Toward Understanding Life-Long Performance of a Sonexion File System

CUG 2015

Mark Swan, Doug Petesch, Cray Inc.
dpetesch@cray.com

Safe Harbor Statement

This presentation may contain forward-looking statements that are based on our current expectations. Forward looking statements may include statements about our financial guidance and expected operating results, our opportunities and future potential, our product development and new product introduction plans, our ability to expand and penetrate our addressable markets and other statements that are not historical facts. These statements are only predictions and actual results may materially vary from those projected. Please refer to Cray's documents filed with the SEC from time to time concerning factors that could affect the Company and these forward-looking statements.

Agenda

- Introduction of concepts
- Experiment to reduce I/O variation
- Test method and results
- Summary

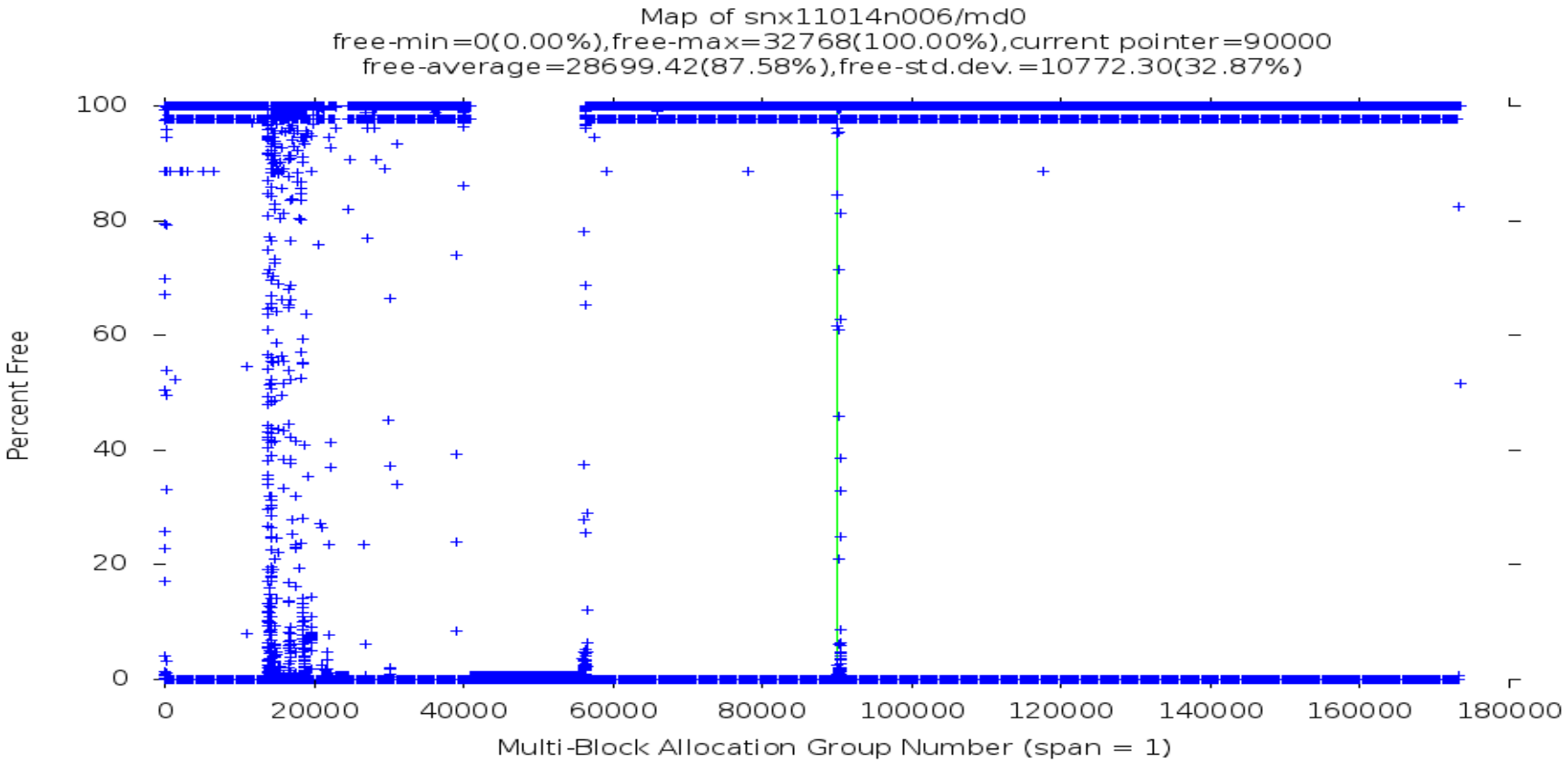
Sources of I/O Performance Variation Over Time

- **Software updates**
  - Lustre client and servers, I/O libraries
- **Transient hardware issues**
  - IB cables, degraded connections
  - Failing disks, bad sectors, RAID rebuilds
- **Disk position and fragmentation**
  - Can it be controlled?
  - Focus of this presentation

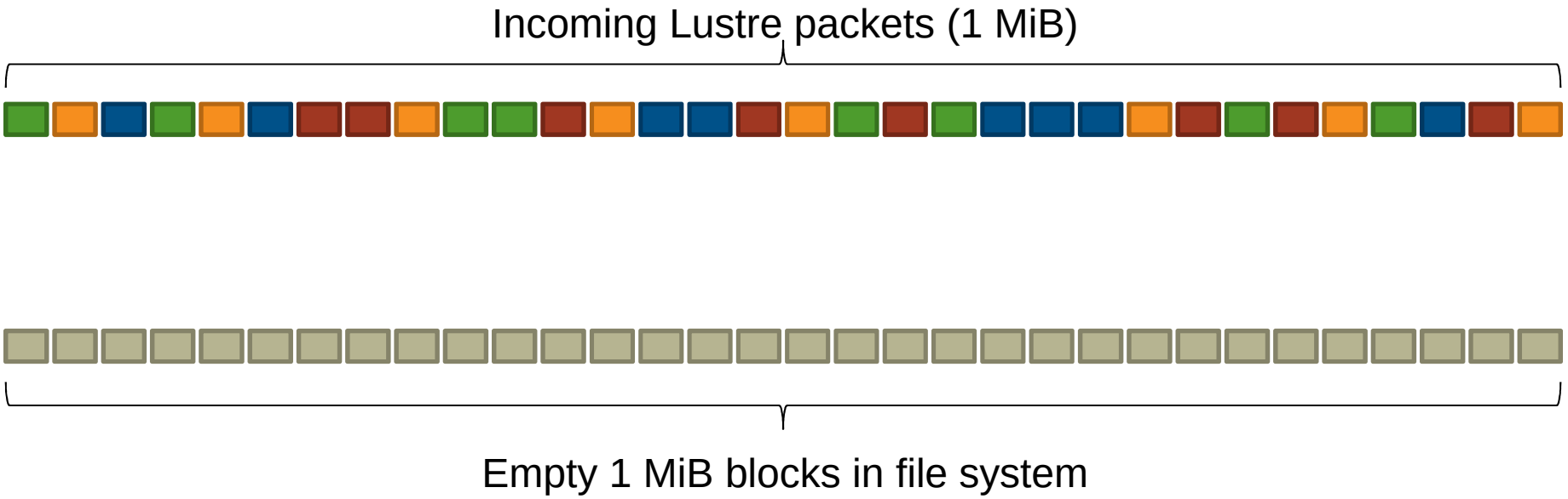
Lustre File System Terms

- **Cray Sonexion 1600 file system**
  - 1 SSU contains 2 servers with 4 OSTs each
- **Files in /proc/fs/ldiskfs/<device>**
  - mb_groups - bitmap of free space
  - mb_last_group - current allocation pointer
  - prealloc_table - disk space allocation size
- **Each multi-block allocation group is 128 MiB**
  - 32768 (4096 byte) disk blocks
  - Over 160k groups in a 23 TB OST (3 TB disks)

