High Availability Operation of Parallel File Systems at the K computer

Yuichi Tsujita
RIKEN AICS

Jun. 22, 2017@HPC-IODC'17

K computer

Outline

- Overview of the K computer and its file systems
- Activities for high availability and performance
  - Alleviation of MDS load using loop-back file systems
  - Elimination of client evicts
  - Optimization for alleviating interference by huge data accesses
- Summary

Overview of the K computer and its file systems

System configuration of the K computer

40 m x 40 m
Full System
Compute Rack × 864

10.6(11.3)PFLOPS
1.27(1.34)PiB

4000mm x 800mm
2 Cabinets
Compute Rack × 4
Disk Racks × 1

49.2(52.4)TFLOPS
6.00(6.38)TiB

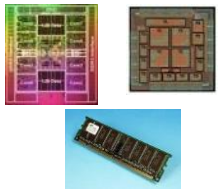
800mm x 800mm
Compute Rack
SB × 24
IOSB × 6

12.3(13.1)TFLOPS
1.50(1.59)TiB

500mm x 500mm
System Board(SB)
Node × 4

512GFLOPS
64GiB

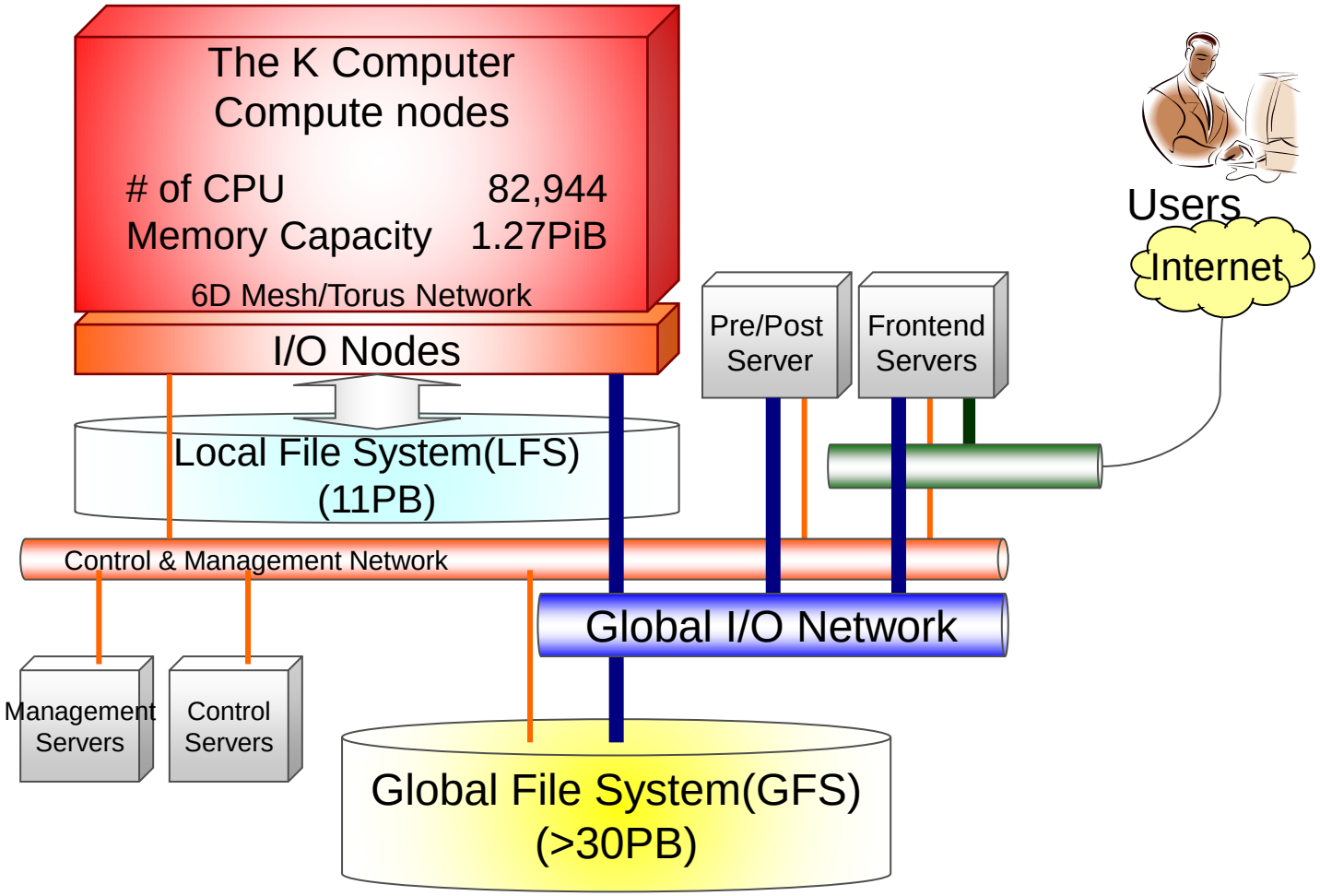
Node
CPU × 1
ICC × 1
memory



128GFLOPS
16GiB

( )included IO node performance and memory capacity

Overview of the K computer



FEFS is used for both LFS and GFS.
(FEFS: Fujitsu Exabyte File System based on Lustre technology)

File system 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 introduced in Apr. 2017.

