NCI

Providing Australian researchers with world-class computing services

Lustre Admins & Developers Workshop 2016

Petascale Data Migration

Daniel Rodwell
Manager, Data Storage Services

September 2016

NCRIS
National Research Infrastructure for Australia
An Australian Government Initiative

Australian Government
Bureau of Meteorology

Australian Government
Geoscience Australia

Australian Government
Australian Research Council

CSIRO

Australian National University

W nci.org.au

@NCInews

Agenda

- **NCI Storage Overview**
  - Systems & Growth
- **Migration Drivers**
  - Redistribution of Content
  - Filesystem Decommissioning / Replacement
- **Performance Profiles**
  - Filesystem Source & Destination
  - Data Migration Nodes
- **Considerations & Planning**
- **Data Migration Tools**
  - Quick Comparison of Utilities
  - Performance
- **Data Migration Process**
  - NCI Approach
  - Performance in Practice
- **Issues and Future Considerations**

30PB High Performance Storage

Storage at NCI

NCI – an overview

- NCI is Australia's national high-performance computing service
  - comprehensive, vertically-integrated research service
  - providing national access on priority and merit
  - driven by research objectives

- Operates as a formal collaboration of ANU, CSIRO, the Australian Bureau of Meteorology and Geoscience Australia

- As a partnership with a number of research-intensive Universities, supported by the Australian Research Council.

NCRIS
National Research
Infrastructure for Australia
An Australian Government Initiative

Australian National University

CSIRO

Australian Government
Bureau of Meteorology

Australian Government
Geoscience Australia

Australian Government
Australian Research Council

DEAKIN UNIVERSITY

INTERSECT

RMIT UNIVERSITY

qcif

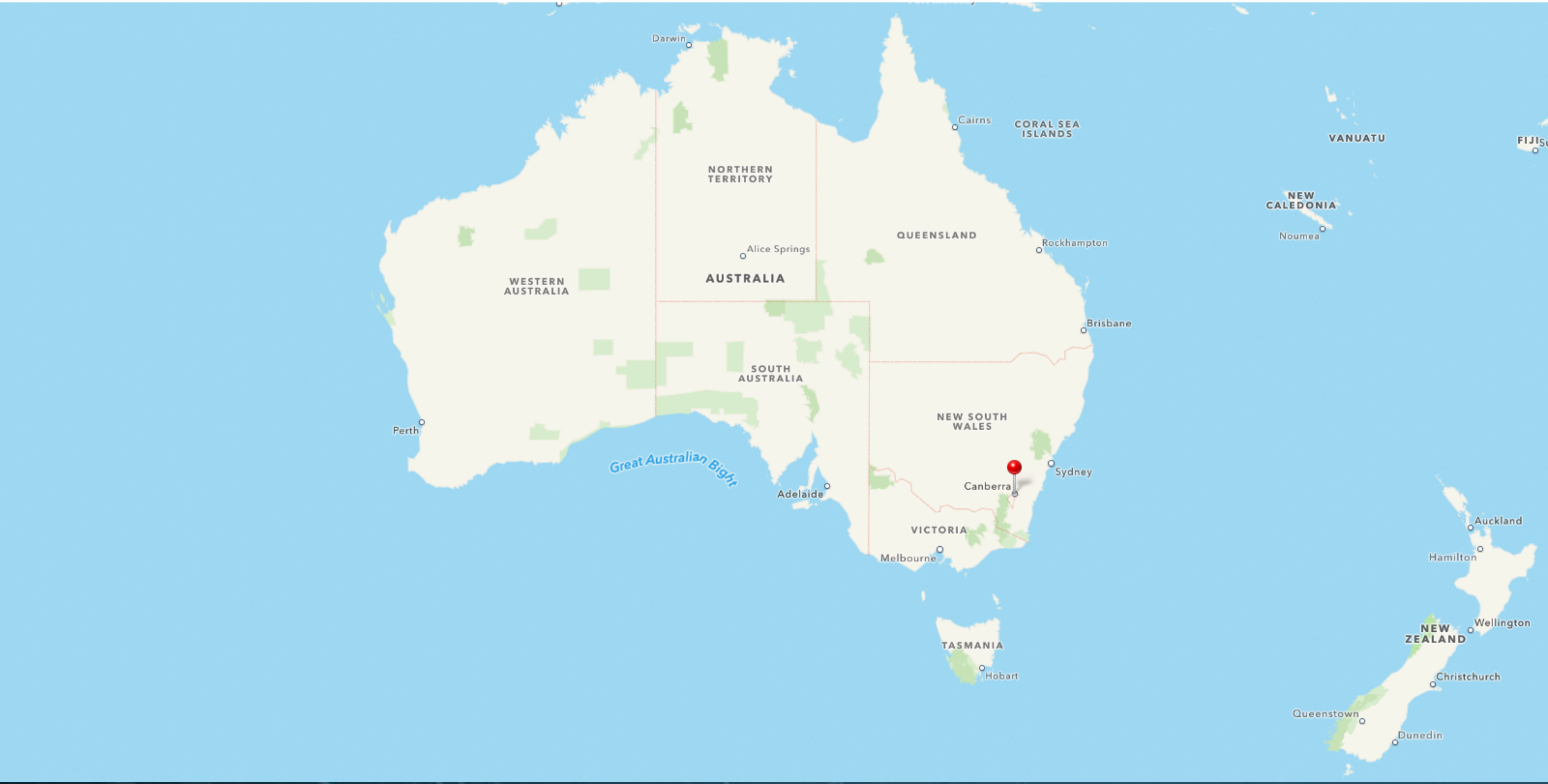
ACE CRC
ANTARCTIC CLIMATE & ECOSYSTEMS
COOPERATIVE RESEARCH CENTRE

UTS
UNIVERSITY OF TECHNOLOGY, SYDNEY

GARVAN INSTITUTE
OF MEDICAL RESEARCH

Where are we located?

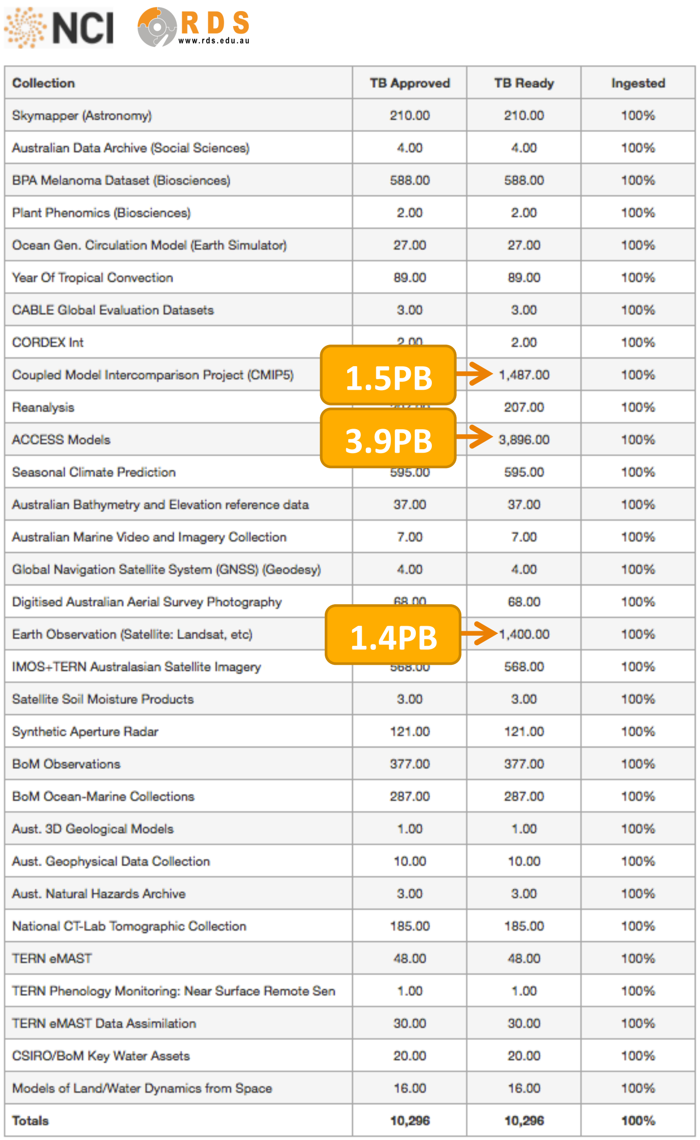
- Canberra, ACT
- The Australian National University (ANU)



What do we store?

- **How big?**
  - Very.
  - Average data collection is 50-100+ Terabytes
  - Larger data collections are multi-Petabytes in size
  - Individual files can exceed 2TB or be as small as a few KB.
  - Individual datasets consist of tens of millions of files
  - Next Generation datasets likely to be 6-10x larger.

  - Gdata1+2+3 = 451 Million inodes stored
  - 1% of /g/data1 capacity = 74TB



NCI RDS www.rds.edu.au

Collection	TB Approved	TB Ready	Ingested
Skymapper (Astronomy)	210.00	210.00	100%
Australian Data Archive (Social Sciences)	4.00	4.00	100%
BPA Melanoma Dataset (Biosciences)	588.00	588.00	100%
Plant Phenomics (Biosciences)	2.00	2.00	100%
Ocean Gen. Circulation Model (Earth Simulator)	27.00	27.00	100%
Year Of Tropical Convection	89.00	89.00	100%
CABLE Global Evaluation Datasets	3.00	3.00	100%
CORDEX Int	2.00	2.00	100%
Coupled Model Intercomparison Project (CMIP5)	[illegible]	1,487.00	100%
Reanalysis	207.00	207.00	100%
ACCESS Models	[illegible]	3,896.00	100%
Seasonal Climate Prediction	595.00	595.00	100%
Australian Bathymetry and Elevation reference data	37.00	37.00	100%
Australian Marine Video and Imagery Collection	7.00	7.00	100%
Global Navigation Satellite System (GNSS) (Geodesy)	4.00	4.00	100%
Digitised Australian Aerial Survey Photography	68.00	68.00	100%
Earth Observation (Satellite: Landsat, etc)	[illegible]	1,400.00	100%
IMOS+TERN Australasian Satellite Imagery	568.00	568.00	100%
Satellite Soil Moisture Products	3.00	3.00	100%
Synthetic Aperture Radar	121.00	121.00	100%
BoM Observations	377.00	377.00	100%
BoM Ocean-Marine Collections	287.00	287.00	100%
Aust. 3D Geological Models	1.00	1.00	100%
Aust. Geophysical Data Collection	10.00	10.00	100%
Aust. Natural Hazards Archive	3.00	3.00	100%
National CT-Lab Tomographic Collection	185.00	185.00	100%
TERN eMAST	48.00	48.00	100%
TERN Phenology Monitoring: Near Surface Remote Sen	1.00	1.00	100%
TERN eMAST Data Assimilation	30.00	30.00	100%
CSIRO/BoM Key Water Assets	20.00	20.00	100%
Models of Land/Water Dynamics from Space	16.00	16.00	100%
Totals	**10,296**	**10,296**	**100%**