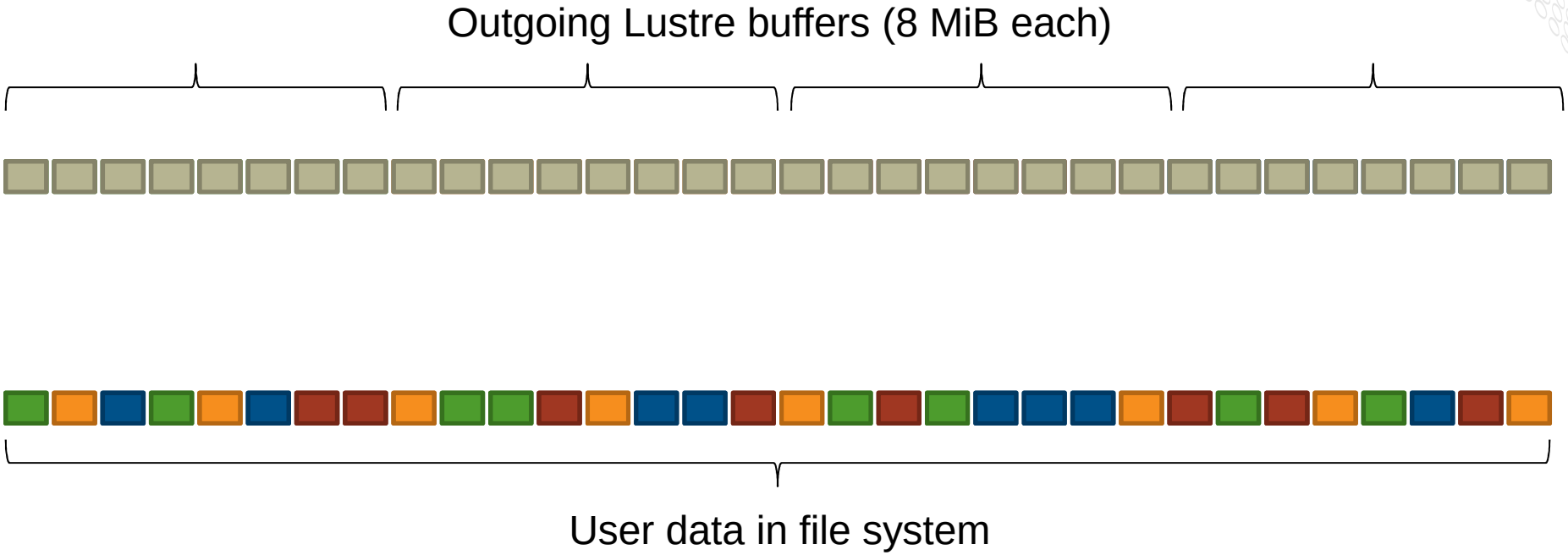
Allocation Bitmap of an OST



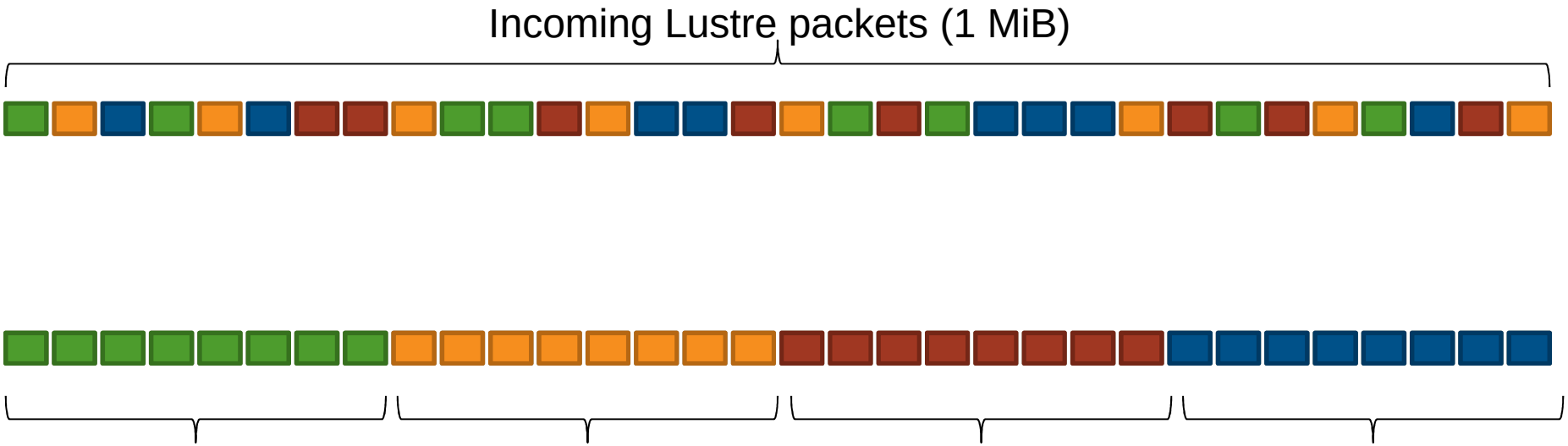
1M OST pre-allocation
4 files, 8M per file, write



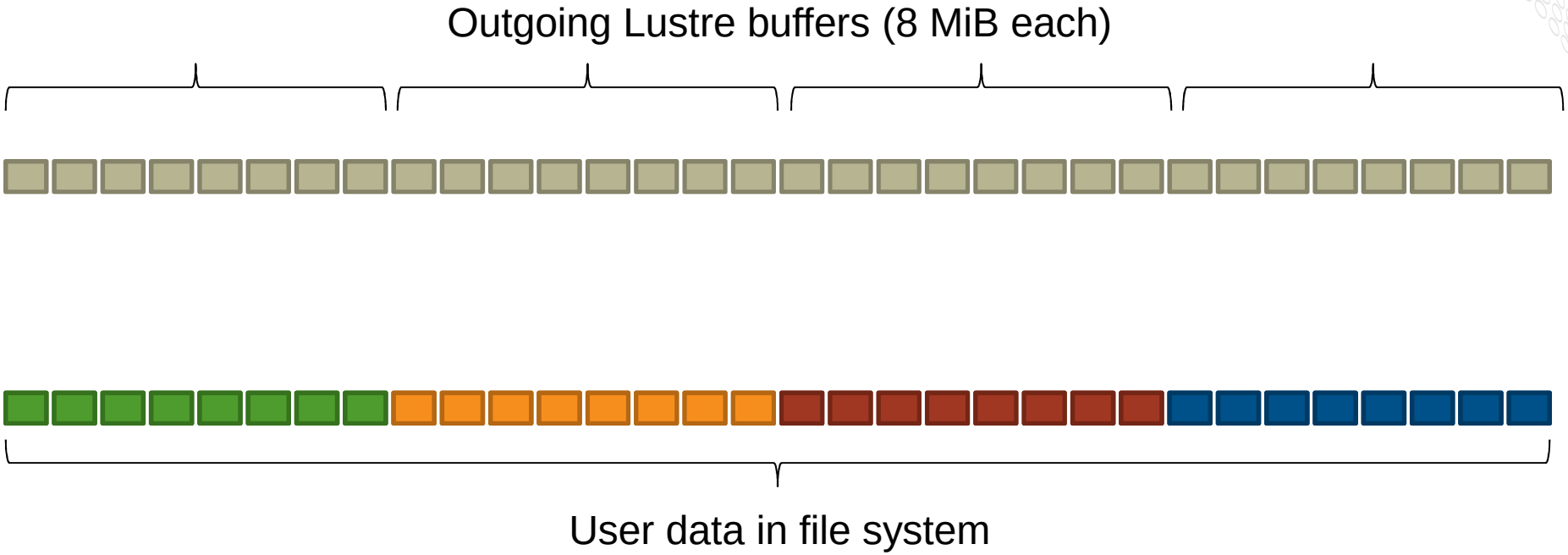
1M OST pre-allocation
4 files, 8M per file, read