[*2] Extended RAID6 by Fujitsu (RAID6 FR) available for the new three volumes

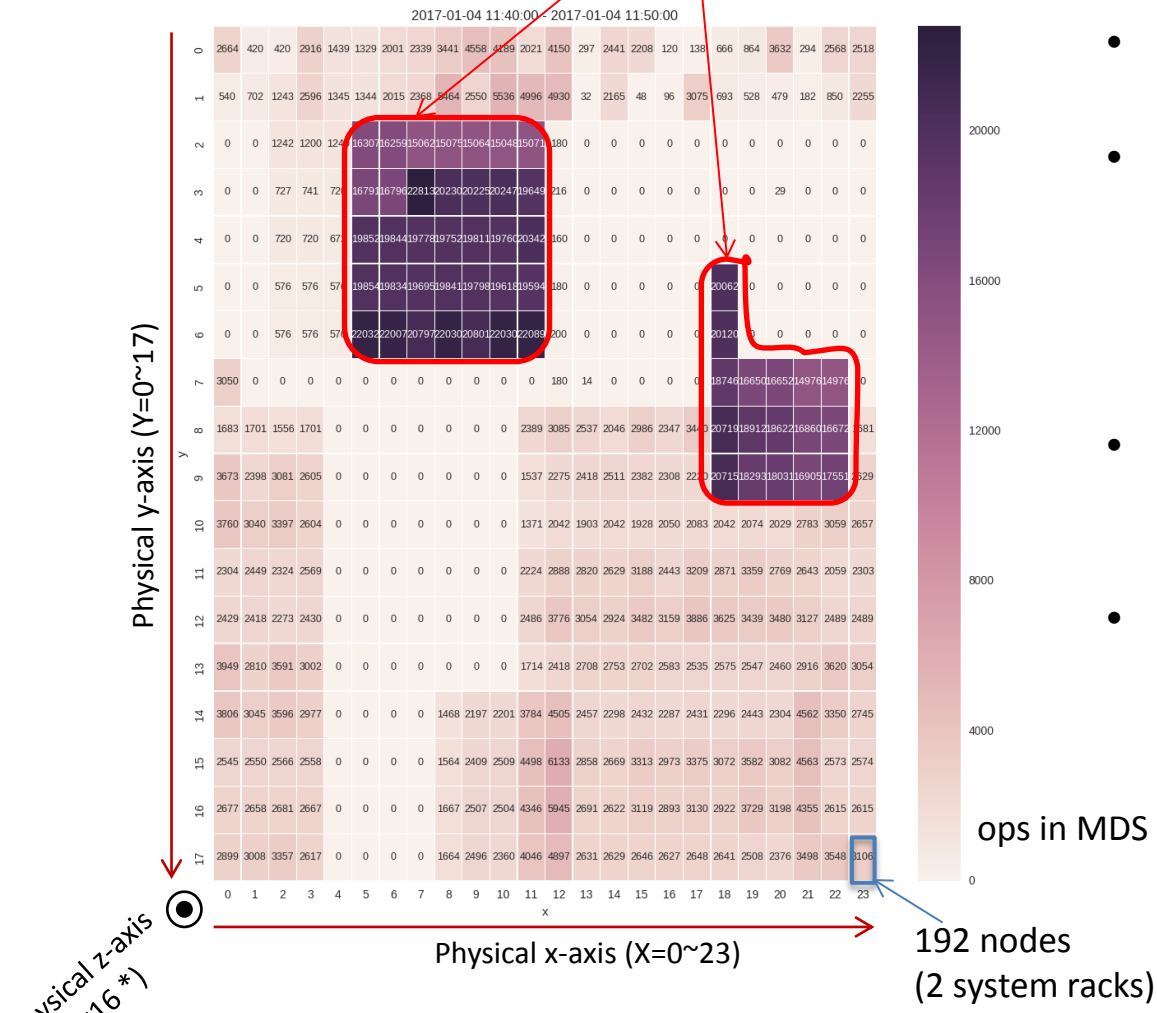
AICS K computer

Activities for high availability and performance

