Lustre User Group 2016

FUJITSU

shaping tomorrow with you

DL-SNAP: A Directory Level SNAPSHOT Facility on Lustre

Shinji Sumimoto

Fujitsu Ltd. a member of OpenSFS

OpenSFS

Outline

- Motivation, Status, Goal and Contribution Plan
- Background
- What is DL-SNAP?
- Use case and Utility Commands
- Implementation
- Evaluation

Motivation, Status, Goal and Contribution Plan

- Motivation:
  - Backup files on large scale file system are an issue to solve. However, existing system level backup requires large storage space and backup time
- Status:
  - We started to develop a snapshot function, and, we have developed a prototype of the function
- Goal of This Presentation:
  - To present our snapshot specification and the prototype implementation
  - To discuss its usability and gather user's requirements
- Contribution Plan:
  - Mid 2017 to Lustre community

Background of DL-SNAP

- It is difficult to make backup on large scale file system
  - PB class file system backup takes long time and requires its backup space
- To reduce backup storage usage and backup time:
  - Using snapshot to reduce duplicate data
  - Not all file system data, selection of backup area
- Two level of backup: System level and User level

Background: System Level vs. User Level Backup

- System level backup:
  - System guarantees to backup data and to restore the backup data
  - Therefore, double sized storage space or another backup device is required to guarantee data backup and restore
  - File Services must be stopped during backup
- User level backup:
  - User can select backup data
  - File Service does not need to be stopped

Background: Selected User Level Backup Scheme

- Customer Requirement:
  - Continuing file system service
  - Difficult to guarantees the backup data to restore in system operation
  - Providing effective backup service with limited storage space

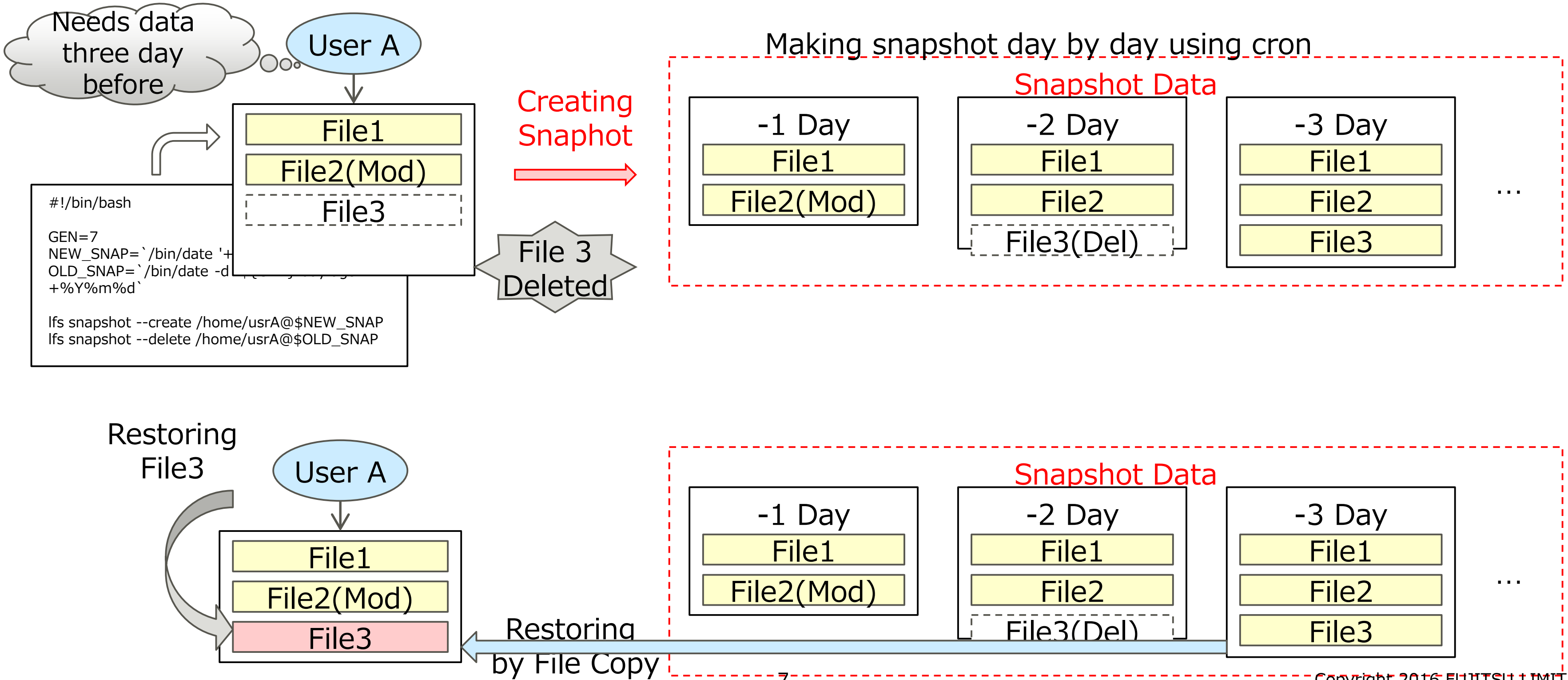
- Therefore, user level backup scheme is selected.
  - We started to develop DL-SNAP which is user and directory level snapshot

