An Overview of Fujitsu‘s Lustre Based File System

Shinji Sumimoto
Fujitsu Limited
Apr.12 2011

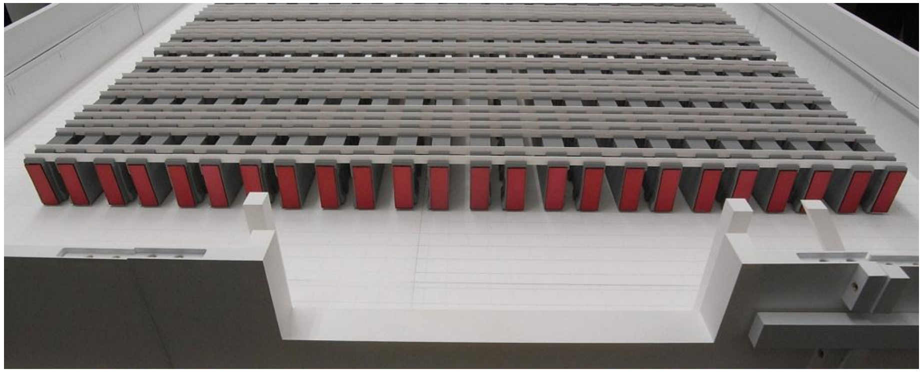
For Maximizing CPU Utilization by Minimizing File IO Overhead

Outline

- Target System Overview
- Goals of Fujitsu's Cluster File System (FEFS)
- IO System Architecture
- Fujitsu's File System Overview

Target System: K computer

- RIKEN and Fujitsu are jointly developing the 'K computer`
  - To be installed at the RIKEN AICS (Advanced Institute for Computational Science), Kobe, by 2012
- Fujitsu is now developing a cluster file system called FEFS for K computer.

