FUJITSU

shaping tomorrow with you

# Lustre and Beyond Panel

Shinji Sumimoto
Fujitsu Limited
Apr.18 2013 @LUG2013, San Diego

# Here are a few questions that I would like you to take some time to think about:

FUJITSU

1) If we were to assume that Lustre is not the correct path forward to Exascale, what would the correct path look like?

- I believe Lustre is the correct path forward to Exascale.
- However, several design points of current Lustre need to be changed

2) You all have experience with implementing Lustre in the service of massive computational systems. Much of that work has been to accommodate defensive I/O. Big Data applications, however, are typically focused on data access and reduction. Will the file system of 2020 be able to serve both worlds?

- Yes,

3) Do you feel that Exascale research will be the main driver for future Lustre innovation or are there other forces that will shape the Lustre roadmap?

- Yes, our one of target is Exascale system

4) What is right and what is wrong with the OpenSFS approach to community development of Lustre? What would you change?

- I am not in position to answer this question.

# How do you satisfy two trade-off targets?

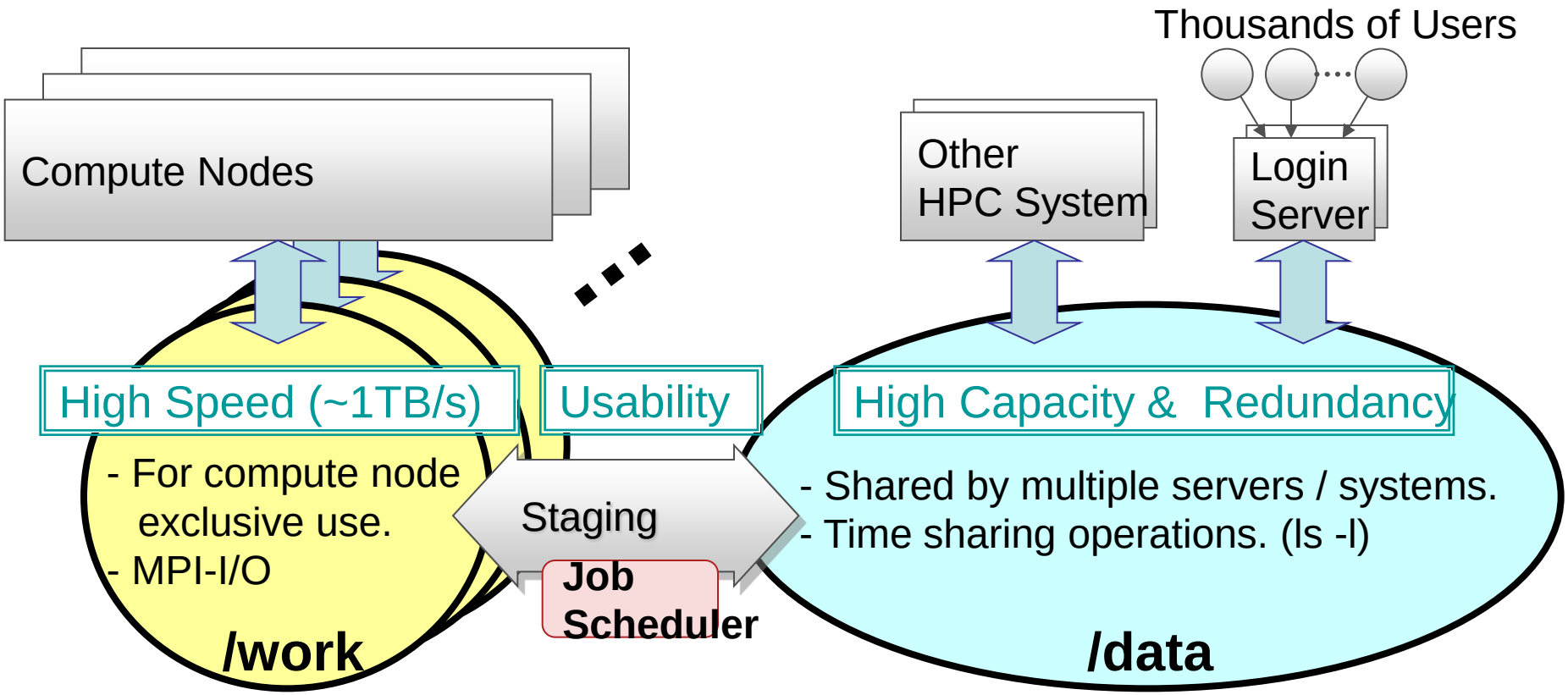
FUJITSU

- Two Trade-off Targets Examples
  - TB Class Large Files vs. KB Class Small Files
  - TB/s Class Bandwidth vs. GOPS Class IOPS
  - Performance vs. Data Integrity
- It is difficult to satisfy two trade-off targets on single file system.
- Therefore, we chose integrated layered file system.

# Integrated Layered Cluster File System

FUJITSU

- 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.



