Gfarmシンポジウム2018

FUJITSU

shaping tomorrow with you

次世代スーパーコンピュータ向けファイルシステムについて

Shinji Sumimoto, Ph.D.
Next Generation Technical Computing Unit
FUJITSU LIMITED

Oct. 26th, 2018

Outline of This Talk

- A64FX: High Performance Arm CPU

- Next Generation File System Design

A64FX: High Performance Arm CPU

- From presentation slides of Hotchips 30th and Cluster 2018
- Inheriting Fujitsu HPC CPU technologies with commodity standard ISA

A64FX Chip Overview

FUJITSU

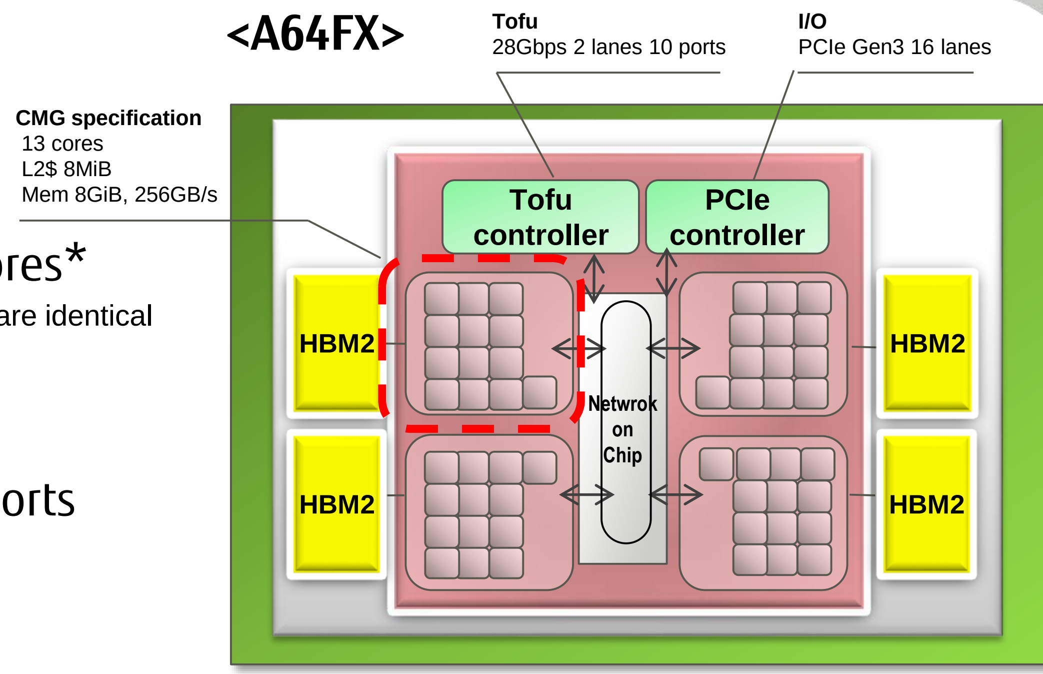
Architecture Features

- Armv8.2-A (AArch64 only)
- SVE 512-bit wide SIMD
- 48 computing cores + 4 assistant cores*

  *All the cores are identical
- HBM2 32GiB
- TofuD 6D Mesh/Torus

  28Gbps x 2 lanes x 10 ports
- PCIe Gen3 16 lanes

<A64FX>



7nm FinFET

- 8,786M transistors
- 594 package signal pins

Peak Performance (Efficiency)

- >2.7TFLOPS (>90%@DGEMM)
- Memory B/W 1024GB/s (>80%@Stream Triad)

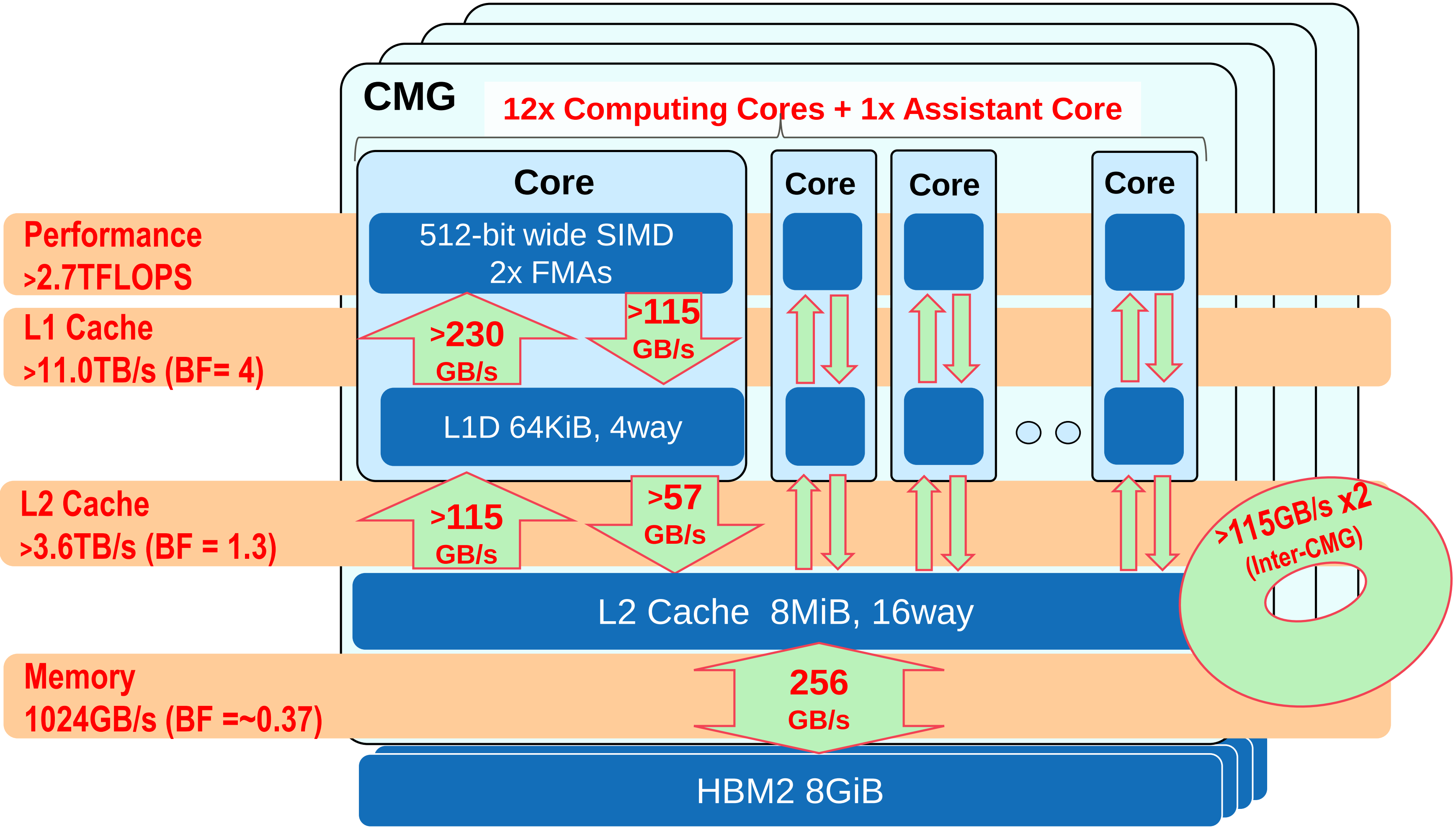
	A64FX (Post-K)	SPARC64 XIfx (PRIMEHPC FX100)
ISA (Base)	Armv8.2-A	SPARC-V9
ISA (Extension)	SVE	HPC-ACE2
Process Node	7nm	20nm
Peak Performance	>2.7TFLOPS	1.1TFLOPS
SIMD	512-bit	256-bit
# of Cores	48+4	32+2
Memory	HBM2	HMC
Memory Peak B/W	1024GB/s	240GB/s x2 (in/out)

A64FX Memory System

FUJITSU

Extremely high bandwidth

- Out-of-order Processing in cores, caches and memory controllers
- Maximizing the capability of each layer's bandwidth