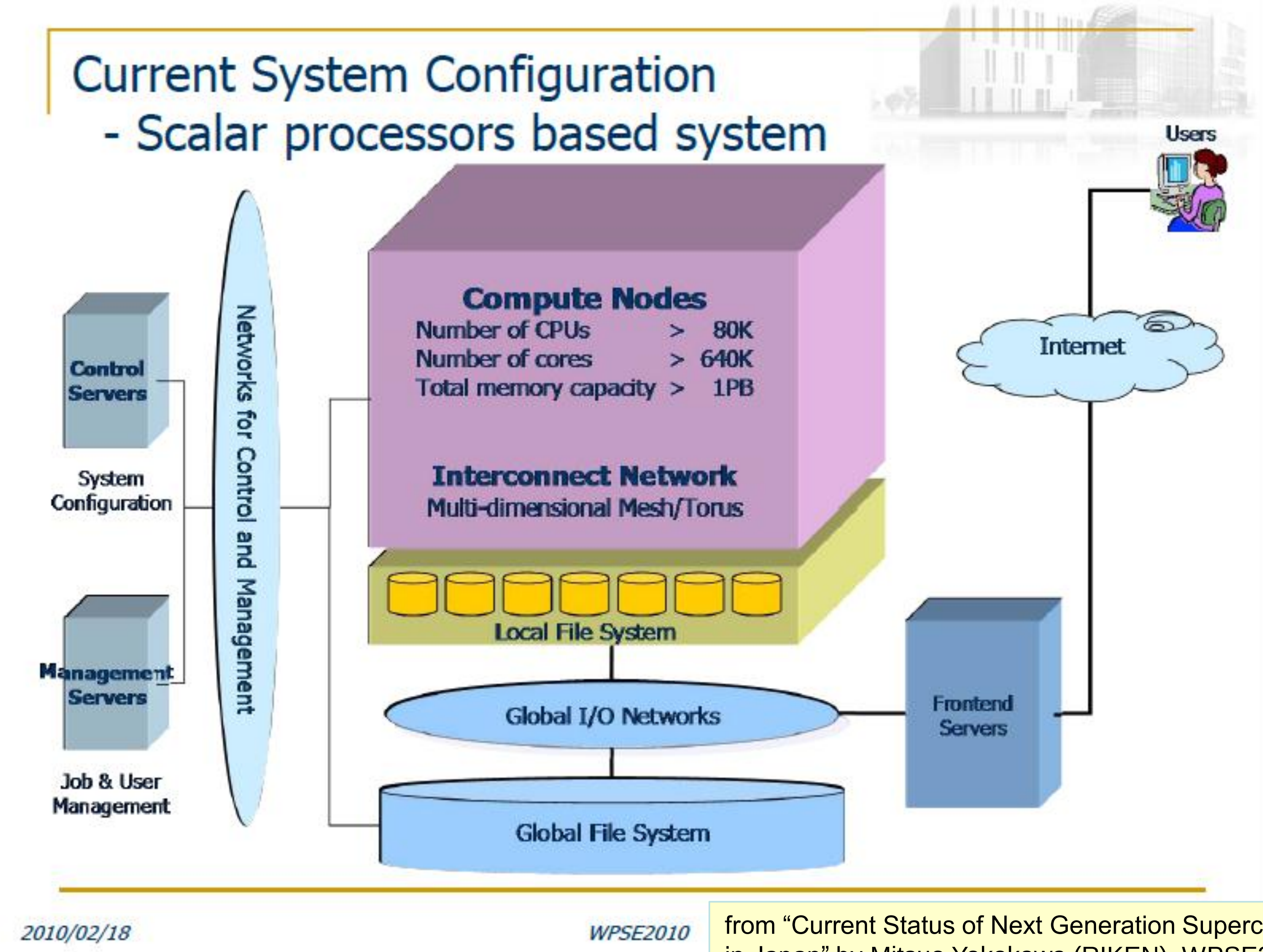


Miniature System Model

The First 8 Racks of K computer

K computer: System Configuration

FUJITSU

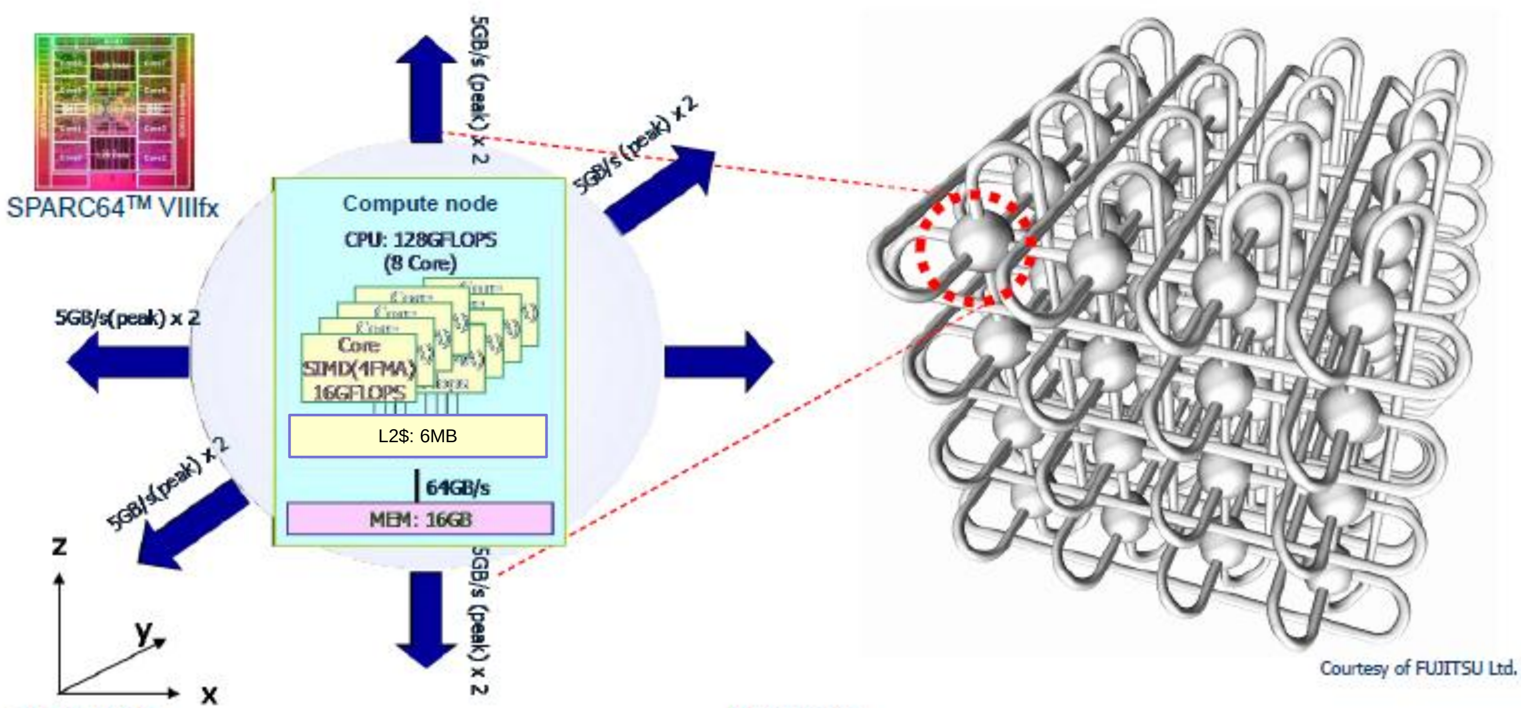


from “Current Status of Next Generation Supercomputer in Japan” by Mitsuo Yokokawa (RIKEN), WPSE2010

K computer: Compute Nodes

Compute Nodes

- Compute nodes (CPUs): > 80,000
  - Number of cores: > 640,000
- Peak performance: > 10PFLOPS
- Memory: > 1PB (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



2010/02/18 WPSE2010

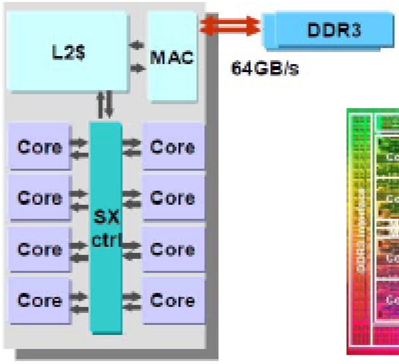
from "Current Status of Next Generation Supercomputer in Japan" by Mitsuo Yokokawa (RIKEN), WPSE2010

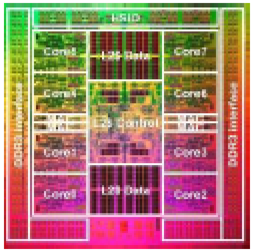
K computer: CPU Features

FUJITSU

CPU Features (Fujitsu SPARC64™ VIIIfx)

- 8 cores
- 2 SIMD operation circuit
  - 2 Multiply & add floating-point operations (SP or DP) are executed in one SIMD instruction
- 256 FP registers (double precision)
- Shared 6MB L2 Cache (12WAY)
  - Hardware barrier
  - Prefetch instruction
  - Software controllable cache
    - Sectored cache
- Performance
  - 16GFLOPS/core, 128GFLOPS/CPU





45nm CMOS process, 2GHz
22.7mm x 22.6mm
760 M transisters
58W (at 30℃ by water cooling)

Reference: SPARC64™ VIIIfx Extensions
http://img.jp.fujitsu.com/downloads/jp/jhpc/sparc64viiifx-extensions.pdf

2010/02/18 *WPSE2010*

from "Current Status of Next Generation Supercomputer in Japan" by Mitsuo Yokokawa (RIKEN), WPSE2010

Goals of Fujitsu's Clustre File System: FEFS

FUJITSU

- FEFS(Fujitsu Exabyte File System) for peta scale and exa-scale supercomputer will achieve:
  - Extremely Large
    - Extra-large volume (100PB~1EB).
    - Massive number of clients (100k~1M) & servers (1k~10k)
  - High Performance
    - Throughput of Single-stream (~GB/s) & Parallel IO (~TB/s).
    - Reducing file open latency (~10k ops).
    - Avoidance of IO interferences among jobs.
  - High Reliability and High Availability
    - Always continuing file service while any part of system are broken down.
- FEFS is optimized for utilizing maximum hardware performance *by minimizing file IO overhead*, and based on Lustre file system.