1.5PB

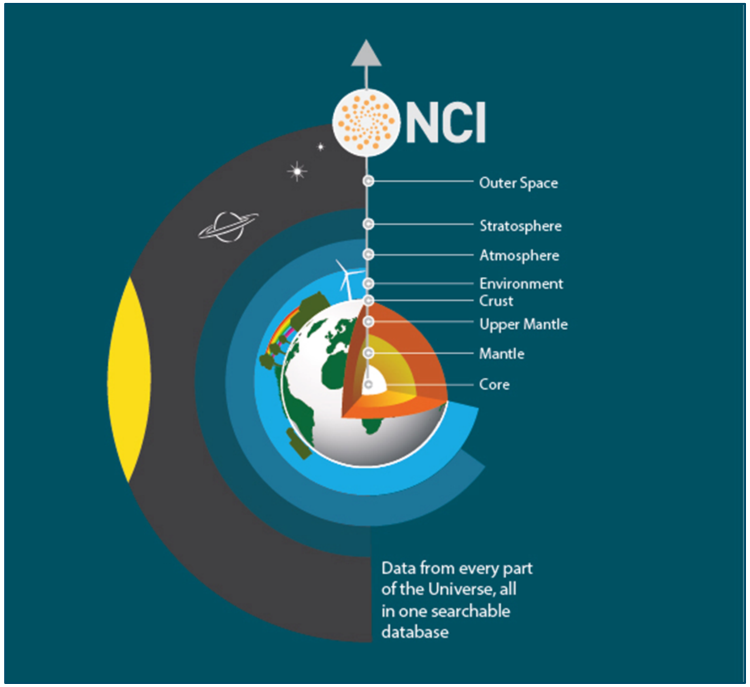
3.9PB

1.4PB

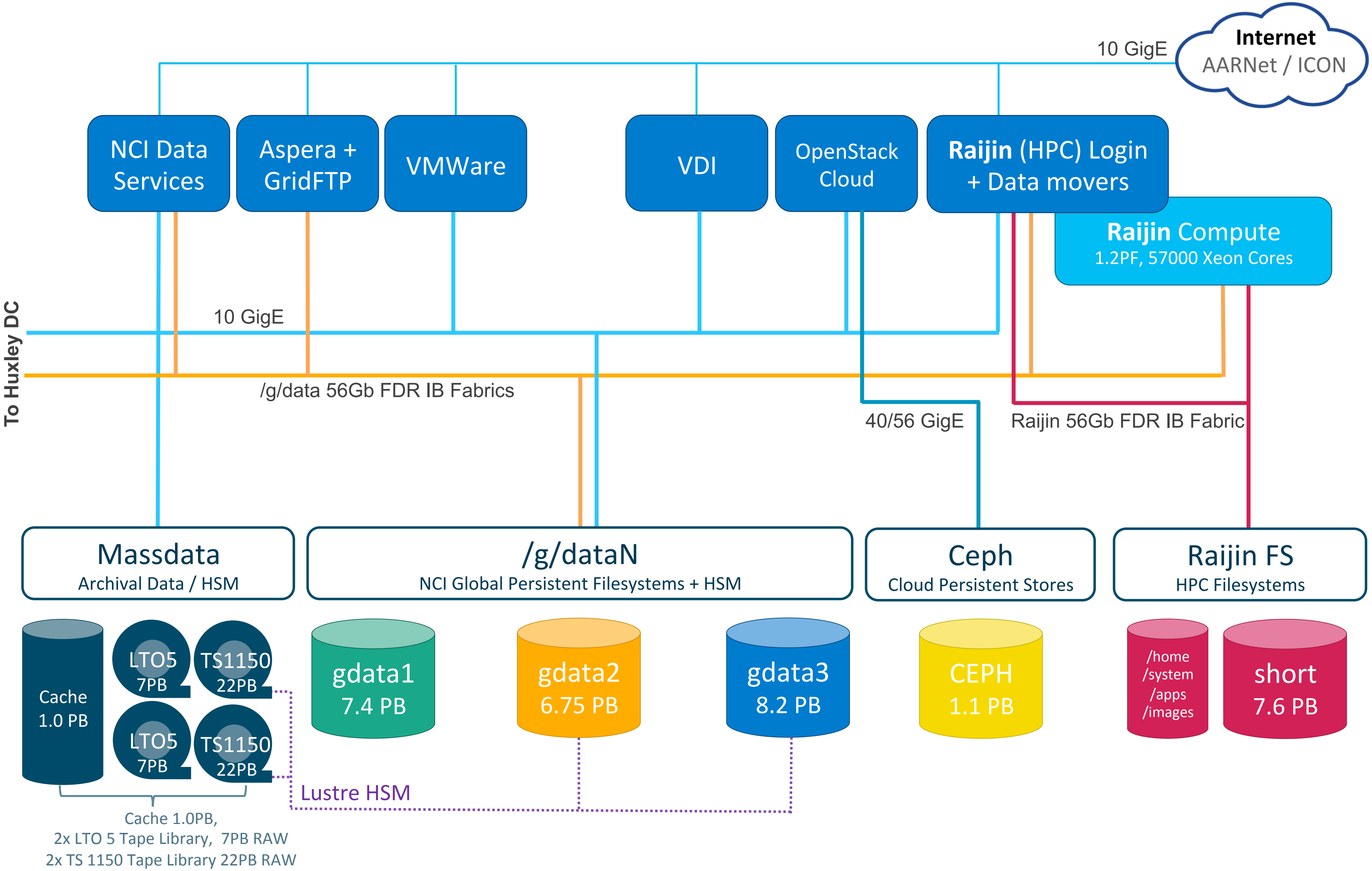
https://www.rds.edu.au/collections

What do we store?

- **What do we store?**
  - High value, cross-institutional collaborative scientific research collections.
  - Nationally significant data collections such as:
    - Australian Community Climate and Earth System Simulator (ACCESS) Models
    - Australian & international data from the CMIP5 and AR5 collection
    - Satellite imagery (Landsat, INSAR, ALOS)
    - Skymapper, Whole Sky Survey/ Pulsars
    - Australian Plant Phenomics Database
    - Australian Data Archive
    - EUMETSAT Copernicus Programme Sentinel Data
  - Large Scale Genomics and Bioinformatics datasets



Systems Overview



Storage Overview

- **Lustre Systems**
  - Raijin Lustre – HPC Filesystems: includes /short, /home, /apps, /images, /system
    - 7.6PB @ 150GB/Sec on /short (IOR Aggregate Sequential Write)
    - Lustre 2.5.23 + Custom patches (NCI + DDN)
  - Gdata1 – Persistent Data: /g/data1
    - 7.4PB @ 54GB/Sec Peak Read
    - Lustre 2.3.11 (IEEL v1)
  - Gdata2 – Persistent Data: /g/data2
    - 6.75PB @ 65GB/Sec Peak Read
    - Lustre 2.5.42.8 (IEEL v2)
  - Gdata3 – Persistent Data: /g/data3 –
    - Stage 1: 5.7PB @ 92GB/sec Peak Read
    - Stage 2: 8.2PB @ 120GB/Sec+ Peak Read
    - (Lustre 2.5.42.8, IEEL v2)

Why migrate data between filesystems?

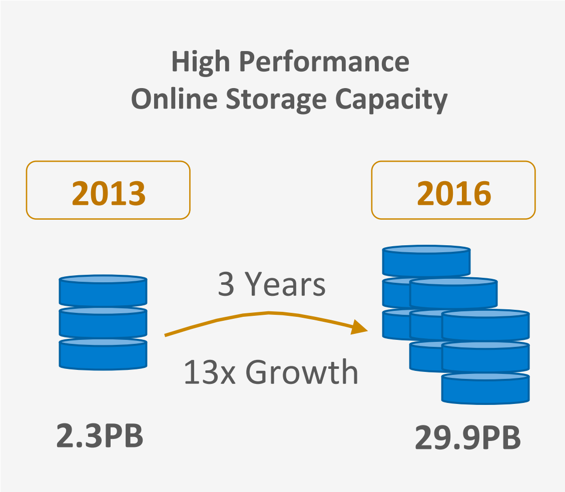
Data Migration

Why Migrate Data?

- **Reasons for migrating data**
  - Migrate data from an old filesystem being decommissioned on to a new system
  - Migrate a dataset or project to a different filesystem for performance, feature or security profile reasons
  - Need to rebalance storage allocation distribution between filesystems to manage overall capacity and growth
  - Duplication of data to multiple filesystems for protection or rollback
  - Staged replacement of Persistent Filesystems – continual rolling replacement schedule.

Previous NCI Systems

- 3 Years ago…
  - Vayu HPC
    - Previous Gen HPC Lustre Filesystem
    - 800TB, 25GB/Sec
  - Gdata (Original)
    - Persistent Lustre Filesystem on Vayu
    - 900TB, 6GB/sec
  - Projects
    - Dual State CXFS/DMF Filesystem
    - 1.4PB, 3GB/sec



High Performance
Online Storage Capacity

2013

2016

3 Years

13x Growth

2.3PB

29.9PB

- Migrated over 8PB, 100+ Million files between various internal data systems
- Need to find a solution that can scale to PetaBytes of data.
- Traditional approaches handle GigaBytes, not PetaBytes.

Nature of the Problem

- We have a High Performance Data Problem
  - An individual Project may be 2-3PB, 40+ Million files in size
  - Each file within the dataset or project needs to be read, written and verified
  - The time to process the data must be reasonable
  - A sequential, linear or traditional approach is unlikely to scale
  - Distributed & parallel processing of the problem is likely required

Component Performance & Resources Available

Performance Profiles

Raijin Test Cluster

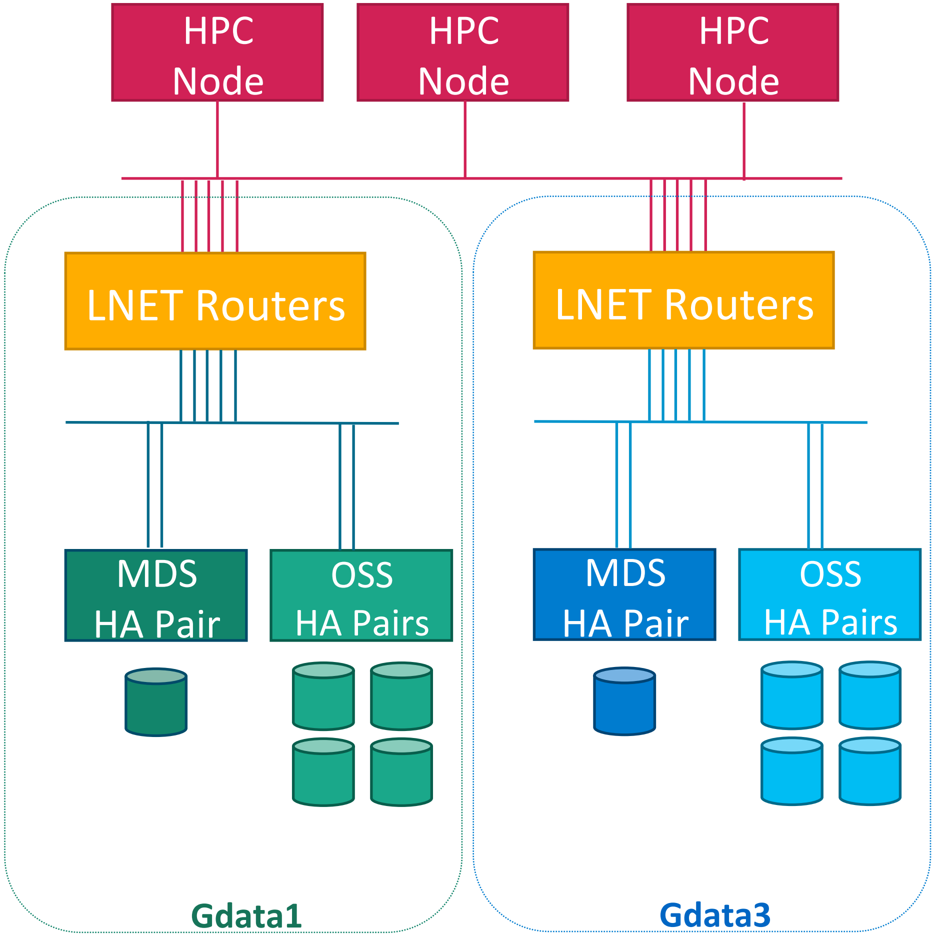
- 36x Fujitsu CX250 Nodes
- Dual Intel Sandy Bridge Xeon E5-2670, 8C, 2.6GHz (same spec as main cluster)
- 32GB DDR3
- InfiniBand FDR interconnect, connected to Raijin HPC Fabric
- All Lustre Filesystems mounted

– **Summary**
- 36 Nodes
- 576 Cores
- 36x IB interfaces at 5GB/sec (180GB/sec agg)
- Exemption - Can ssh between nodes
- Exemption - Can run jobs as root
- Administrative / Test Jobs do not block user jobs.
- Failed Administrative / Test jobs not shared on nodes with user jobs

Filesystem Performance

- **Source**
  - Gdata1
  - 54GB/sec peak read performance
  - 520 OSTs, 400MB/sec peak R each
  - Lustre 2.3.11

- **Destination**
  - Gdata3
  - 70GB/sec+ peak write performance
  - 252 OSTs, 800MB/sec peak W each
  - Lustre 2.5.42.8



Preparing to Migrate User Project Data

Considerations & Planning

Things to Consider

Filesystem Bandwidth

- Typically, regular user filesystem access will still be present while an individual project migration is in progress
- We don't want to choke the filesystem with administrative data migration activities, or cause user jobs to go into heavy IO wait
- Use long term monitoring to predict average user bandwidth requirements during migration period. Typically use max 50% of available filesystem bandwidth for project migrations.

Dry Run

- Dry run in business hours when all system administration staff are present
- Something **will** break unexpectedly during testing

Run as Root

- Unless the administrator's account has user/group access permissions to the files, typically you'll be running as root. Plan carefully. Think twice, run once.
- Build a 'Flight Plan' of all commands that you plan to run before you run them.
- There is the potential to overwrite the wrong data, at speed, disastrously, as root.

Things to Consider

- **Dataset**
  - Determine the size and count of files before the migration, build a test data set to evaluate scaling and timing estimates
  - Use a test data set **to prove to yourself** that the mechanisms/utilities work as expected

- **Batch Process**
  - Break up the data into smaller batches within a project, i.e. run each first level subdirectory within the project as its own copy
  - Its easier to restart / resume on failure.
  - You can have hardware / software failures as any other HPC job could

- **Data Custodians**
  - Agreed time with data custodian for migration to occur
  - Use dry runs and scaling tests to estimate time required
  - Have a rollback plan
  - Preserve the original data (restricted access) on the source filesystem for a time period after the initial migration has completed.

Datasets

– Determine count and size of dataset

- Use find, du, lfs quota
- Use rbh-lhsm-report for quick summary

```
group     ,     type,      count,     volume,     status,   avg_size
proj1     ,  symlink,          4,         69,       n/a,         17
proj1     ,      dir,     149351,  929.67 MB,       n/a,    6.37 KB
proj1     ,     file,    6238341,  990.45 TB,       new,  166.48 MB
Total: 6387696 entries, 1089016908285488 bytes used (990.46 TB)


group     ,     type,      count,     volume,     status,   avg_size
proj2     ,  symlink,      10358,  600.30 KB,       n/a,         59
proj2     ,      dir,      11792,  157.83 MB,       n/a,   13.71 KB
proj2     ,     file,    1570527,  229.37 TB,       new,  153.14 MB
Total: 1592677 entries, 252192953669719 bytes used (229.37 TB)


group     ,     type,      count,     volume,     status,   avg_size
proj3     ,  symlink,        404,   13.91 KB,       n/a,         35
proj3     ,      dir,    1279878,    5.98 GB,       n/a,    4.90 KB
proj3     ,     file,   43594251,    3.29 PB,       new,   81.03 MB
Total: 44874533 entries, 3704084238427042 bytes used (3.29 PB)
```

Comparison of toolsets

Data Migration Tools

– Many different options available

- Lustre has some Lustre-to-Lustre filesystem replication mechanisms
- Many different copy approaches available
- Most filesystem migrations at NCI occur on a project by project basis – a gradual migration of projects from a filesystem being decommissioned or rebalanced.
- Options presented here have been found viable for Project / dataset copies between high performance filesystems.
- Utilities are independent of Filesystem type (work for non-lustre) and do not require a specific release of lustre, and are available today.

Data Migration Tools

– Traditional cp

- **`cp –Rp /path/source /path/dest`**
- Always an option
- Manual handling required to get performance out of it – build and split lists, or assign subdirectories.

– Traditional Rsync

- **`rsync -aAXS --numeric-ids -–many-many-options /path/source /path/dest`**
- Smarter than cp
- Not particularly well optimized for very large files or high bandwidth conditions
- Accurate, reliable, well understood
- Can use initial copy to stage data into place, followed by differential sync
- Manual handling / scripting required to distribute work over multiple nodes
- Large amounts of data will take a long time if not automated and distributed to multiple nodes.