What is DL-SNAP?

- DL-SNAP is designed for user and directory level file backups
- Users can create a snapshot of a directory using lfs command with snapshot option and create option like a directory copy
- The user creates multiple snapshot of the directory and manage the snapshots including merge of the snapshots
- DL-SNAP also supports quota to limit storage usage of users

DL-SNAP Use-case 1

- Avoiding file deletion or corruption by file operation



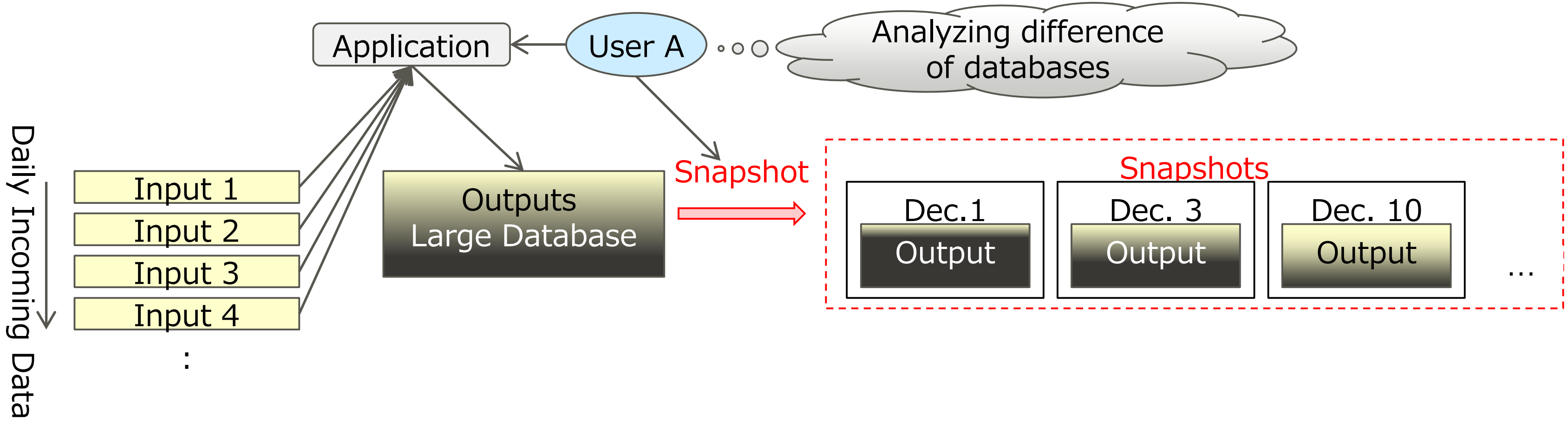
```
#!/bin/bash

GEN=7
NEW_SNAP=`/bin/date '+
OLD_SNAP=`/bin/date -d
+%Y%m%d`

lfs snapshot --create /home/usrA@$NEW_SNAP
lfs snapshot --delete /home/usrA@$OLD_SNAP
```