A64FX Core Features

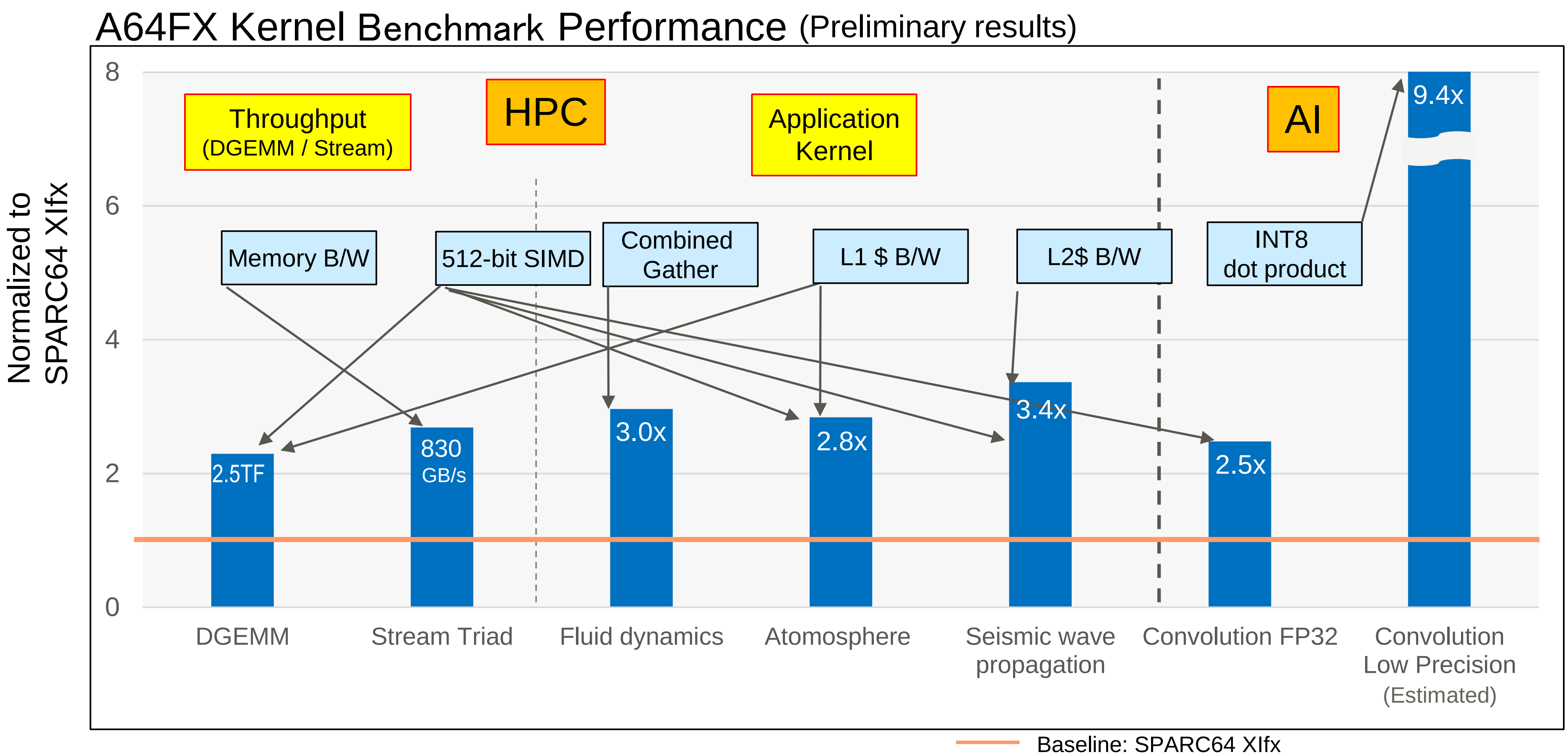
- Optimizing SVE architecture for wide range of applications with Arm including AI area by FP16 INT16/INT8 Dot Product
- Developing A64FX core micro-architecture to increase application performance

	A64FX (Post-K)	SPARC64 XIfx (PRIMEHPC FX100)	SPAR64 VIIIfx (K computer)
ISA	**Armv8.2-A + SVE**	**SPARC-V9 + HPC-ACE2**	**SPARC-V9 + HPC-ACE**
SIMD Width	512-bit	256-bit	128-bit
Four-operand FMA	✓ Enhanced	✓	✓
Gather/Scatter	✓ Enhanced	✓	
Predicated Operations	✓ Enhanced	✓	✓
Math. Acceleration	✓ Further enhanced	✓ Enhanced	✓
Compress	✓ Enhanced	✓	
First Fault Load	✓ New		
FP16	✓ New		
INT16/ INT8 Dot Product	✓ New		
HW Barrier* / Sector Cache*	✓ Further enhanced	✓ Enhanced	✓

* Utilizing AArch64 implementation-defined system registers

A64FX Chip Level Application Performance

- Boosting application performance up by micro-architectural enhancements, 512-bit wide SIMD, HBM2 and semi-conductor process technologies
  - > 2.5x faster in HPC/AI benchmarks than that of SPARC64 XIfx tuned by Fujitsu compiler for A64FX micro-architecture and SVE

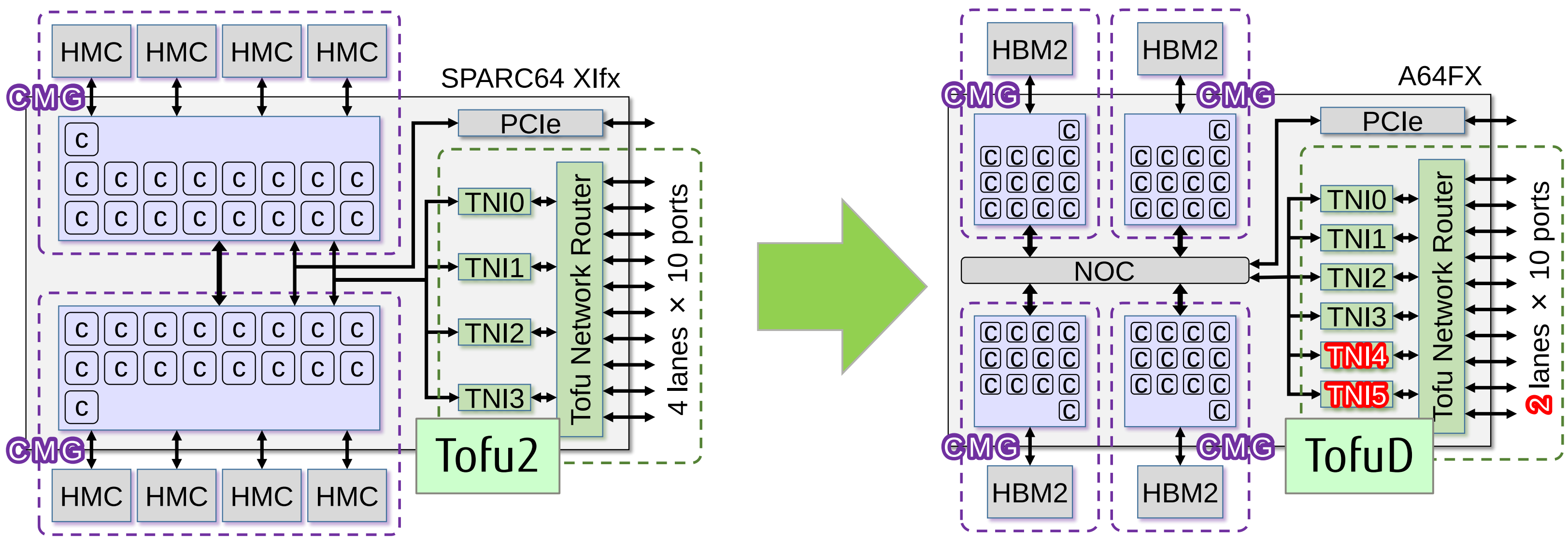


A64FX TofuD Overview

FUJITSU

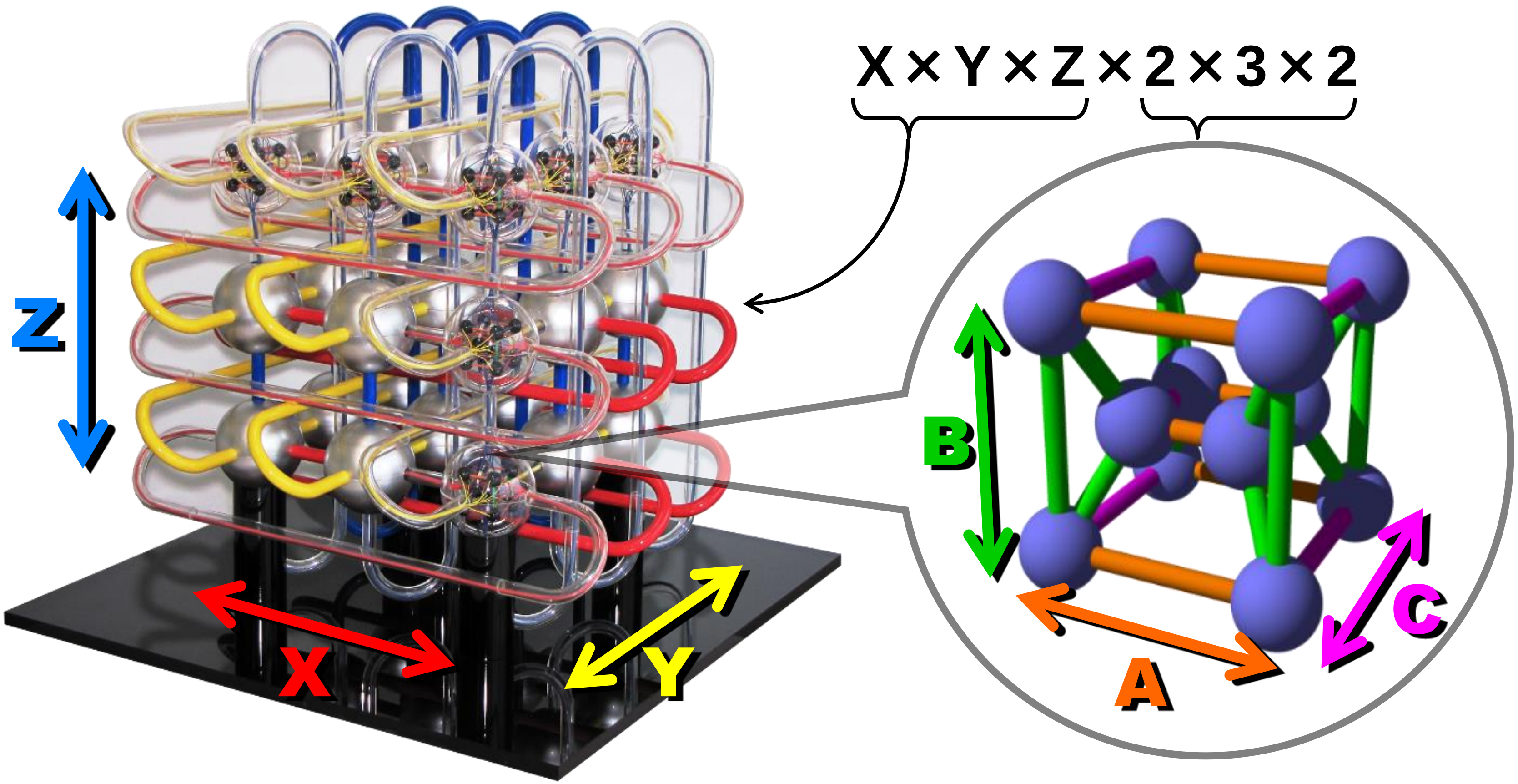
- **Halved Off-chip Channels**
  - Power and Cost Reduction
- **Increased Communication Resources**
  - TNIs from 2 to 4
  - Tofu Barrier Resources
- **Reduced Communication Latency**
  - Simplified Multi-Lane PCS
- **Increased Communication Reliability**
  - Dynamic Packet Slicing: Split and Duplicate