- 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)

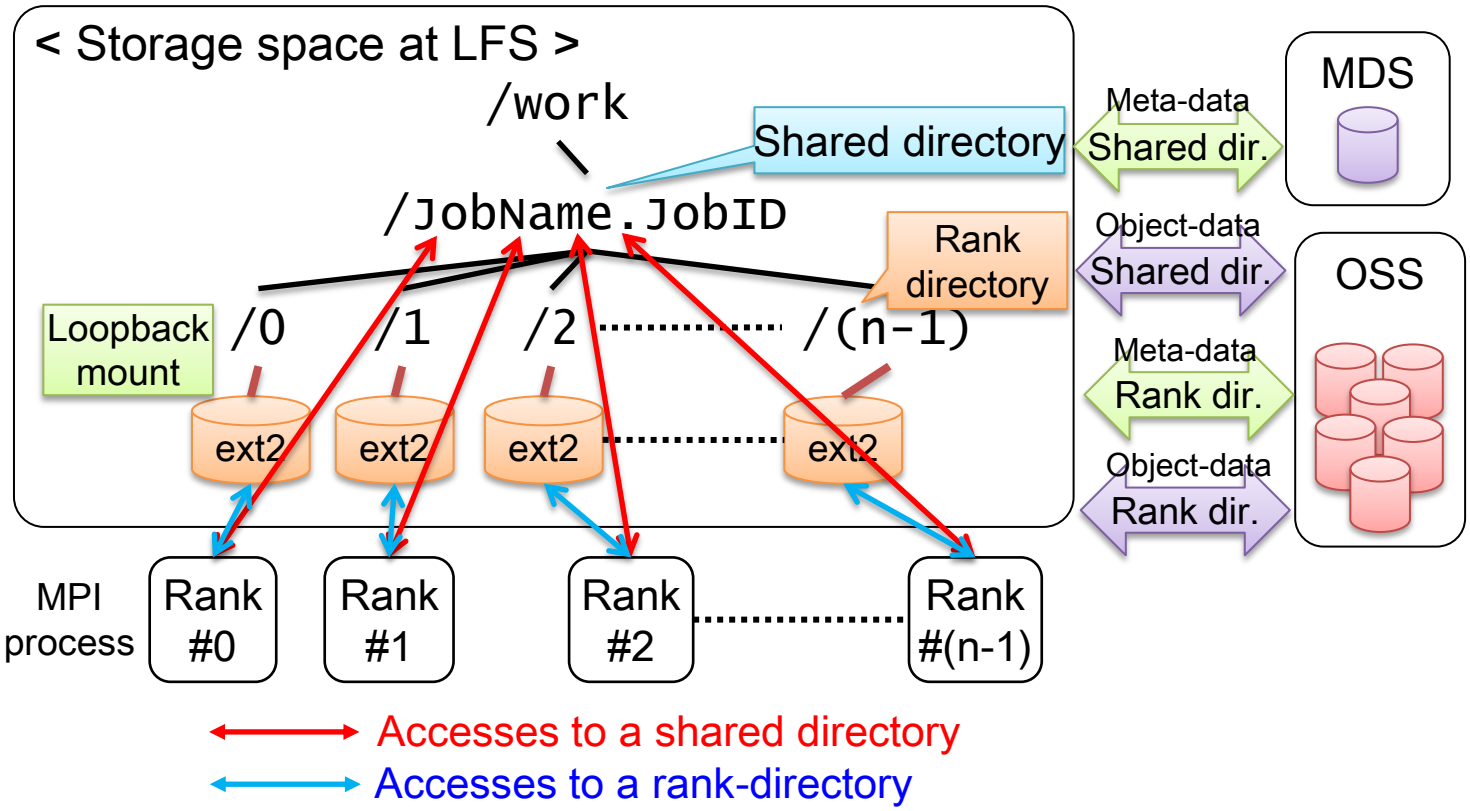
Compute nodes which generated huge number of requests to MDS of LFS