DL-SNAP Use-case 2

- Maintaining large database with partially different data
  - Updating database by an application using DL-SNAP



Quota Support and Utility Commands

- Quota function is also provided to manage storage usage of users
  - a little bit complicate when the owner of the snapshot is different among the original and some snapshot generations
- Utility Commands: lfs snapshot, lctl snapshot
  - Enabling Snapshot: lctl snapshot on <fsname>
  - Getting Status of Snapshot: lctl snapshot status <fsname>
  - Creating a snapshot: lfs snapshot --create [-s <snapshot>] [-d <directory>]
  - Listing snapshot: lfs snapshot --list [-R] [-d <directory>]
  - Deleting snapshot: lfs snapshot --delete [-f] -s <snapshot> [-d <directory>]

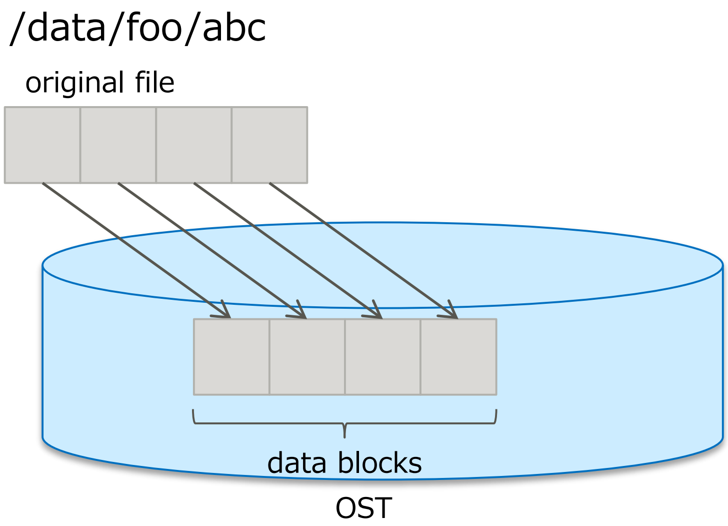
DL-SNAP Implementation

- The implementation of DL-SNAP is copy on write base
  - Implemented on top of current Lustre ldiskfs and limited in OST level modification
  - Without modification of ext4 disk format
  - Adding special function to create snapshot to MDS.
- OST level modification (more detail on next page):
  - Add Function which creates extra-references on OSTs.
  - Add Copy-on-Write capability to the backend-fs.
- Two Methods to Manage Copy-on-Write Region Blocks
  - Block Bitmap Method
  - Extent Region Method (Our Method)

Basic Mechanism of DL-SNAP by Extent Region (1)

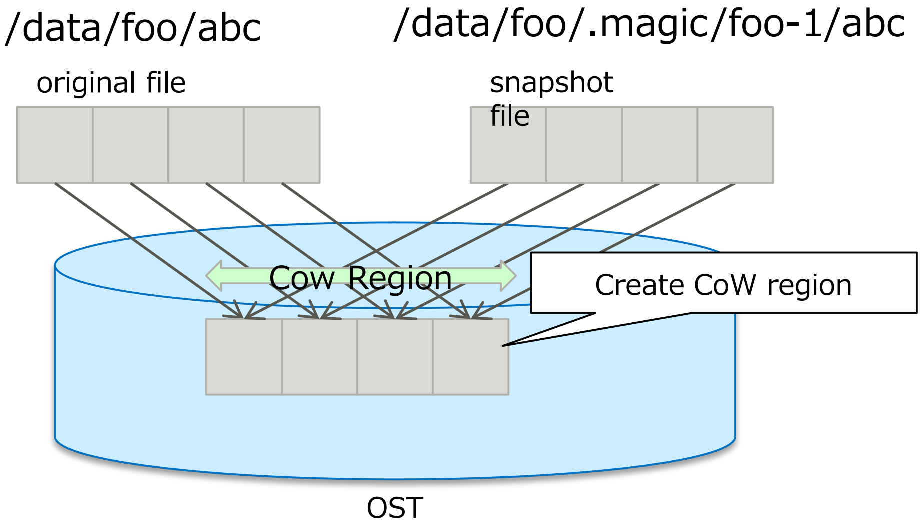
Initial state:

