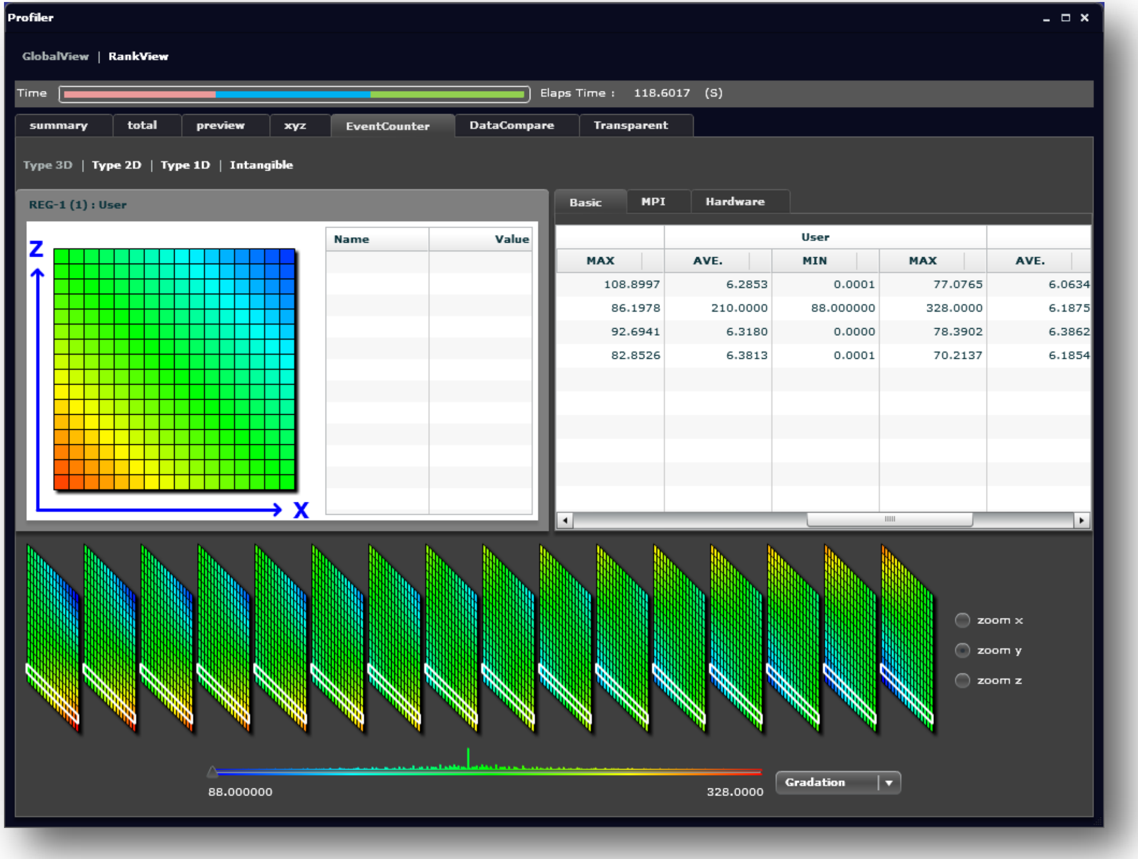
Profiler snapshot



Name	Value

	User			
MAX	AVE.	MIN	MAX	AVE.
108.8997	6.2853	0.0001	77.0765	6.0634
86.1978	210.0000	88.000000	328.0000	6.1875
92.6941	6.3180	0.0000	78.3902	6.3862
82.8526	6.3813	0.0001	70.2137	6.1854

Performance Tuning (Event Counter Example)



Time

Elaps Time : 118.6017 (S)

		User		
MAX	AVE.	MIN	MAX	AVE.
108.8997	6.2853	0.0001	77.0765	6.0634
86.1978	210.0000	88.000000	328.0000	6.1875
92.6941	6.3180	0.0000	78.3902	6.3862
82.8526	6.3813	0.0001	70.2137	6.1854

- 3-D job example
  - Display 4096 procs in 16 x 16 x 16 cells
  - Cells painted in colors according to the proc status (e.g. CPU time)
  - Cut a slice of jobs along x-, y-, or z-axis to view

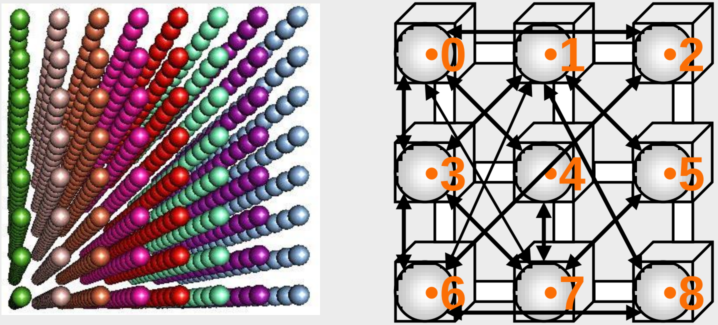
Rank Mapping Optimization

(RMATT : Rank Map Automatic Turning Tool)

Bruck algorism(Allgather type) Communication

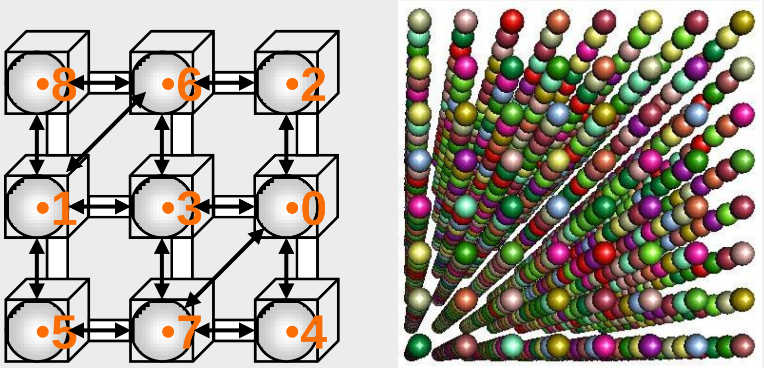
- Number of ranks : 4096 ranks
- Network configuration : 16x16x16 nodes (4096)

Standard x,y,z order mapping 22.3ms



Remapping with RMATT 5.5ms

4 times faster



Communication analysis by Vampir-trace

Log file of network configuration and communication pattern (Communication weight between ranks)

input

RMATT

output

Re-execution using optimized rank map file

Optimized rank map file
Reduce number of hops and congestion

Conclusion:
FX10 enables practical high-level parallel environment

- LINUX for SPARC64 processors
  - Reducing the system noise effect & power usage
- Highly available job/system management facilities
- VISIMPACT^{TM} lets you treat multi-core CPU as one single high-speed core.
  - Collaboration by the CPU architecture and the compiler.
    - High-speed hardware barrier to reduce the overhead of synchronization
    - Shared L2 cache to improve memory access
    - Automatic parallelization to recognize parallelism and accelerate your program
- Open MPI based MPI to utilize “Tofu” interconnect.
- Tuning facility shows the activity of parallel programs.

FUJITSU

shaping tomorrow with you