Stable and Scalable Operation of Parallel File Systems at the K computer

Yuichi Tsujita
RIKEN AICS

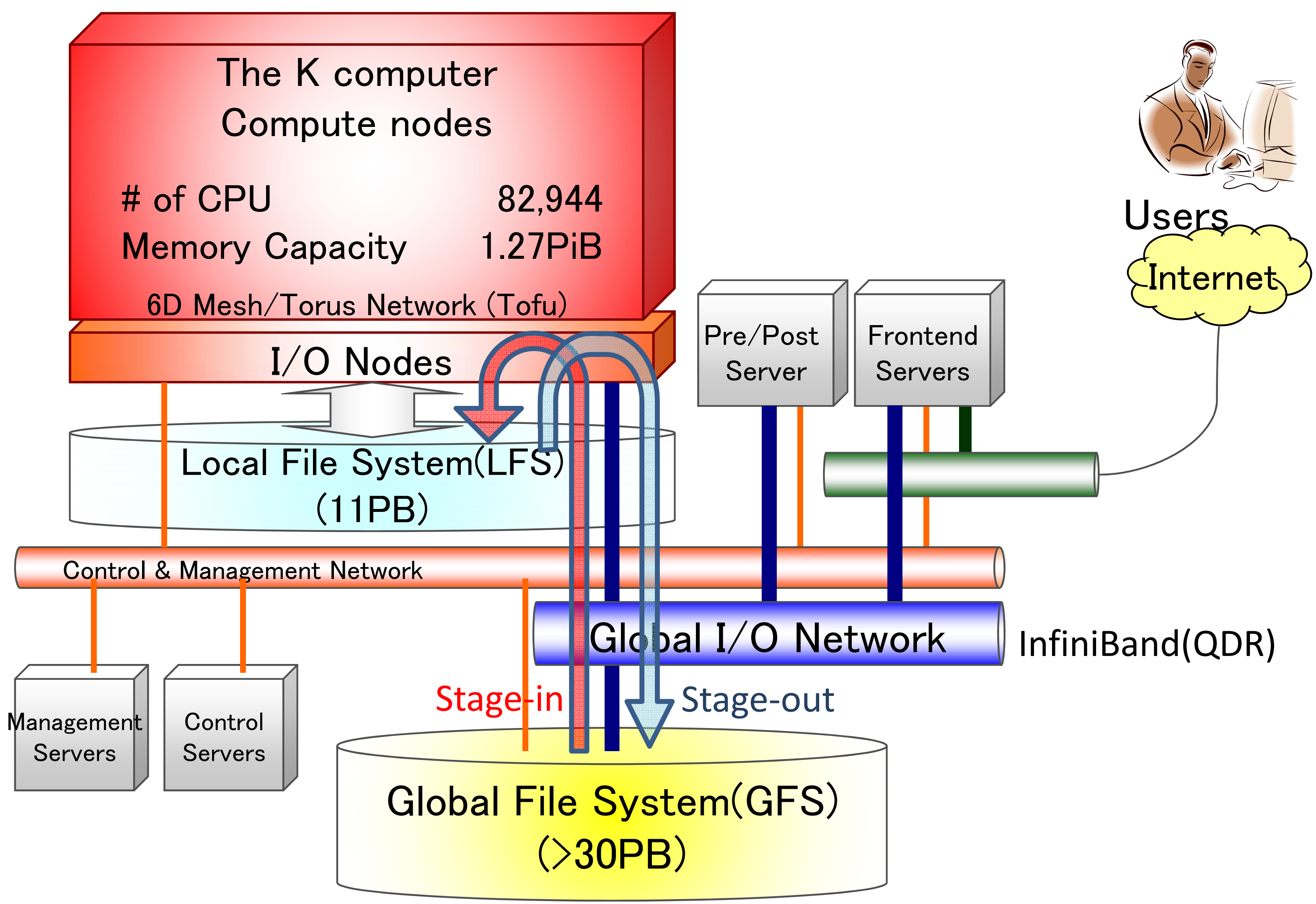
Nov. 2, 2017@JLUG2017

Outline

- Overview of the K computer and its file systems
- Activities for stable and scalable operation
  - Alleviation of MDS load by using loop-back file systems
  - Elimination of client evicts
  - Alleviation of I/O interference by huge data accesses
- Summary

Overview of the K computer and its file systems

Overview of the K computer



FEFS is used for both LFS and GFS.

(FEFS: Fujitsu Exabyte File System based on Lustre technology)

File systems at the K computer

- Organization of file systems at the K computer
  - LFS : Performance oriented
    - for high performance I/O during computation
  - GFS : Capacity oriented
    - for huge data storing and high redundancy

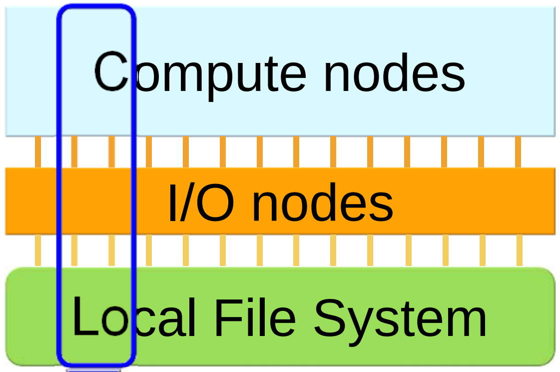
File system	LFS	GFS *1
Total volume size	~ 11 PB	> 30 PB
# volumes	1	11
# OSSs	2,592	108
# OSTs	5,184	3,024
Disk system of OST	RAID5+0	RAID6 RAID6 FR (new three volumes only) *2

*1 New three volumes have been operated since Apr. 2017.