Pfsync

- https://github.com/martymac/fpart
- http://manpages.ubuntu.com/manpages/wily/man1/fpsync.1.html

- Automate work distribution and queuing over the top of rsync
- Uses fpart to build filelists
- Has queue manager to distribute filelists as jobs to worker rsync processes

- Can use most rsync options
- … `-aAXS --numeric-ids --many-many-options /path/source /path/dest`

- Still not particularly optimized for very large files or high bandwidth conditions
- Easy to understand what is going on as it is based on rsync
- Can use initial an copy and difference sync to stage data into place

- Need to figure out partition size parameters for optimal performance and well balanced workload distribution

Data Migration Tools

- **dcp2 (distributed copy)**
  - http://fileutils.io
  - https://github.com/hpc/fileutils/blob/master/doc/markdown/dcp.1.md
  - Main Contributors – LANL, LLNL, ORNL

  - MPI application - scales very well
  - MPI application – single node failure is fatal
  - May need to tune mpirun parameters.
    - Can exceed memory on node
    - May need to adjust number of processes per host
    - May need to set mpirun bind-to options

  - Limited options compared to rsync
  - Can break lower performing filesystems with load
  - Recommendation - start low with fewer nodes and processes, then scale up tests

Example Data Set

– Example dataset built for test

- Typical NCI project has millions of files
- Individual file size is commonly in the 100MB-150MB range
- Files created using
  – `dd bs=1048576 count=100 if=/dev/urandom of=/randomfile.$number`

/g/data1/proj/exampledata	(4 Million files, 400TB)
> /Bin1	(500,000 x 100MB files, 50TB)
> /Bin 1A	(100,000 x 100MB files, 10TB)
> /Bin 111	(10,000 x 100MB files, 1TB)
> /Bin1111	(1000 x100MB files, 100GB)
> /Bin2	(500,000 x 100MB files, 50TB)
> …	
> ...	

- Lustre stripe count = 1 for all files (gdata filesystem default)
- Gdata1 = 1.6PB free, 79% Used (when test dataset created)
- Gdata3 = 1.8PB free, 76% Used (at start of each copy run)

Small Scale Test (1T, 10 000 files)

– Small Scale Test – Traditional cp

- 1TB
- 10,000 x 100MB files
- 66 Minutes, 12 seconds.

```
# cp -Rp /g/data1/fu2/exampledata/Bin1/Bin1A/Bin111 /g/data3/fu2/
exampletransfer/cptest/
```

```
bash-4.1# date; time cp -Rp /g/data1/fu2/exampledata/Bin1/Bin1A/Bin111 /g/data3/fu2/
exampletransfer/cptest/; date

Tue Sep  6 22:19:18 AEST 2016
          real    66m12.661s
          user    0m0.514s
          sys     40m27.818s
Tue Sep  6 23:25:31 AEST 2016

bash-4.1#
```

- iotop - cp performing a single process copy at approx. 290-340MB/sec
- 350MB/sec is about the average write performance we expect from a gdata1 OST

Small Scale Test (1T, 10 000 files)

– Small Scale Test – Traditional Rsync

- 1TB
- 10,000 x 100MB files
- 4 Hours, 35 Minutes

```
# rsync -aAXS --numeric-ids /g/data1/fu2/exampledata/Bin1/Bin1A/Bin111 /g/
data3/fu2/exampletransfer/rsynctest/
```

```
bash-4.1# date; time rsync -aAXS --numeric-ids /g/data1/fu2/exampledata/Bin1/Bin1A/
Bin111 /g/data3/fu2/exampletransfer/rsynctest/; date

Fri Sep  9 15:30:28 AEST 2016
          real    261m14.329s
          user    79m46.624s
          sys     222m10.312s
Fri Sep  9 19:51:42 AEST 2016

bash-4.1#
```

- iotop – rsync performing a single process sync at approx 35-75MB/sec
- Default Rsync 3.0.6 from Centos 6.7 Repo. No custom compiler options.

Medium Scale Test (10T, 100 000 files)

Medium Scale Test – fpsync, 16 nodes

- 10TB
- 100,000 x 100MB files

- Fpsync requires the size and count of the worker partitions to be passed to it as command parameters.
- -n is the number of partitions (jobs)
- -f is the filecount for each partition
- -s is the size in bytes for each partition

- 100,000 files, 10484019363840 bytes (9.535 TiB)

- - n = nodes x cores x 2 = 16 nodes x 16 cores x 2 = **512**

- - f = number files / # of partitions = 100000 / 512 = **200** (round up)

- - s = number of bytes / # of partitions = 10484019363840 / 512
  = 20476600320 = **20480000000** (round up)

Medium Scale Test (10T, 100 000 files)

– Medium Scale Test – fpsync, 16 nodes

- 10TB
- 100,000 x 100MB files

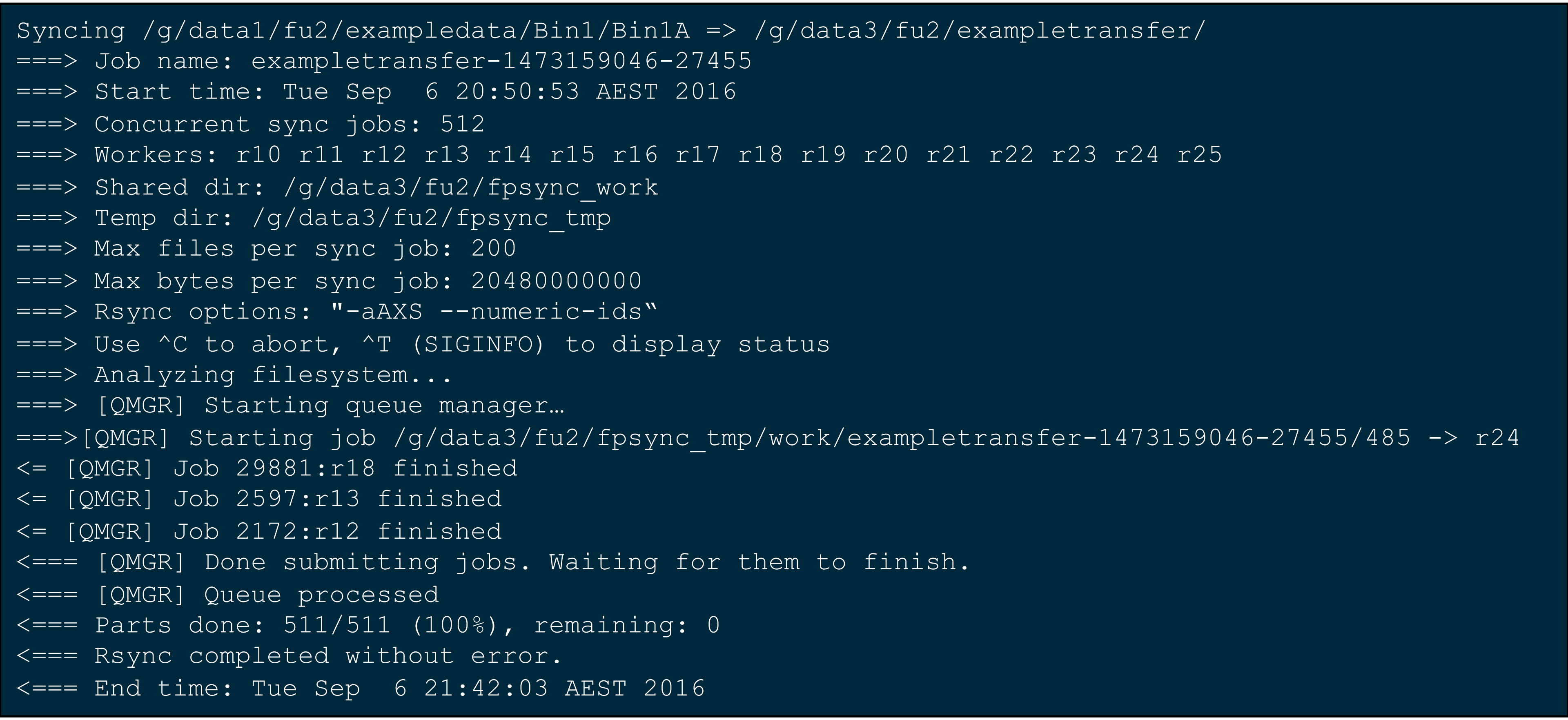
```
# /sbin/fpsync \-w 'r10 r11 r12 r13 r14 r15 r16 r17 r18 r19 r20 r21
r22 r23 r24 r25'
-d /g/data3/fu2/fpsync_work
-t /g/data3/fu2/fpsync_tmp
-vv -n 512 -s 20480000000
-f 200 -o '-aAXS --numeric-ids’
/g/data1/fu2/exampledata/Bin1/Bin1A
/g/data3/fu2/exampletransfer/
| tee /g/data3/fu2/fpsync_16_node_10T_test.out
#
```

- Run fpsync in a `screen` or `tmux` session
- `tee` stdout/stderr to a text file for review

Medium Scale Test (10T, 100 000 files)

– Medium Scale Test – fpsync, 16 nodes

- 16 Nodes, 512 Partitions (concurrent sync jobs)
- 10T, 100000 files copied
- 52 Minutes



```
Syncing /g/data1/fu2/exampledata/Bin1/Bin1A => /g/data3/fu2/exampletransfer/
===> Job name: exampletransfer-1473159046-27455
===> Start time: Tue Sep  6 20:50:53 AEST 2016
===> Concurrent sync jobs: 512
===> Workers: r10 r11 r12 r13 r14 r15 r16 r17 r18 r19 r20 r21 r22 r23 r24 r25
===> Shared dir: /g/data3/fu2/fpsync_work
===> Temp dir: /g/data3/fu2/fpsync_tmp
===> Max files per sync job: 200
===> Max bytes per sync job: 20480000000
===> Rsync options: "-aAXS --numeric-ids"
===> Use ^C to abort, ^T (SIGINFO) to display status
===> Analyzing filesystem...
===> [QMGR] Starting queue manager…
===>[QMGR] Starting job /g/data3/fu2/fpsync_tmp/work/exampletransfer-1473159046-27455/485 -> r24
<= [QMGR] Job 29881:r18 finished
<= [QMGR] Job 2597:r13 finished
<= [QMGR] Job 2172:r12 finished
<=== [QMGR] Done submitting jobs. Waiting for them to finish.
<=== [QMGR] Queue processed
<=== Parts done: 511/511 (100%), remaining: 0
<=== Rsync completed without error.
<=== End time: Tue Sep  6 21:42:03 AEST 2016
```

Medium Scale Test (10T, 100 000 files)

– Medium Scale Test – dcp, 16 nodes

- 10TB
- 100,000 x 100MB files

- dcp only needs the number of processes to run, and the hosts to run on
- Typically use all 16 cores per node, 16

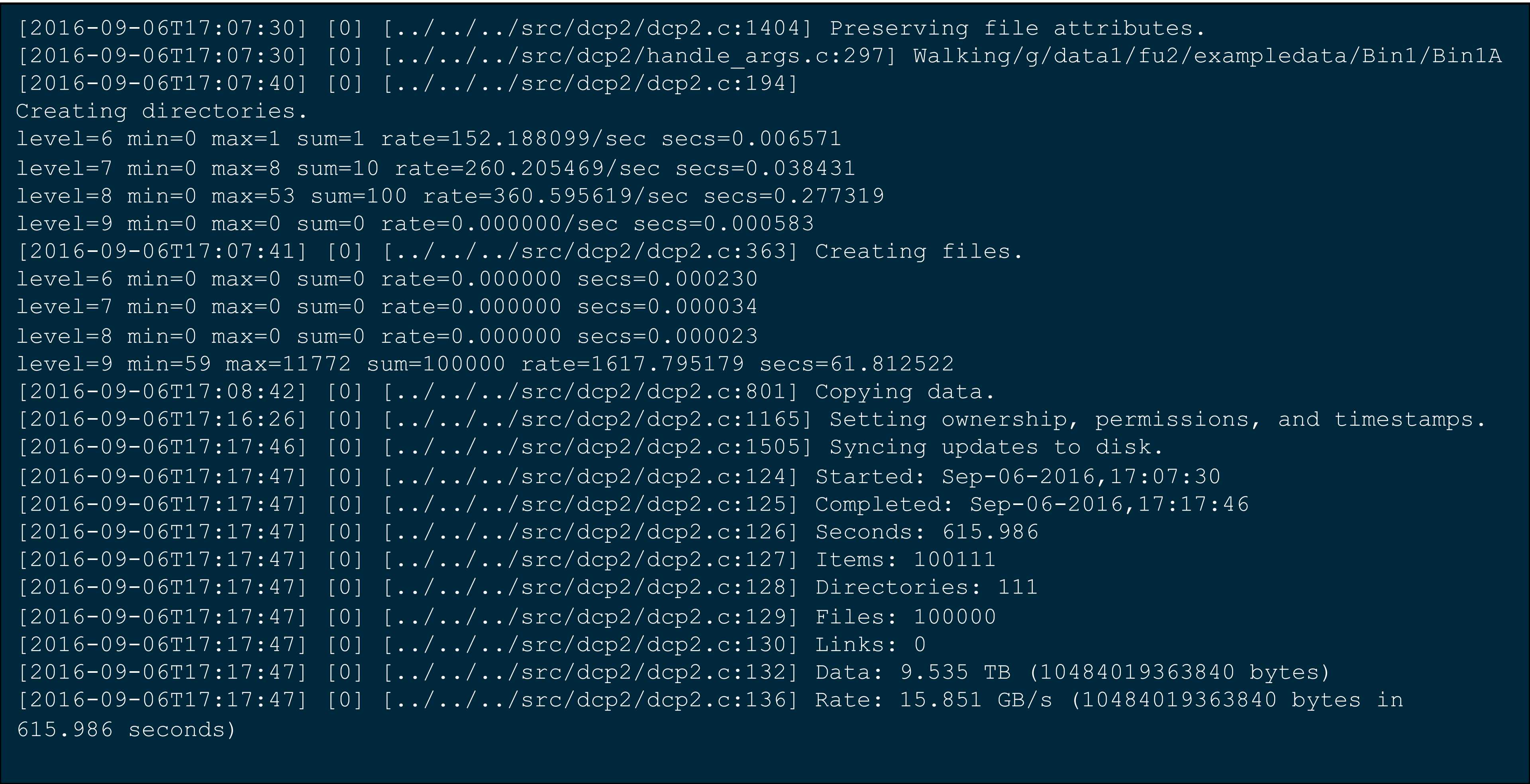
```
# module load dcp/1.0-NCI2
# mpirun --allow-run-as-root -np 256 -H
r10,r11,r12,r13,r14,r15,r16,r17,r18,r19,r20,r21,r22,r23,r24,r25 /
apps/dcp/1.0-NCI2/bin/dcp2 -p /g/data1/fu2/exampledata/Bin1/Bin1A /
g/data3/fu2/exampletransfer/
#
```