8M OST pre-allocation
4 files, 8M per file, write



8M OST pre-allocation
4 files, 8M per file, read



Hypothesis

Using larger values of OST pre-allocation will:

- **Create more contiguous space when files are created**
- **Leave behind fewer, bigger, holes when files are remove**
- **Increase read rates due to more contiguous space**
- **Decrease write rates due to seeking**
- **The increase seen in read rates will be larger than the decrease in write rates, providing better balance for large-block sequential I/O**

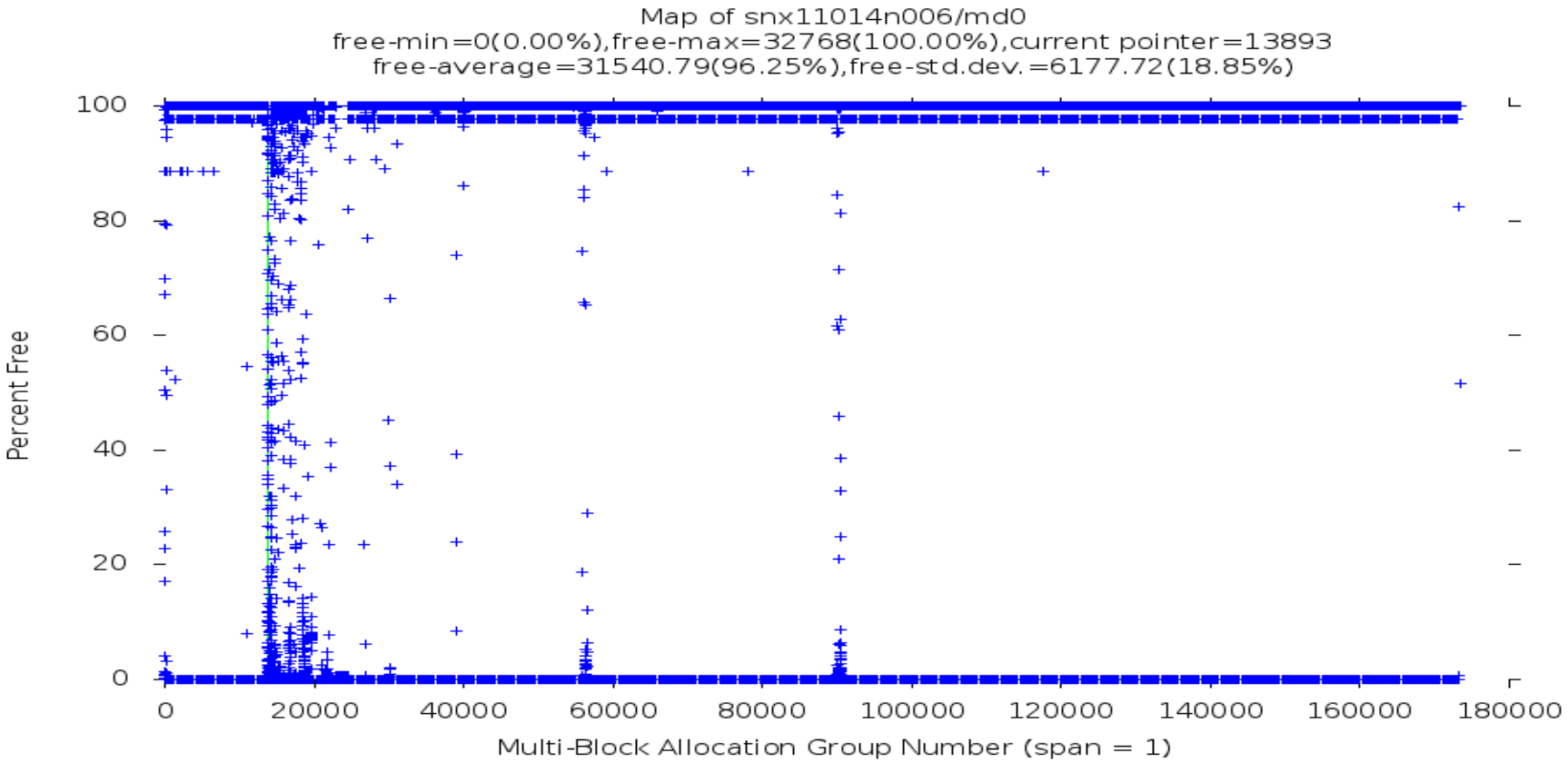
Testing Scenario

- Set 4 SSUs to 4 different OST pre-allocation sizes
- Choose a clean section of each OST
- Run optimal write and optimal read IOR jobs in each SSU to determine base rates at 100% free
- Simultaneously write 20 files on each OST so that data from all 20 files is intermingled
- Show distribution of contiguous space in each SSU

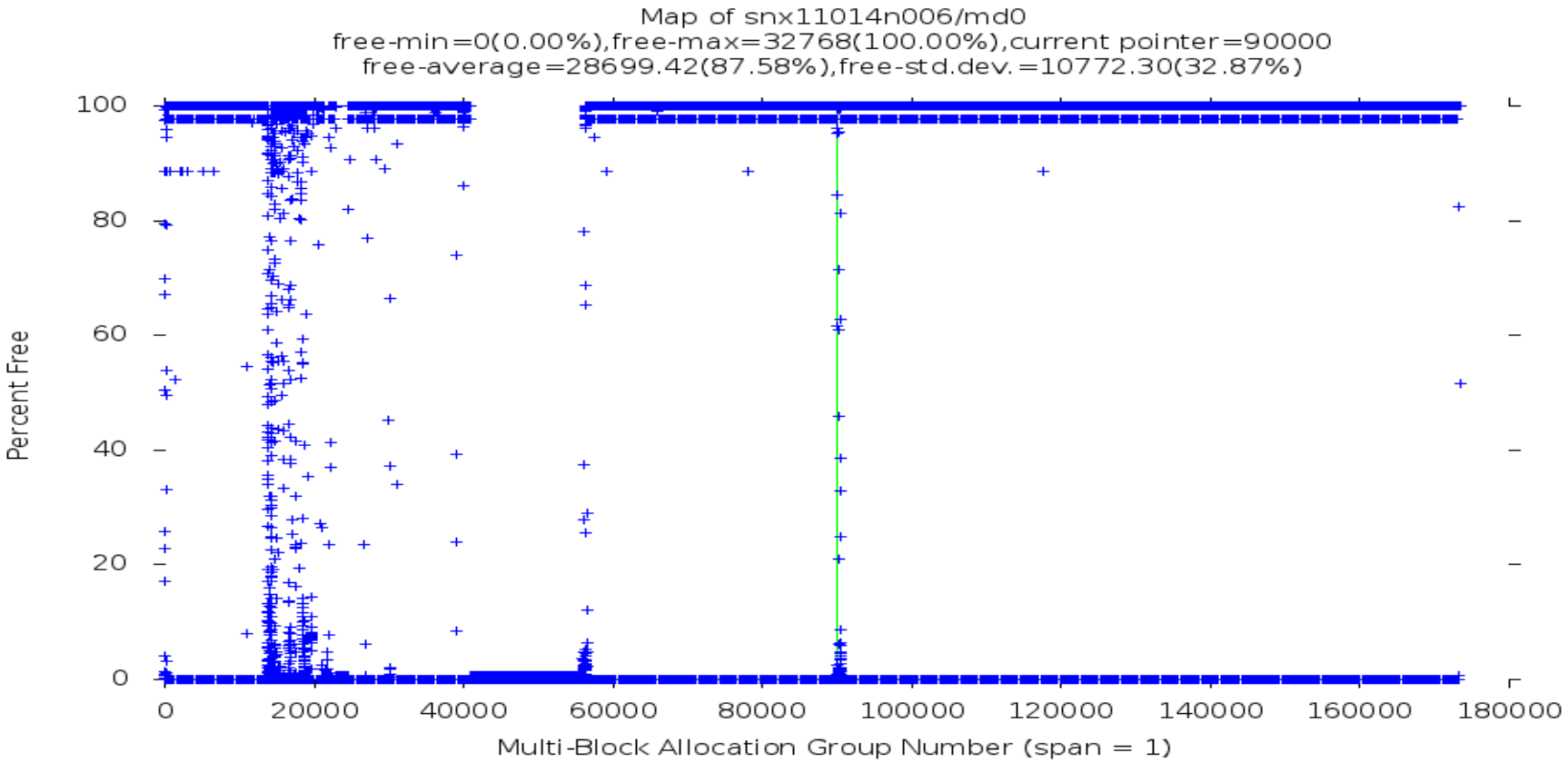
Testing Scenario (continued)

