



Providing Australian researchers with
world-class computing services

Petascade Data Migration

Lustre User Group 2016 (Canberra)

Daniel Rodwell

Manager, Data Storage Services

September 2016



Australian Government
Department of Education



Australian
National
University



Australian Government
Bureau of Meteorology



Australian Government
Geoscience Australia



Australian Government
Australian Research Council



nci.org.au



@NCInews

- **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

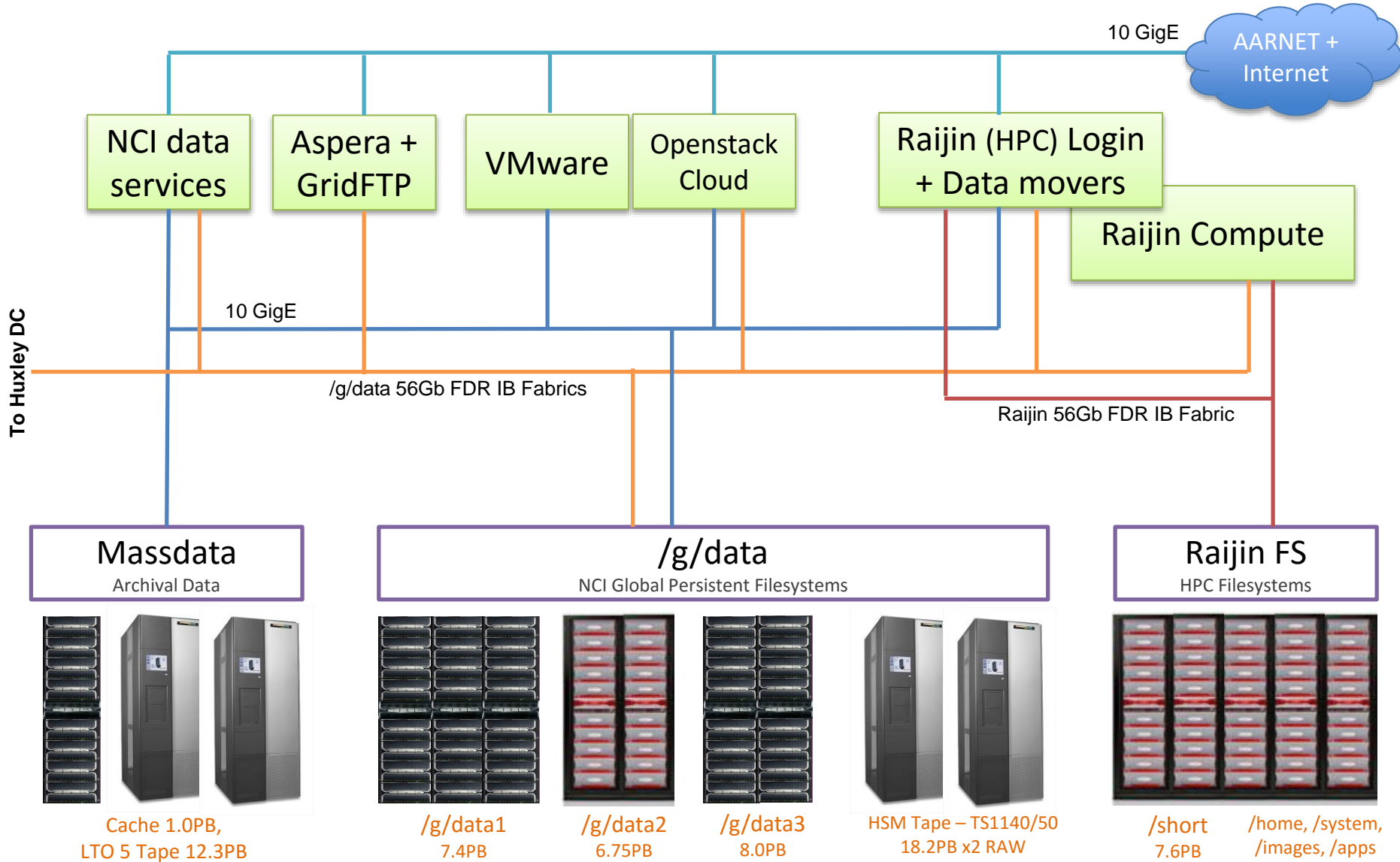
- How big?
 - Very.
 - Average data collection is 50-100+ Terabytes
 - Larger data collections are multi-Petabytes in size
 - Individual files can exceed 2TB or be as small as a few KB.
 - Individual datasets consist of tens of millions of files
 - Next Generation datasets likely to be 6-10x larger.
 - Gdata1+2+3 = 451 Million inodes stored
 - 1% of /g/data1 capacity = 74TB