- The original file points to the data blocks on OSTs



Taking snapshot:

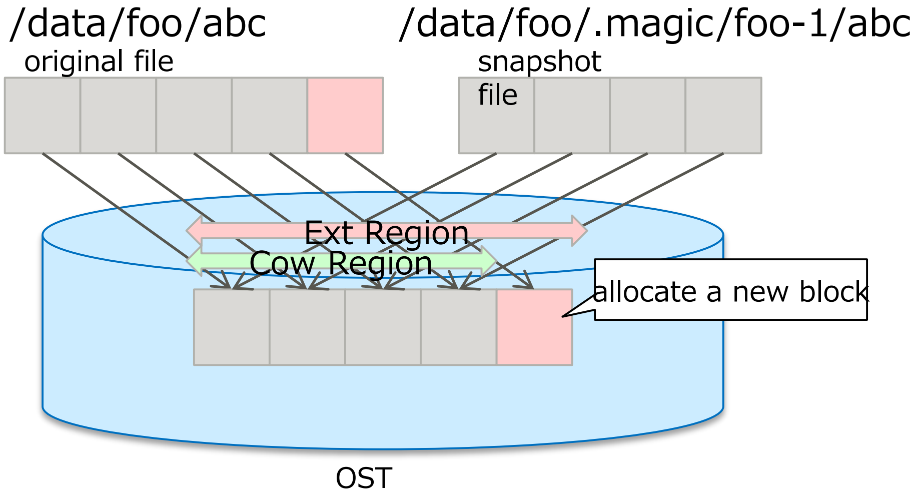
- Adds another reference and it points the blocks the original file points to

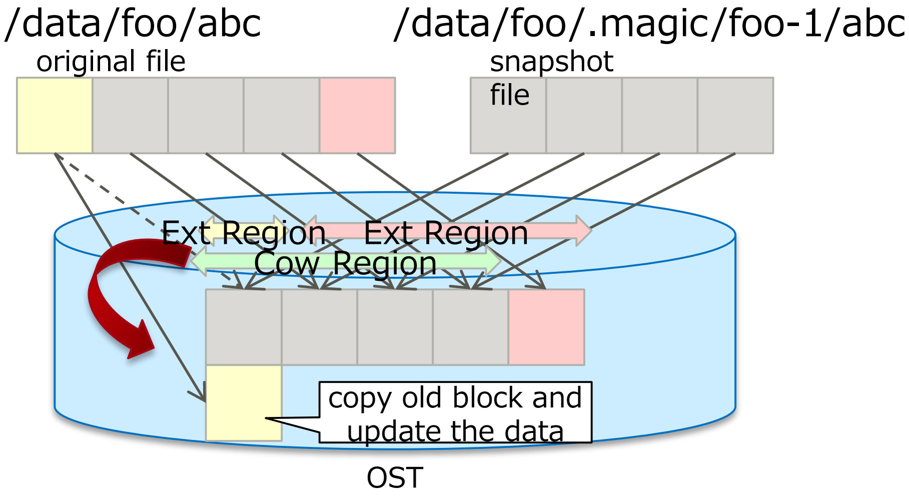


Basic Mechanism of DL-SNAP by Extent Region(2)

- Append-writing the original file:
  - Allocates a new data block on the OST and writes the data to the data block. Also, creating the original file modification extent of the data block