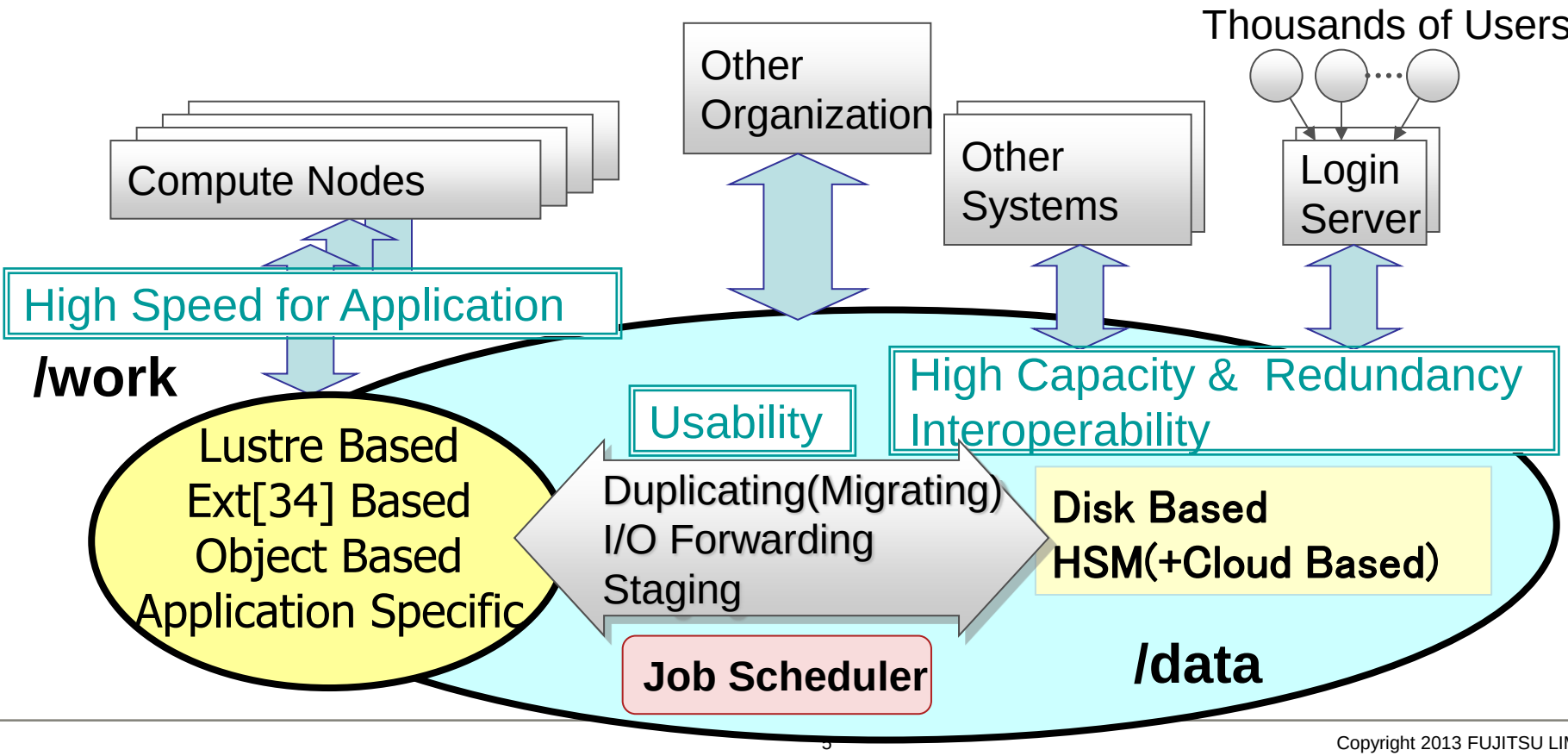
# Towards Luster for Exascale Systems

FUJITSU

- Co-design is required for exascale systems among hardware, system software and application.
  - This means exascale file system must fit to each exascale application precisely
- Therefore, customization of file system for each application is important issue to realize
  - Not only file system types but also file cache size, block size etc…