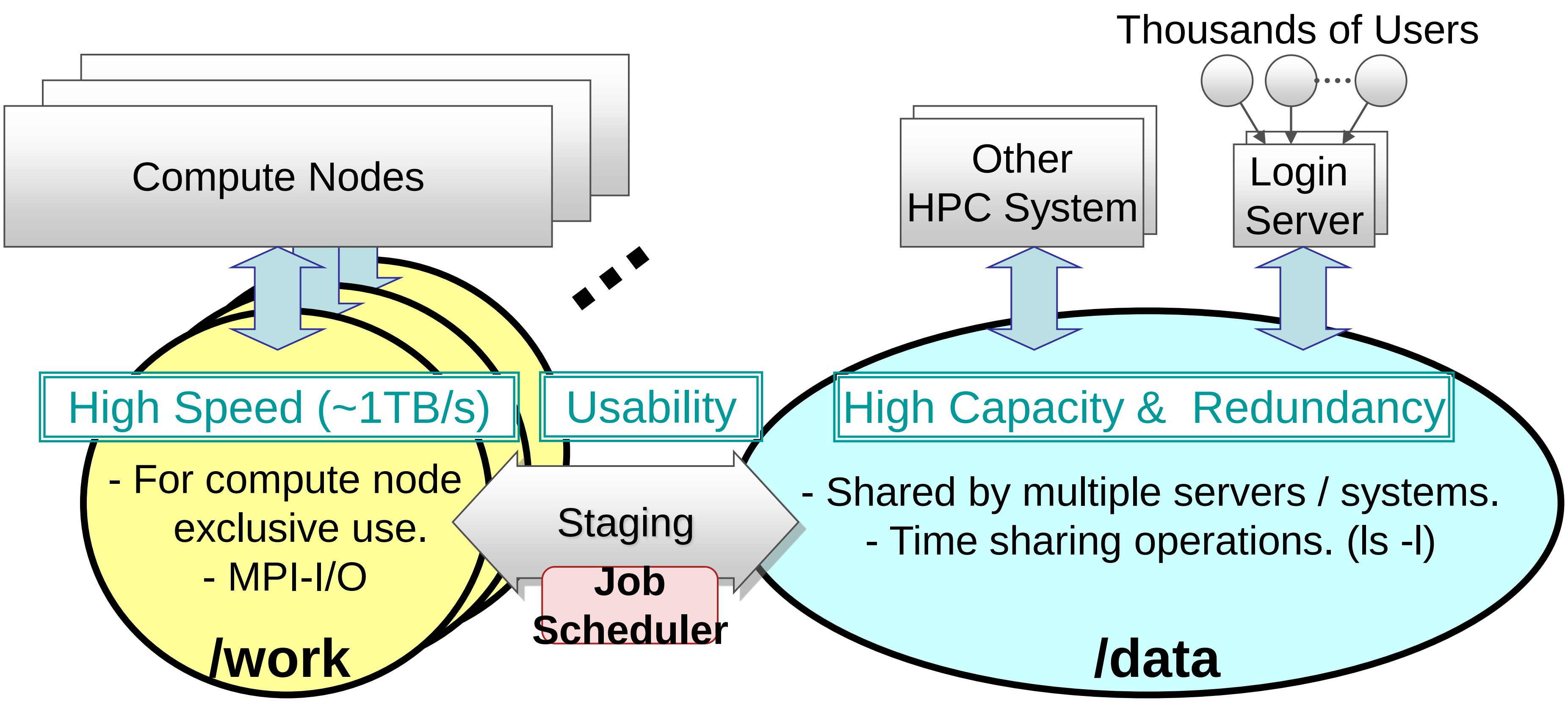
Design Challenges of FEFS for K computer

- How should we realize High Speed and Redundancy together?
  - There is design trade off.
- How do we realize Ultra High Scalability?
  - Over 1K IO servers
  - Accessed by over 100Ks of Compute nodes.
- How do we avoid I/O conflicts between Jobs?
  - Storage devices should be occupied for each Job
  - I/O operations by Multiple Users and Multiple Jobs
- How do we keep Robustness and High Data Integrity?
  - Eliminate single point failure

- To realize these challenges, we have introduced Integrated Layered File system with Lustre extensions.