	Tofu K.comp	Tofu2 FX100	TofuD
Data rate (Gbps)	6.25	25.78	28.05
# of signal lanes per link	8	4	2
Link bandwidth (GB/s)	5.0	12.5	6.8
# of TNIs per node	4	4	6
Injection bandwidth per node (GB/s)	**20**	**50**	**40.8**



TofuD: 6D Mesh/Torus Network

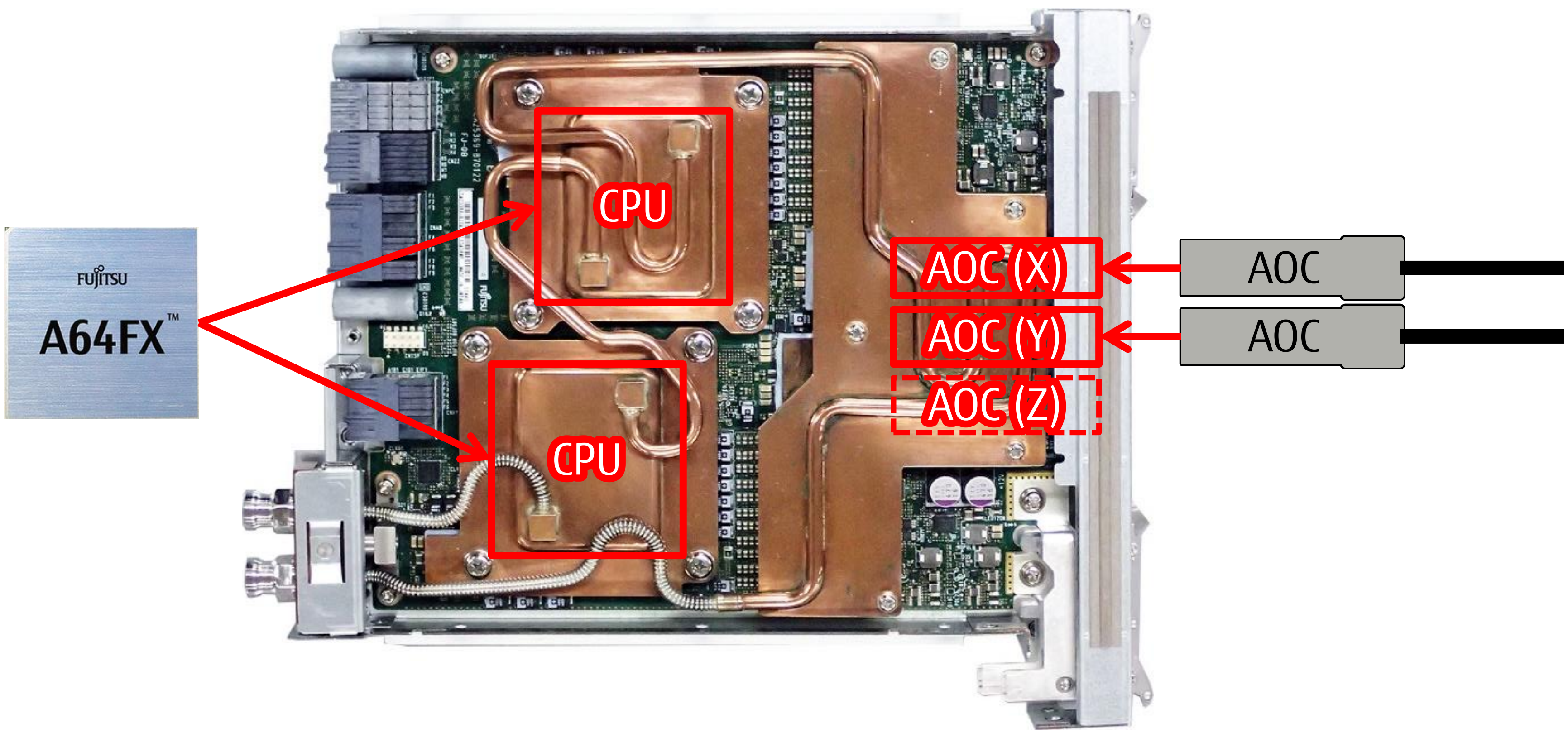
- Six coordinates: (X, Y, Z) × (A, B, C)
  - X, Y and Z: sizes are depends on the system size
  - A, B and C: sizes are fixed to 2, 3, and 2 respectively
- Tofu stands for "torus fusion"



TofuD: Packaging – CPU Memory Unit

FUJITSU

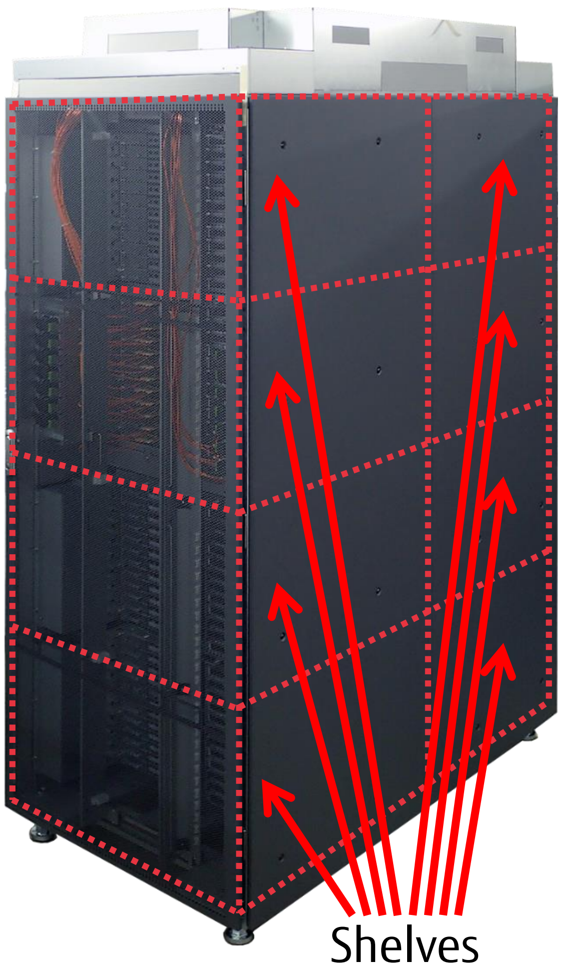
- Two CPUs connected with C-axis
  - X × Y × Z × A × B × C = 1 × 1 × 1 × 1 × 1 × 2
- Two or three active optical cable cages on board
  - Each cable is shared by two CPUs



TofuD: Packaging – Rack Structure

FUJITSU

- Rack
  - 8 shelves
  - 192 CMUs or 384 CPUs
- Shelf
  - 24 CMUs or 48 CPUs
  - X × Y × Z × A × B × C = 1 × 1 × 4 × 2 × 3 × 2
- Top or bottom half of rack
  - 4 shelves
  - X × Y × Z × A × B × C = 2 × 2 × 4 × 2 × 3 × 2



TofuD: Put Latencies & Throughput& Injection Rate

FUJITSU

- TofuD: Evaluated by hardware emulators using the production RTL codes
  - Simulation model: System-level included multiple nodes

	Communication settings	Latency
Tofu	Descriptor on main memory	1.15 μs
	Direct Descriptor	0.91 μs
Tofu2	Cache injection OFF	0.87 μs
	Cache injection ON	0.71 μs
TofuD	To/From far CMGs	0.54 μs
	To/From near CMGs	0.49 μs

	Put throughput	Injection rate
Tofu	4.76 GB/s (95%)	15.0 GB/s (77%)
Tofu2	11.46 GB/s (92%)	45.8 GB/s (92%)
TofuD	6.35 GB/s (93%)	38.1 GB/s (93%)