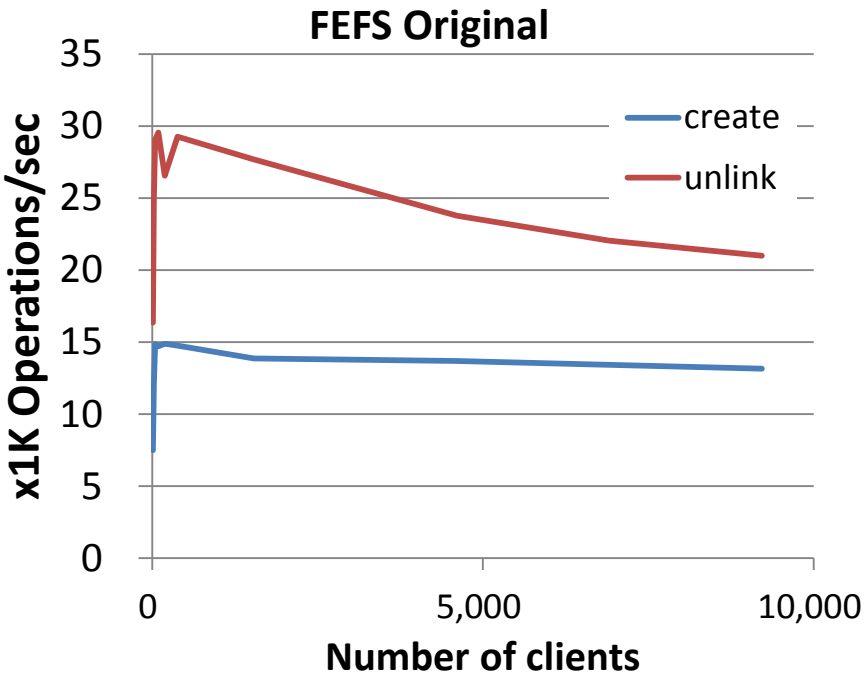
# The Next Integrated Layered File System Architecture

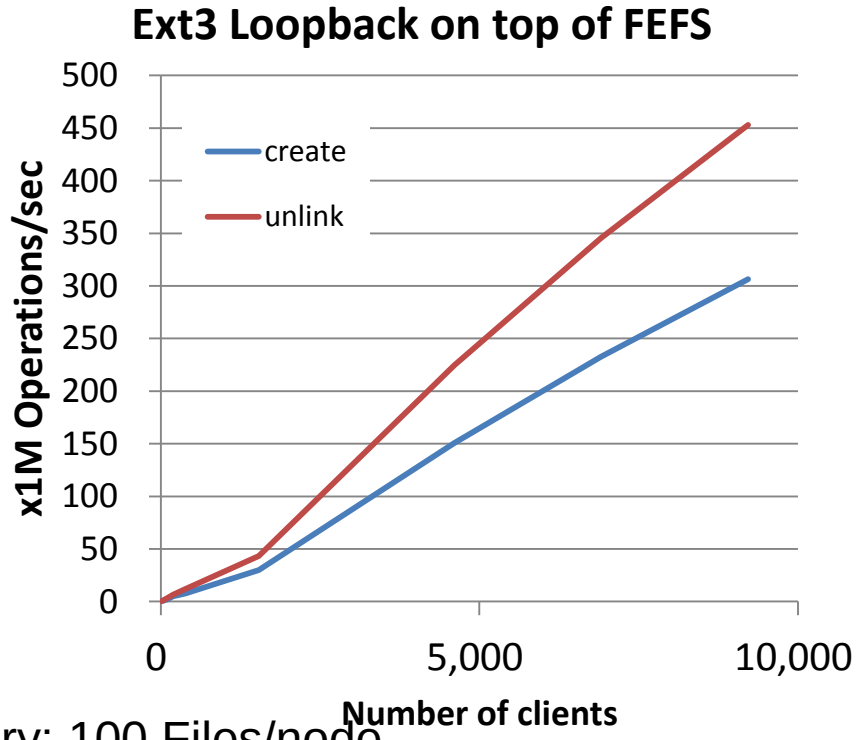
- Local File System (/work): Memory , SSD, Disk Based
  - Lustre Based, Ext[34], Object Based, Application Specific etc..
- Global File System (/data): Disk Based, HSM(Disk+Tape+Cloud)
  - Lustre Based



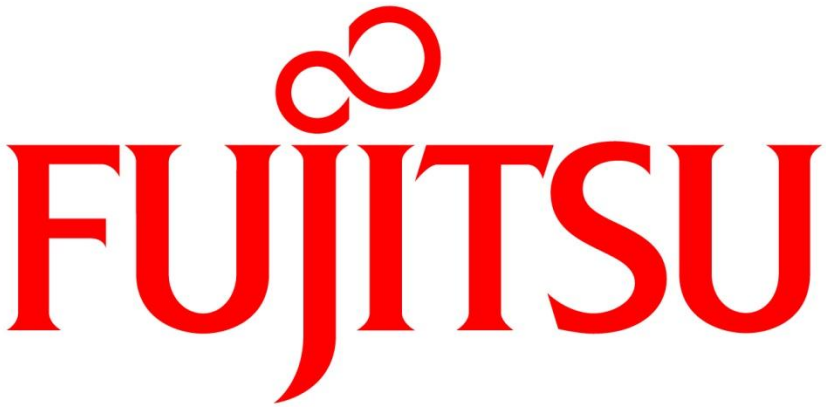
# Metadata Performance on Layered File System Using Loopback

- Single MDS not scale,
  but Layered FS using Loopback does
  - create ~26K ops/node scalability
  - unlink ~37K ops/node scalability




Using mdtest Different Directory: 100 Files/node




shaping tomorrow with you