Current Status of FEFS for the K computer

Shinji Sumimoto
Fujitsu Limited
Apr.24 2012 LUG2012@Austin

Outline

- RIKEN and Fujitsu are jointly developing the “K computer”*
  - Development continues with system software tuning for completion in June 2012.
- Outline of This Talk
  - K computer and FEFS Overview
  - Development Status of FEFS
  - Performance

* Nickname of the “Next Generation Supercomputer” developed by RIKEN and Fujitsu

World's No.1 at SC11

Again on TOP500 List

Performance of over 10 Peta*flops

K computer

4 of HPC Challenge Awards

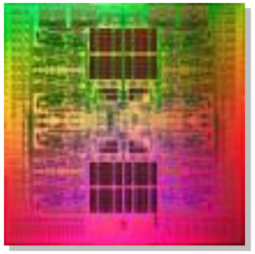
SC11 Gordon Bell Prize

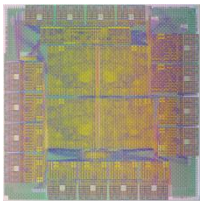
SC11 TOP500 #1

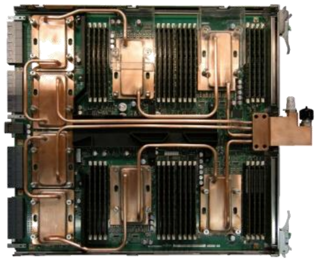
TOP500 awarding in SC11

***10 Peta = 10,000,000,000,000,000**

System Overview of K computer







Processor: SPARC64™ VIIIfx

- Fujitsu's 45nm technology
- 8 Core, 6MB Cache Memory and MAC on Single Chip
- High Performance and High Reliability with Low Power Consumption

Interconnect Controller:ICC

- 6 dims-Torus/mesh (Tofu Interconnect)

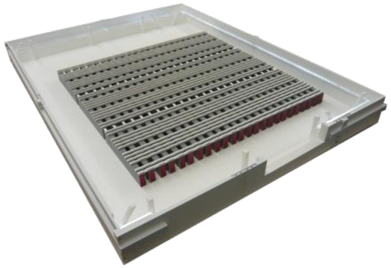
System Board: High Efficient Cooli

- With 4 Computing Nodes
- Water Cooling: Processors, ICCs etc
- Increasing component lifetime and reducing electric leak current by low temperature water cooling

Rack:High Density

- 102 Nodes on Single Rack
  - 24 System Boards
  - 6 IO System Boards
  - System Disk
  - Power Units

(10PFlops: >800 Racks)