*2 Extended RAID6 from Fujitsu (RAID6 FR) available for the new three volumes

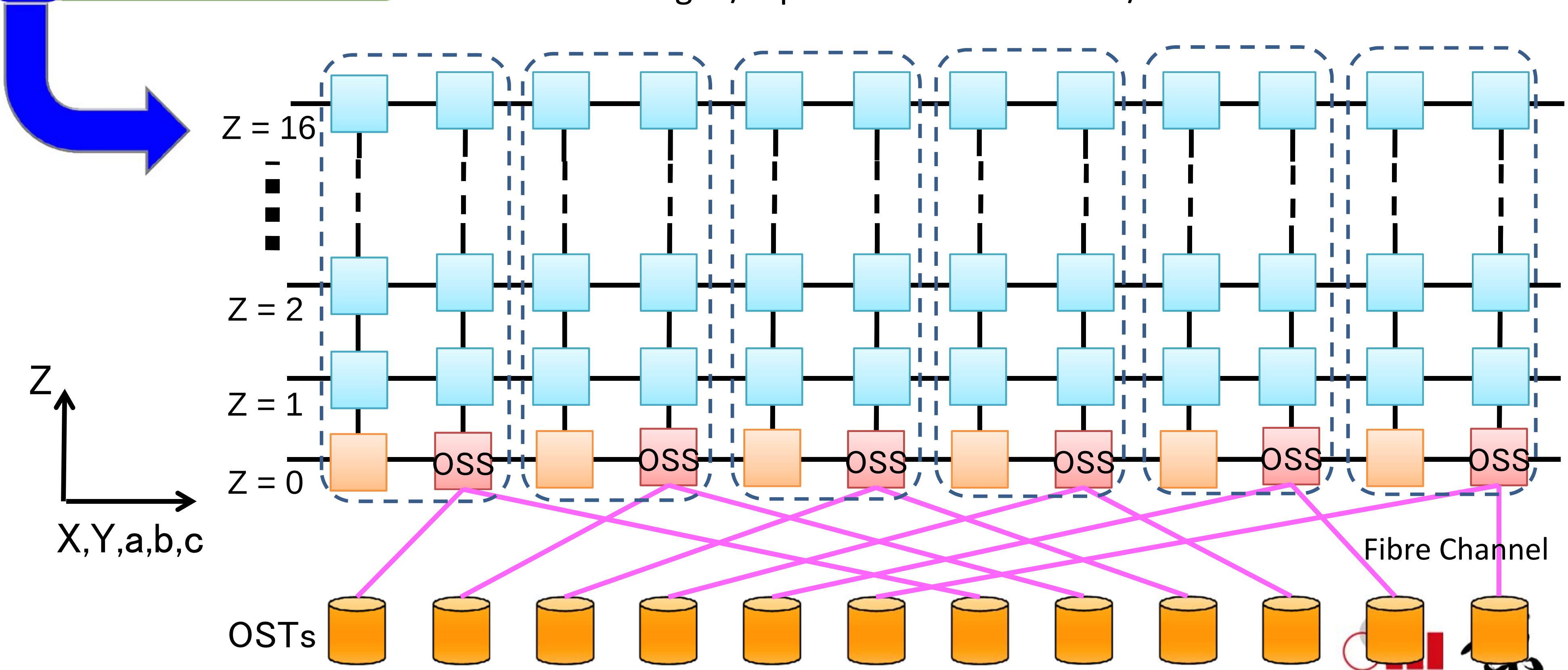
LFS and I/O zoning

- Configuration of a LFS



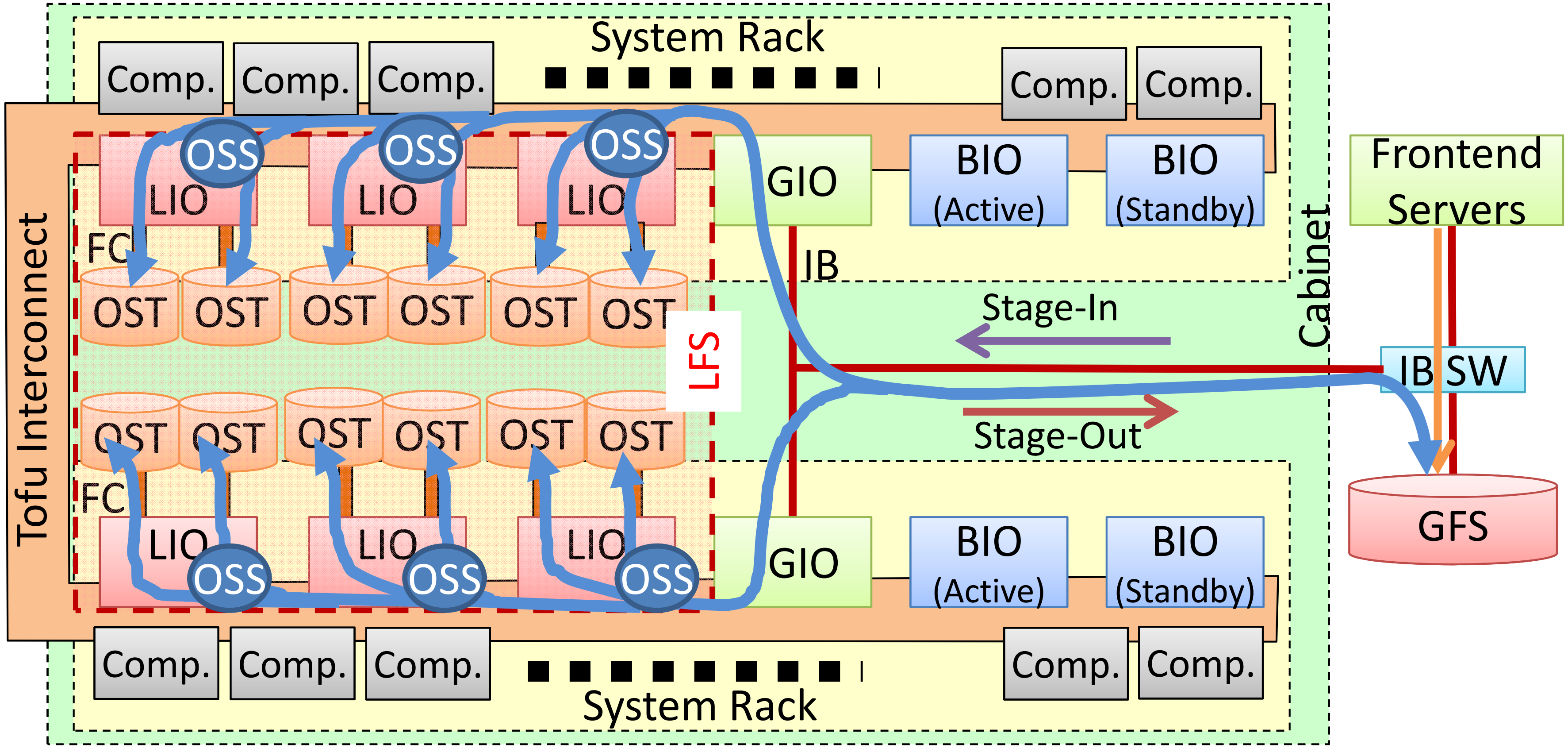
- I/O Zoning for LFS
  - File I/O is separated among jobs and processed by I/O nodes located at Z=0. (OSS is running on I/O nodes.)
  - Every OSTs are accessible from I/O nodes via Fibre Channel.
  - Z link is used for file I/O.