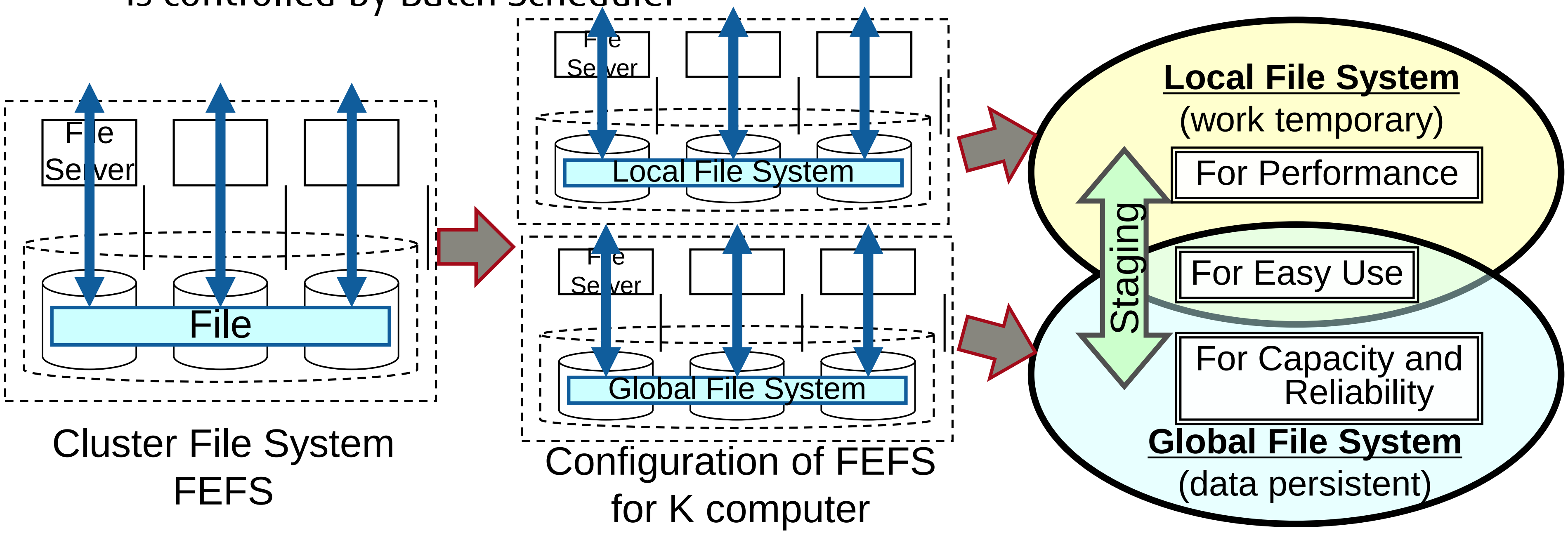
Next Generation File System Design

- File System Design for the K computer
- Next Generation File System Structure and Design
- Next-Gen 1st Layer File System Overview

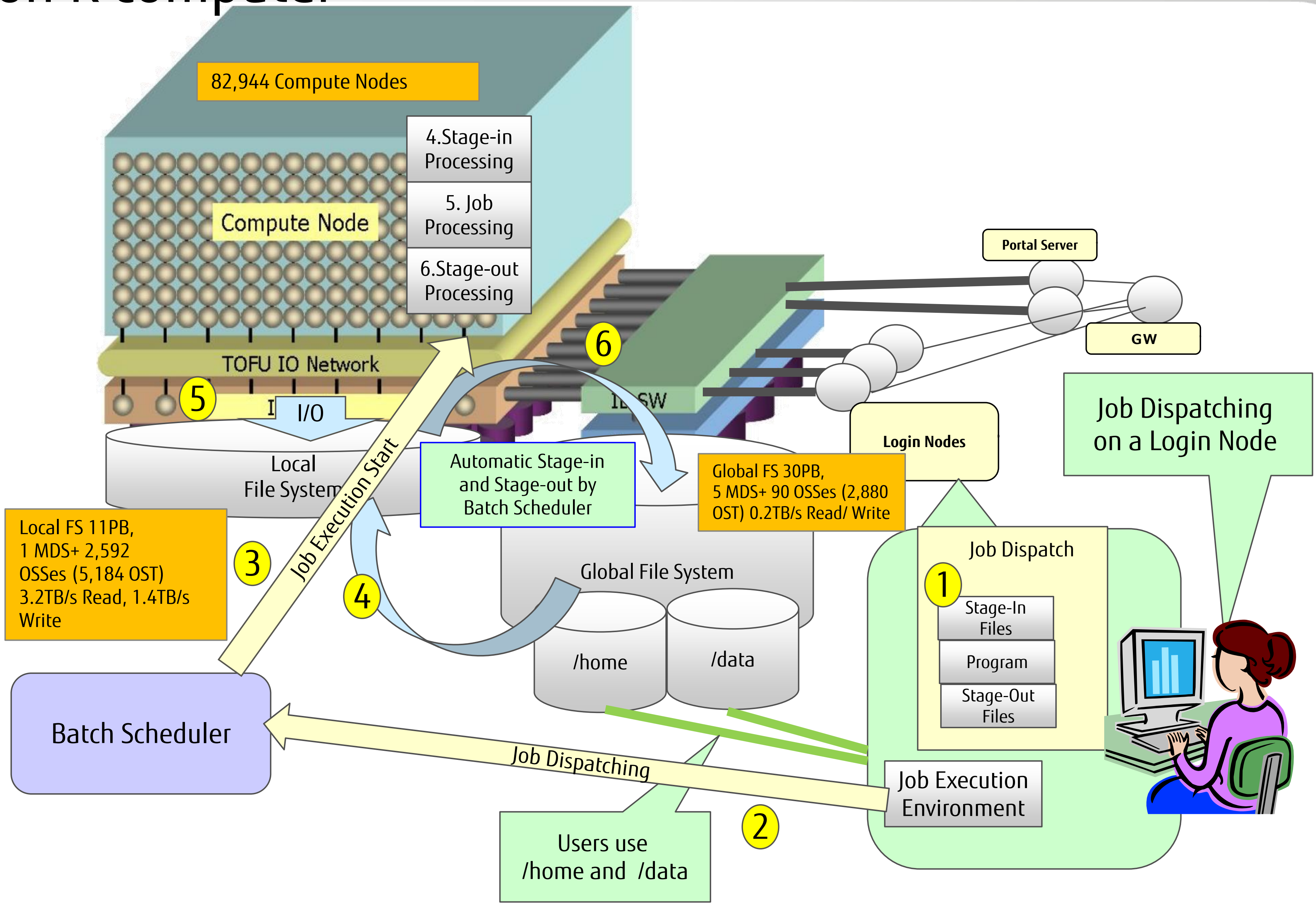
Overview of FEFS for K computer

FUJITSU

- Goals: To realize World Top Class Capacity and Performance File system
  100PB, 1TB/s
- Based on Lustre File System with several extensions
  - These extensions are now going to be contributed to Lustre community.
- Introducing Layered File system for each file layer characteristics
  - Temporary Fast Scratch FS(Local) and persistent Shared FS(Global)
  - Staging Function which transfers between Local FS and Global FS is controlled by Batch Scheduler



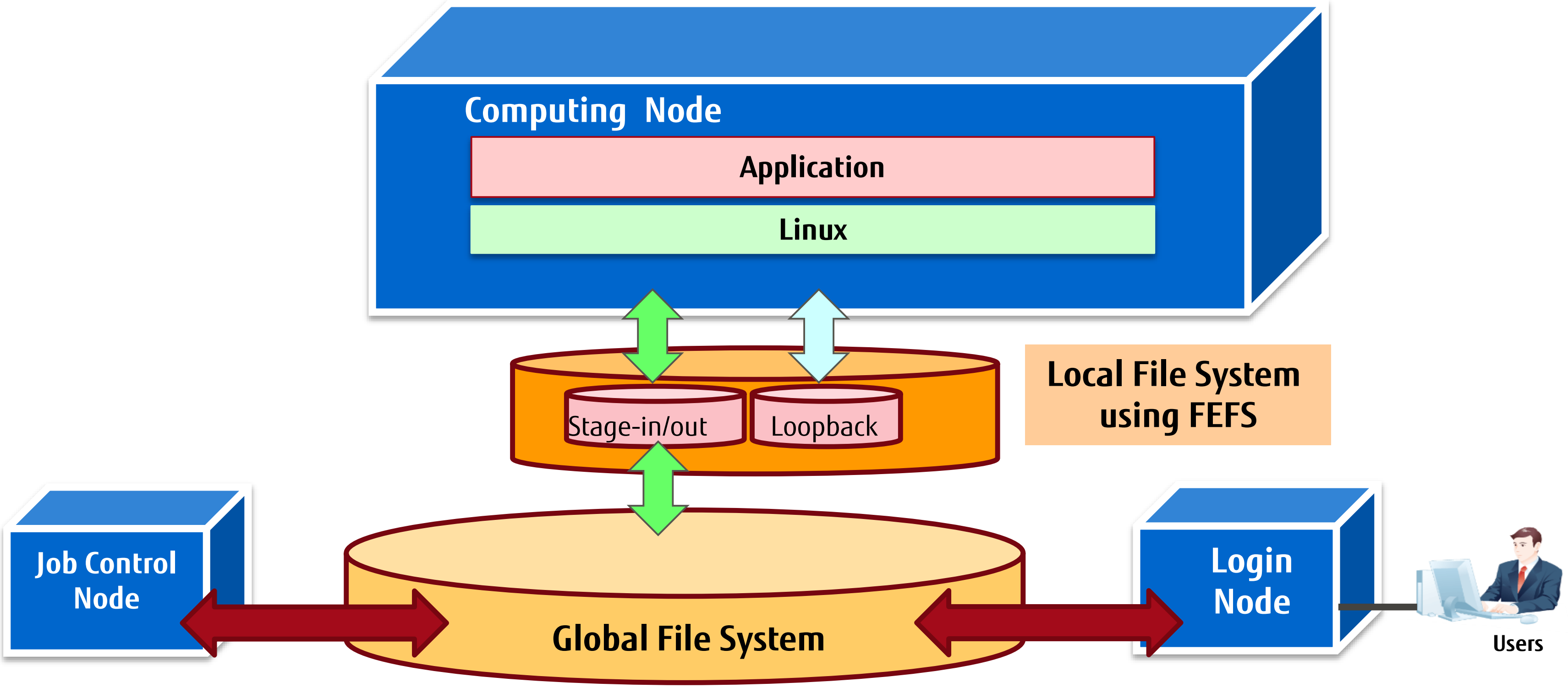
Job Execution and File System Accesses on K computer