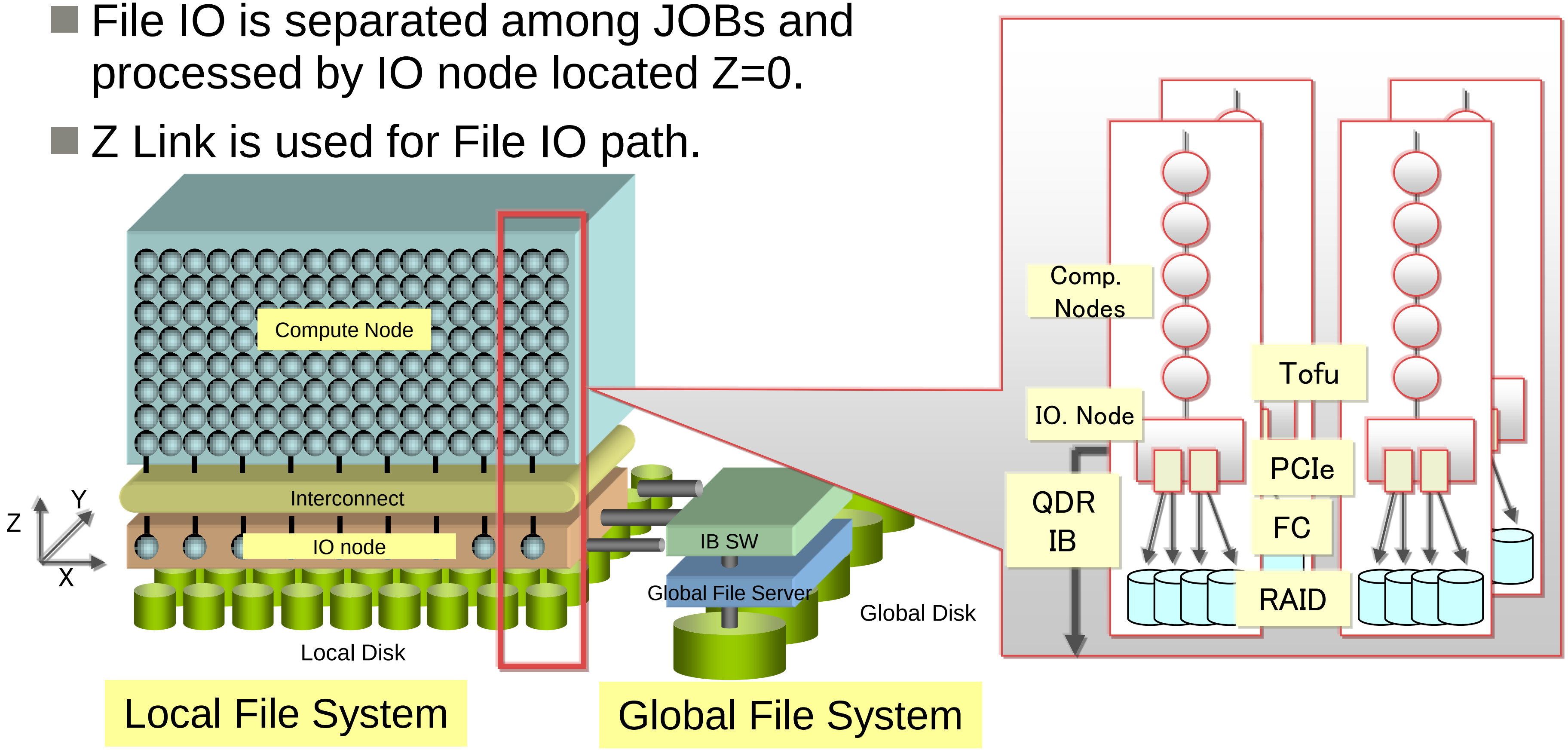
Integrated Layered Cluster File System

- Incompatible features is implemented by introducing Layered File System.
  - Local File System (/work): High Speed FS for dedicated use for jobs.
  - Global File System (/data): High Capacity and Redundancy FS for shared use.



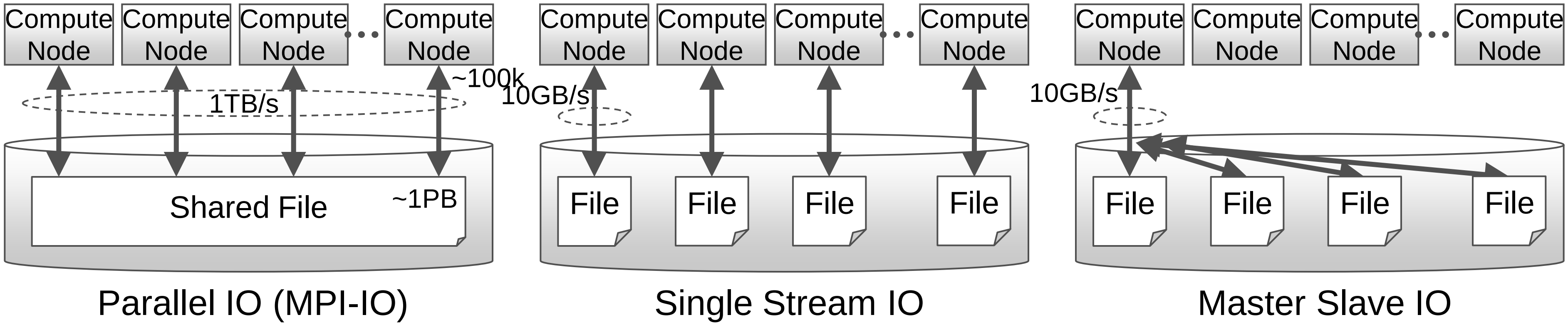
File IO Architecture

- Optimized for Scalable File IO Operation
  - Achieving Scalable Storage Volume and Performance
  - Eliminating IO Conflicts from Every Components
- IO Zoning Technology for Local File System
  - File IO is separated among JOBs and processed by IO node located Z=0.
  - Z Link is used for File IO path.