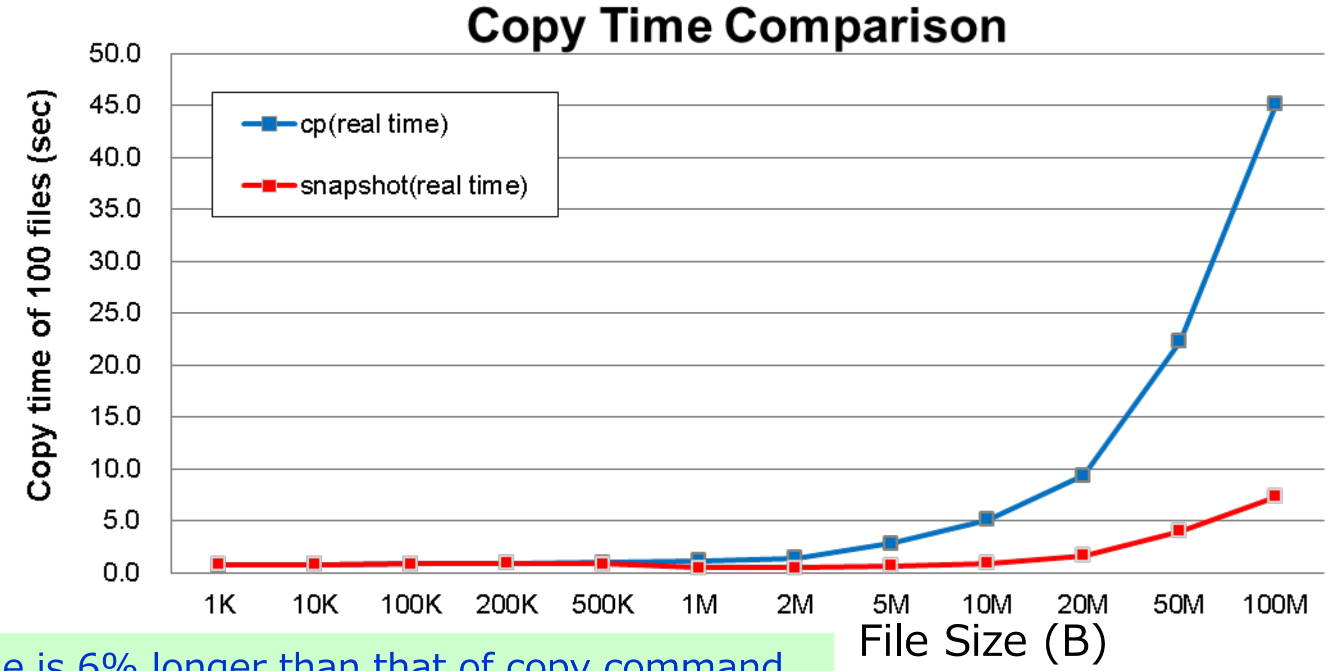
- Over-writing the original file:
  - Allocates a new data block on the OST and copy the original data block. Then, the file point the data block