- Run dcp in a `screen` or `tmux` session
- `tee` stdout/stderr to a text file for review

Medium Scale Test (10T, 100 000 files)

– Medium Scale Test – dcp, 16 nodes

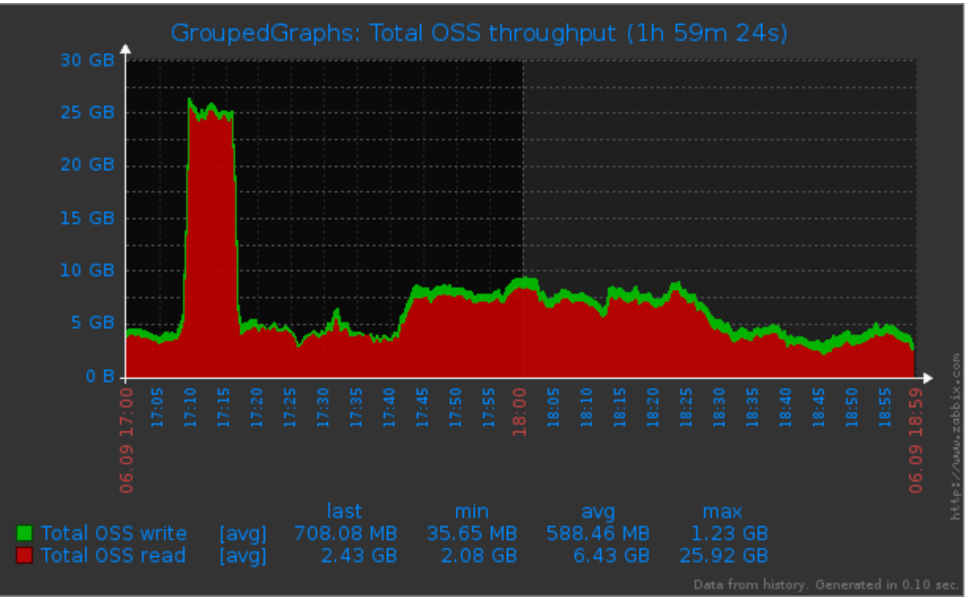
- 16 Nodes, 256 Processes
- 10T, 100000 files copied
- 10 minutes, 16 seconds.



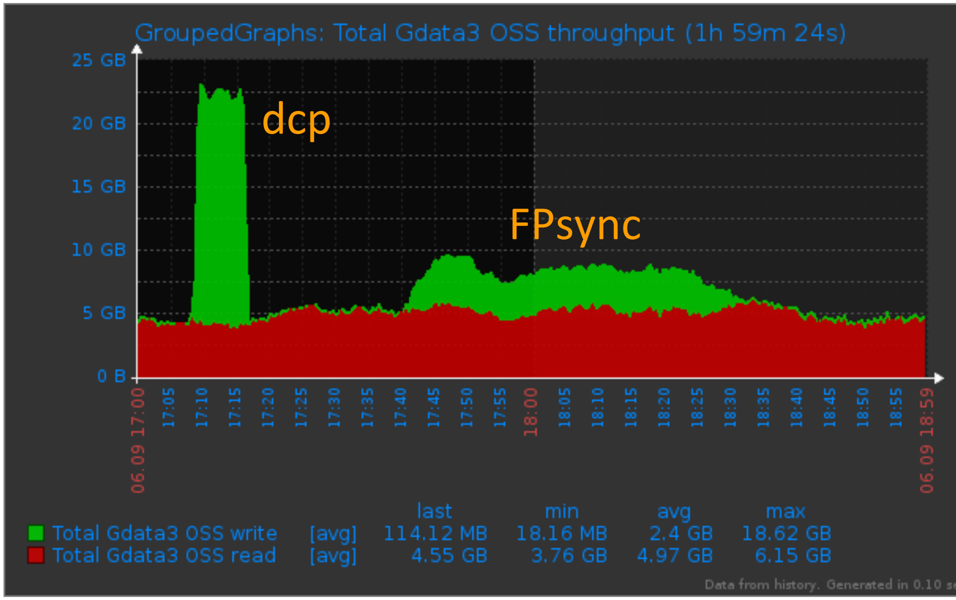
```
[2016-09-06T17:07:30] [0] [../../../src/dcp2/dcp2.c:1404] Preserving file attributes.
[2016-09-06T17:07:30] [0] [../../../src/dcp2/handle_args.c:297] Walking/g/data1/fu2/exampledata/Bin1/Bin1A
[2016-09-06T17:07:40] [0] [../../../src/dcp2/dcp2.c:194]
Creating directories.
level=6 min=0 max=1 sum=1 rate=152.188099/sec secs=0.006571
level=7 min=0 max=8 sum=10 rate=260.205469/sec secs=0.038431
level=8 min=0 max=53 sum=100 rate=360.595619/sec secs=0.277319
level=9 min=0 max=0 sum=0 rate=0.000000/sec secs=0.000583
[2016-09-06T17:07:41] [0] [../../../src/dcp2/dcp2.c:363] Creating files.
level=6 min=0 max=0 sum=0 rate=0.000000 secs=0.000230
level=7 min=0 max=0 sum=0 rate=0.000000 secs=0.000034
level=8 min=0 max=0 sum=0 rate=0.000000 secs=0.000023
level=9 min=59 max=11772 sum=100000 rate=1617.795179 secs=61.812522
[2016-09-06T17:08:42] [0] [../../../src/dcp2/dcp2.c:801] Copying data.
[2016-09-06T17:16:26] [0] [../../../src/dcp2/dcp2.c:1165] Setting ownership, permissions, and timestamps.
[2016-09-06T17:17:46] [0] [../../../src/dcp2/dcp2.c:1505] Syncing updates to disk.
[2016-09-06T17:17:47] [0] [../../../src/dcp2/dcp2.c:124] Started: Sep-06-2016,17:07:30
[2016-09-06T17:17:47] [0] [../../../src/dcp2/dcp2.c:125] Completed: Sep-06-2016,17:17:46
[2016-09-06T17:17:47] [0] [../../../src/dcp2/dcp2.c:126] Seconds: 615.986
[2016-09-06T17:17:47] [0] [../../../src/dcp2/dcp2.c:127] Items: 100111
[2016-09-06T17:17:47] [0] [../../../src/dcp2/dcp2.c:128] Directories: 111
[2016-09-06T17:17:47] [0] [../../../src/dcp2/dcp2.c:129] Files: 100000
[2016-09-06T17:17:47] [0] [../../../src/dcp2/dcp2.c:130] Links: 0
[2016-09-06T17:17:47] [0] [../../../src/dcp2/dcp2.c:132] Data: 9.535 TB (10484019363840 bytes)
[2016-09-06T17:17:47] [0] [../../../src/dcp2/dcp2.c:136] Rate: 15.851 GB/s (10484019363840 bytes in
615.986 seconds)
```

Medium Scale Test (10T, 100 000 files)

- **Medium Scale Test – dcp vs fpsync, 16 nodes**
- Aggregate OST throughput



		last	min	avg	max
🟩 Total OSS write	[avg]	708.08 MB	35.65 MB	588.46 MB	1.23 GB
🟥 Total OSS read	[avg]	2.43 GB	2.08 GB	6.43 GB	25.92 GB



		last	min	avg	max
🟩 Total Gdata3 OSS write	[avg]	114.12 MB	18.16 MB	2.4 GB	18.62 GB
🟥 Total Gdata3 OSS read	[avg]	4.55 GB	3.76 GB	4.97 GB	6.15 GB

Source Filesystem:
Gdata1 – 25.92GB/sec Read peak

Destination Filesystem:
Gdata3 – 18.62GB/sec Write peak

Medium Scale Test

- **Medium Scale Test – dcp vs fpsync**
  - 10TB
  - 100,000 x 100MB files
  - 16 Nodes

- Results
  - Fpsync – 52 Minutes
  - dcp – 10 Minutes, 16 Seconds

- What about Fpsync, re-run with no changes?
  - 2nd pass Fpsync, no data changes – 3 Minutes, 33 Seconds

- But…
  - Beware of potential partition imbalance with fpsync if planning a 2 phase transfer, ie bulk data staged into place in the background, then ‘offline’ differential sync run.
  - A large number of ”new files” may end up in few bins for the differential sync, which will result in just a few rsync processes doing the work.

Large Scale Test (50T, 500 000 files)

Large Scale Test – dcp, 16 nodes

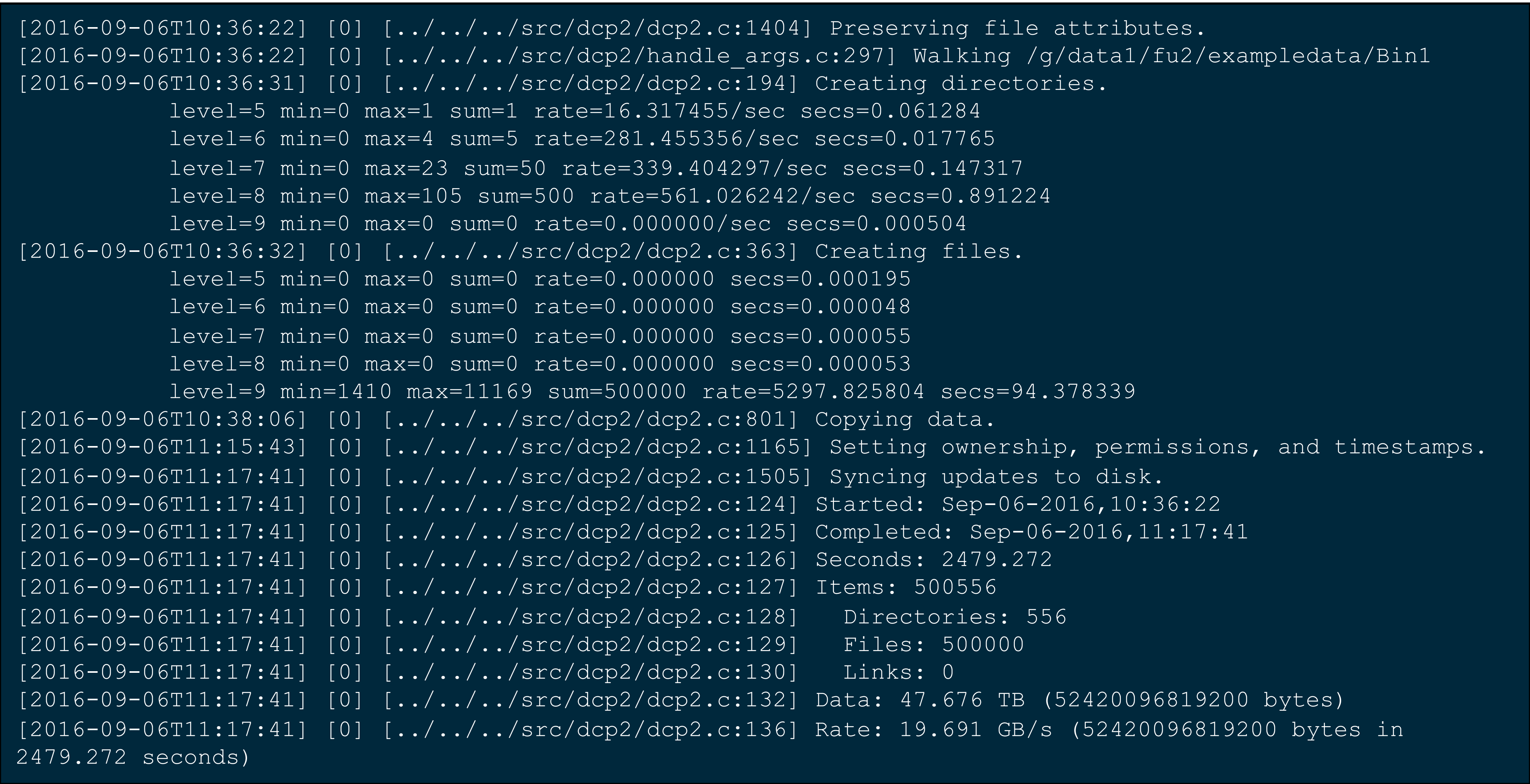
- 50TB
- 500,000 x 100MB files

```
# module load dcp/1.0-NCI2
# mpirun --allow-run-as-root -np 256 -H
r10,r11,r12,r13,r14,r15,r16,r17,r18,r19,r20,r21,r22,r23,r24,r25 /
apps/dcp/1.0-NCI2/bin/dcp2 -p /g/data1/fu2/exampledata/Bin1 /g/
data3/fu2/exampletransfer/
#
```

Large Scale Test (50T, 500 000 files)