- What ?
 - 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



Collection	TB Approved	TB Ready	Ingested
Skymapper (Astronomy)	227.00	140.00	62%
Australian Data Archive (Social Sciences)	4.00	3.00	75%
BPA Melanoma Dataset (Biosciences)	129.00	123.00	95%
Plant Phenomics (Biosciences)	110.00	2.00	2%
Ocean Gen. Circulation Model (Earth Simulator)	29.00	27.00	93%
Year Of Tropical Convection	41.00	41.00	100%
CABLE Global Evaluation Datasets	24.00	2.00	8%
CORDEX Int	57.00	1.00	2%
Coupled Model Intercomparison Project (CMIP5)	2,600.00	1,488.00	57%
Reanalysis	146.00	146.00	100%
ACCESS Models	2,538.00	2,099.00	83%
Seasonal Climate Prediction	595.00	369.00	62%
Australian Bathymetry and Elevation reference data	113.00	23.00	20%
Australian Marine Video and Imagery Collection	7.00	7.00	100%
Global Navigation Satellite System (GNSS) (Geodesy)	5.00	4.00	80%
Digitised Australian Aerial Survey Photography	77.00	74.00	96%
Earth Observation (Satellite: Landsat, etc)	1,486.00	1,413.00	95%
IMOS+TERN Australasian Satellite Imagery (NOAA/AVHRR, MODIS, VIIRS and AusCover)	436.00	257.00	59%
Satellite Soil Moisture Products	5.00	1.00	20%
Synthetic Aperture Radar	29.00	29.00	100%
BoM Observations	366.00	175.00	48%
BoM Ocean-Marine Collections	429.00	77.00	18%
Aust. 3D Geological Models	3.00	1.00	33%
Aust. Geophysical Data Collection	330.00	7.00	2%
Aust. Natural Hazards Archive	27.00	3.00	11%
National CT-Lab Tomographic Collection	205.00	171.00	83%
TERN eMAST	90.00	15.00	17%
TERN Phenology Monitoring: Near Surface Remote Sensing	12.00	1.00	8%
TERN eMAST Data Assimilation	110.00	9.00	8%
CSIRO/BoM Key Water Assets	44.00	18.00	41%
Models of Land/Water Dynamics from Space	22.00	11.00	50%
Totals	10,296	6,737	65%

<https://www.rdsi.edu.au/collections-stored>



- **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
 - Lustre 2.3.11 (IEEL v1)
 - **Gdata2** – Persistent Data: /g/data2
 - 6.75PB @ 65GB/Sec
 - Lustre 2.5.42.8 (IEEL v2)
 - **Gdata3** – Persistent Data: /g/data3 –
 - Stage 1: 5.7PB @ 92GB/sec
 - Stage 2: 8.0PB @ 120GB/Sec+
 - (Lustre 2.5.42.8, IEEL v2)



Why migrate data between filesystems?

Data Migration Drivers

- 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.

- 3 Years ago...
 - Vayu HPC – HPC Lustre Filesystem
 - 800TB, 25GB/Sec
 - Gdata (Original) – Persistent Lustre Filesystem on Vayu
 - 900TB, 12GB/sec
 - Projects – Dual State CXFS/DMF Filesystem
 - 1.4PB, 5GB
- Increased Persistent online storage capacity from 2.3PB to 22.3 PB in 3 years
- Migrated over 8PB, 100+ Million files between various data systems
- Need to find a solution that can scale to petabytes of data.
- Traditional approaches handle gigabytes, not petabytes.