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

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

Rank-directory (loopback file system)

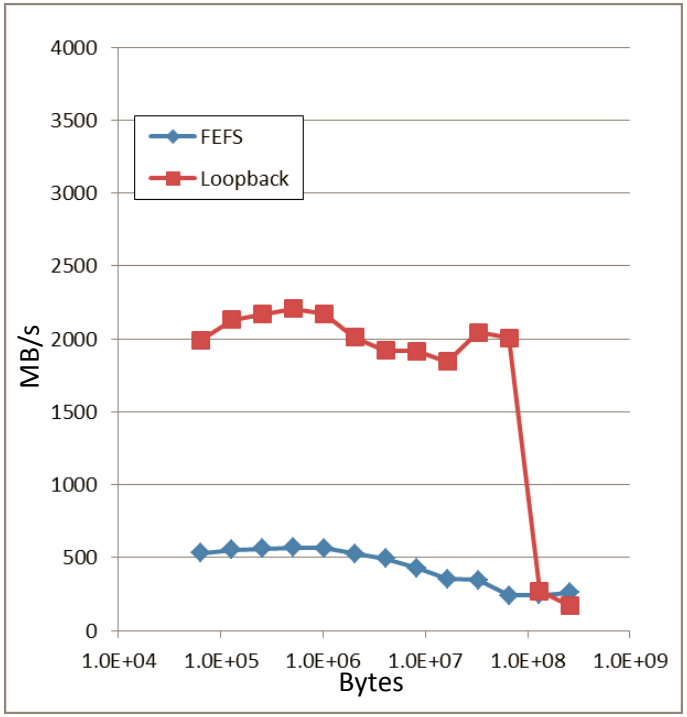


- Reducing MDS accesses leads to effective utilization of LFS.
- I/O accesses in rank-directories are free from slowdown of MDS performance.

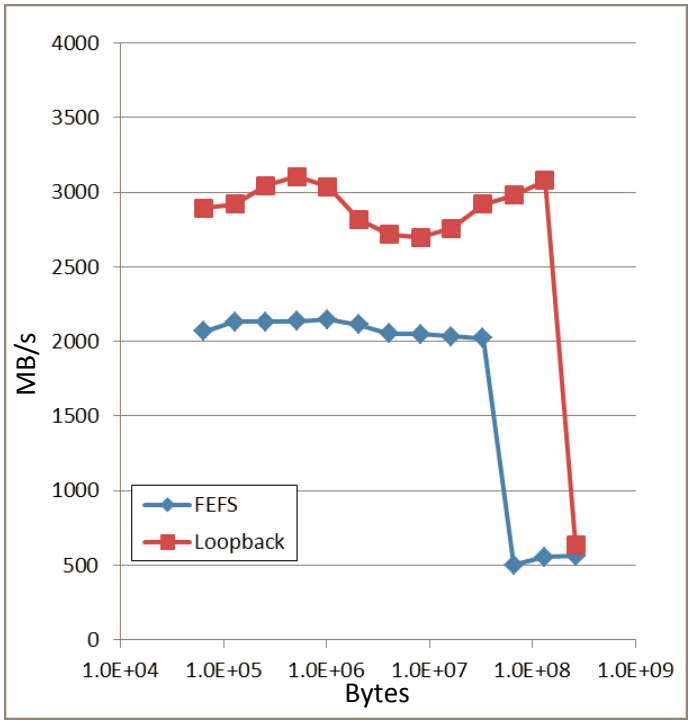
Single node I/O performance evaluation by using IOzone