– Large Scale Test – dcp, 16 nodes

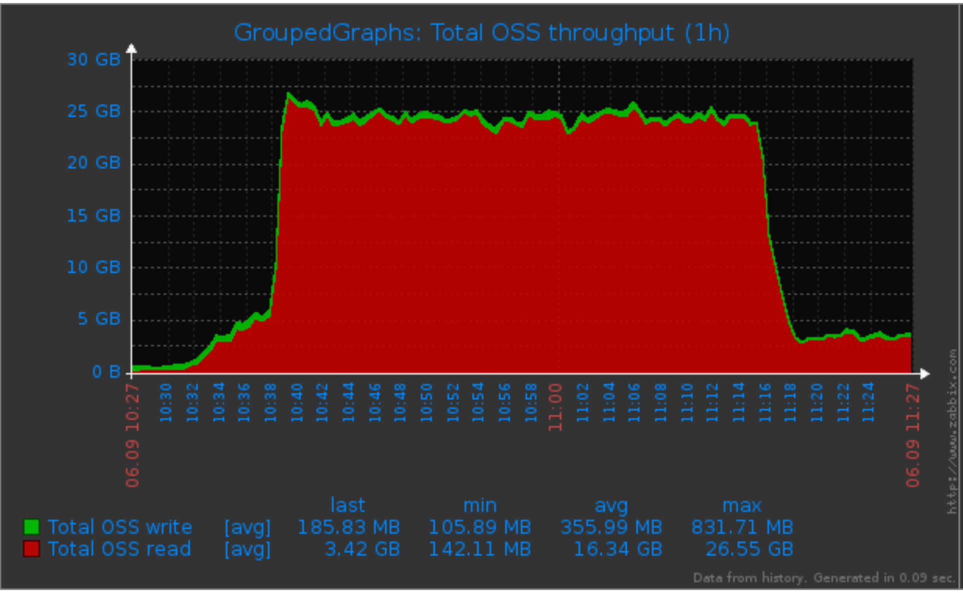
- 16 Nodes, 256 Processes
- 50T, 500 000 files copied
- 41 minutes, 10 seconds.



```
[2016-09-06T10:36:22] [0] [../../../src/dcp2/dcp2.c:1404] Preserving file attributes.
[2016-09-06T10:36:22] [0] [../../../src/dcp2/handle_args.c:297] Walking /g/data1/fu2/exampledata/Bin1
[2016-09-06T10:36:31] [0] [../../../src/dcp2/dcp2.c:194] Creating directories.
           level=5 min=0 max=1 sum=1 rate=16.317455/sec secs=0.061284
           level=6 min=0 max=4 sum=5 rate=281.455356/sec secs=0.017765
           level=7 min=0 max=23 sum=50 rate=339.404297/sec secs=0.147317
           level=8 min=0 max=105 sum=500 rate=561.026242/sec secs=0.891224
           level=9 min=0 max=0 sum=0 rate=0.000000/sec secs=0.000504
[2016-09-06T10:36:32] [0] [../../../src/dcp2/dcp2.c:363] Creating files.
           level=5 min=0 max=0 sum=0 rate=0.000000 secs=0.000195
           level=6 min=0 max=0 sum=0 rate=0.000000 secs=0.000048
           level=7 min=0 max=0 sum=0 rate=0.000000 secs=0.000055
           level=8 min=0 max=0 sum=0 rate=0.000000 secs=0.000053
           level=9 min=1410 max=11169 sum=500000 rate=5297.825804 secs=94.378339
[2016-09-06T10:38:06] [0] [../../../src/dcp2/dcp2.c:801] Copying data.
[2016-09-06T11:15:43] [0] [../../../src/dcp2/dcp2.c:1165] Setting ownership, permissions, and timestamps.
[2016-09-06T11:17:41] [0] [../../../src/dcp2/dcp2.c:1505] Syncing updates to disk.
[2016-09-06T11:17:41] [0] [../../../src/dcp2/dcp2.c:124] Started: Sep-06-2016,10:36:22
[2016-09-06T11:17:41] [0] [../../../src/dcp2/dcp2.c:125] Completed: Sep-06-2016,11:17:41
[2016-09-06T11:17:41] [0] [../../../src/dcp2/dcp2.c:126] Seconds: 2479.272
[2016-09-06T11:17:41] [0] [../../../src/dcp2/dcp2.c:127] Items: 500556
[2016-09-06T11:17:41] [0] [../../../src/dcp2/dcp2.c:128]   Directories: 556
[2016-09-06T11:17:41] [0] [../../../src/dcp2/dcp2.c:129]   Files: 500000
[2016-09-06T11:17:41] [0] [../../../src/dcp2/dcp2.c:130]   Links: 0
[2016-09-06T11:17:41] [0] [../../../src/dcp2/dcp2.c:132] Data: 47.676 TB (52420096819200 bytes)
[2016-09-06T11:17:41] [0] [../../../src/dcp2/dcp2.c:136] Rate: 19.691 GB/s (52420096819200 bytes in
2479.272 seconds)
```

Large Scale Test (50T, 500 000 files)

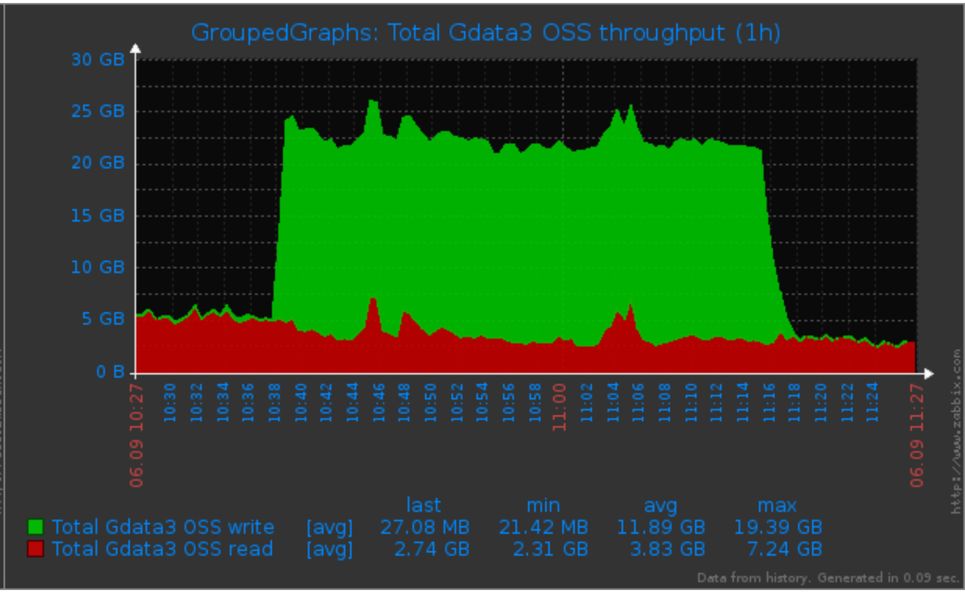
- **Large Scale Test – dcp, 16 nodes**
- OSS Activity



GroupedGraphs: Total OSS throughput (1h)

		last	min	avg	max
🟩 Total OSS write	[avg]	185.83 MB	105.89 MB	355.99 MB	831.71 MB
🟥 Total OSS read	[avg]	3.42 GB	142.11 MB	16.34 GB	26.55 GB

Data from history. Generated in 0.09 sec.



GroupedGraphs: Total Gdata3 OSS throughput (1h)

		last	min	avg	max
🟩 Total Gdata3 OSS write	[avg]	27.08 MB	21.42 MB	11.89 GB	19.39 GB
🟥 Total Gdata3 OSS read	[avg]	2.74 GB	2.31 GB	3.83 GB	7.24 GB

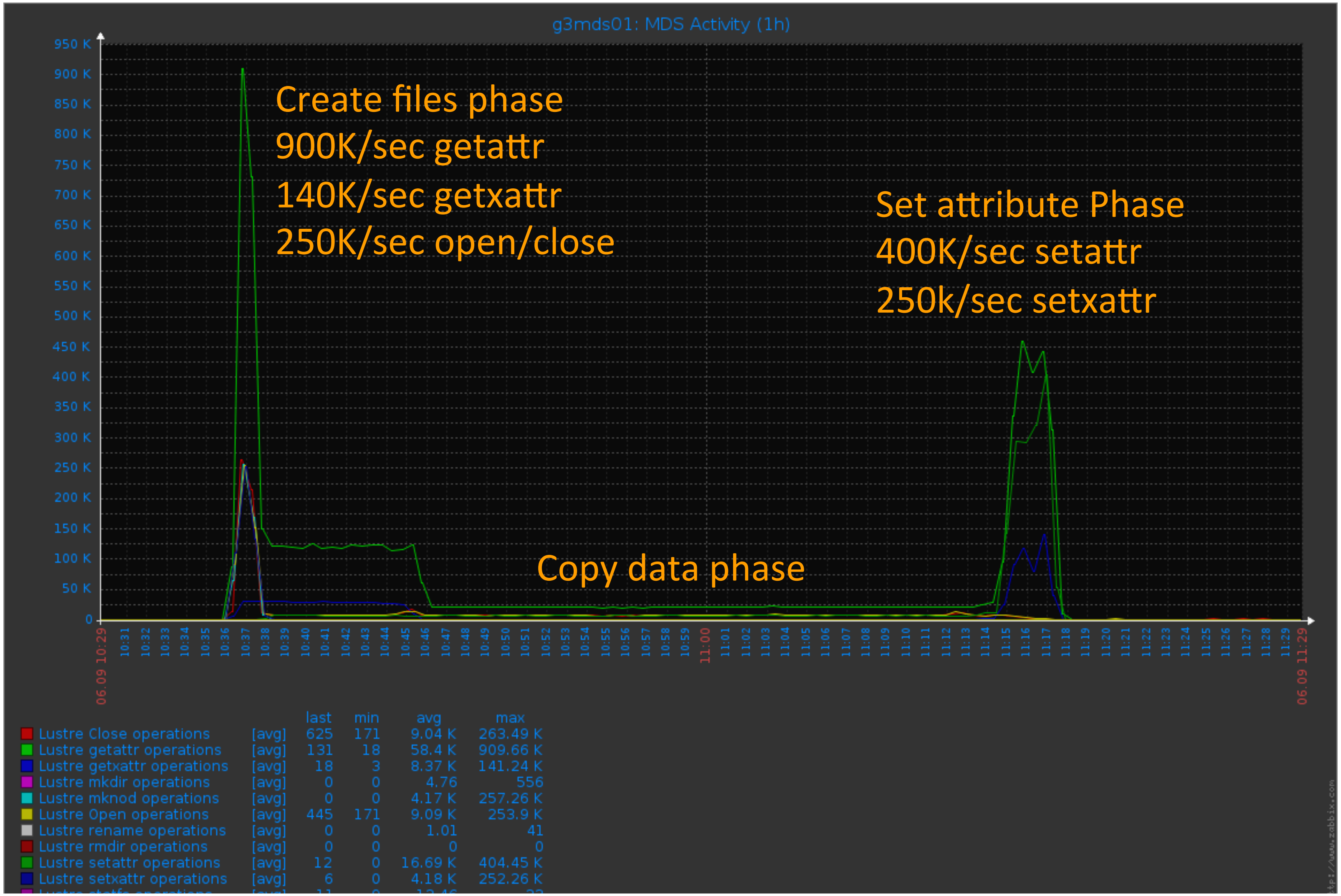
Data from history. Generated in 0.09 sec.

Source Filesystem:
Gdata1 – 26.55GB/sec Read peak

Destination Filesystem:
Gdata3 – 19.39GB/sec Write peak

Large Scale Test (50T, 500 000 files)