- **Repeat until 100% free again:**
  - Delete 1 file from each OST which releases 5% of the used space
  - Reset allocation pointer to beginning of test section on each OST
  - Run optimal write and optimal read IOR jobs
- **Record and plot results**
- **Test system**
  - Cray XE6, 12 LNET routers
    - Lustre 2.5.1 client
  - 4 SSU Sonexion 1600
    - NEO software level 1.2.1 (MDRAID)
    - 3 TB disk drives

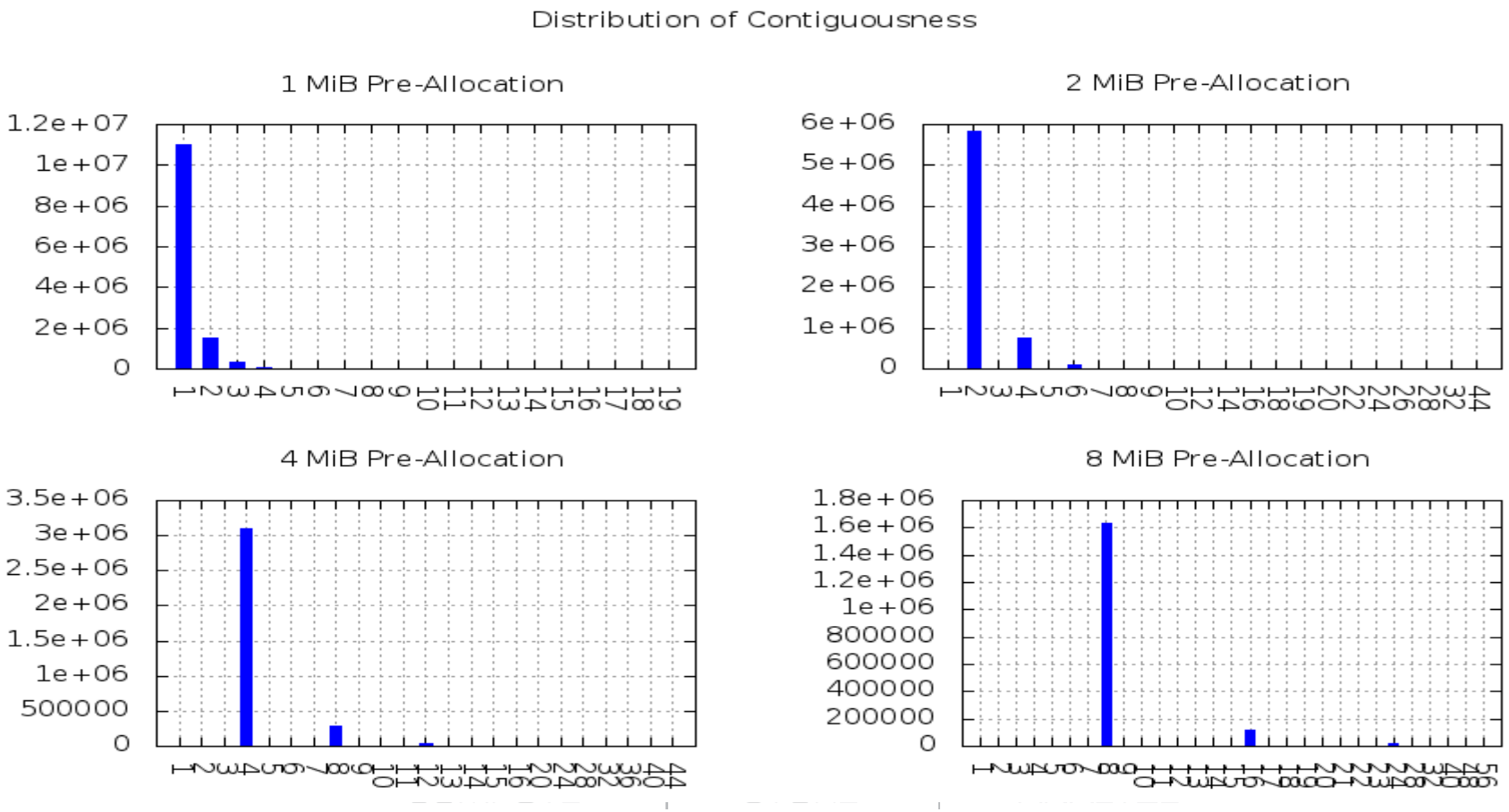
Bitmap of an OST (before)