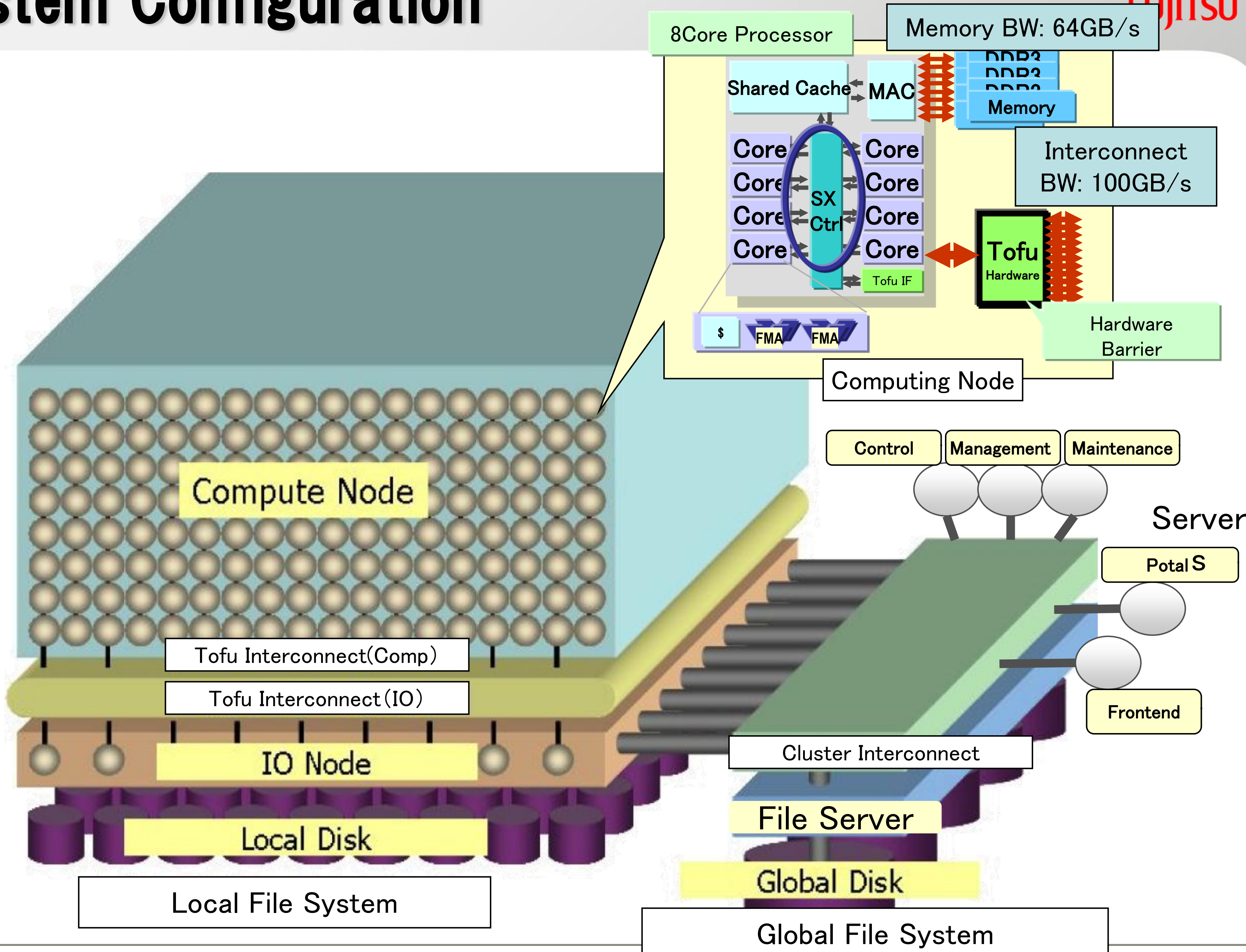


Our Goals

- Challenging to Realize World's Top 1 Performance
- Keeping Stable System Operation over 80K Node System