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

write (64KB I/O block)



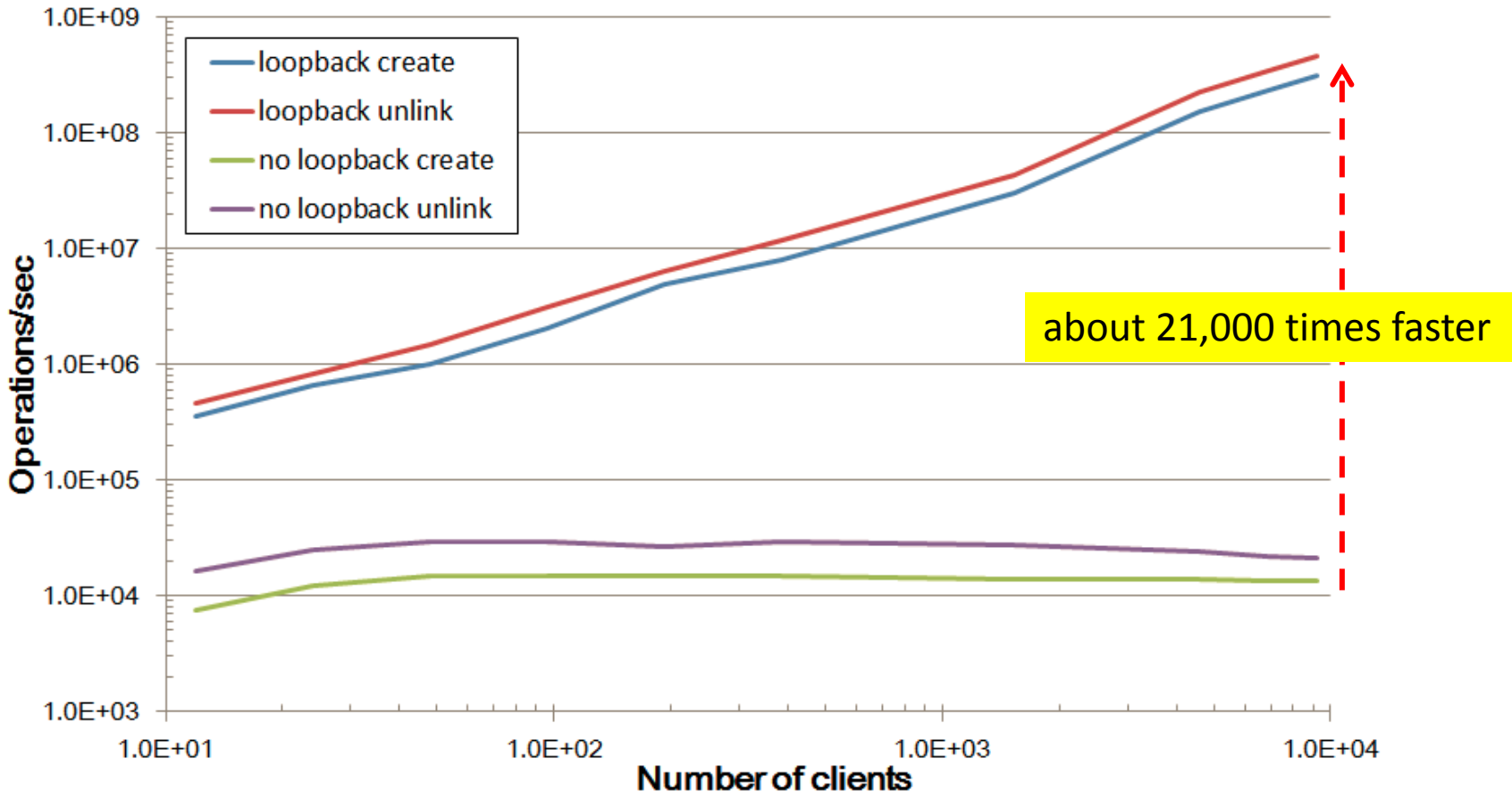
read (64KB I/O block)