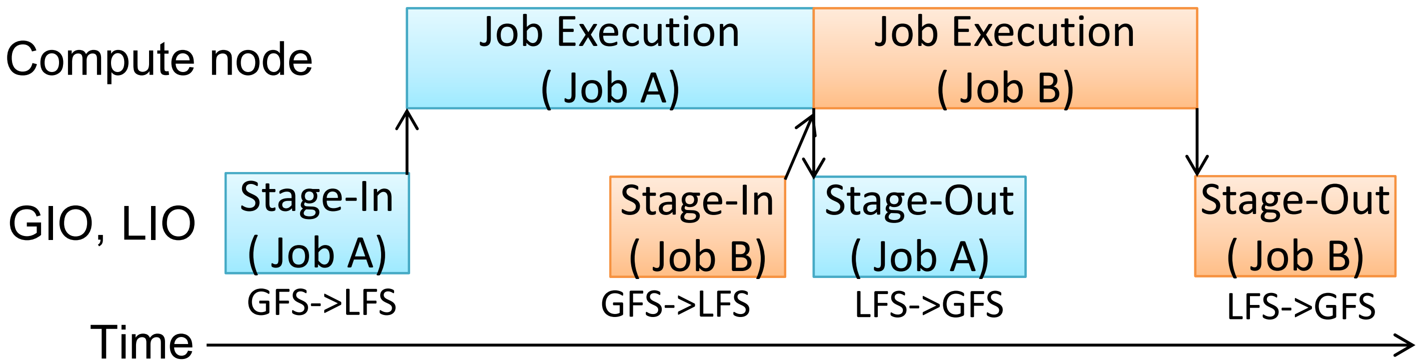
High I/O performance and low I/O interference



Data-staging

- Asynchronous data staging for high efficient job scheduling

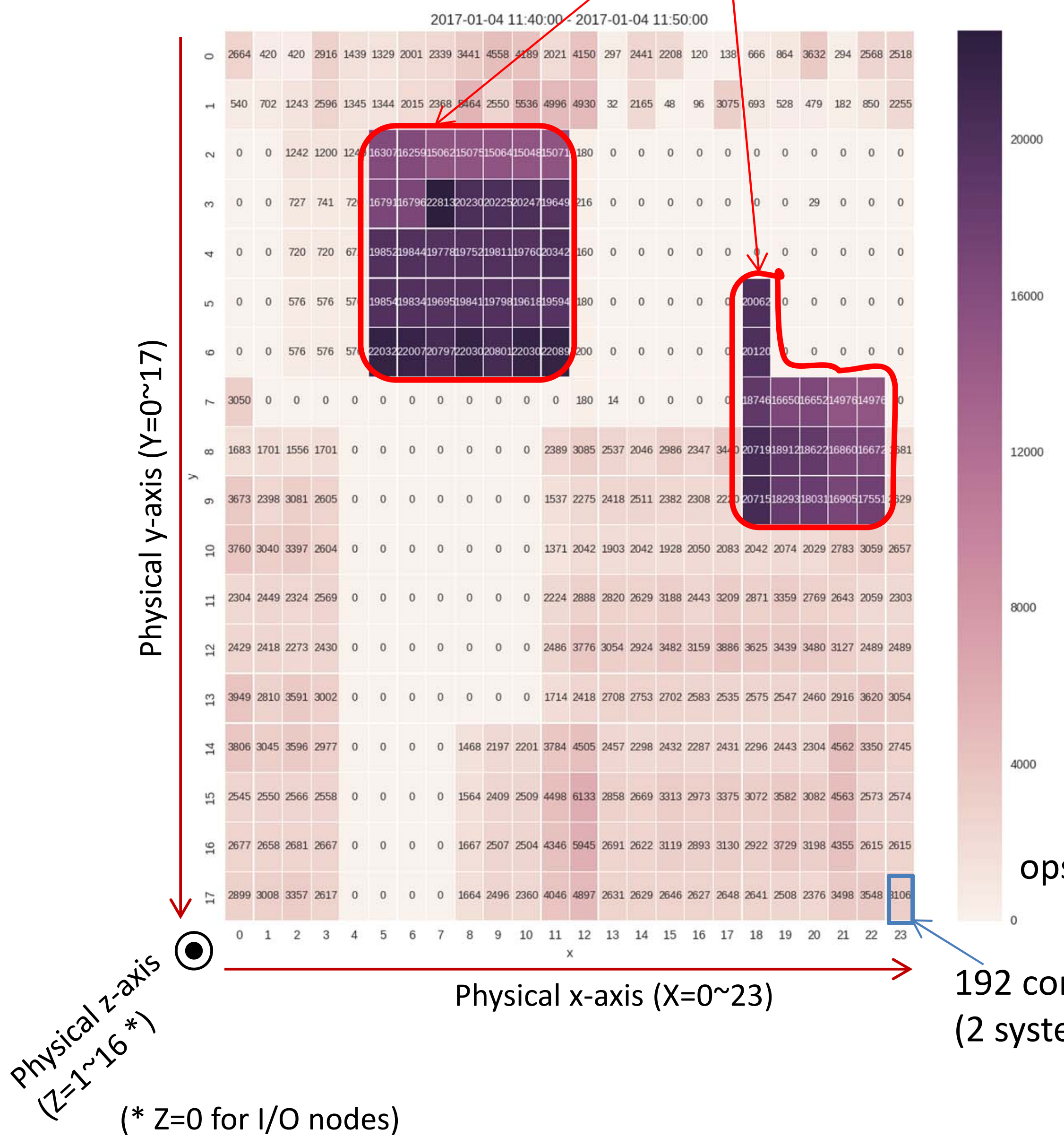




Activities for stable and scalable operation

- Alleviation of MDS load using loop-back file systems
- Elimination of client evicts
- Optimization for alleviating interference by huge data accesses

High load of MDS (LFS)