– **Large Scale Test –** MDS activity



Create files phase
900K/sec getattr
140K/sec getxattr
250K/sec open/close

Set attribute Phase
400K/sec setattr
250k/sec setxattr

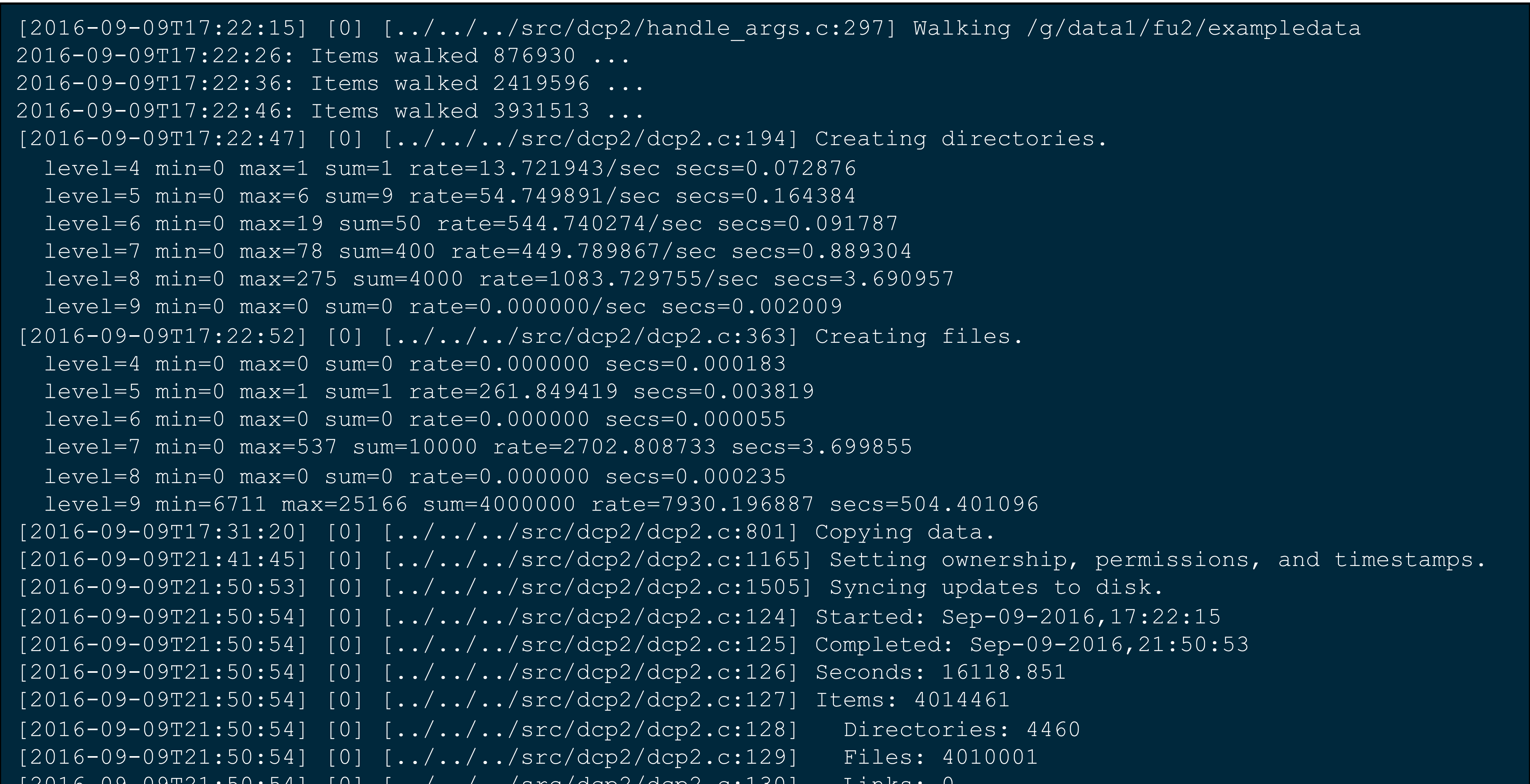
Copy data phase

		last	min	avg	max
🟥 Lustre Close operations	[avg]	625	171	9.04 K	263.49 K
🟩 Lustre getattr operations	[avg]	131	18	58.4 K	909.66 K
🟦 Lustre getxattr operations	[avg]	18	3	8.37 K	141.24 K
🟪 Lustre mkdir operations	[avg]	0	0	4.76	556
🟦 Lustre mknod operations	[avg]	0	0	4.17 K	257.26 K
🟨 Lustre Open operations	[avg]	445	171	9.09 K	253.9 K
⬜ Lustre rename operations	[avg]	0	0	1.01	41
🟥 Lustre rmdir operations	[avg]	0	0	0	0
🟩 Lustre setattr operations	[avg]	12	0	16.69 K	404.45 K
🟦 Lustre setxattr operations	[avg]	6	0	4.18 K	252.26 K
🟪 Lustre statfs operations	[avg]	11	0	12.46	22

XLarge Scale Test (400T, 4 000 000 files)

– XLarge Scale Test – dcp, 32 nodes

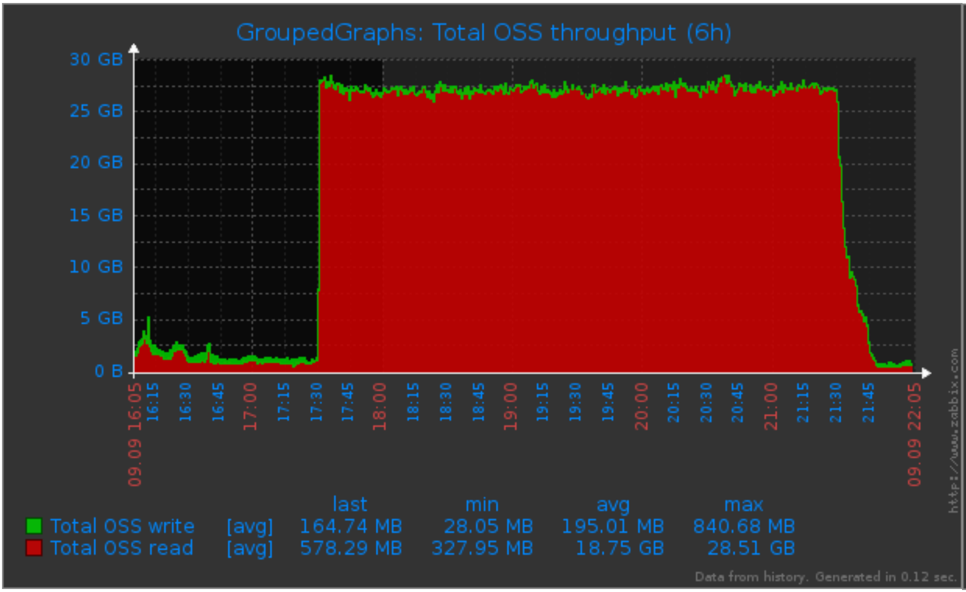
- 32 Nodes, 512 Processes
- 400T, 4 000 000 files copied
- 4 hours, 28 minutes.



```
[2016-09-09T17:22:15] [0] [../../../src/dcp2/handle_args.c:297] Walking /g/data1/fu2/exampledata
2016-09-09T17:22:26: Items walked 876930 ...
2016-09-09T17:22:36: Items walked 2419596 ...
2016-09-09T17:22:46: Items walked 3931513 ...
[2016-09-09T17:22:47] [0] [../../../src/dcp2/dcp2.c:194] Creating directories.
  level=4 min=0 max=1 sum=1 rate=13.721943/sec secs=0.072876
  level=5 min=0 max=6 sum=9 rate=54.749891/sec secs=0.164384
  level=6 min=0 max=19 sum=50 rate=544.740274/sec secs=0.091787
  level=7 min=0 max=78 sum=400 rate=449.789867/sec secs=0.889304
  level=8 min=0 max=275 sum=4000 rate=1083.729755/sec secs=3.690957
  level=9 min=0 max=0 sum=0 rate=0.000000/sec secs=0.002009
[2016-09-09T17:22:52] [0] [../../../src/dcp2/dcp2.c:363] Creating files.
  level=4 min=0 max=0 sum=0 rate=0.000000 secs=0.000183
  level=5 min=0 max=1 sum=1 rate=261.849419 secs=0.003819
  level=6 min=0 max=0 sum=0 rate=0.000000 secs=0.000055
  level=7 min=0 max=537 sum=10000 rate=2702.808733 secs=3.699855
  level=8 min=0 max=0 sum=0 rate=0.000000 secs=0.000235
  level=9 min=6711 max=25166 sum=4000000 rate=7930.196887 secs=504.401096
[2016-09-09T17:31:20] [0] [../../../src/dcp2/dcp2.c:801] Copying data.
[2016-09-09T21:41:45] [0] [../../../src/dcp2/dcp2.c:1165] Setting ownership, permissions, and timestamps.
[2016-09-09T21:50:53] [0] [../../../src/dcp2/dcp2.c:1505] Syncing updates to disk.
[2016-09-09T21:50:54] [0] [../../../src/dcp2/dcp2.c:124] Started: Sep-09-2016,17:22:15
[2016-09-09T21:50:54] [0] [../../../src/dcp2/dcp2.c:125] Completed: Sep-09-2016,21:50:53
[2016-09-09T21:50:54] [0] [../../../src/dcp2/dcp2.c:126] Seconds: 16118.851
[2016-09-09T21:50:54] [0] [../../../src/dcp2/dcp2.c:127] Items: 4014461
[2016-09-09T21:50:54] [0] [../../../src/dcp2/dcp2.c:128]   Directories: 4460
[2016-09-09T21:50:54] [0] [../../../src/dcp2/dcp2.c:129]   Files: 4010001
[2016-09-09T21:50:54] [0] [../../../src/dcp2/dcp2.c:130]   Links: 0
```

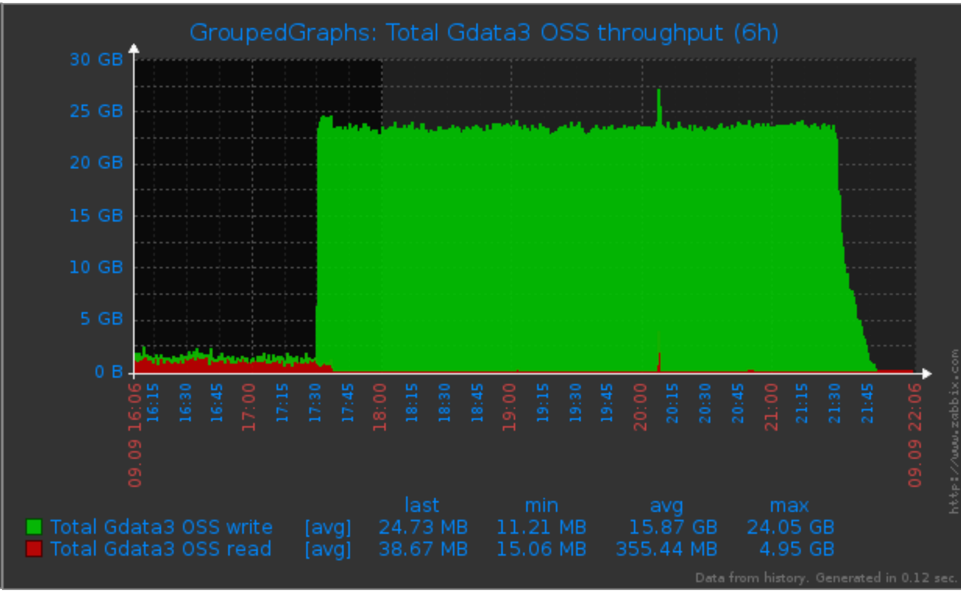
XLarge Scale Test (400T, 4 000 000 files)

- **XLarge Scale Test – dcp, 32 nodes**
- OSS Activity


GroupedGraphs: Total OSS throughput (6h)

		last	min	avg	max
🟩 Total OSS write	[avg]	164.74 MB	28.05 MB	195.01 MB	840.68 MB
🟥 Total OSS read	[avg]	578.29 MB	327.95 MB	18.75 GB	28.51 GB

Data from history. Generated in 0.12 sec.


GroupedGraphs: Total Gdata3 OSS throughput (6h)

		last	min	avg	max
🟩 Total Gdata3 OSS write	[avg]	24.73 MB	11.21 MB	15.87 GB	24.05 GB
🟥 Total Gdata3 OSS read	[avg]	38.67 MB	15.06 MB	355.44 MB	4.95 GB

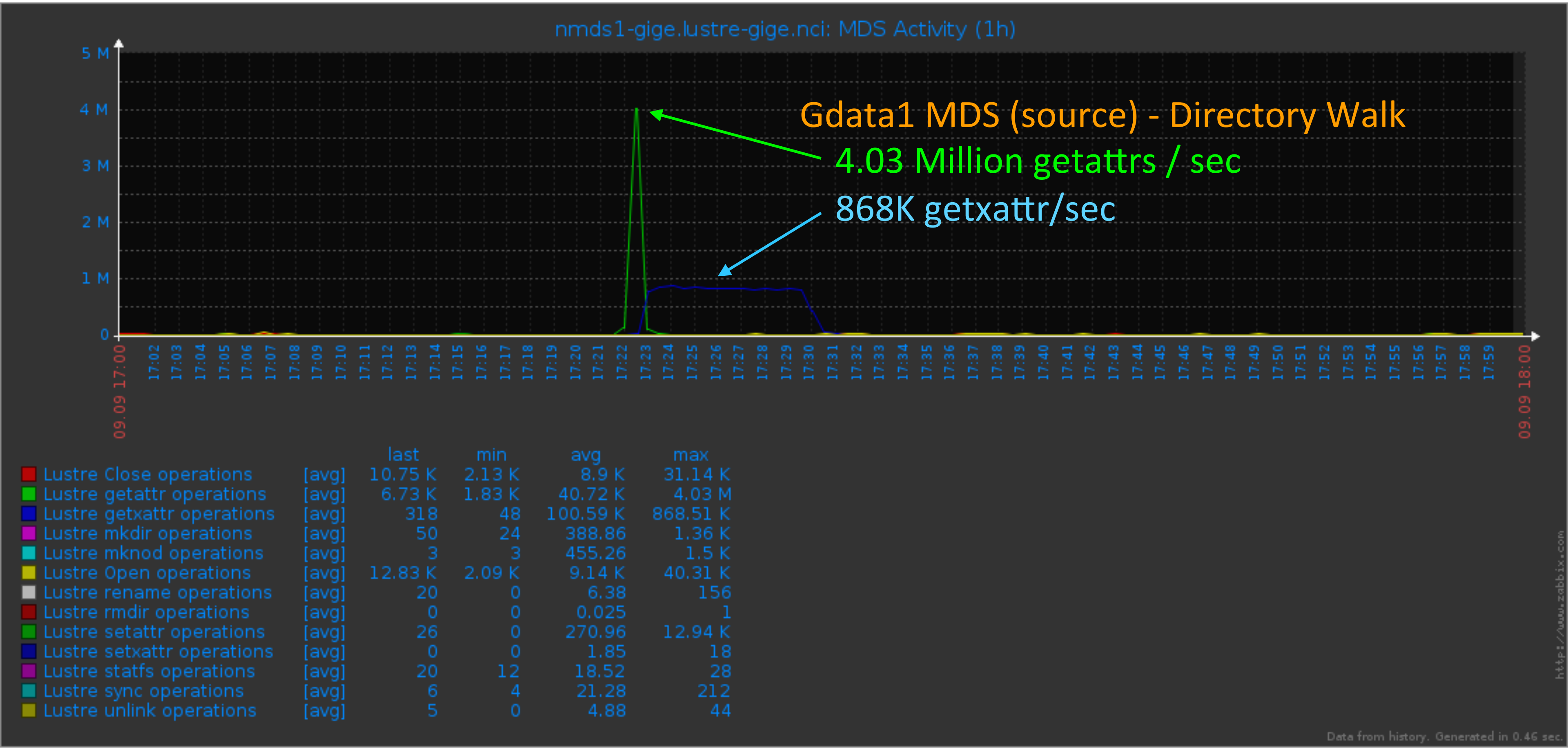
Data from history. Generated in 0.12 sec.