System Image

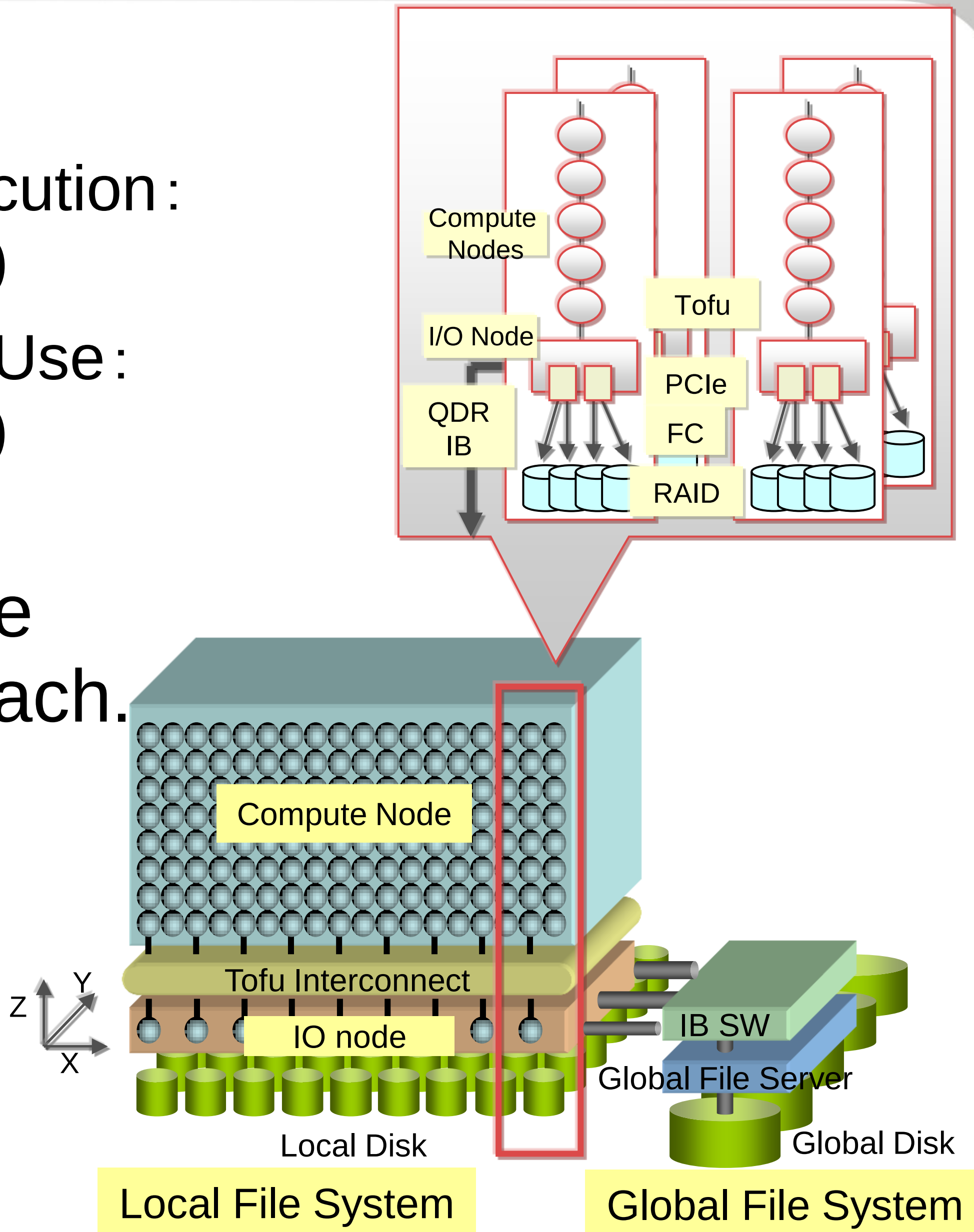
System Configuration



IO Architecture of "K computer"

FUJITSU

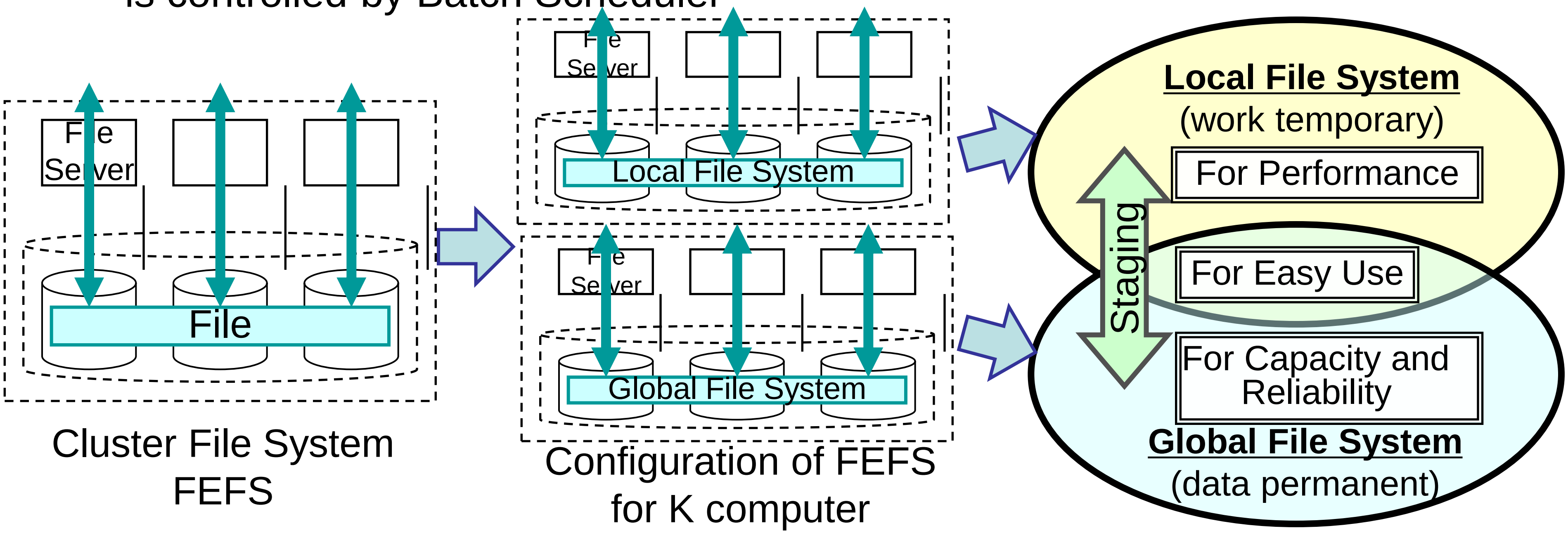
- IO System Architecture
  - Local Storage for JOB Execution：ETERNUS(2.5inch, RAID5)
  - Global Storage for Shared Use：ETERNUS(3.5inch, RAID6)



- Configurations of each file system is optimized for each.
  - Local File System：Over 2400-OSS (for Highly Parallel)
  - Global File System：Over 80-OSS (for Big Capacity)