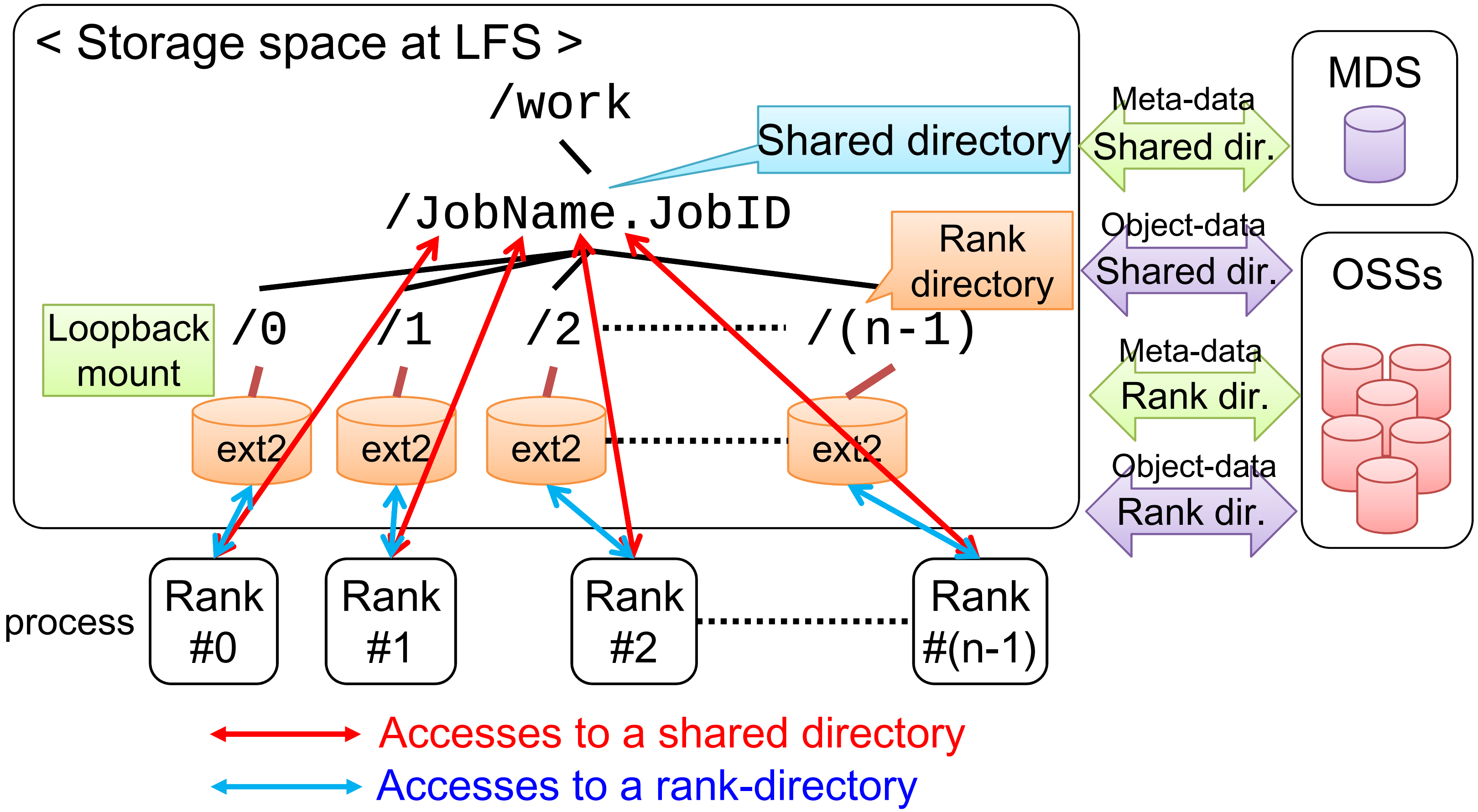


- Many file accesses(open, close, …) lead to high MDS load.
- High MDS load of LFS may affect many applications accessing LFS.

- Providing loop-back file systems (rank-directory) to alleviate high MDS load
- Rank-directory is recommended for applications which access many files and file per rank case.

AICS K computer

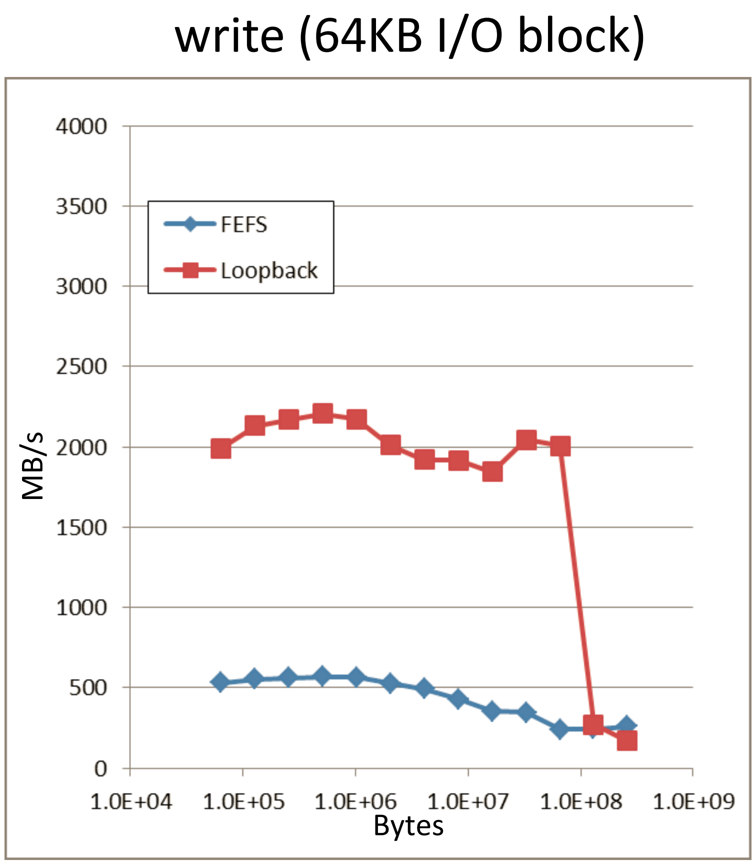
Rank-directory (loopback file system)