Source Filesystem:
Gdata1 – 28.51GB/sec Read peak

Destination Filesystem:
Gdata3 – 24.05GB/sec Write peak

XLarge Scale Test (400T, 4 000 000 files)

- **XLarge Scale Test –** MDS activity



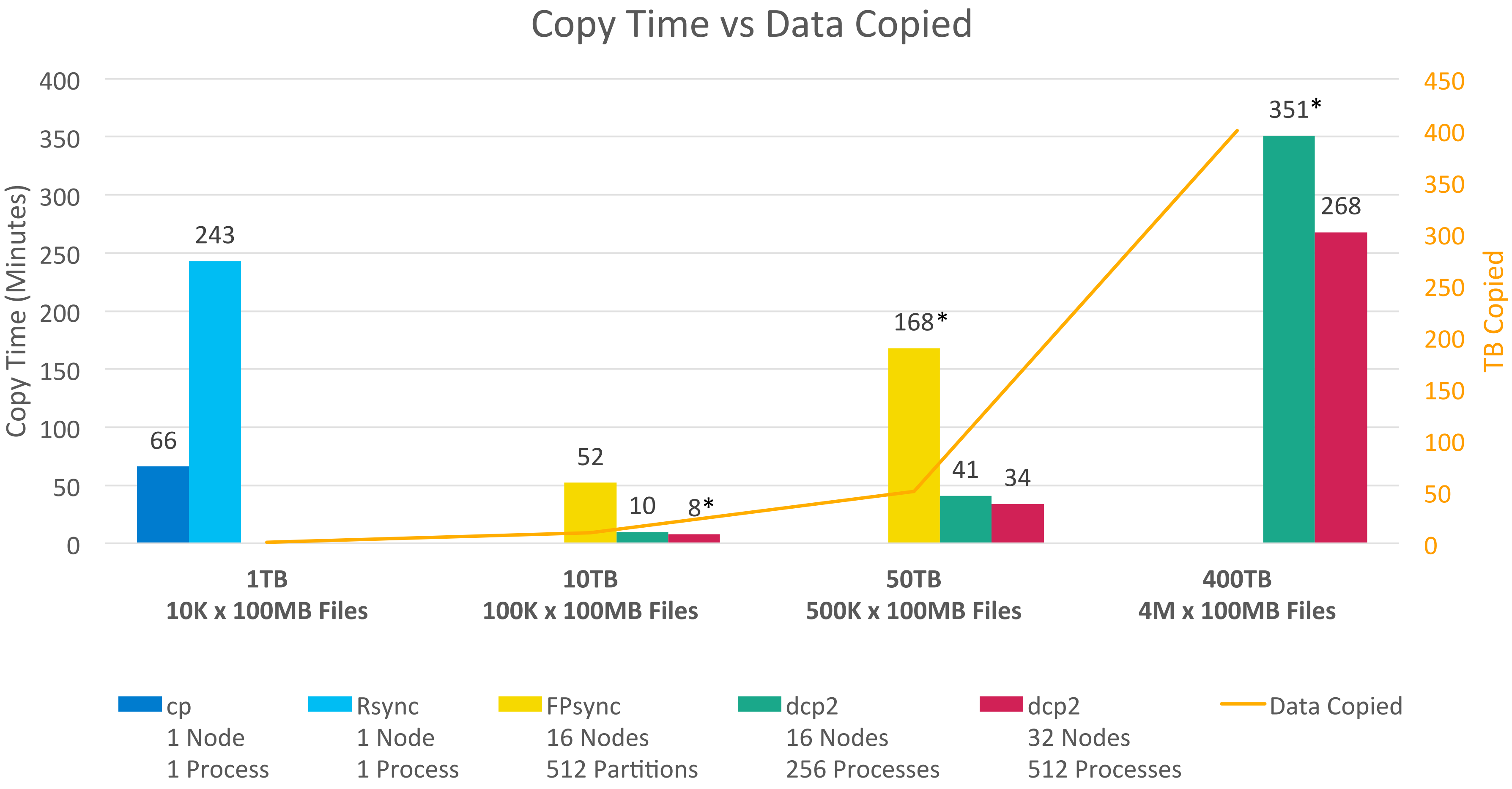
Gdata1 MDS (source) - Directory Walk

4.03 Million getattrs / sec

868K getxattr/sec

		last	min	avg	max
🟥 Lustre Close operations	[avg]	10.75 K	2.13 K	8.9 K	31.14 K
🟩 Lustre getattr operations	[avg]	6.73 K	1.83 K	40.72 K	4.03 M
🟦 Lustre getxattr operations	[avg]	318	48	100.59 K	868.51 K
🟪 Lustre mkdir operations	[avg]	50	24	388.86	1.36 K
🟦 Lustre mknod operations	[avg]	3	3	455.26	1.5 K
🟨 Lustre Open operations	[avg]	12.83 K	2.09 K	9.14 K	40.31 K
⬜ Lustre rename operations	[avg]	20	0	6.38	156
🟥 Lustre rmdir operations	[avg]	0	0	0.025	1
🟩 Lustre setattr operations	[avg]	26	0	270.96	12.94 K
🟦 Lustre setxattr operations	[avg]	0	0	1.85	18
🟪 Lustre statfs operations	[avg]	20	12	18.52	28
🟦 Lustre sync operations	[avg]	6	4	21.28	212
🟨 Lustre unlink operations	[avg]	5	0	4.88	44

– Results Comparison



* Indicates additional tests not included in presentation for comparison purposes

NCI Approach to Data Migration

Data Migration Process

Data Migration Process

- **NCI approach for a typical project level data migration**
  - Set Quota on Destination Filesystem
  - Stop NFS exports for project directory being relocated
  - Move project into restricted access source migration directory
    - Root only accessible directory with obvious name

    -
      ```
      drwx------    5 root    root           4096 Mar 10 02:53 migration_in_progress_g1_to_g3
      ```

    -
      ```
      mv /g/data1/projectID /g/data1/migration_in_progress_g1_to_g3
      ```

  - Target directory is similarly configured – root access only

  - Break up project contents (usually on first level subdir) into smaller dcp runs

  - Compare / Checksum source and destination
    - Build a list of all files in both source and dest using find with printf, Combine lists, awk | sed | sort. Diff output
    - Run fpart and feed NCI custom built MPImd5sum tool.

  - Correct any mismatched files – create a filelist with paths, split the list, use rsync. Typically none or very few files need re-sync.

  - Move data out of restricted target directory on destination filesystem
  - Re-establish NFS exports

Data Migration Process

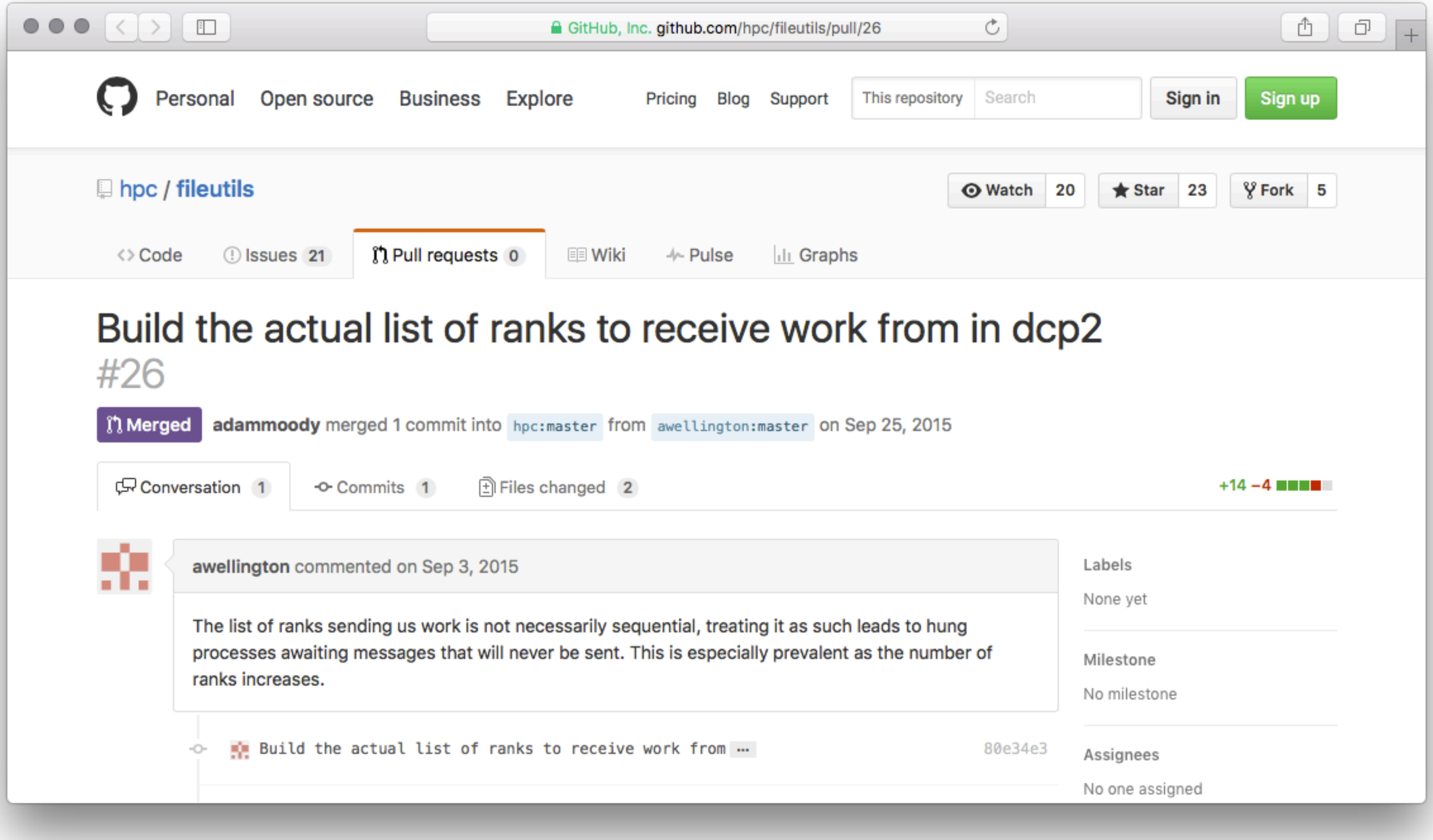
– Data Migration in Practice - example

- September 2015
- 2.4PB Project, migrated from Gdata1 to Gdata3
- Duration includes all stages (Data Copy and Verify)

- Start: 1800 Tues 8 Sept 2015
- End: 1330 Thurs 10 Sept 2015
- Duration: 43h 30m
- Data Copied: 2473TB
- Items: 39255806 (39.26 Million)

Issues & Future work

- Encountered a bug in dcp2, fix committed upstream



Build the actual list of ranks to receive work from in dcp2 #26

Merged adammoody merged 1 commit into `hpc:master` from `awellington:master` on Sep 25, 2015

awellington commented on Sep 3, 2015

The list of ranks sending us work is not necessarily sequential, treating it as such leads to hung processes awaiting messages that will never be sent. This is especially prevalent as the number of ranks increases.

`Build the actual list of ranks to receive work from …` 80e34e3

Labels
None yet

Milestone
No milestone

Assignees
No one assigned

Issues & Future work

HSM Integration

- If the data is part of a Lustre HSM system (dual state), how do we avoid re-writes on a shared tape system.
- If the data is migrating offline (tape resident), how do we avoid recalling all data from tape.
- Need to test both scenarios.

Improve scalability of Validation Processes

- Test dcmp (distributed compare) as part of the fileutils.io suite

Build dedicated Migration /Filesystem Load Test Cluster

- Gdata1 will be decommissioned late 2016 – early 2017.
- Reuse 44x OSSes from gdata1 when decommissioned
- OSSes are Dell R620 Dual E5-2620, 6C, 2.0GHz, 256GB RAM, FDR Interconnect
- Rebuild as IO Load Test and Migration Cluster

Questions ?

NCI

Providing Australian researchers with
world-class computing services

NCI Contacts

General enquiries: +61 2 6125 9800
Media enquiries: +61 2 6125 4389
Support: help@nci.org.au

Address:

NCI, Building 143, Ward Road
The Australian National University
Canberra ACT 2601
Australia

NCRIS
National Research
Infrastructure for Australia
An Australian Government Initiative

Australian Government
Bureau of Meteorology

Australian Government
Geoscience Australia

Australian Government
Australian Research Council

CSIRO

Australian
National
University

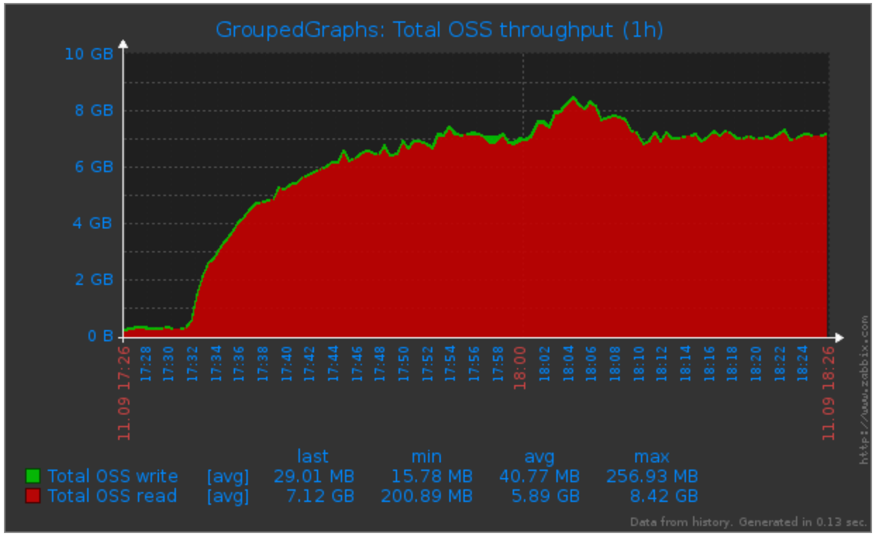
nci.org.au

@NCInews

Extra Slides

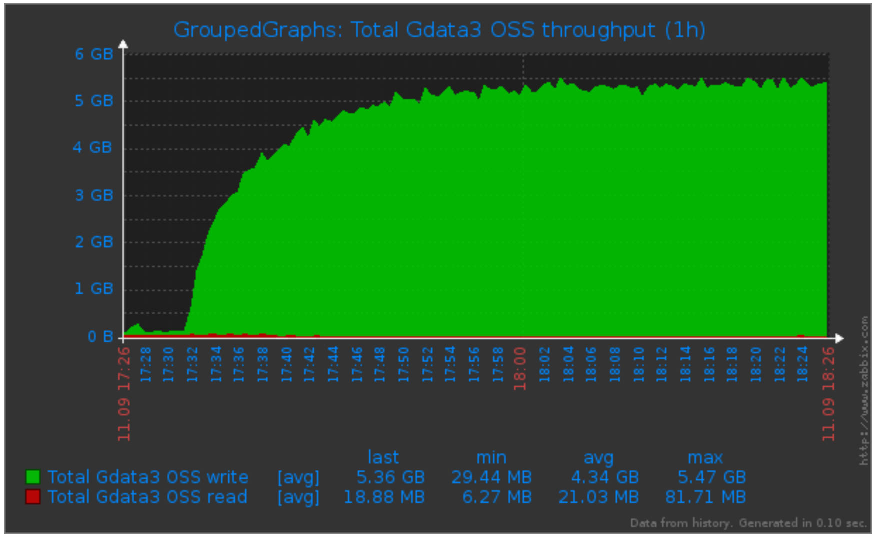
FPsync slow ramp

- **FPsync has a gradual (gentle) ramp. 16 node, 50T, 500 000 files**
- Sunday evening, very little background IO. Fpsync achieving 5.5GB/sec.