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

Total metadata access performance

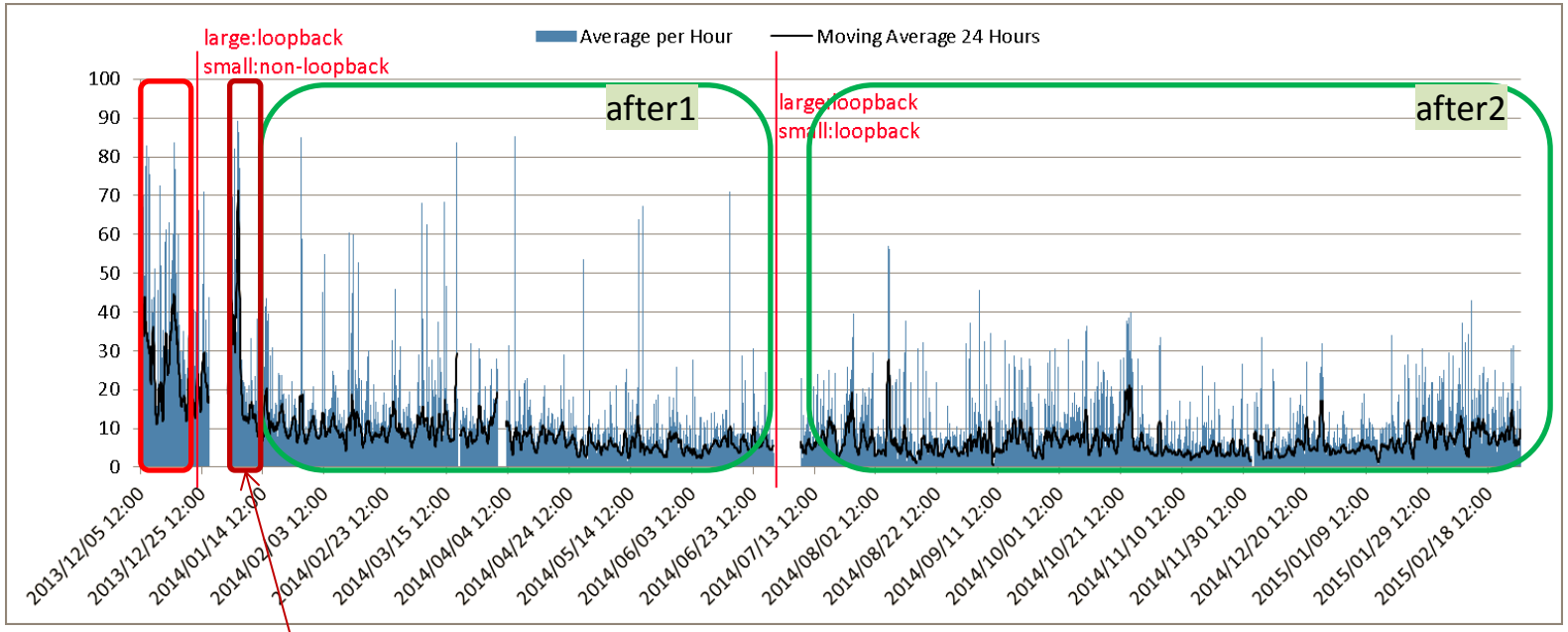
- Create 26K ops/node, unlink 37K ops/node by mdtest (100 files/node)
- Rank directory (loopback) scales with a large number of processes.