Evaluation of DL-SNAP

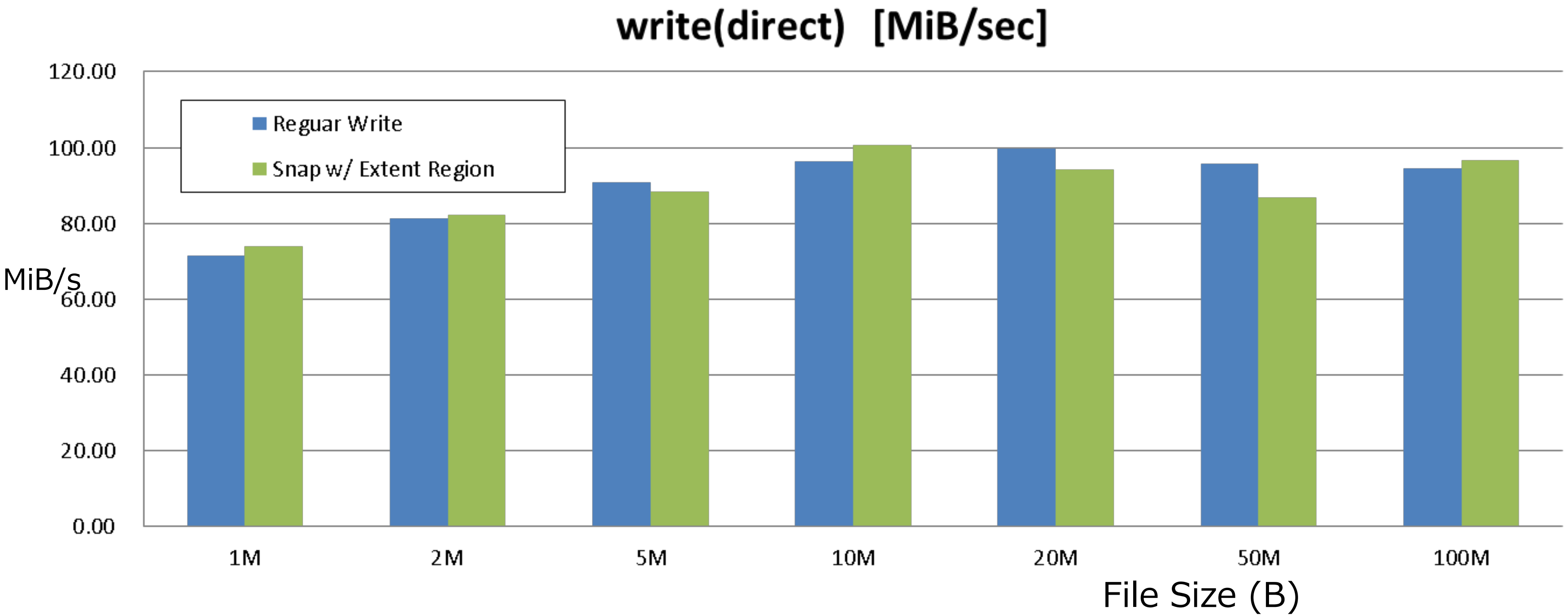
- DL-SNAP is faster than normal copy



1K byte file is 6% longer than that of copy command, but the time on 100 MB file is over 5 times faster