K computer: Pre-Staging-In/Post-Staging-Out Method

FUJITSU

- Pros:
  - Stable Application Performance for Jobs
- Cons:
  - Requiring three times amount of storage which a job needs
  - Pre-defining file name of stage-in/out processing lacks of usability
  - Data-intensive application affects system usage to down because of waiting pre-staging-in/out processing



Next-Gen File System Requirement and Issues

FUJITSU

Requirements

- 10 times higher access performance
- 100 times larger file system capacity
- Lower power and footprint

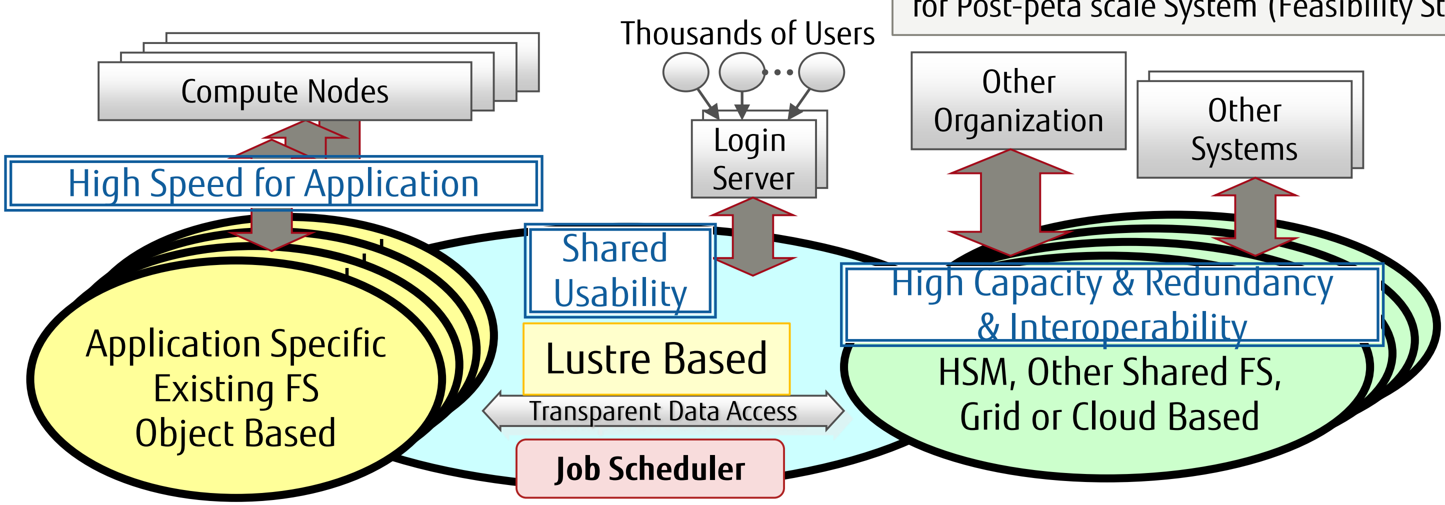
Issues

- How to realize 10 times faster and 100 times larger file access at a time?

Next-Gen. File System Design

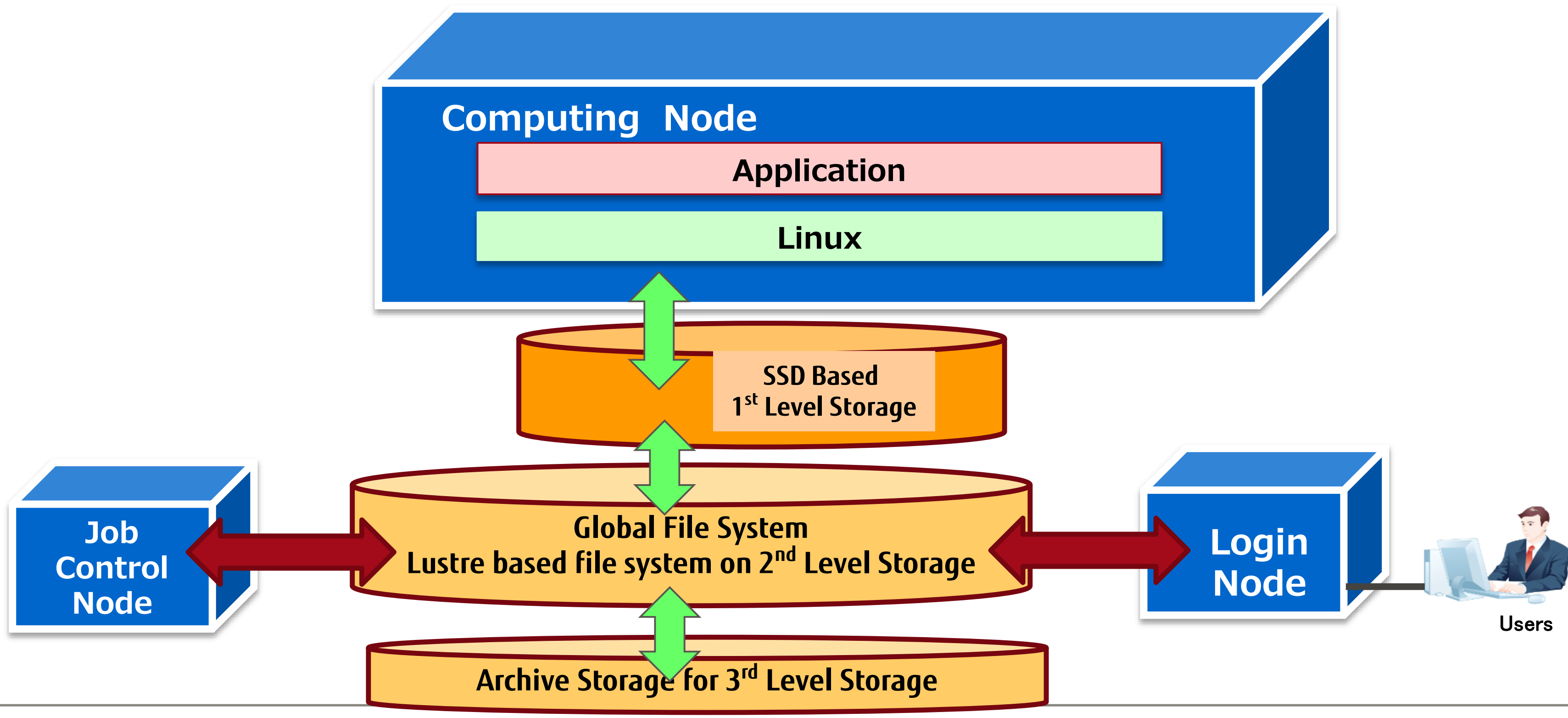
- K computer File System Design
  - How should we realize High Speed and Redundancy together?
  - Introduced Integrated Two Layered File System.
- Next-Gen. File System/Storage Design
  - Another trade off targets: Power, Capacity, Footprint
    - Difficult to realize single Exabyte and 10TB/s class file system in limited power consumption and footprint.
  - Additional Third layer Storage for Capacity is needed:

The Next Integrated Layered File System Architecture for Post-peta scale System (Feasibility Study 2012-2013)



Next Gen. File System Design

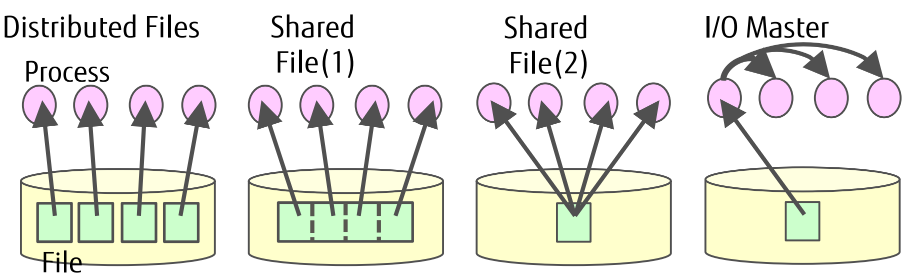
- Introducing three level hierarchical storage.
  - 1st level storage: Accelerating application file I/O performance (Local File System)
  - 2nd level storage: Sharing data using Lustre based file system (Global File System)
  - 3rd level storage: Archive Storage (Archive System)
- Accessing 1st level storage as file cache of global file system and local storage
  - File cache on computing node is also used as well as 1st level storage



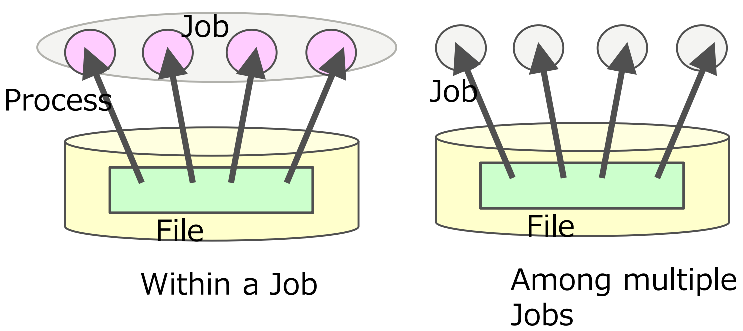
Scopes of File Usages for Post-K File System Design