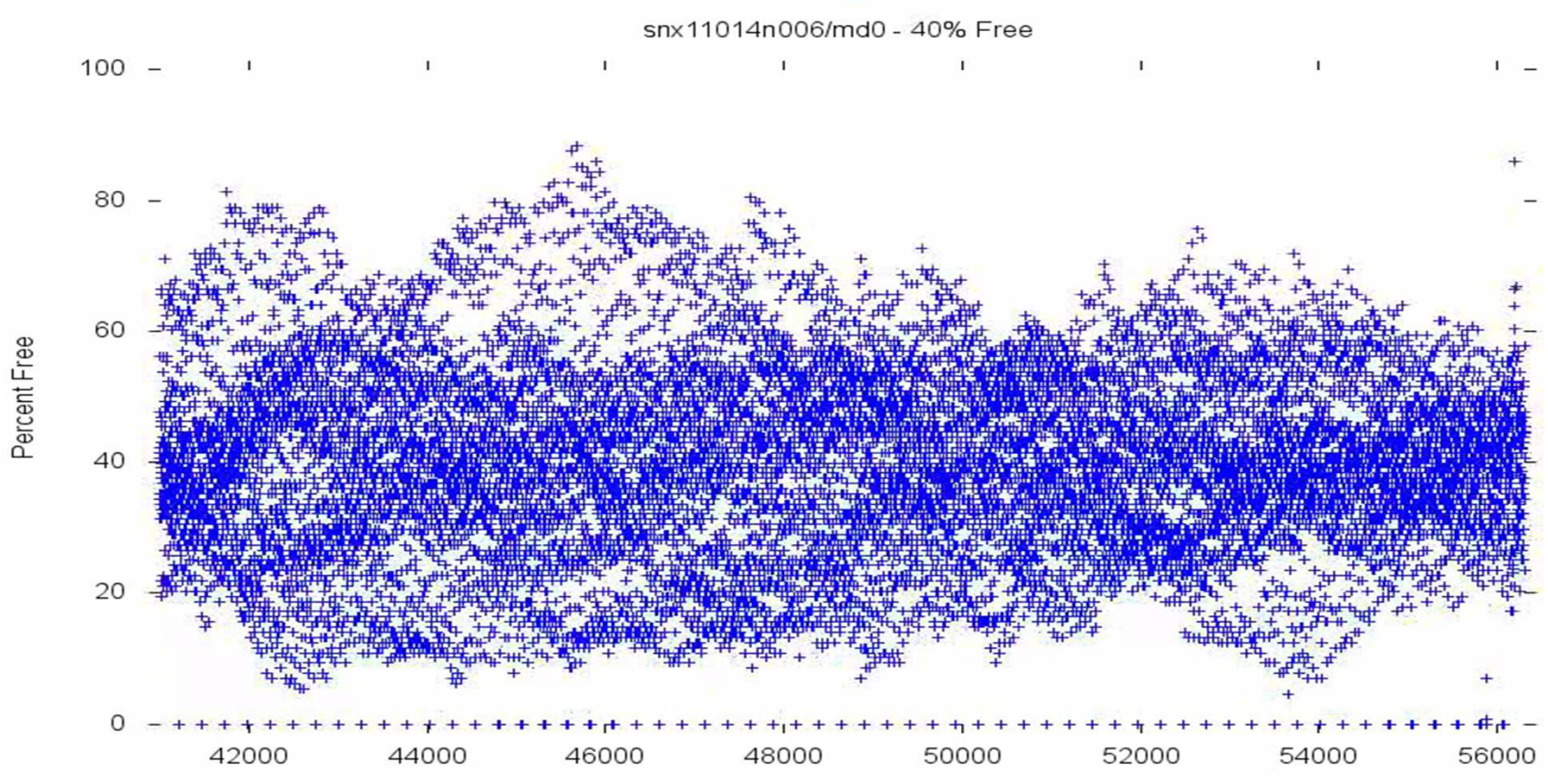
Bitmap of an OST (after)