Overview of FEFS

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



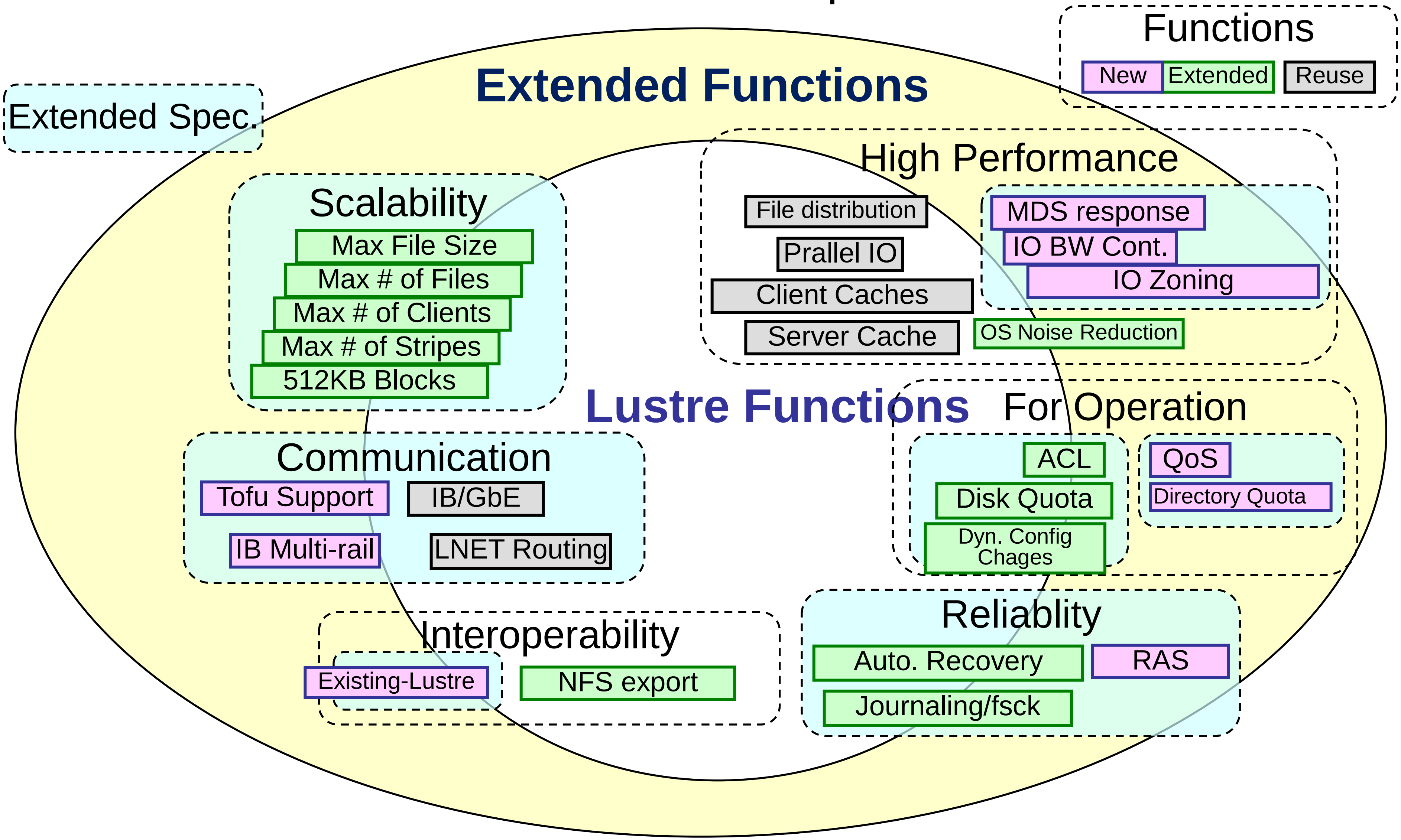
Lustre Specification and Goal of FEFS

Features		Current Lustre	Our 2012 Goals
System Limits	Max file system size	64PB	100PB (8EB)
	Max file size	320TB	1PB (8EB)
	Max #files	4G	32G (8E)
	Max OST size	16TB	100TB (1PB)
	Max stripe count	160	20k
	Max ACL entries	32	8191
Node Scalability	Max #OSSs	1020	20k
	Max #OSTs	8150	20k
	Max #Clients	128K	1M
Block Size of *ldiskfs* (Backend File System)		4KB	~512KB