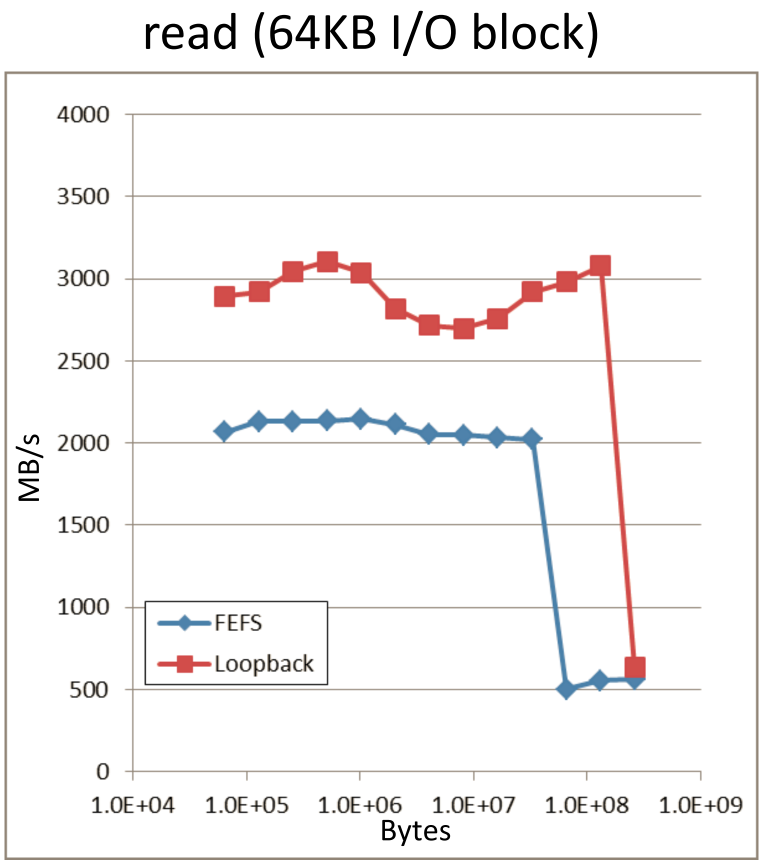


- Reducing MDS accesses leads to effective utilization of LFS.
- I/O accesses in rank-directories are free from slowdown of MDS performance.

Single node I/O performance evaluation by using IOzone

- FEFS (shared directory among nodes) vs. loopback
- Loopback outperformed FEFS for smaller data size with the help of file system cache.



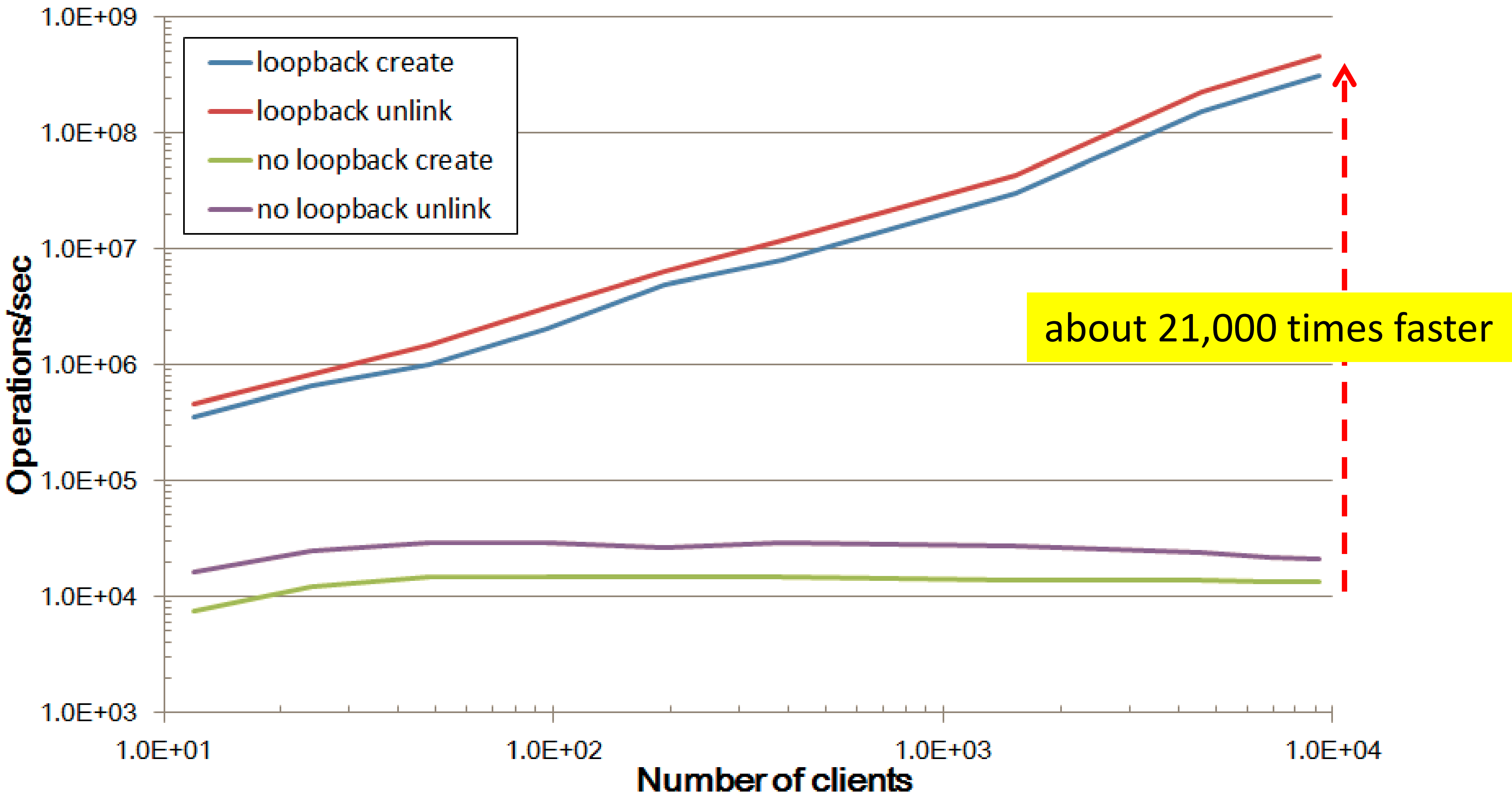


K. Yamamoto F. Shoji, A. Uno, S. Matsui, K. Sakai, F. Sueyasu, and S. Sumimoto,
"Analysis and Elimination of Client Evictions on a Large Scale Lustre Based File System," LUG'15