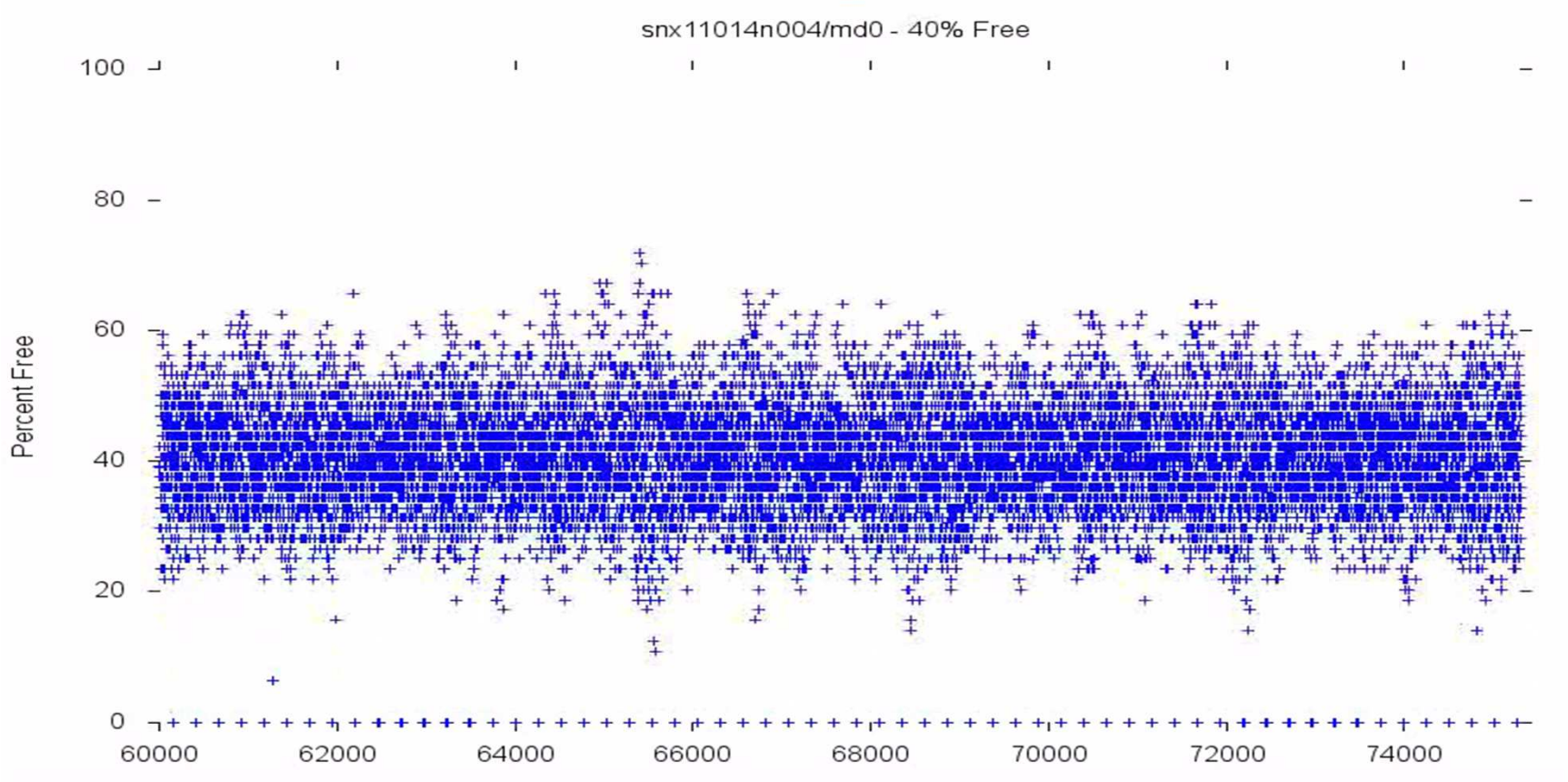
Histograms of extent sizes of IOR files



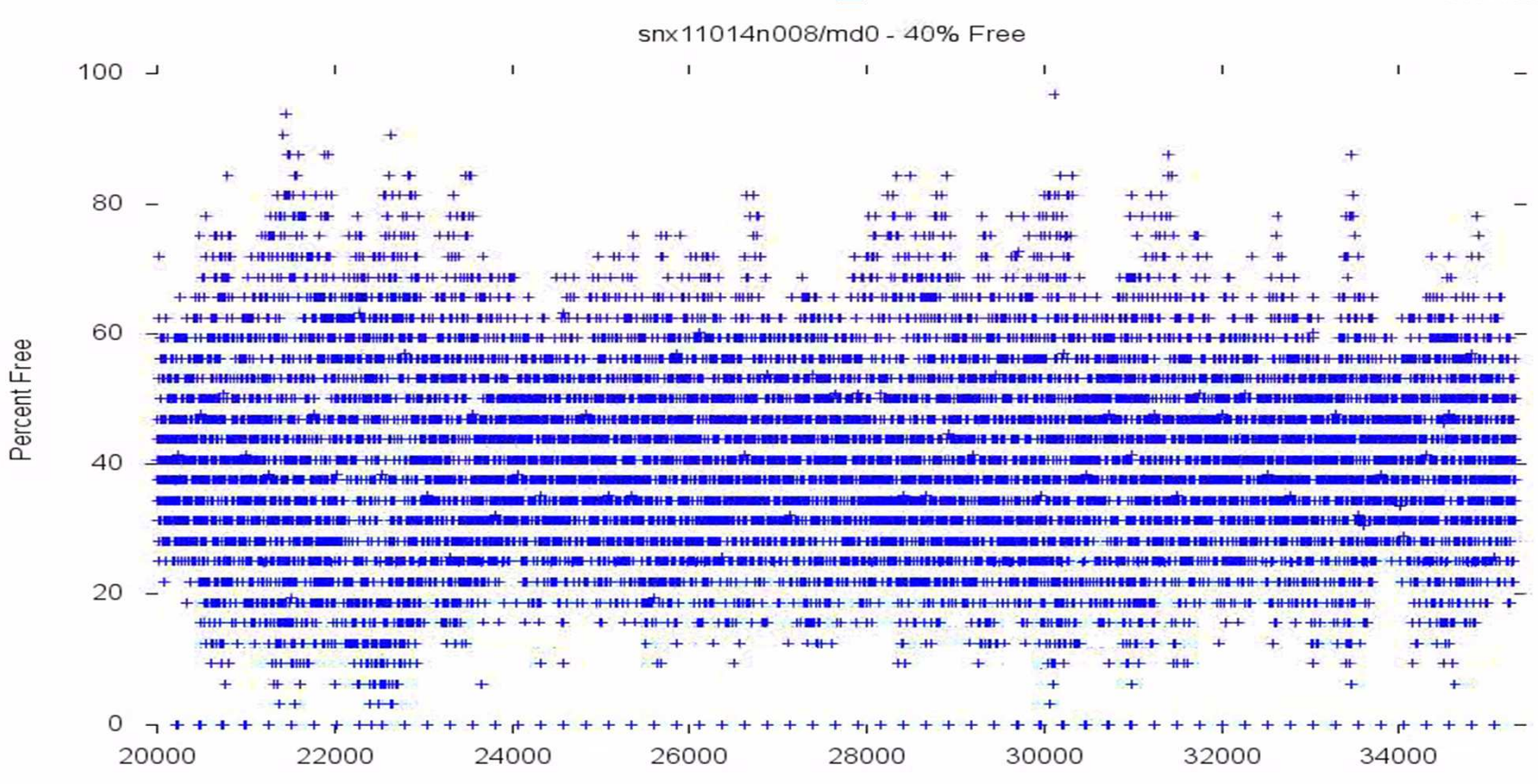
1 MiB OST Pre-allocation Fragmentation



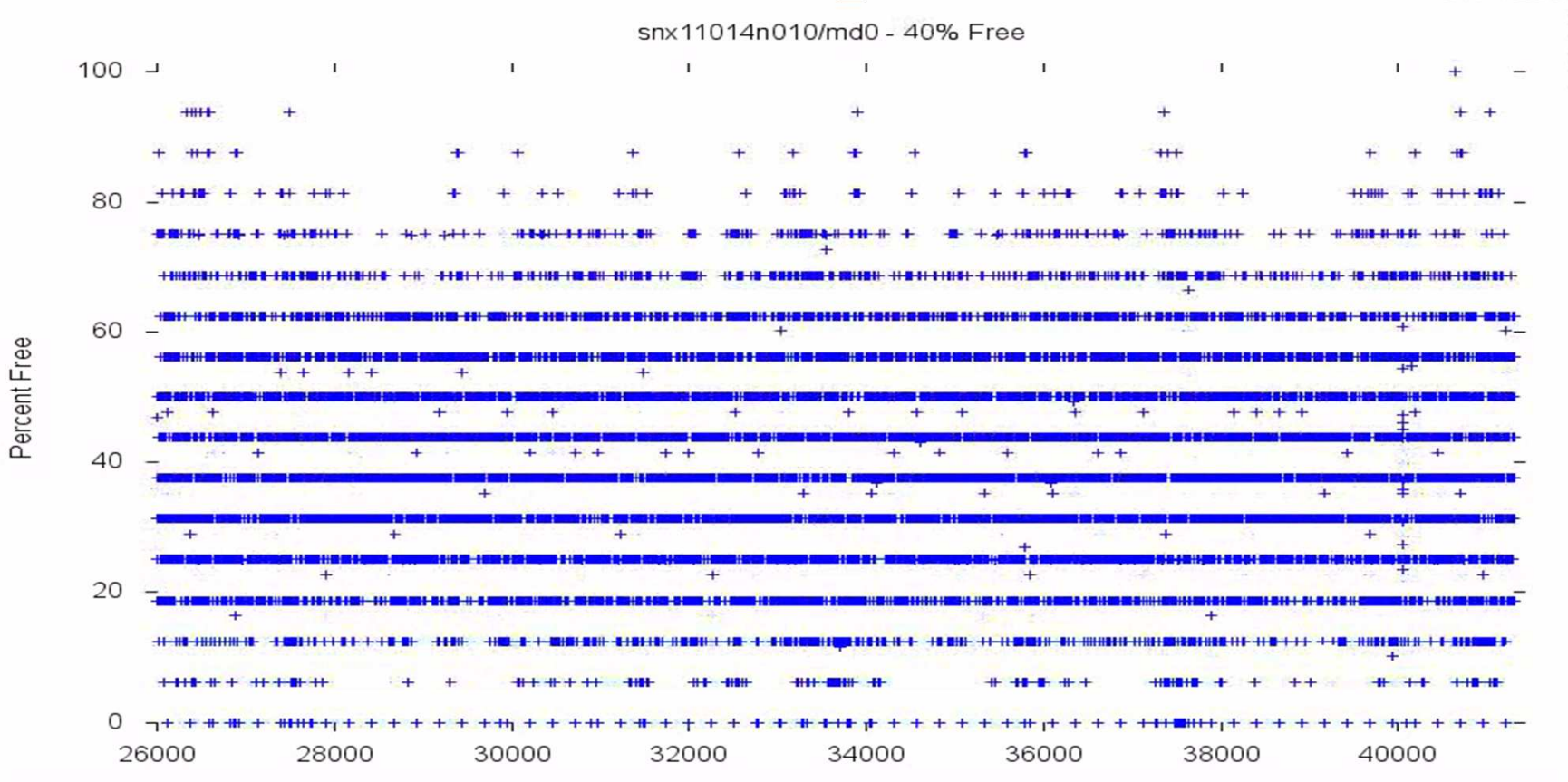
2 MiB OST Pre-allocation Fragmentation

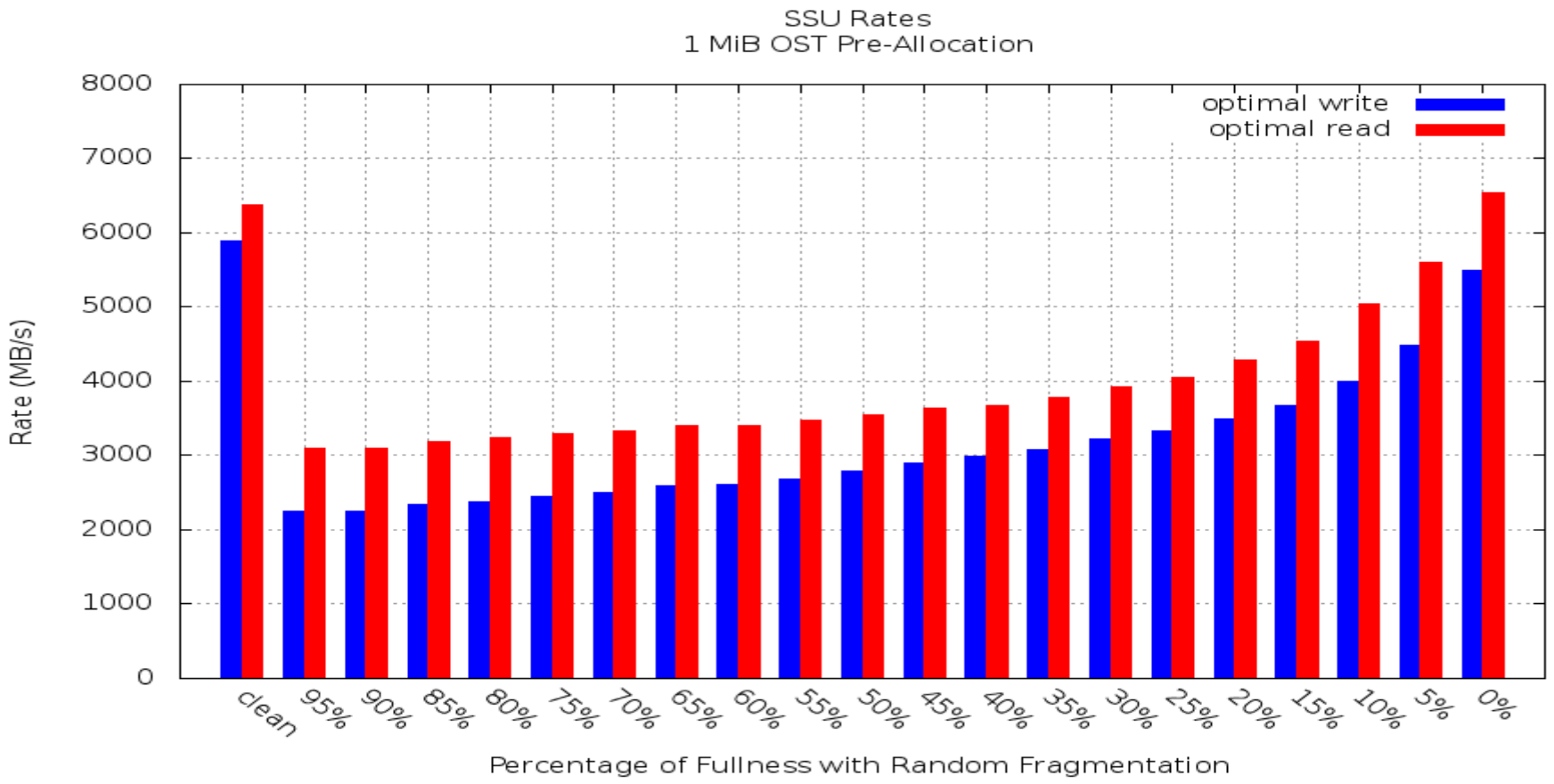


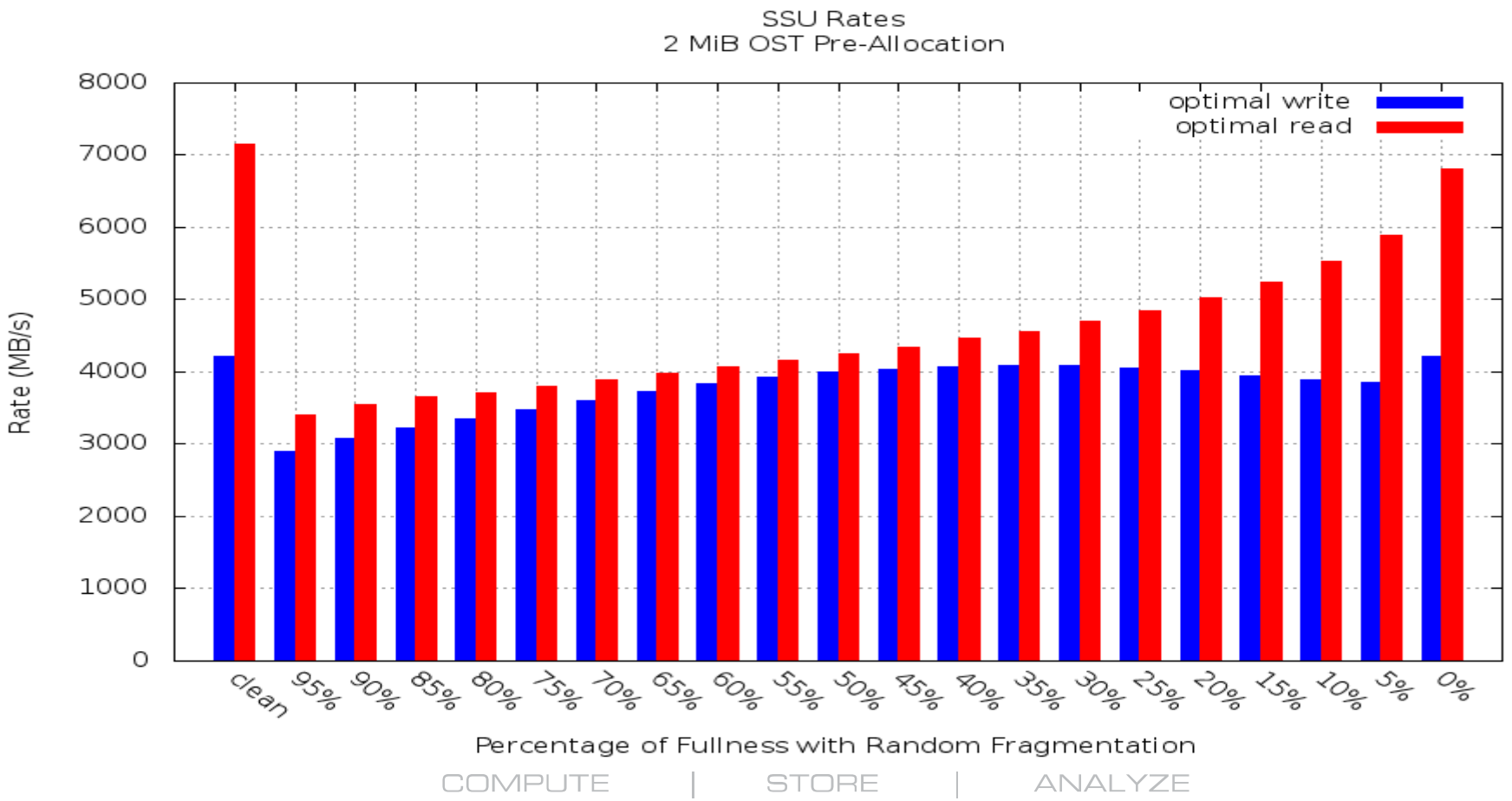
4 MiB OST Pre-allocation Fragmentation

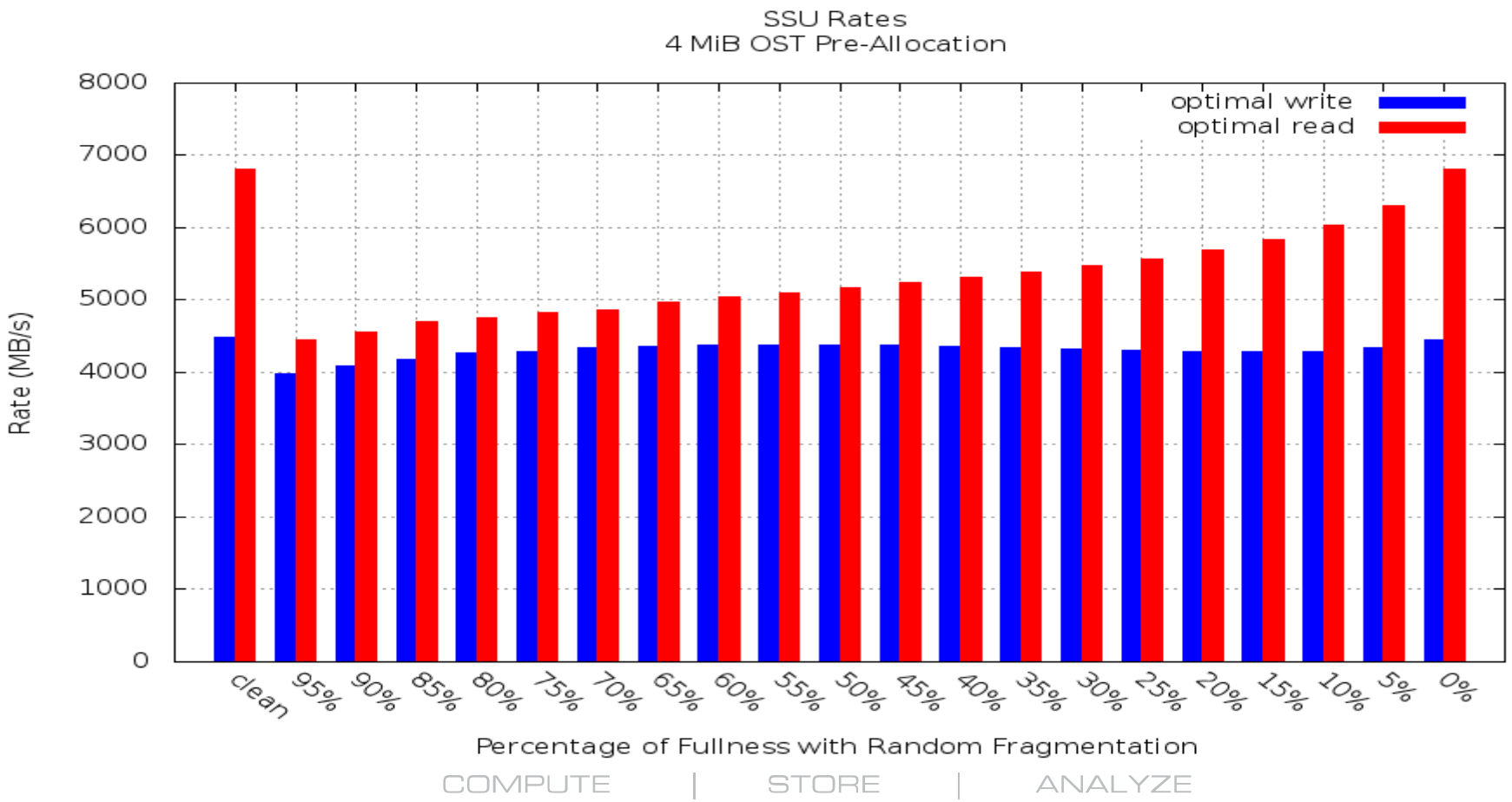


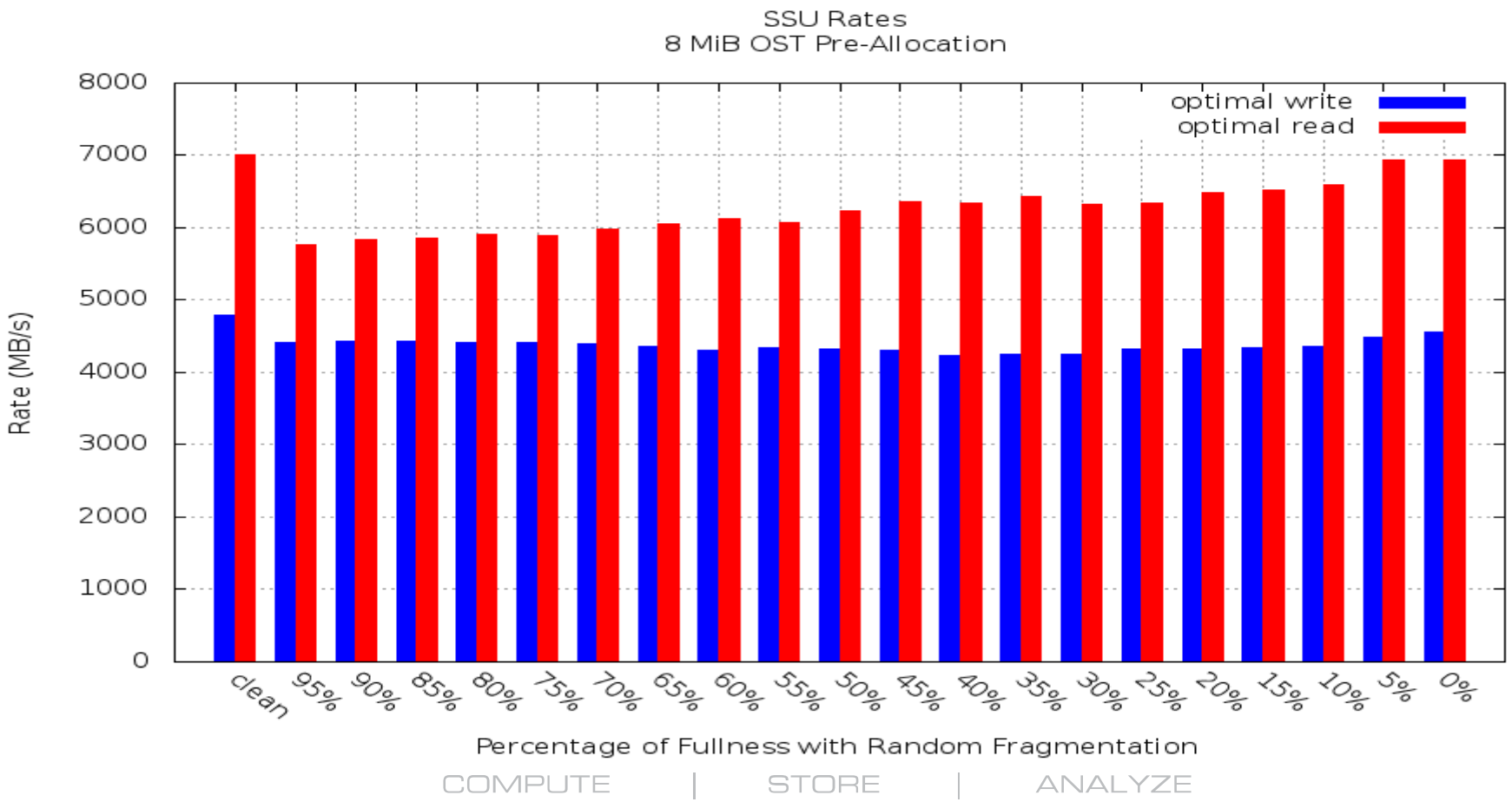
8 MiB OST Pre-allocation Fragmentation











Summary

- **The performance variation due to fragmentation is biggest for 1 MiB OST pre-allocation.**
  - This is the default for Sonexion 1600.
- **Using 8 MiB OST pre-allocation, no matter what level of fragmentation:**
  - The optimal write rate is at least 89% of a clean file system
  - The optimal read rate is at least 82% of a clean file system
  - Optimal write rates are always above 4 GB/s/SSU
  - Optimal read rates are always above 5 GB/s/SSU

Checking Settings on OSS Nodes

- **cat /proc/fs/ldiskfs/md*/prealloc_table**
  - 32 64 128 # default prealloc
  - 128 256 512 # prealloc of 2 MiB
  - 256 512 1024 # prealloc of 4 MiB
  - 256 512 1024 2048 # prealloc of 8 MiB
- **cat /proc/fs/ldiskfs/md*/mb_last_group**
  - Which of the multi-block allocation groups was last used
  - Example: 8000 means about 1 TB from start of OST

Thank You

- Questions ?