These were contributed to OpenSFS 2/2011

Lustre Extension of FEFS

- We have extended several Lustre specifications and functions



FEFS Development Status on the K computer

- Currently Almost of All the functions are implemented and in testing phase.
  - Current Base Version: 1.8.5 + α
  - 8PF Class Scalability Testing

- Main Testing Target:
  - User's available memory over 90% of physical memory
  - Minimizing impact of OS jitter to application performance
  - IO performance
  - RAS
- We will discuss memory issues and OS jitter

To Solve Memory Issues

- Goal: System memory usage < 1.6GB

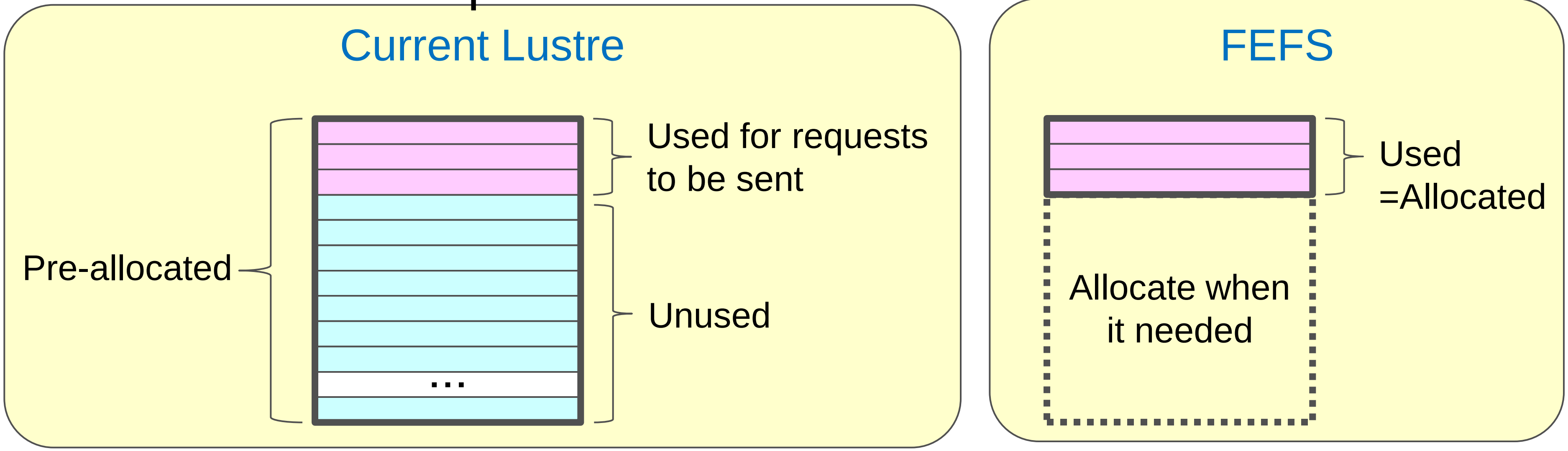
- Minimizing memory usage for buffer cache on each FEFS client
  - Added some functionality to limit dirty pages on write operation.
- Minimizing memory consumption of FEFS for ultra large scale system
  - Investigating the reasons and fix them.
  - We found that there are several issues on current basic Lustre designs

Memory Issue: Request Buffer (1)

- Issue: Request buffer on client is **pre-allocated by #OSTs** in Lustre.
  - Buffer size = **8KB x 10 x #OSTs / request**