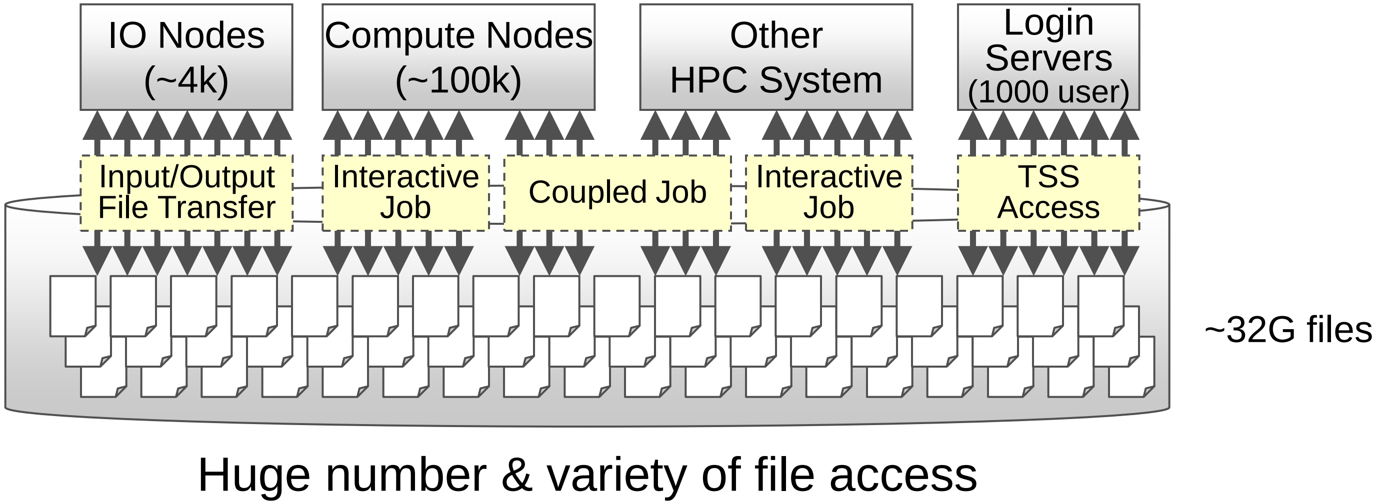


File System: IO Use Cases

- Local File System (/work)



- Global File System (/data)



FEFS Overview

- Lustre File System Based
  - Currently, 1.8.1.1 based
- Optimized for IO system architecture:
  - Integrated Layered Cluster File System
  - Drawing out ’s Hardware Performance and Reliability

- Main Features
  - Ultra high file IO and MPI-IO performance. (~1TB/s)
  - Low time impact on job execution by file IO.
  - Huge file system capacity. (100PB~)
  - Scalability of performance & capacity. (~4K OSS)
  - High reliability (Continuous service and Data integrity)
  - High usability (Easy system operation and management)
  - Dedicated resource allocation to avoid interferences among jobs

Lustre Extension of FEFS

FUJITSU

- Several functions are extended for our requirements.

Targets	Issues		Extension
Large Scale FS	File Size, Number of Files, Number of OSSs etc.		•File Size > 1PB to 8EB, Number of Files: 8 Exa •Number of OSSs: Thousands of OSSs
Performance	TSS Response		• TSS Priority Scheduling
	Meta Access Performance	Common	• Upgrading of Hardware Specification （Communication, CPU, File Cache, Disk) • Reducing Software Bottleneck
		Local File System	• MDS Distribution： Allocating Dedicated File System for each JOB
		Global File System	• Fairness among Users： QOS Scheduling for Users
	IO Separation among JOBs for Local File System		•IO Zoning: Processing IO nodes just below the computing nodes •Priority Scheduling
Availability	Recovering Sequence		• Recovering Sequences with Hardware Monitoring Support

Requirements for FEFS Lustre Extension(1/2)

Features		Current Lustre	2012 Goals
System Limits	Max file system size	64PB	100PB
	Max file size	320TB	1PB
	Max #files	4G	32G
	Max OST size	16TB	100TB
	Max stripe count	160	10k
	Max ACL entries	32	8191
Node Scalability	Max #OSSs	1020	10k
	Max #OSTs	8150	10k
	Max #Clients	128K	1M
Block Size of *ldiskfs*	(Backend File System)	4KB	~512KB
Patch-less Server		NA	Support

Requirements for FEFS Lustre Extension(2/2)

- (Cont'd)

Features	Current Lustre	2012 Goals
Big-endian support	NA	Support
Quota OST storage limit	<= 4TB	No limitation
Directory Quota	NA	Support
InfiniBand bonding	NA	Support
Arbitrary OST assignment	NA	Support
QOS	NA	Support

FEFS Technical Issues

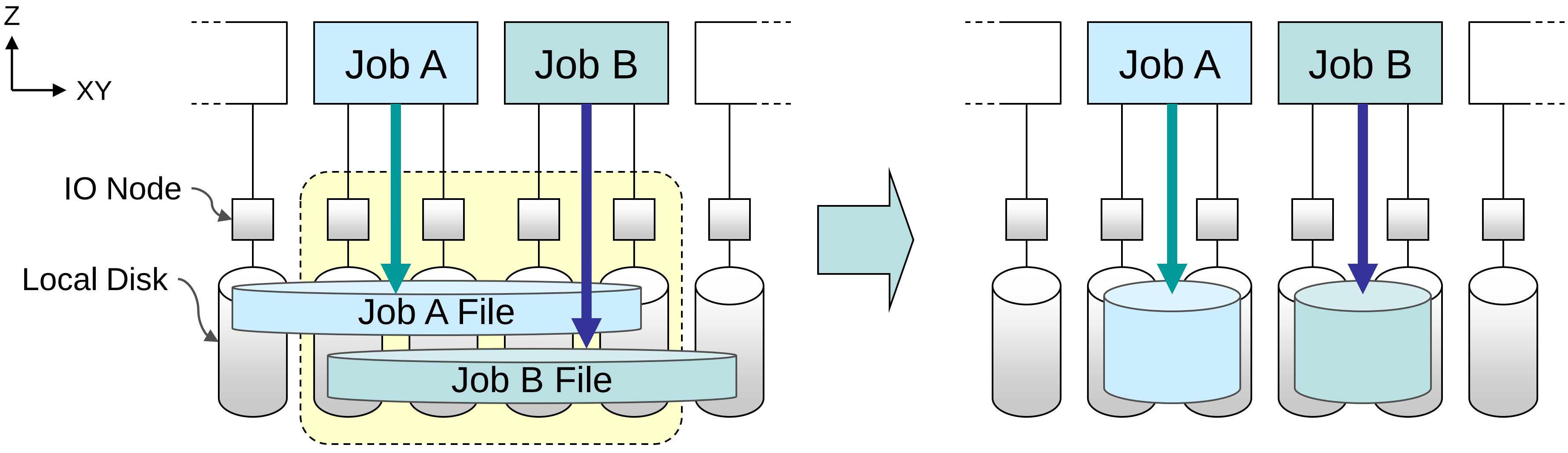
- MDS and OSS Scalability
- IO Zoning
- QOS
- Staging (K computer Specific)
- Other Issues