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 decrease in node utilization seriously.

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.

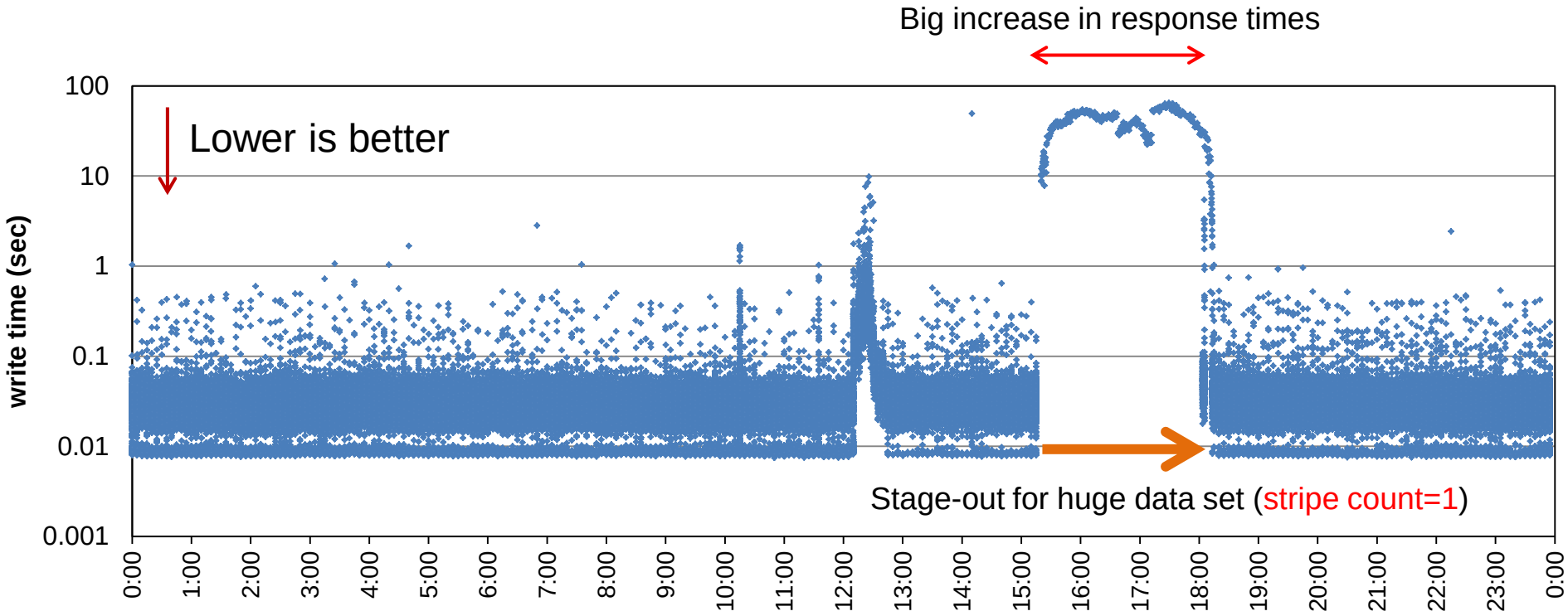
Eviction occurrence ratio/node

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

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

- We have already adopted stripe count selection simply based on file size in stage-out phase. => Success in mitigation interference so far.
- For more optimization, we have examined impact of stripe count and QoS function of FEFS.