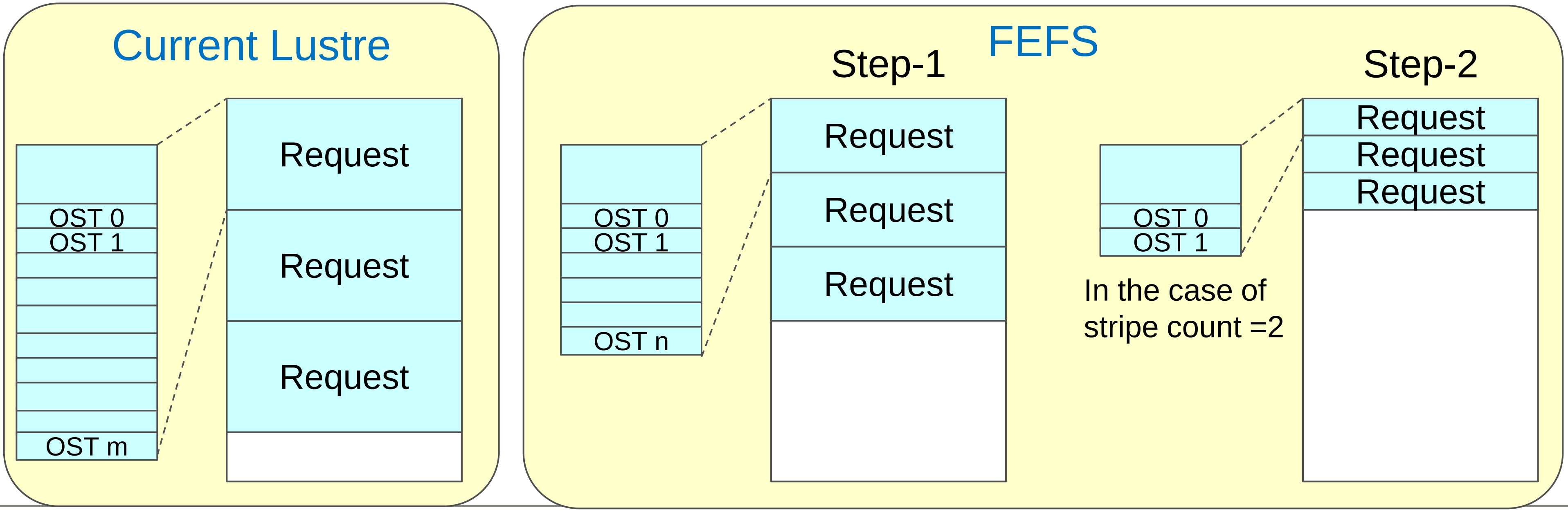
    #OST=1,000 ⇒ 80MB / request

    #OST=10,000 ⇒ 800MB / request
- Our Approach: On demand allocation: Allocate request buffer when it required.



Memory Issue: Request Buffer (2)

- Issue: When create a file, client allocates "**24B x Max. OST index**" size of request buffer.
  - **OST index = 1,000 ⇒ 23KB / request**
  - **OST index = 10,000 ⇒ 234KB / request**
- Our Approach
  - Step-1: Reduce buffer size to "**24B x #Existing OSTs**".
  - Step-2: Minimize to "**24B x #Striped OSTs**".



Memory Issue: Granted Cache

Issue

- Dirty pages† on the clients have to be written to target OST. † Default is 32MB (max_dirty_mb)
- Each OST keeps free disk space for all clients.
  - Required free space = **32MB/client x #Clients / OST**

    10,000 Clients ⇒ 320GB/OST

    100,000 Client ⇒ 3200GB/OST (3.2TB!!)

Our Approach

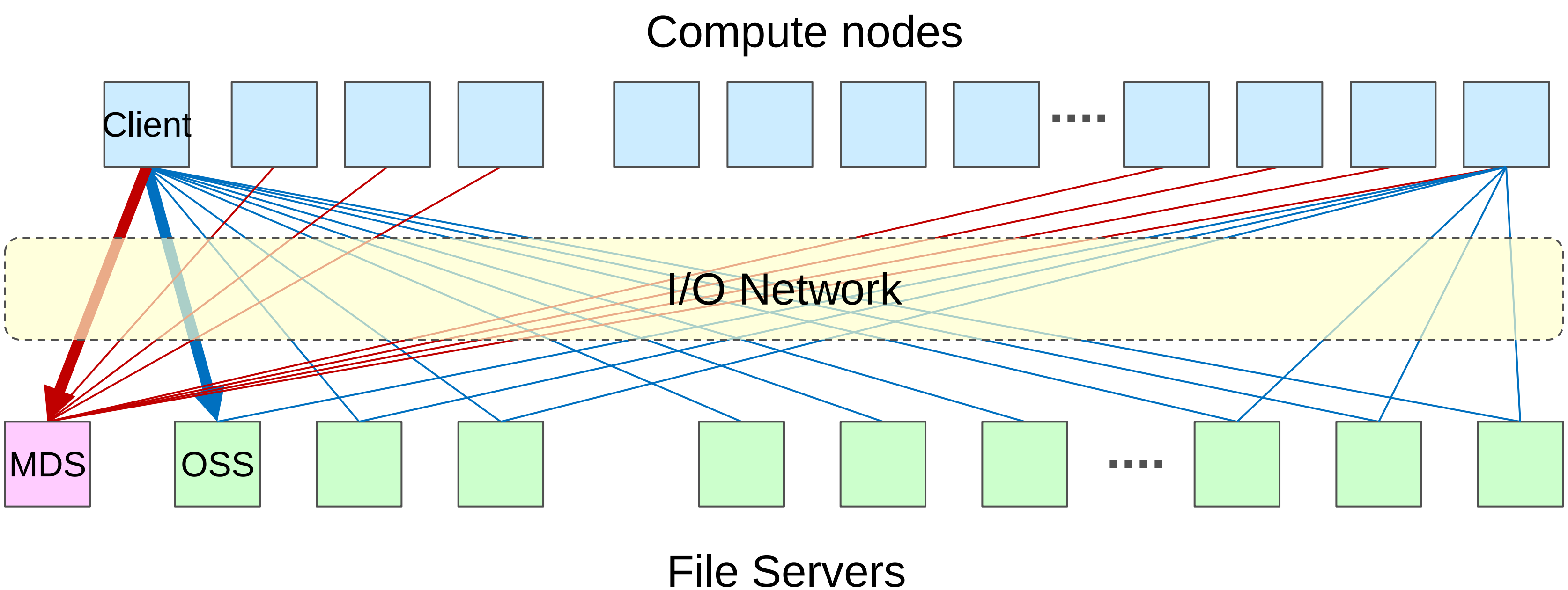
- Shrinking dirty pages limitation: 32MB ⇒ 1~4MB
  - This **causes serious degradation** of I/O performance. (Not acceptable)
- It's supposed to be fixed in Lustre 2.x