MDS and OSS Scailability

- Locality is very important for increasing scalability
  - Locality : Storage, OSS, Interconnect, Server
- Our strategy: Utilizing these locality as much as possible
  - MDS Scalability:
    - Dividing file systems
    - Clustered MDS (Future Lustre 2.x Based)
  - OSS Scalability over 1000 of servers:
    - Minimizing OSS server response
    - Avoiding Interconnect congestion
    - Minimizing Storage(RAID) response

IO Zoning: IO Separation among JOBs

- Issue: Sharing disk volumes, network links among jobs cause IO performance degradation because of their conflicts.
- Our Approach: Separating of disk volumes, network links among jobs as much as possible.



× w/ IO Conflict

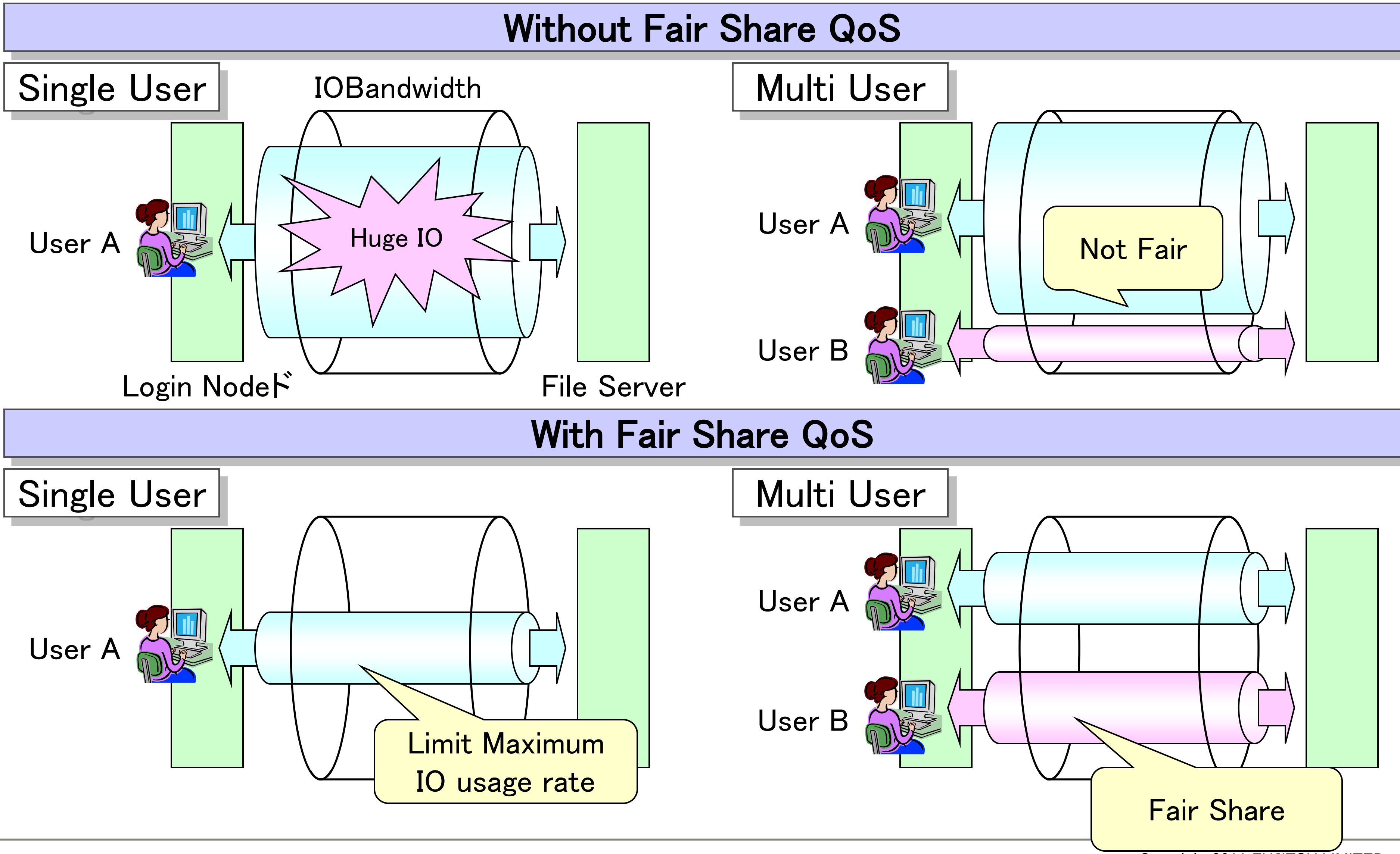