FUJITSU

- File Lifetime:
  - Persistent Files: Input Files, Output Files
  - Temporary Files: Input Files, Output File
- Access Pattern:
  - Distributed Files: for each process
  - Shared File :
    - partial access
    - concentrate access to same data
  - I/O Master: Master does whole File I/O



- Data Sharing:
  - Within a job
  - Among multiple jobs(under designing)



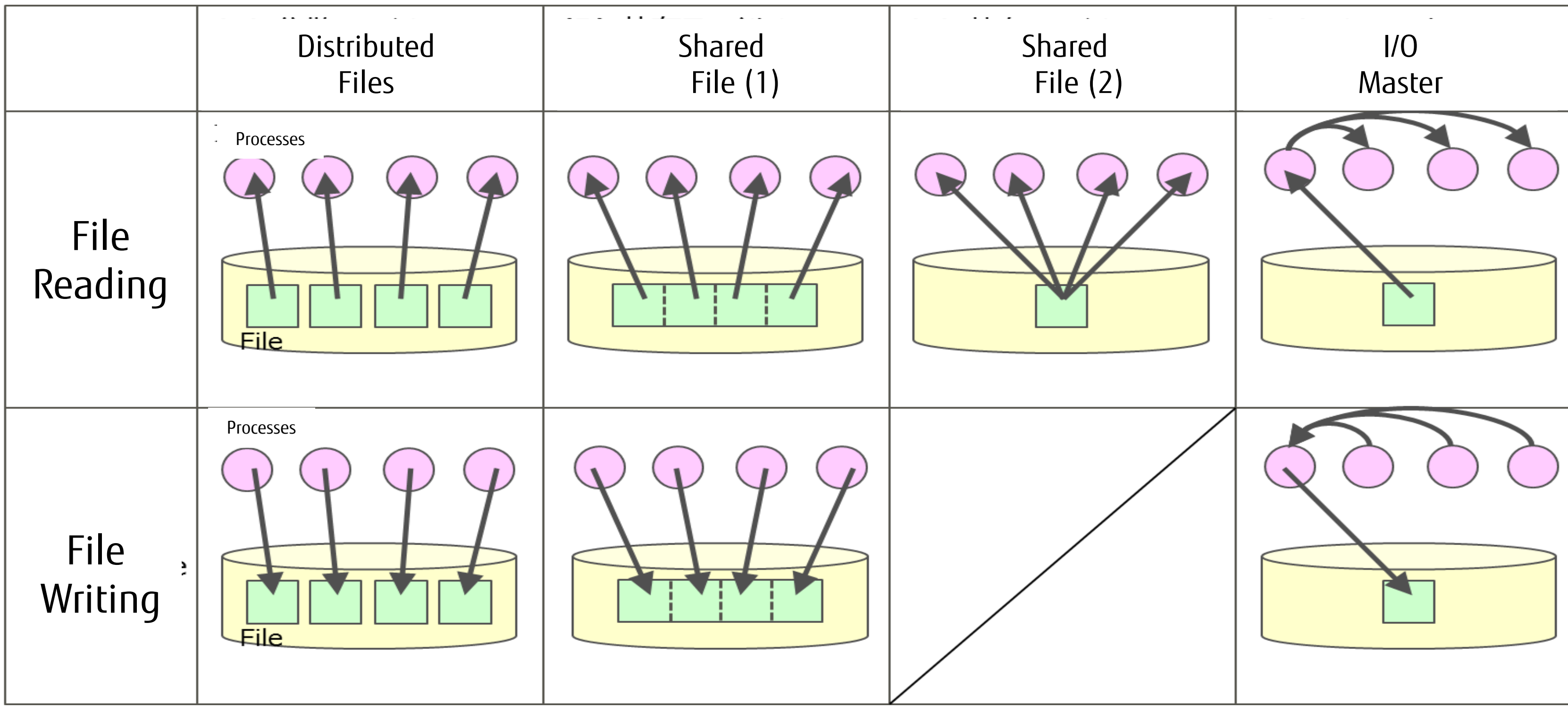
File Lifetime for Effective SSD Use

- Persistent files in a job are located on SSD as file cache
  - Asynchronous data transfer is used between SSD and global file system
- Temporary files in a job should be located on SSD to eliminate global file system accesses
- But, how persistent file cache on SSD should be used?
  - It depends on file access patterns

Application's Access Pattern and SSD Cache Effects

FUJITSU

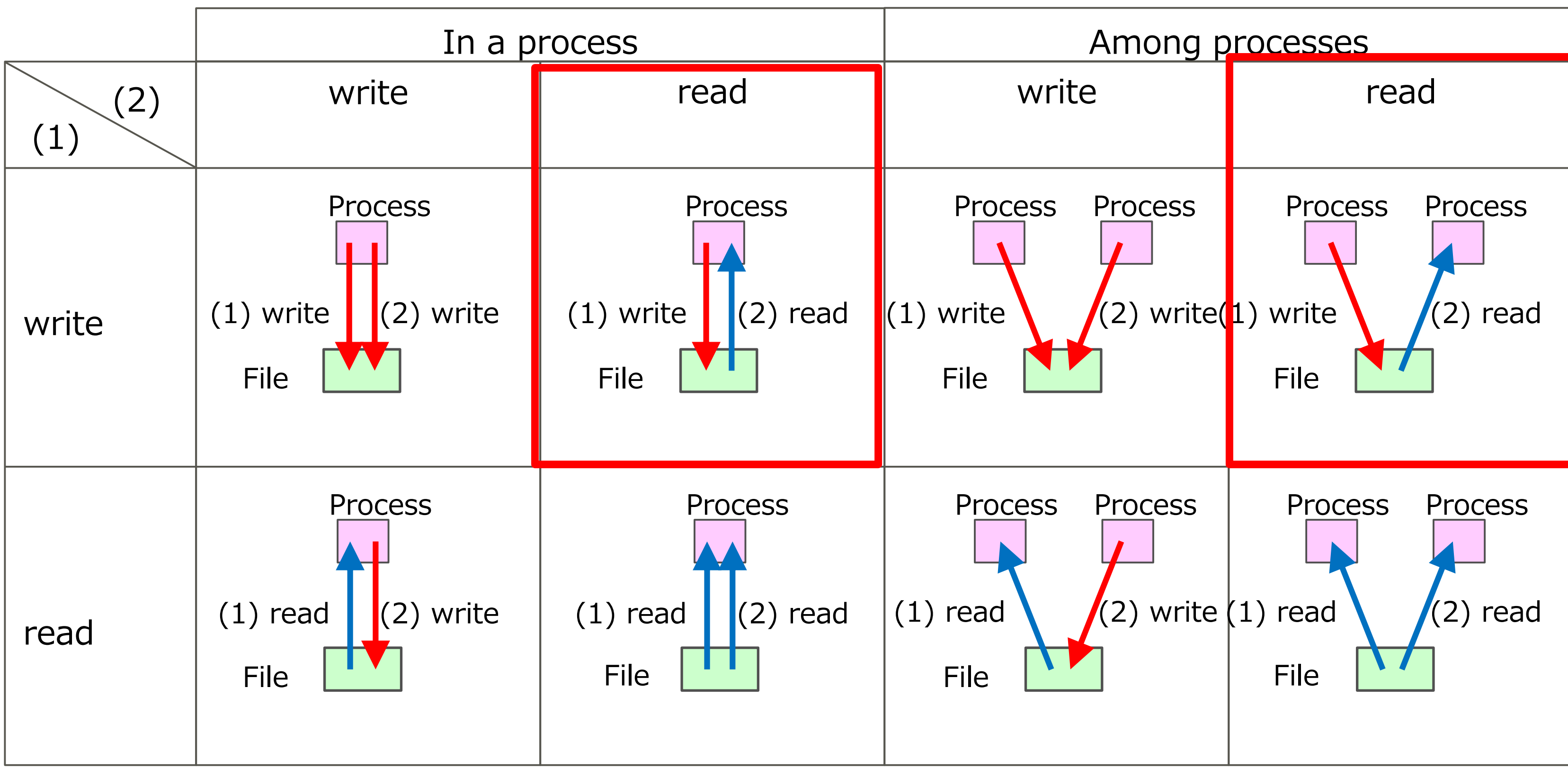
Comparison of Effective Pattern for SSD based storage



File Read: Effects	Rereading Case: ◎ Non Rereading : ×	Rereading Case: ◎ Non Rereading : ×	Rereading Case: ◎ Non Rereading : ×	Rereading Case: ◎ Non Rereading : ×
File Write: Effects	Rewriting Case: ◎ Non Rewriting : ○	Rewriting Case: ◎ Non Rewriting : ○		Rewriting Case: ◎ Non Rewriting : ○

Data Sharing in a Job

- Write-Read in a process and among processes are effective to use SSD
- For Persistent Files: File cache of global file system should be shared among processes
- For Temporary Files: Two types of temporary file systems are effective to use SSD
  - Temporary Local System (in a process)
  - Temporary Shared File System (among processes)



Data Sharing among Multiple Jobs

- Write-Read among multiple jobs are effective to use SSD

- To be designed how to share file cache on global file system and temporary shared file system data

SSD Lifetime Issue

FUJITSU

- Current SSDs mainly use NAND based cells and have an issue of limited number of lifetime writes(DWPD)
  - Consumer products can not be used because of lack of DWPD
  - Enterprise products must be used
- Operating period of Post-K will be planed at least 5 years
- The DWPD of most I/O intensive target application is 7.1TB/Day
  - Intel P3700 is the best choice in these products

	Enterprise Products		Consumer Products				
	Intel P3700	Intel P3608	Intel 750	Intel 600p	Samsung 950 pro	Samsung 960 Pro	Samsung 960 EVO
Capacity	800GB	1.6TB	1.2TB	1TB	512GB	1TB	1TB
Warranty Period	5 years	5 years	5 years	5 years	5 years	5 years	3 years
MTBF	2.0M	1.0M	1.2M	1.5M	1.5M	1.5M	1.5M
AFR	0.44%	0.87%	0.73%	0.54%	0.58%	0.58%	0.58%
DWPD	8TB/Day	4.8TB/Day	70GB/Day	40GB/Day	210GB/Day	430GB/Day	360GB/Day

How about Intel Optane Products?