To Solve the OS Jitter Problem

- Minimizing Network Traffic Congestion
  - ll_ping
- Minimizing Background Daemons
  - ll_ping
  - ldlm_poold

Minimizing Network Traffic Congestion by ll_ping

- Network congestion and request timeout cause: #of monitoring pings ∝ “#of clients x #of servers”
  - MPI and file I/O communication degradation.
  - Application performance degradation by OS jitter.
- Our Solution: Stopping interval ll_ping

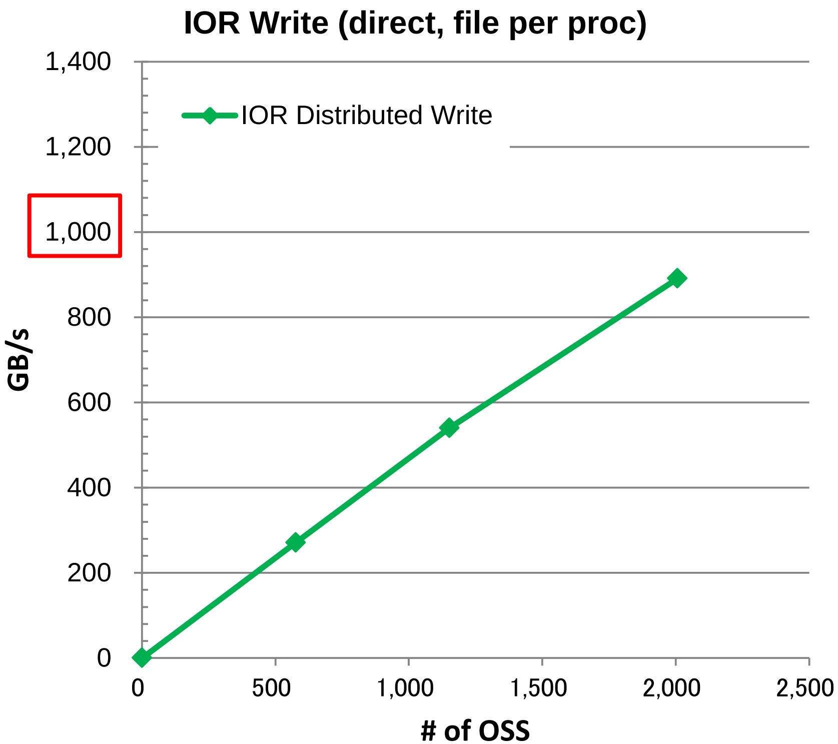


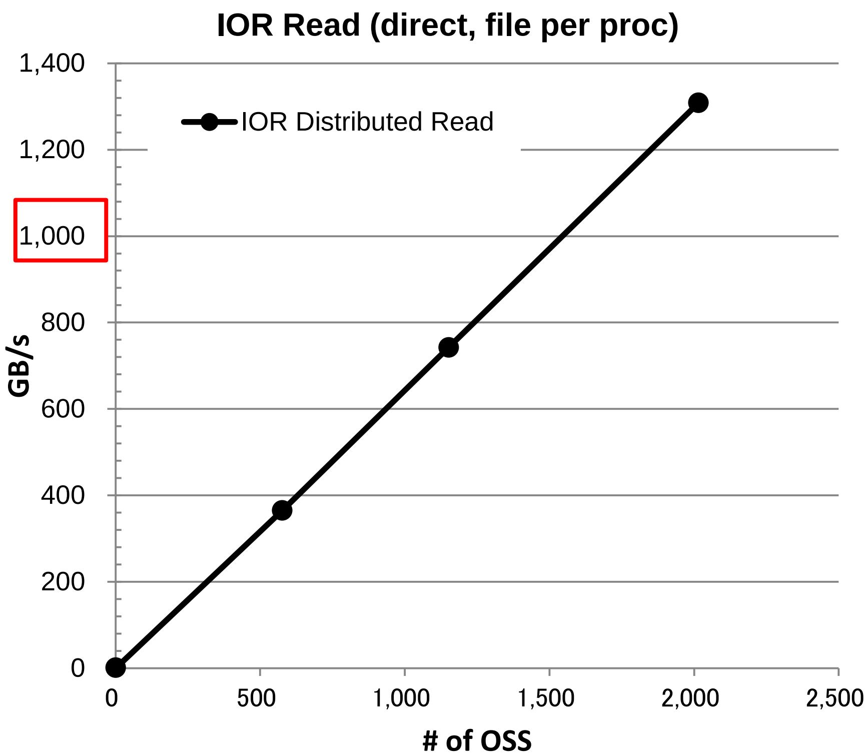
Minimizing Background Daemons

- ll_ping Problem:
  - All clients broadcast monitoring pings to all OSTs (not OSS) at regular intervals of 25 seconds.
    100K Clients x 10K OSTs ⇒ 1G pings every 25 seconds.
    - Sending Ping and receiving Ping become OS zitter.
  - Our Solution: Stopping broadcasting pings on clients.
    - Other pings, for recovery and for I/O confirmation, etc., are kept
    - Node failure is detected by system function of K computer
- ldlm_poold Problem:
  - Operation time of *ldlm_poold* on client increases in proportion to the number of OSTs. *It* manages the pool of LDLM locks. It wakes up regular interval of 1sec.
  - Our Solution: Reduce the processing time per operation of *ldlm_poold* by divide the deamon’s internal operation.

Preliminary Benchmark Results on 8PF Local File System

- Over 1TB/s IOR Bandwidth on the K computer.
  1.3TB/s IOR Read on 80% of K computer





	Measured (2007 OSS)	Full Estimated (2592 OSS)
Read	1.31 TB/s	1.68 TB/s
Write	0.83 TB/s	1.06 TB/s

Performance Evaluation of FEFS (2)
(Collaborative work with RIKEN on K computer)

Metadata performance of mdtest. (unique directory)

- FEFS (K computer)

  MDS:RX300S6 (X5680 3.33 GHz 6core x2, 48GB, IB(QDR)x2)

- FEFS, Lustre (IA)

  MDS:RX200S5 (E5520 2.27GHz 4core x2, 48GB, IB(QDR)x1)

	FEFS		Lustre	
IOPS	K computer	IA	IA	
	FEFS	FEFS	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

*We will evaluate latest Lustre 2.1.0 performance.