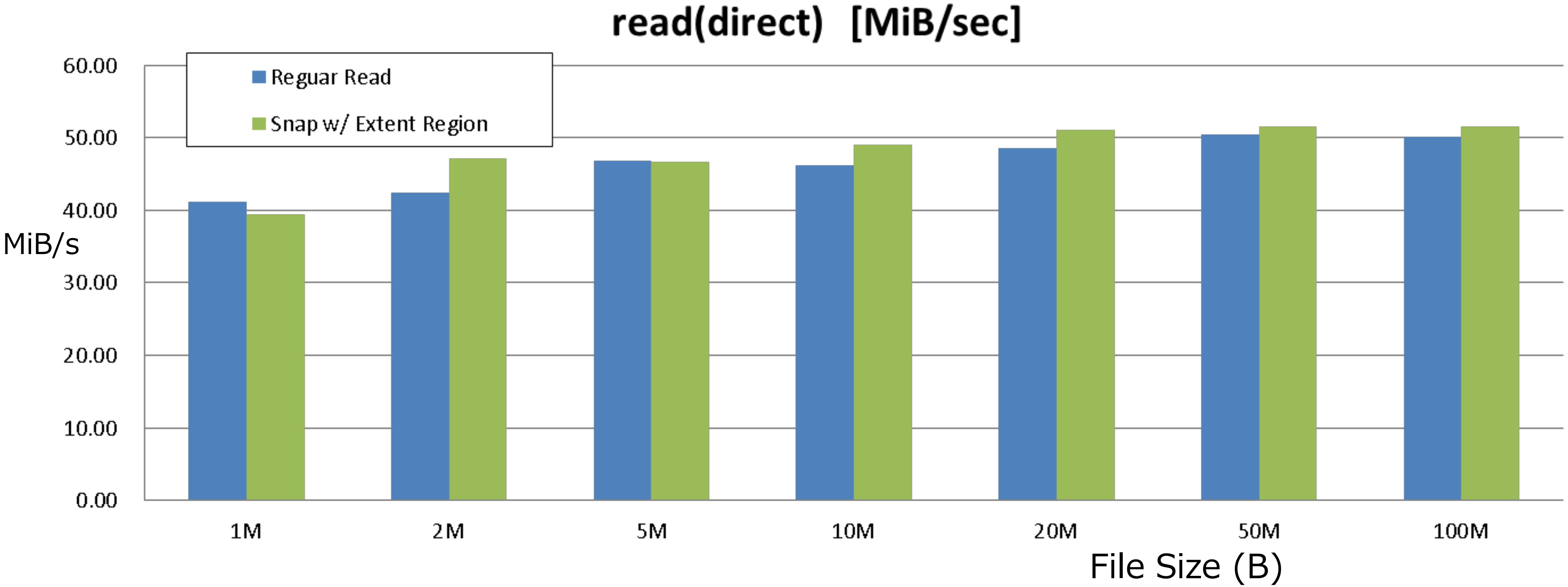
Write Performance by IOR

- Comparable performance to regular write



Read Performance by IOR

- Comparable performance to regular read

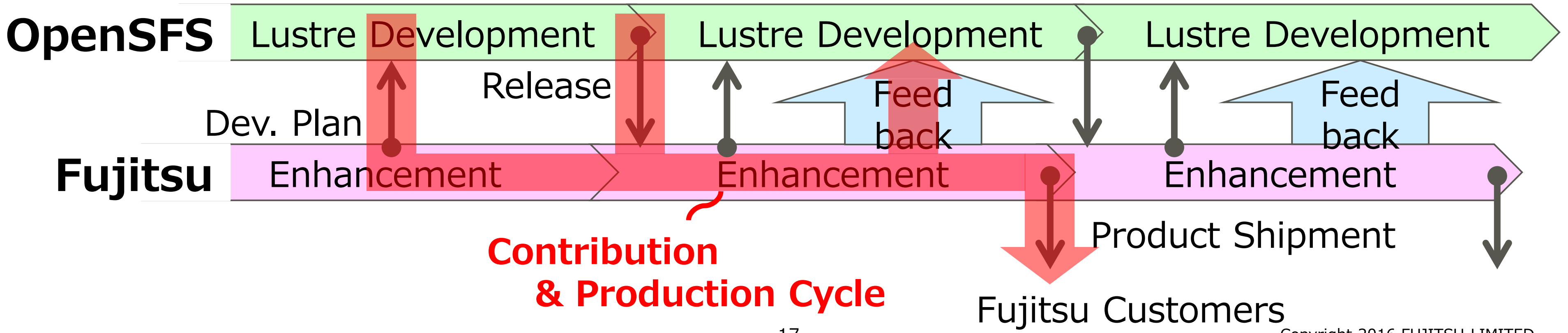


Contribution Plan and Vendor neutrality

- Contribution Plan:
  - Mid 2017 to Lustre community, several months after shipping as a product
- Vendor Neutrality:
  - The implementation of DL-SNAP is absolutely vendor-neutral because no special hardware is required and based on standard Lustre code based implementation

Fujitsu' Lustre Contribution Policy (Presented as LAD 14)

- Fujitsu will contribute open its development plan and feed back it's enhancement to Lustre community
- Fujitsu's basic contribution policy:
  - Opening development plan and Contributing Production Level Code
  - Feeding back its enhancement to Lustre community no later than after a certain period when our product is shipped.



Summary

- We are now developing DL-SNAP and evaluated its performance. The performance results show that the creating snapshot time is much better than that using copy command in longer files
  - Creating snapshot time on 1K byte file is 6% longer than that of copy command,
    but the time on 100 MB file is over 5 times faster than that of copy
- Our contribution of DL-SNAP will be planned in mid 2017

FUJITSU

shaping tomorrow with you