I/O workload-aware stripe count

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

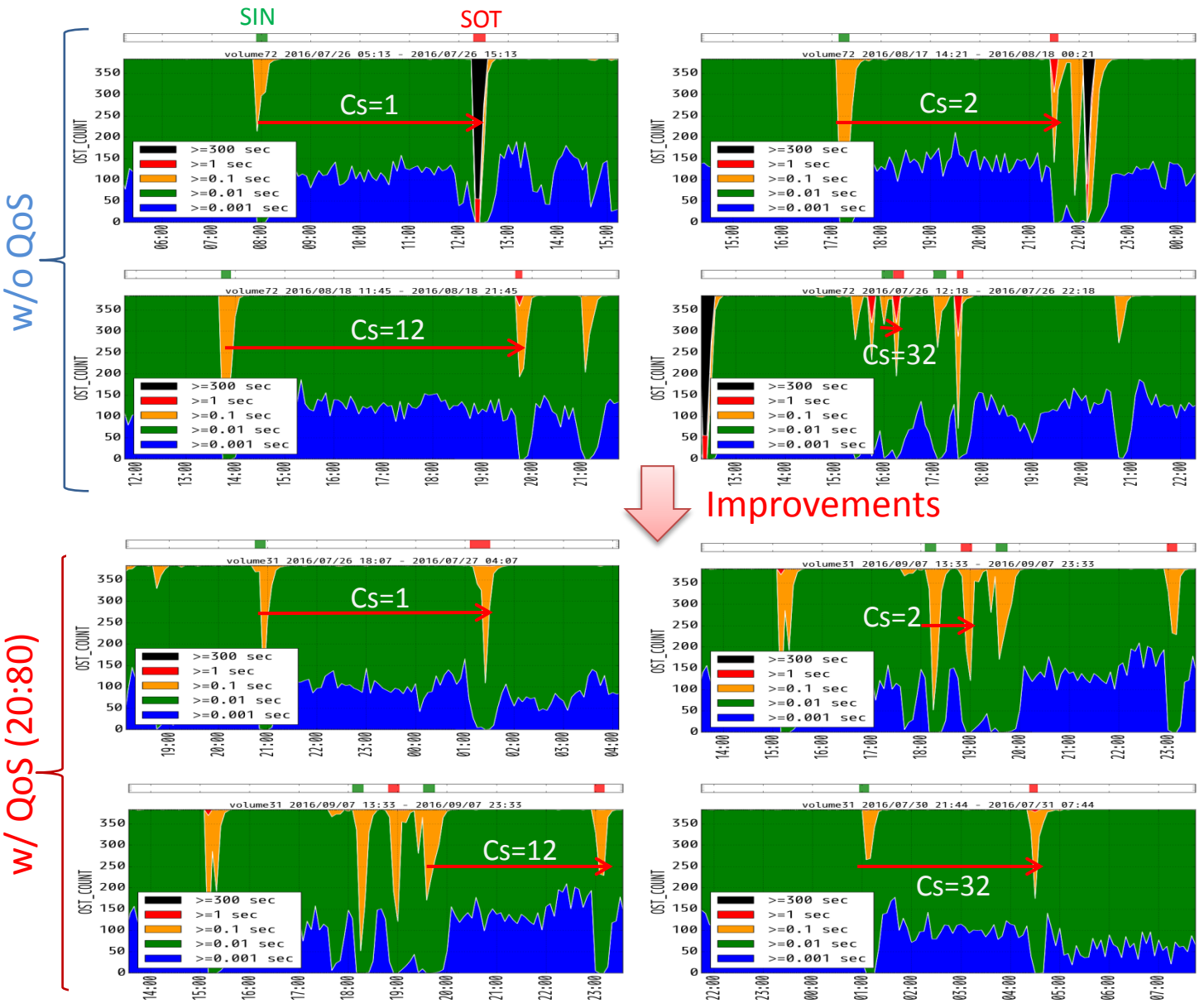
$$C_S = \left\lceil \frac{\alpha}{\beta} \times \frac{N_{OST}}{N_{IO} \times k_{stg}} \right\rceil \quad \text{, where} \quad \alpha = \left\lceil \frac{n_{stg}}{N_{OSS} \times l_{thr}} \right\rceil \quad \text{and} \quad k_{stg} = \min\left(\frac{n_{stg}}{N_{IO}}, k_{stg}^{\max}\right)$$

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

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.

Summary

- Our efforts done for 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 in 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
  - H. Hida, N. Ikeda, S. Matsui, R. Miyazaki, M. Okamoto, R. Sekizawa, F. Sueyasu, S. Sumimoto, T. Yoshizaki

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