Summary and Future Work

- We described overview of FEFS 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 Work
  - Stable Operation on K computer
  - Rebase to newer version of Lustre (2.x)
  - Contribution our extension to Lustre Community

Press Release at SC11

FUJITSU

Whamcloud and Fujitsu to Collaborate on Lustre Development

Fujitsu to advance Lustre development for HPC

Danville, CA – November 15, 2011 – Whamcloud, a venture-backed company formed from a worldwide network of high-performance computing (HPC) storage industry veterans, and Fujitsu, the global IT products and services company, and together with RIKEN, the joint developer of the world's fastest supercomputer, the K computer[(1)], announced today that both parties agreed to the principal terms of joint Lustre development. This collaboration will include scalability and file system work for Lustre, and merging Fujitsu's Lustre enhancements into the Lustre 2.x community release.

"Lustre is a central technology in our supercomputing products, and we look forward to working closely with Whamcloud, the leader in file system software technologies, to advance performance, add features and push supercomputing capabilities to new levels," said Yuji Oinaga, Head of Next Generation Technical Computing Unit at Fujitsu. "Fujitsu is committed to being at the forefront of supercomputing technologies."

"Working with Fujitsu is an extreme honor, and we look forward to their Lustre enhancements benefiting the entire community," said Brent Gorda, CEO of Whamcloud. "Lustre is the most widely used file system in HPC and is deployed in the most extreme computing environments. Fujitsu's rigorous quality standards are well-known and this agreement is a great vote of confidence for the future of Lustre.

For more details on Whamcloud and its Lustre support and development services, please see: http://www.whamcloud.com.

FUJITSU

shaping tomorrow with you