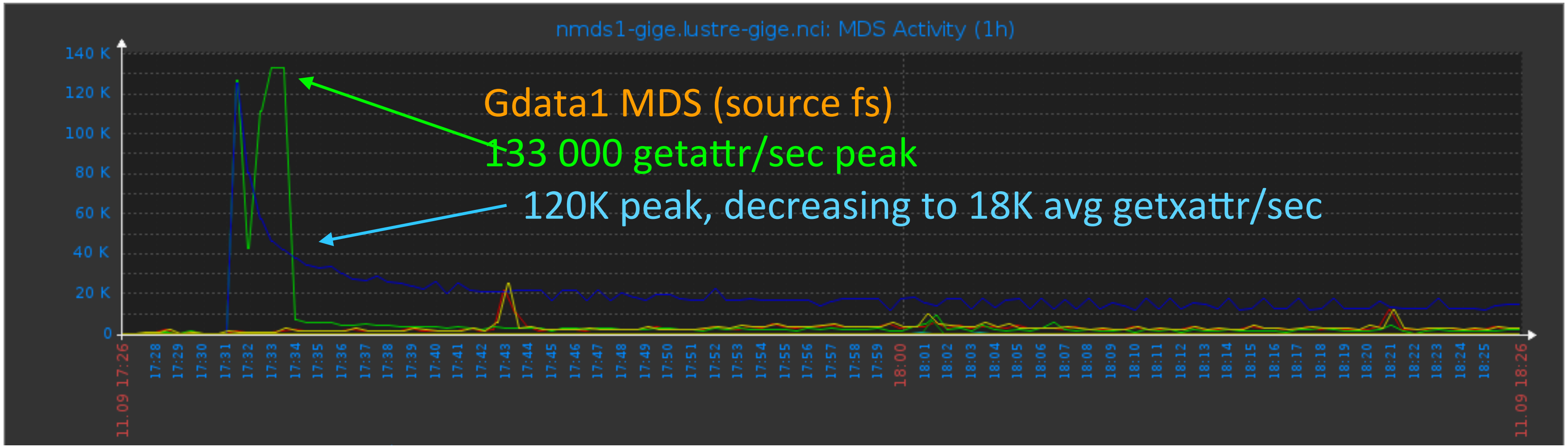
		last	min	avg	max
🟩 Total OSS write	[avg]	29.01 MB	15.78 MB	40.77 MB	256.93 MB
🟥 Total OSS read	[avg]	7.12 GB	200.89 MB	5.89 GB	8.42 GB

Data from history. Generated in 0.13 sec.



		last	min	avg	max
🟩 Total Gdata3 OSS write	[avg]	5.36 GB	29.44 MB	4.34 GB	5.47 GB
🟥 Total Gdata3 OSS read	[avg]	18.88 MB	6.27 MB	21.03 MB	81.71 MB

Data from history. Generated in 0.10 sec.



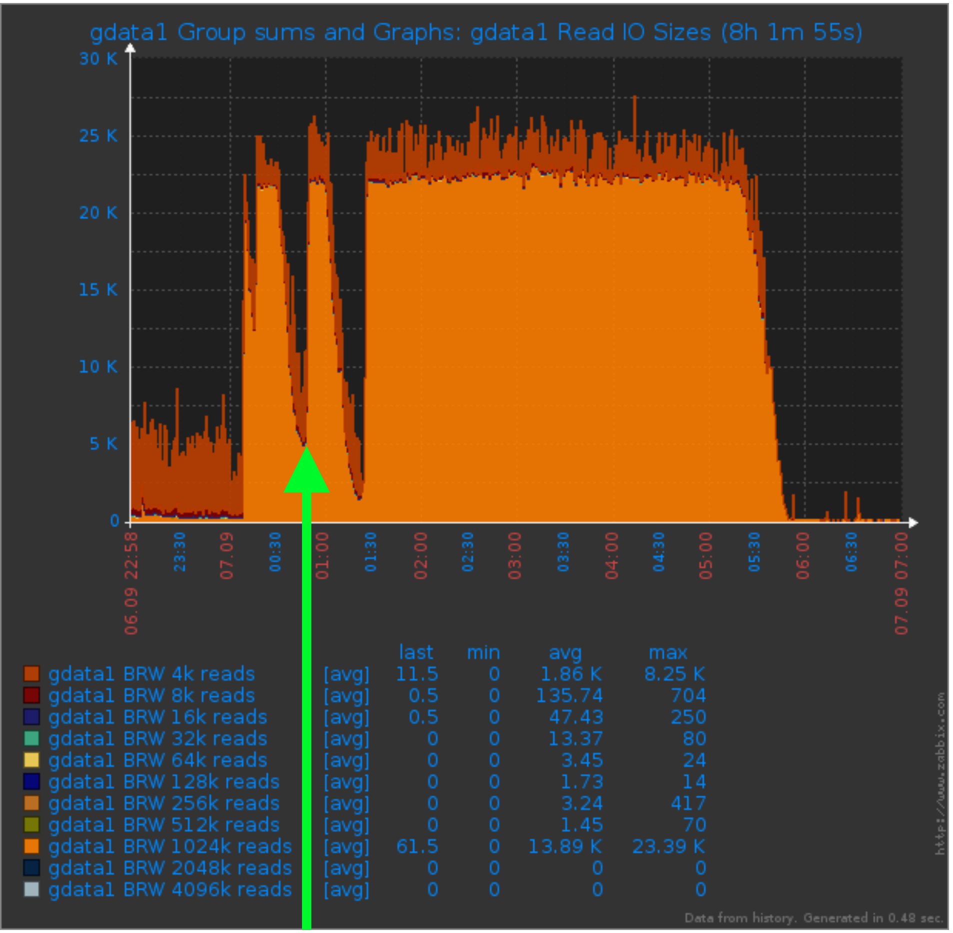
Gdata1 MDS (source fs)

133 000 getattr/sec peak

120K peak, decreasing to 18K avg getxattr/sec

Filesystem Stall

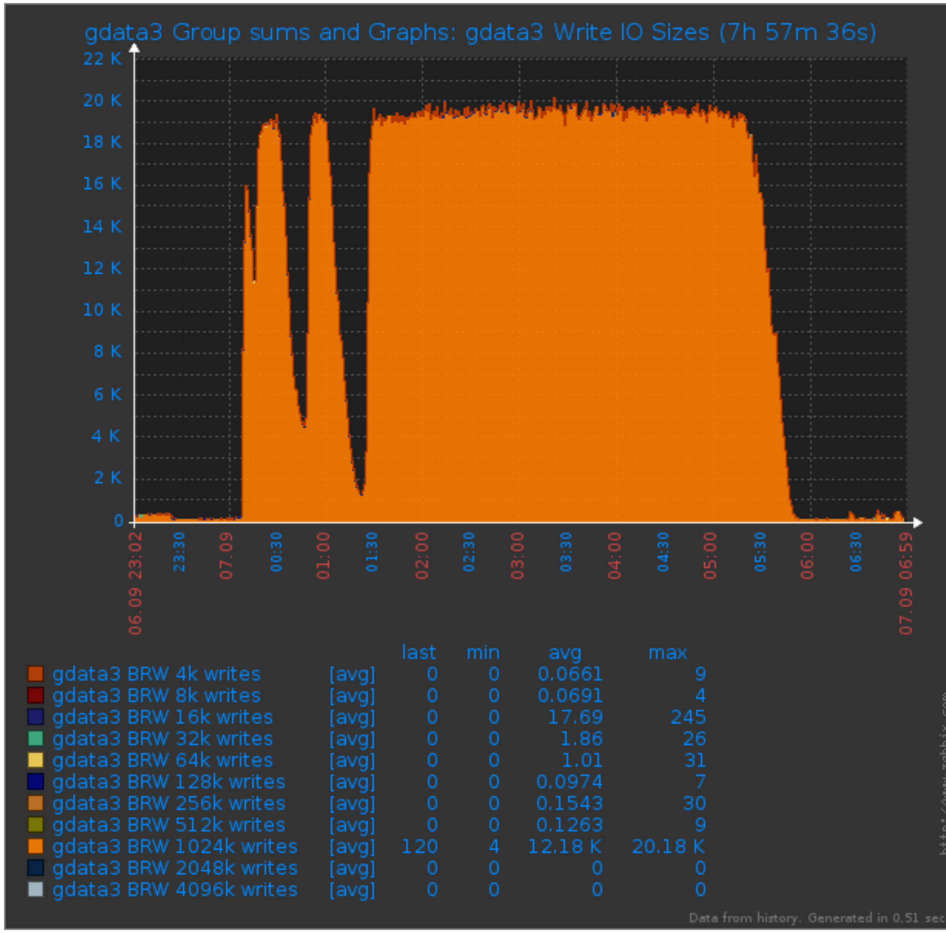
- **Large Scale Test – dcp, 16 node, 400T, 4Mill. files -** BRW Sizes
- 1x OSS in Gdata1 is in heavy IO wait, all processes affected.



gdata1 Group sums and Graphs: gdata1 Read IO Sizes (8h 1m 55s)

		last	min	avg	max
🟧 gdata1 BRW 4k reads	[avg]	11.5	0	1.86 K	8.25 K
🟥 gdata1 BRW 8k reads	[avg]	0.5	0	135.74	704
🟦 gdata1 BRW 16k reads	[avg]	0.5	0	47.43	250
🟩 gdata1 BRW 32k reads	[avg]	0	0	13.37	80
🟨 gdata1 BRW 64k reads	[avg]	0	0	3.45	24
🟦 gdata1 BRW 128k reads	[avg]	0	0	1.73	14
🟫 gdata1 BRW 256k reads	[avg]	0	0	3.24	417
🟫 gdata1 BRW 512k reads	[avg]	0	0	1.45	70
🟧 gdata1 BRW 1024k reads	[avg]	61.5	0	13.89 K	23.39 K
⬛ gdata1 BRW 2048k reads	[avg]	0	0	0	0
⬜ gdata1 BRW 4096k reads	[avg]	0	0	0	0

Data from history. Generated in 0.48 sec.



gdata3 Group sums and Graphs: gdata3 Write IO Sizes (7h 57m 36s)

		last	min	avg	max
🟧 gdata3 BRW 4k writes	[avg]	0	0	0.0661	9
🟥 gdata3 BRW 8k writes	[avg]	0	0	0.0691	4
🟦 gdata3 BRW 16k writes	[avg]	0	0	17.69	245
🟩 gdata3 BRW 32k writes	[avg]	0	0	1.86	26
🟨 gdata3 BRW 64k writes	[avg]	0	0	1.01	31
🟦 gdata3 BRW 128k writes	[avg]	0	0	0.0974	7
🟫 gdata3 BRW 256k writes	[avg]	0	0	0.1543	30
🟫 gdata3 BRW 512k writes	[avg]	0	0	0.1263	9
🟧 gdata3 BRW 1024k writes	[avg]	120	4	12.18 K	20.18 K
⬛ gdata3 BRW 2048k writes	[avg]	0	0	0	0
⬜ gdata3 BRW 4096k writes	[avg]	0	0	0	0

Data from history. Generated in 0.51 sec.

Zabbix event ID 960602: Wed 7 Sept 2016 - 00:38

Trigger: Disk I/O is overloaded on noss62-gige.lustre-gige.nci

Item values: CPU iowait time (noss62-gige.lustre-gige.nci:system.cpu.util[,iowait]): 67.66 %

Memory Errors on Node

– MCE / ECC Correctable errors

- High Correctable memory error count detected by system monitoring
- Occurred during September 2015 migration example
- Multi-petabyte migration separated into smaller dcp jobs to minimize impact of hardware failure.
- Node removed from service
- Affected job restarted to exclude node

```
compute r9: STATUS 8c000048000800c2 MCGSTATUS 0
compute r9: MCGCAP 1000c14 APICID 0 SOCKETID 0
compute r9: CPUID Vendor Intel Family 6 Model 45
compute r9: Hardware event. This is not a software error.
compute r9: MCE 4
compute r9: CPU 0 BANK 5
compute r9: MISC 21400e0e86 ADDR 17b81ad80
compute r9: TIME 1441704575 Tue Sep  8 19:29:35 2015
compute r9: MCG status:
compute r9: MCi status:
compute r9: Corrected error
compute r9: MCi_MISC register valid
compute r9: MCi_ADDR register valid
compute r9: MCA: MEMORY CONTROLLER RD_CHANNEL2_ERR
compute r9: Transaction: Memory read error
```