AICS K computer

Total metadata access performance

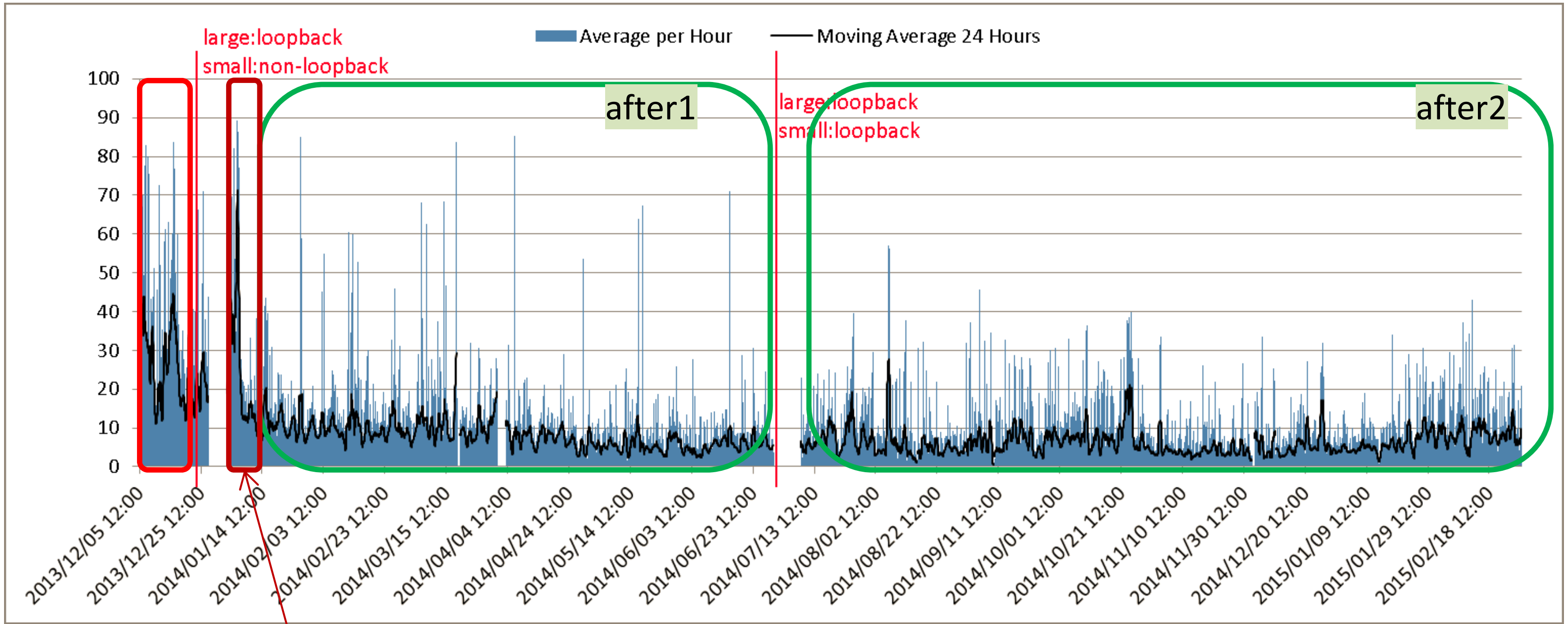
- Create 26K ops/node, unlink 37K ops/node by mdtest (100 files/node)
- Rank directory (loopback) scales with a large number of processes.



K. Yamamoto, F. Shoji, A. Uno, S. Matsui, K. Sakai, F. Sueyasu, and S. Sumimoto, "Analysis and Elimination of Client Evictions on a Large Scale Lustre Based File System," LUG'15

Impact for MDS load average

- MDS CPU load



< MDS CPU load over time before and after loopback introduction through two steps(after1 and after2) >

* Some large class job did not use loopback.

- MDS load average per hour: reduced to 1/3.5
- Peak occurrence times per day (over 50%, 70%): reduced to 1/30

K. Yamamoto, F. Shoji, A. Uno, S. Matsui, K. Sakai, F. Sueyasu, and S. Sumimoto, "Analysis and Elimination of Client Evictions on a Large Scale Lustre Based File System," LUG'15

Eviction problem

- Eviction
  - File server evicts a client when a client does not work properly, e.g. no response to requests from servers.
- Impact of eviction
  - I/O accesses of running jobs on the node will fail.
    - In many cases, jobs affected by evictions are aborted.

↓

  - Frequent evictions led to a serious decrease in node utilization.

Mitigation of evictions

- Elimination of client evictions that we have done
  – Step 1: Eliminating evictions during system board maintenance by system operation level
  – Step 2: Eliminating evictions during system board maintenance by improvement of file system level

⇩

- The two fixes reduced eviction occurrence ratio by a 1/72.

Eviction occurrence ratio/node

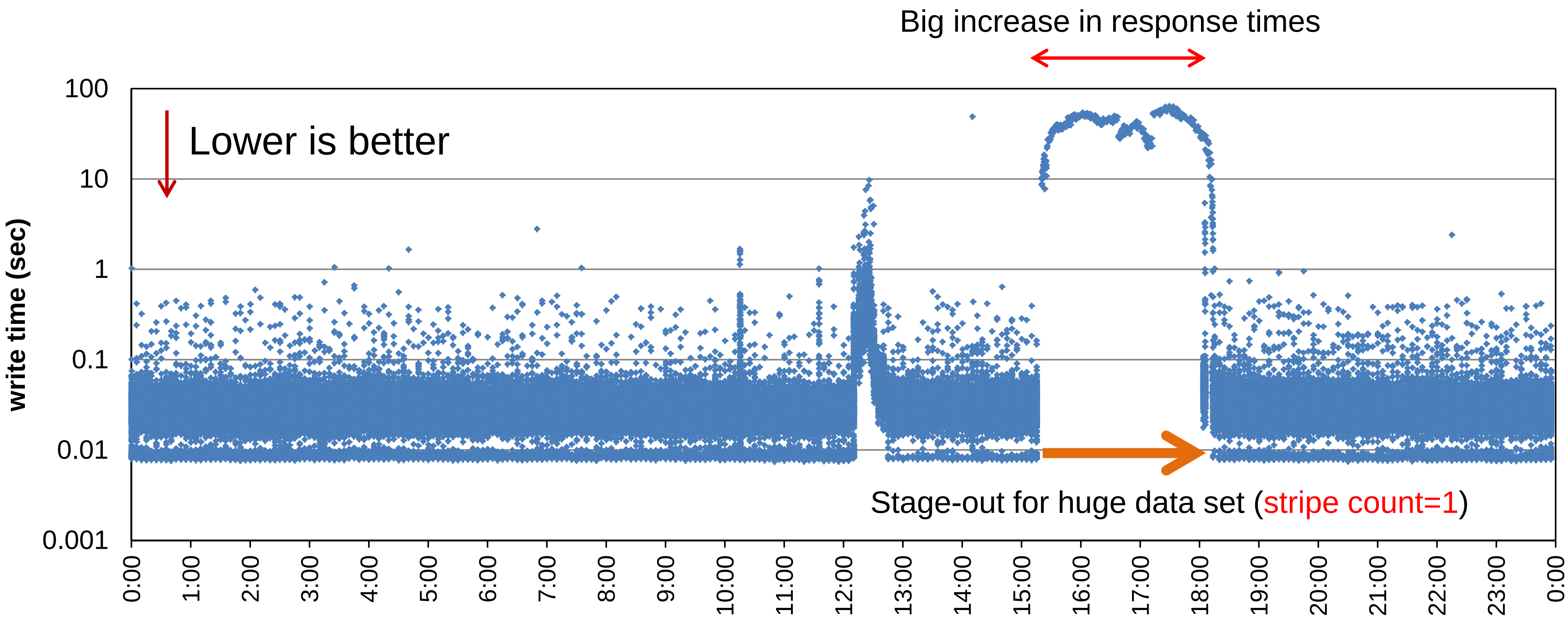
Before	After	Improvements
0.47	0.0065	1/72