O w/o IO Conflict

SP-QoS： Selectable Policy QoS

- System manager is able to select:
  - Fair Share QoS
  - Best Effort QoS

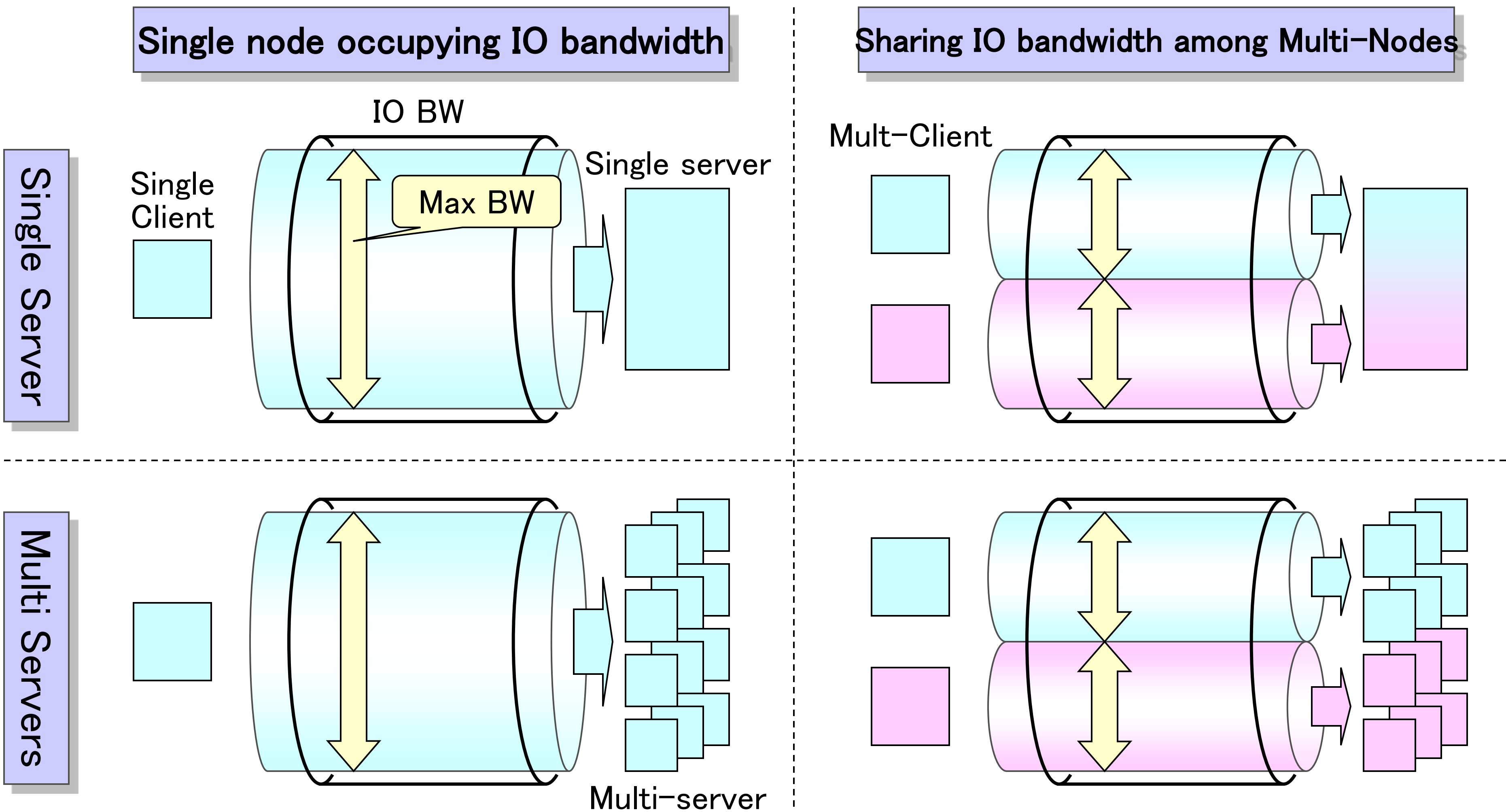
Fair Share QoS

- Avoiding from some one’s occupying file IO resources



Best Effort QoS

- Fair Share among users



Staging: System Controlled File Transfer

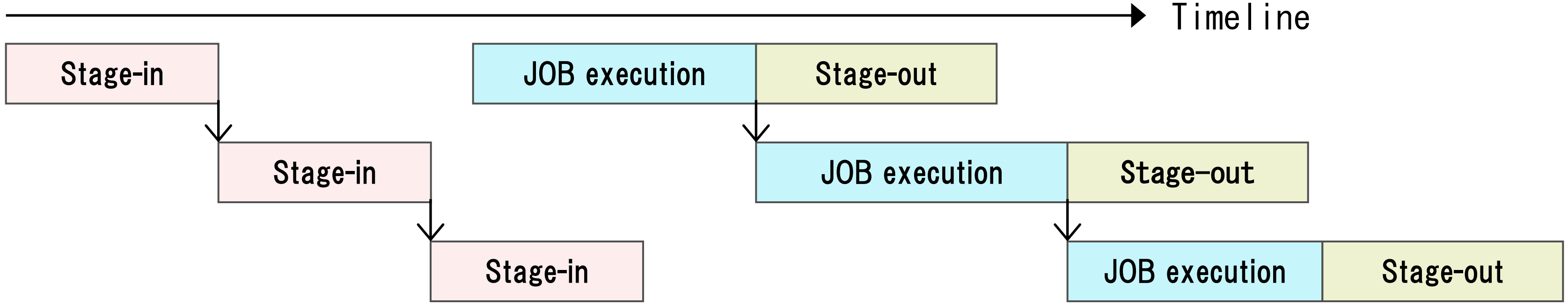
FUJITSU

- Goal: Minimizing IO conflict by controlling copying speed and IO timing.
  - Stage-in: from Global File System to Local File System.
  - Stage-out: from Local File System to Global File System.
- Staging Directive: Written by user in JOB script
  - Ex.) Transferring a.out file to rank 0-15 nodes.

```
#XXX -I "0-15@ ./a.out %r : ./"
```