FUJITSU

	Enterprise Products					Enthusiast
	Intel P3700	Intel P3608	Intel P4600	Intel P4500	Intel Optane P4800X	Intel Optane 900P
Capacity	800GB	1.6TB	1.6TB	1TB	375GB	480GB
Read Perf.	2.7GB/s	5.0GB/s	3.3GB/s	3.3GB/s	2.4GB/s	2.5GB/s
Write Perf.	1.9GB/s	2.0GB/s	1.4GB/s	0.6GB/s	2.0GB/s	2.0GB/s
K IOPS(R/W)	460/90	850/150	587/184	394/32	550/500	550/500
Latency(R/W)	20/20us	20/20us	79/34us	80/29us	10/10 us	10/10us
Warranty	5 years	5 years	5 years	5 years	5 years	5 years
MTBF	2.0M	1.0M	2.0M	2.0M	2.0M	1.6M
AFR	0.44%	0.87%	0.44%	0.44%	0.44%	0.54%
DWPD	8TB/Day	4.8TB/Day	4.7TB/Day	0.72TB/Day	11.2TB/Day	4.7TB/Day

https://www.intel.com/content/www/us/en/products/memory-storage/solid-state-drives/data-center-ssds.html

- Intel Optane:
  - Write IOPs is 2.7 times higher than that of P4600, but 375GB capacity is too small to use
  - DWPD 11.2TB/Day is not higher than expected, (3 times better than P3700/800G) but actual number of cells should be investigated.
  - Current cost is 30% higher than that of P3700 800GB (Amazon.com)

Next-Gen. File System Design: How SSD based storage should be used?

FUJITSU

Life Time
- Persistent files in a job are located on SSD as file cache
- Temporary files in a job should be located on SSD to eliminate global file system accesses

Application's Access Pattern
- Non reusable file in file reading should not use SSD based storage

Data Sharing in a Job
- Write-Read in a process and among processes are effective to use SSD
- For Persistent Files: File cache of global file system should be shared among processes
- For Temporary Files: Two types of temporary file systems are effective to use SSD
  - Temporary Local System (in a process)
  - Temporary Shared File System (among processes)

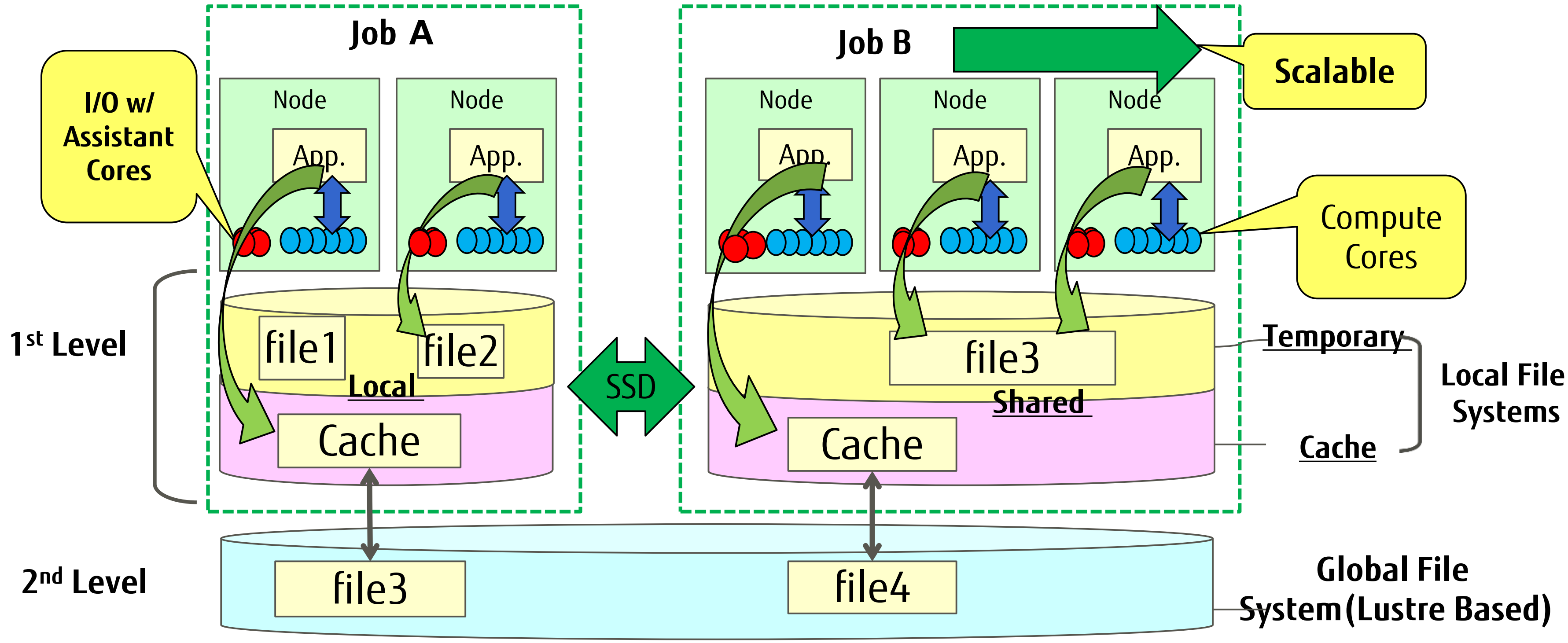
Data Sharing among Multiple Jobs
- Write-Read among multiple jobs are effective to use SSD
- To be designed how to share file cache on global file system and temporary shared file system data

SSD Lifetime Issue
- Enterprise SSD with higher DWPD than that of all applications will be selected

Next-Gen 1st Layer File System Overview

FUJITSU

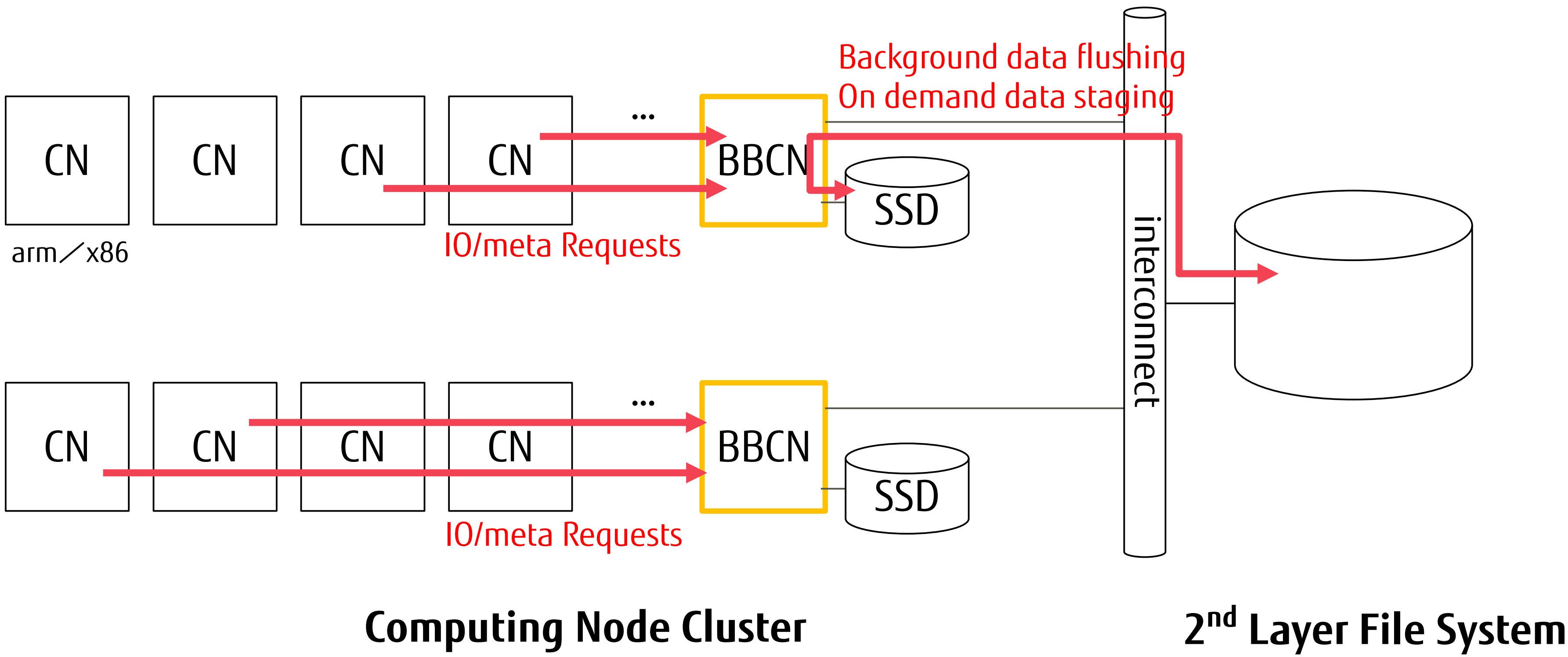
- Goal: Maximizing application file I/O performance
- Features:
  - Easy access to User Data: File Cache of Global File System
  - Higher Data Access Performance: Temporary Local FS (in a process)
  - Higher Data Sharing Performance: Temporary Shared FS (among processes)
- Now developing LLIO (Lightweight Layered IO-Accelerator) Prototype