K. Yamamoto, F. Shoji, A. Uno, S. Matsui, K. Sakai, F. Sueyasu, and S. Sumimoto,
"Analysis and Elimination of Client Evictions on a Large Scale Lustre Based File System," LUG'15

AICS K computer

Interference due to heavy data staging

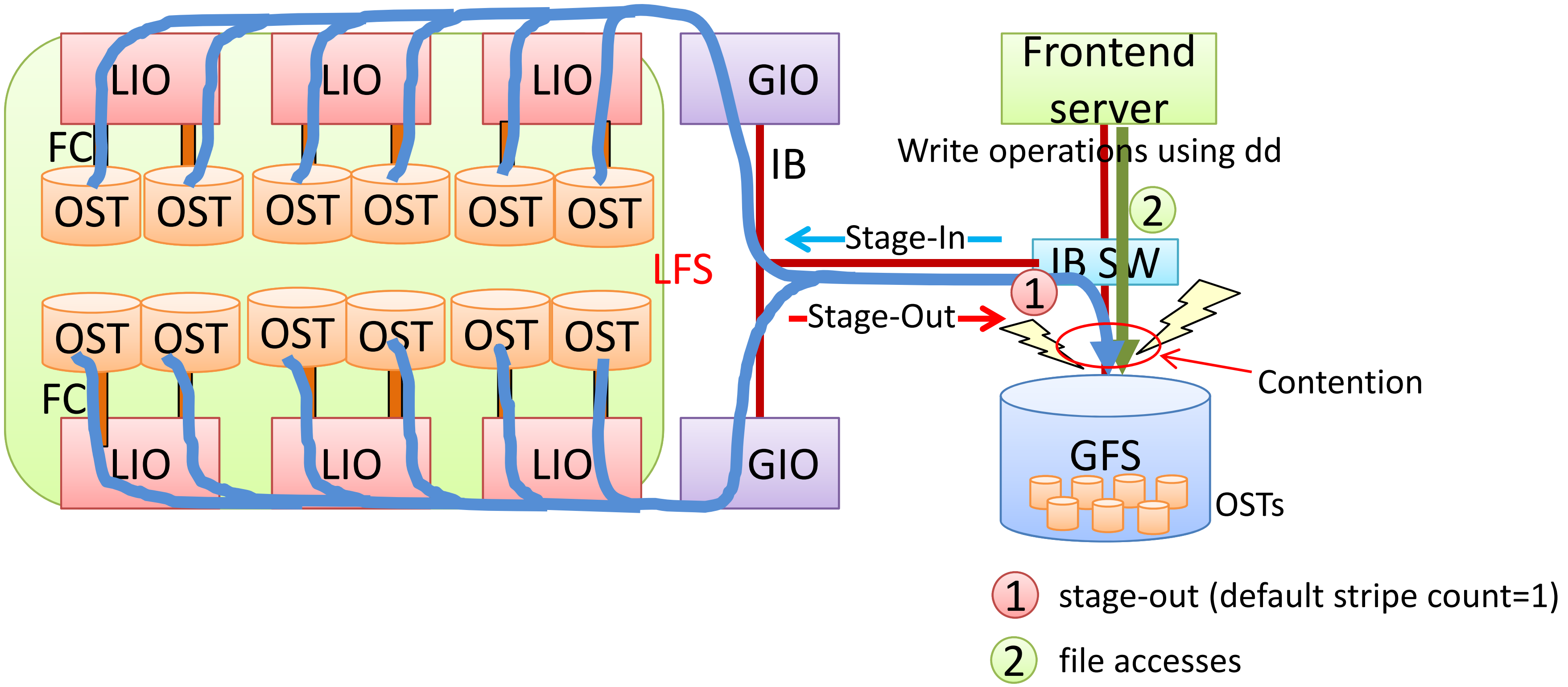
- Increase in response time in GFS accesses due to heavy data staging



Response times of 1MiB write from a frontend server to every OST in every 5 minutes on 2015/2/5

I/O interference in data staging

- Big performance degradation in file accesses from frontend servers due to huge scale of stage-out operations



- How do we mitigate performance degradation ?
  1. I/O workload-aware stripe count
     -> Balanced I/O workload among OSTs
  2. Load-balancing among clients (QoS of FEFS)

I/O workload-aware stripe count

- Tuning scheme of stripe count (Cs) in stage-out

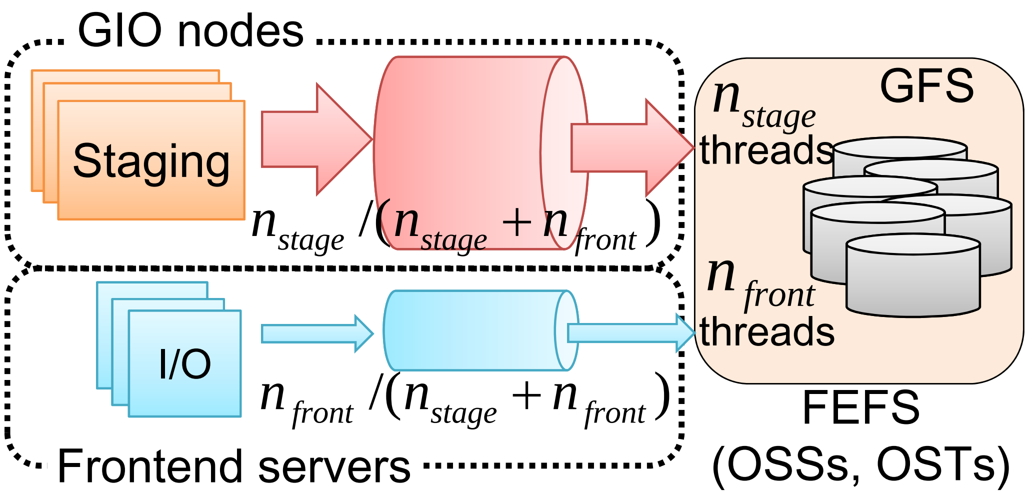
$$C_S = \left\lceil \frac{\alpha}{\beta} \times \frac{N_{OST}}{N_{IO} \times k_{stg}} \right\rceil \text{, where } \alpha = \left\lceil \frac{n_{stg}}{N_{OSS} \times l_{thr}} \right\rceil \text{ and } k_{stg} = \min(\frac{n_{stg}}{N_{IO}}, k_{stg}^{\max})$$