K computer Specific

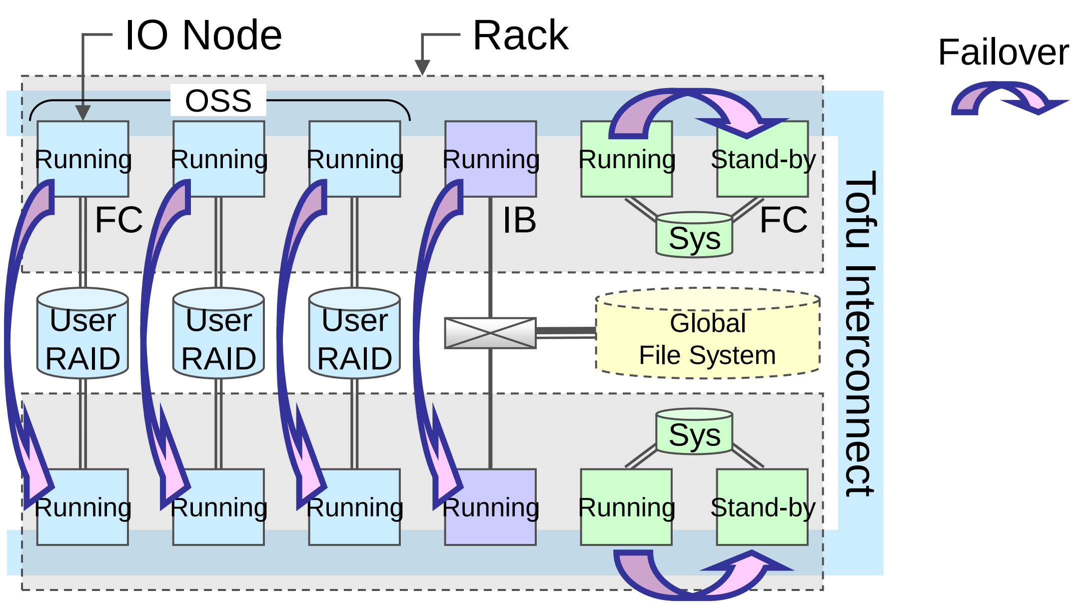
- Staging Timing:
  - Pre-Staging is controlled by JOB Scheduler
  - Stage-out is processed during JOB execution



High Reliability and High Availability

FUJITSU

- Issue: Keeping System Reliability and Availability against failures
- Our Approach：
  - Full Duplex Hardware Architecture for 24 hour 365 day system running against single point of failure
    - Duplex paths of IB, FC, IO Server, Data Robustness using RAID Disks（MDS：RAID1+0, OSS（IO node）：RAID5）
  - Server and communication link are dynamically switched against their failure by software.
    - File Service is not stopped at IO node, FC or IB failure and maintenance.



Other Issues

- Kernel Independent Lustre Servers(MDS, OSS)
  - Current Implementation depends on kernel source codes, especially ext file system.
- LNET, OSS, MDS setting for 100 thousands of Clients and servers
  - Automatic configuration is needed.
  - Checking connectivity among servers
- Data Integrity
  - Fsck of whole data is impractical for over petabyte file system.

Experimental Evaluation Results of FEFS

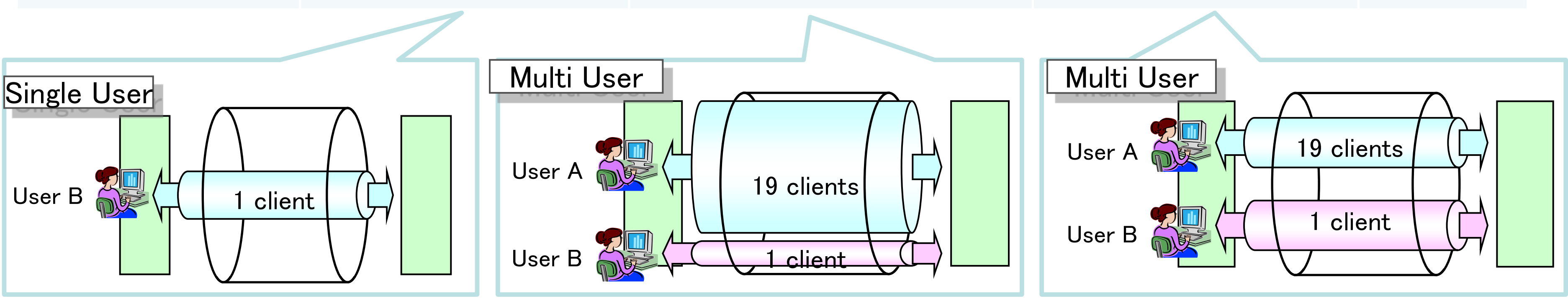
FUJITSU

- IOR Results (864 OSTs, 10000 Clients):
POSIX WRITE 96GiB/s, POSIX READ 148GiB/s

IOR Result of POSIX IO, 10000 clients, 864 OSTs
Command line used: /mnt/client/IOR/IOR –F –C –i 3 –b 1g –t 16m –o
Max Write: 95990.84 MiB/sec (100653.69 MB/sec)
Max Read: 147962.87 MiB/sec (155150.31 MB/sec)

- QOS Results on PC Cluster: Best Effort and Fair Share for two users (User A: 19 node Job, User B: 1 node Job)

User B 10000 Files	w/o QOS Single User	w/o QOS Multi User	w/ QOS Multi User	
Create Files	4.089	10.125	3.932	sec
Remove Files	4.235	14.018	5.534	sec



Summary and Future Works

- We described overview of FEFS (Fujitsu's Lustre Based File System) for the 'K computer' developed by RIKEN and Fujitsu.
  - High-speed file I/O and MPI-IO with low time impact on the job execution.
  - Huge file system capacity, scalable capacity & speed by adding hardware.
  - High-reliability (service continuity, data integrity), Usability (easy operation & management)
- Future Works
  - Rebase to newer version of Lustre (1.8.5, 2.x)

FUJITSU

shaping tomorrow with you