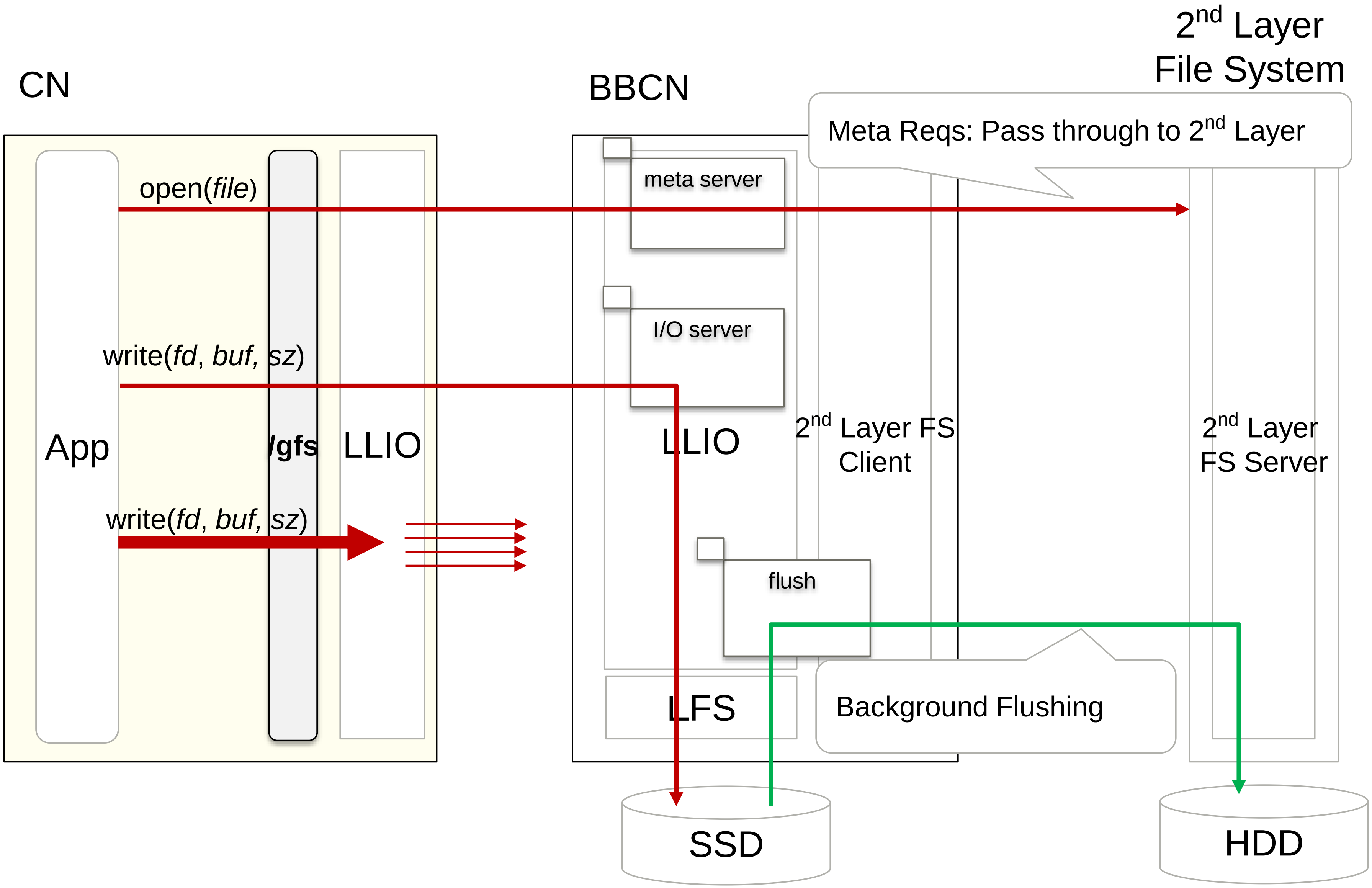
LLIO Prototype Implementation

FUJITSU

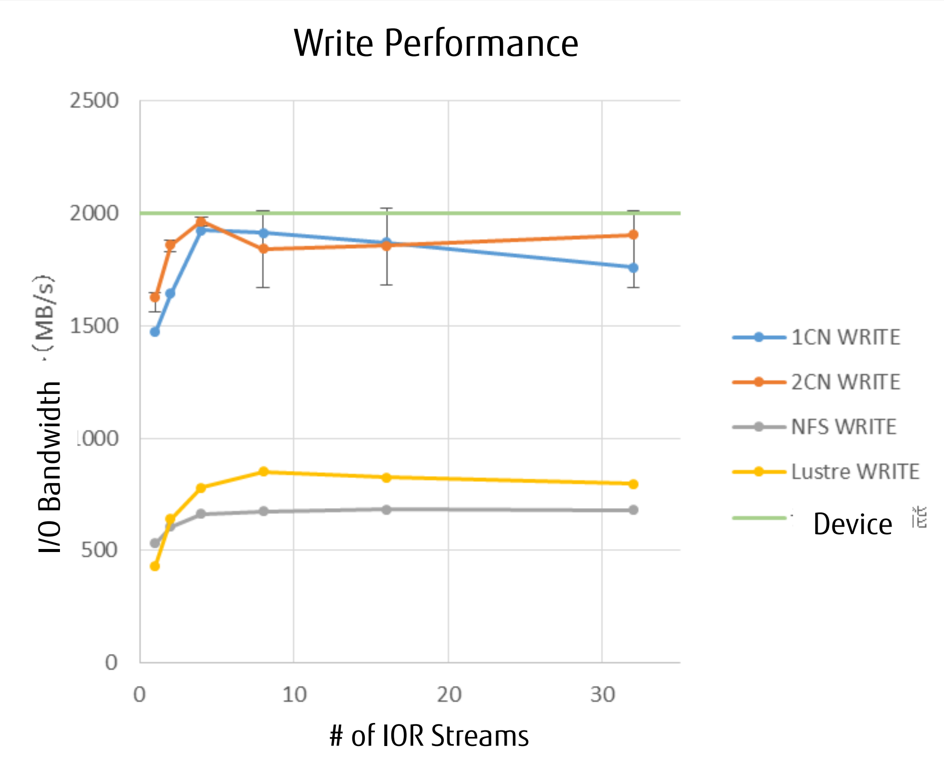
- Two types of Computing Nodes
  - Burst Buffer Computing Node(BBCN)
    - Burst Buffer System Function with SSD Device
  - Computing Node(CN)
    - Burst Buffer Clients: File Access Request to BBCN as burst buffer server

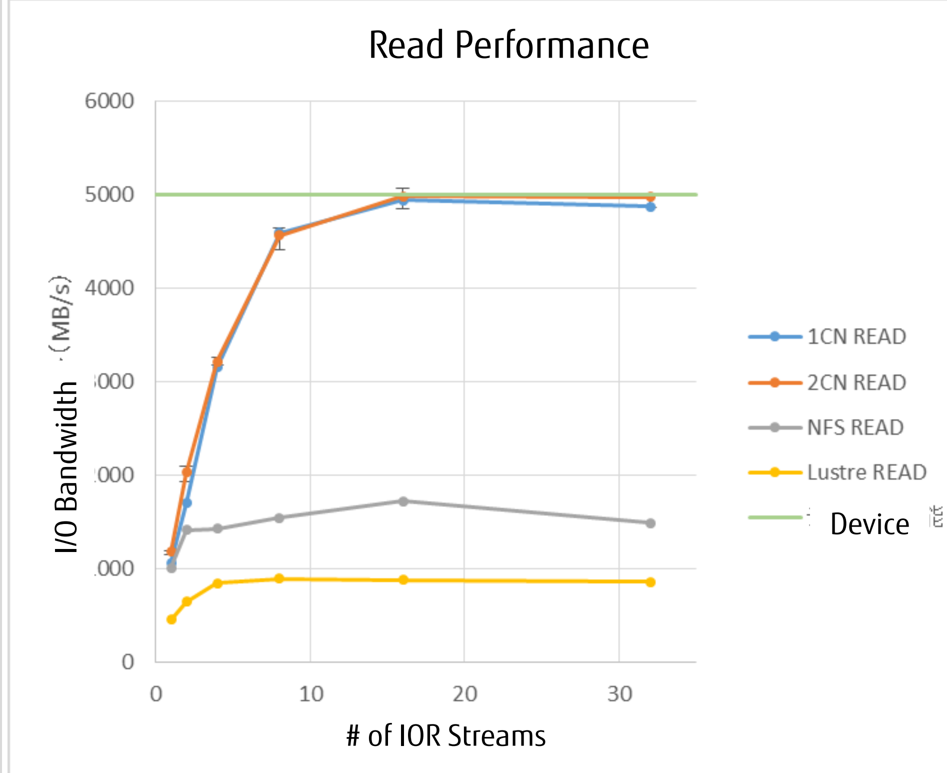


File Access Sequences using LLIO (Cache Mode)



LLIO Prototype I/O Performance





Evaluated on IA servers using Intel P3608

- Higher I/O performance than those of NFS, Lustre
- Utilizing maximum physical I/O device performance by LLIO

FUJITSU

shaping tomorrow with you