α	The number of files that each OSS service thread manages
β	Maximum acceptable variance in I/O workload among OSTs
N_{OSS}	The number of OSSs
N_{OST}	The number of OSTs
N_{IO}	The number of I/O (GIO) nodes
l_{thr}	Maximum number of service threads on each OSS
k_{stg}	The number of files in staging at each GIO
$k_{stg}^{\max}$	Maximum number of files that one GIO can manage

Y. Tsujita, T. Yoshizaki, K. Yamamoto, F. Sueyasu, R. Miyazaki, and A. Uno, “Alleviating I/O Interference Through Workload-Aware Striping and Load-Balancing on Parallel File Systems,” Proceedings of ISC’17

QoS in GFS accesses

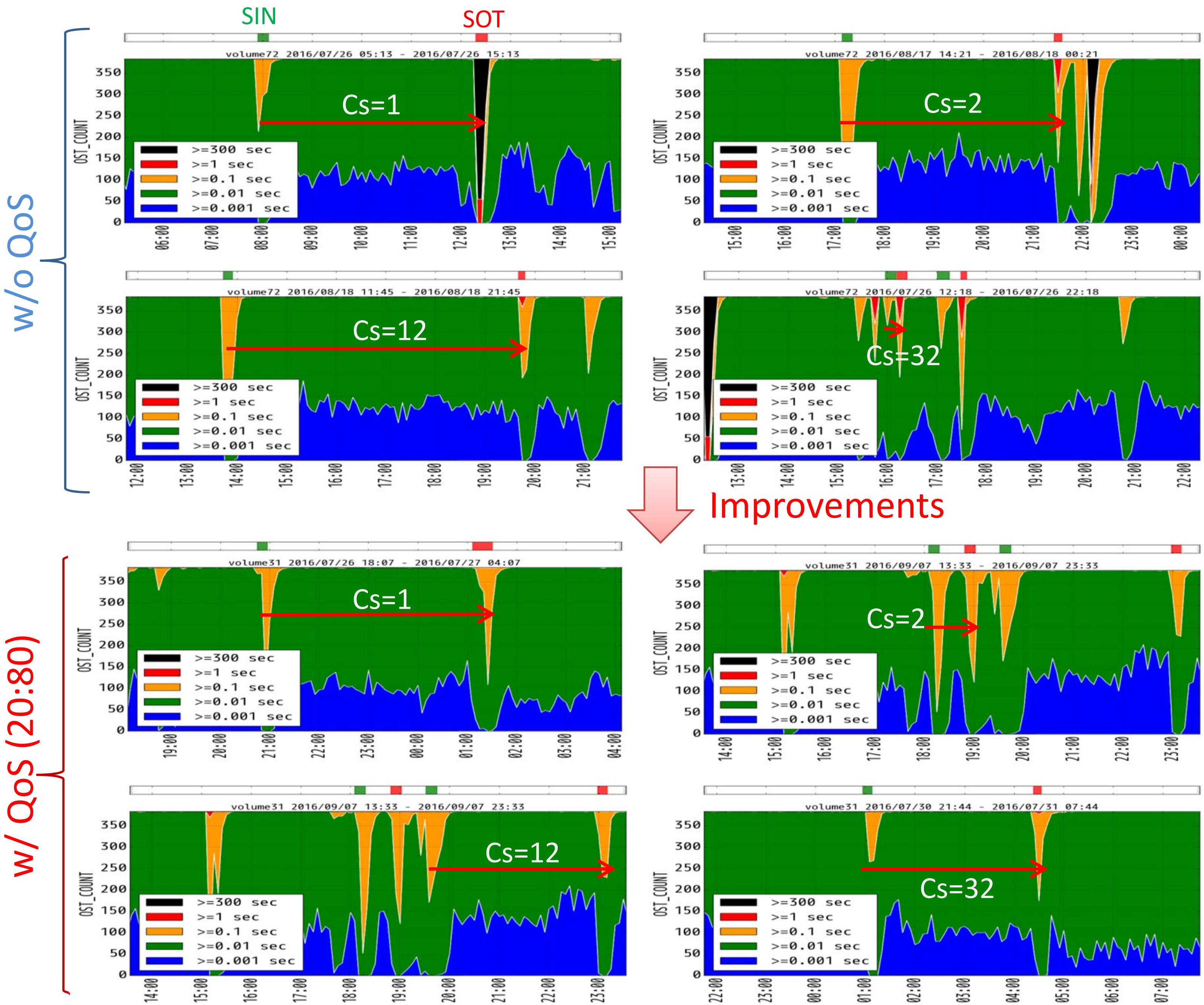
- QoS function of FEFS utilized for data staging at the K computer



Limiting the maximum number of service threads for each client
-> Guarantee I/O bandwidth for each client

Performance improvements in GFS accesses with QoS function

96 GIOs(12x24x2), 576 files (12GB/file)



Our model predicted that Cs=14 was the best.

Performance evaluation showed that Cs=12 was the best.

QoS function was turned out to be effective in I/O interference mitigation.

Contribution to Lustre development by Fujitsu

- Incorporated function derived from R&D works for the K computer by Fujitsu

Jira [1]	Function	Landing
LU-2467	Ability to disable pinging	Lustre 2.4
LU-2466	LNET networks hashing	Lustre 2.4
LU-2934	LNET router priorities	Lustre 2.5
LU-2950	LNET read routing list from file	Lustre 2.5
LU-2924	Reduce ldlm_poold execution time	Lustre 2.5
LU-3221	Endianness fixes (SPARC support)	Lustre 2.5
LU-2743	Errno translation tables (SPARC support)	Lustre 2.5
LU-4665	lfs setstripe to specify OSTs	Lustre 2.7

1. https://jira.hpdd.intel.com/projects/LU/issues/

Summary

- Our efforts about FEFS as shown below have led to high availability and high I/O performance.
  1. Loop-back file system
  2. Eviction treatment,
  3. Stripe count tuning and QoS function, and so forth
- Further efforts for high availability of file systems are in progress.

Acknowledgment

Special thanks to

- RIKEN AICS
  - F. Inoue, M. Iwamoto, F. Shoji, K. Sugeta, A. Uno, K. Yamamoto

- FUJITSU Limited
  - Y. Furutani, H. Hida, N. Ikeda, S. Matsui, R. Miyazaki, M. Okamoto, R. Sekizawa, F. Sueyasu, S. Sumimoto

for giving many useful information about their efforts described in this presentation.