- 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

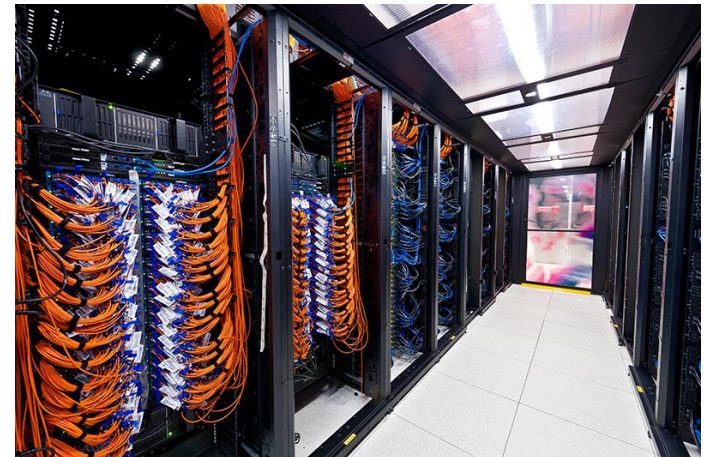
Performance Profiles

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 User jobs

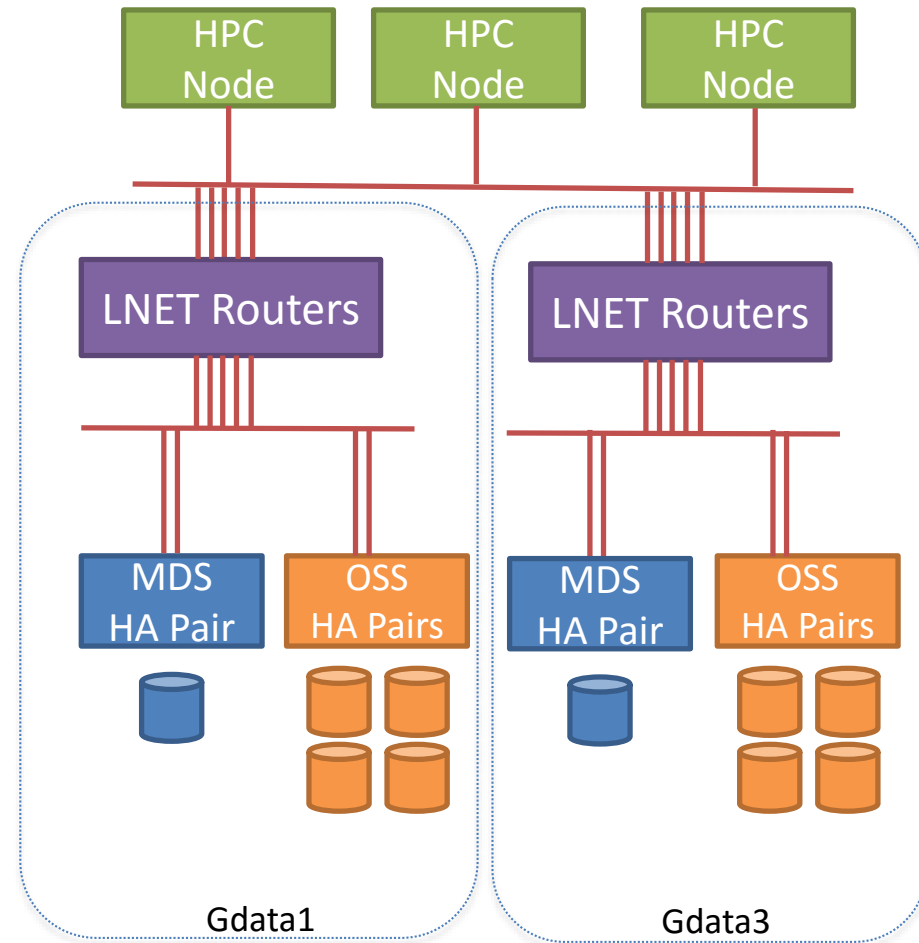


- Source

- Gdata1
- 54GB/sec peak read performance
- 520 OSTs, 300-400MB/sec each
- Lustre 2.3.11

- Destination

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





Preparing to Migrate User Project Data

Considerations & Planning

— 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

— Dry Run

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

— Run as Root

- Unless the administrator 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, as root.

— Dataset

- Determine the size and count of files before the migration, use 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

— 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.

— 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 optimised 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>
- <https://github.com/stefanv/fpsync>
- 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 optimised 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 bin size parameters for optimal performance and well balanced workload distribution

— dcp2 (distributed copy)

- <http://fileutils.io>
- <https://github.com/hpc/fileutils/blob/master/doc/markdown/dcp.1.md>
- 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 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**
 - > /Bin1 (4 Million files, 400TB)
 - > /Bin 1A (500,000 x 100MB files, 50TB)
 - > /Bin 111 (100,000 x 100MB files, 10TB)
 - > /Bin1111 (10,000 x 100MB files, 1TB)
 - > /Bin11111 (1000 x100MB files, 100GB)
 - > /Bin2 (500,000 x 100MB files, 50TB)
 - > ...
 - > ...

— 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
- Which is about the average performance we expect from a gdata1 OST.

— **Small Scale Test – Traditional rsync**

- 1TB
- 10,000 x 100MB files

- ...
- Rsync takes a long time as a single rsync `-aAXS <source> <dest>`
- Didn't finish testing in time for presentation today
- (Result will be in published slide deck)

— Medium Scale Test – fpsync, 16 nodes

- 10TB
- 100,000 x 100MB files
- Fpsync requires the size of the distribution to be passed to it as parameters.
- -n is the number of processes
- -f is the filecount bin size for each work task
- -s is the size in bytes bin size for each work task
- 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 processes = 100000 / 512 = **200** (round up)
- - s = number of bytes / # of processes = 10484019363840 / 512
 - = 20476600320 = **20480000000** (round up)

— 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/dkr900/fpsync_16_node_10T_test.out
```

— Medium Scale Test – fpsync, 16 nodes

- 16 Nodes, 512 Processes
- 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 – 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

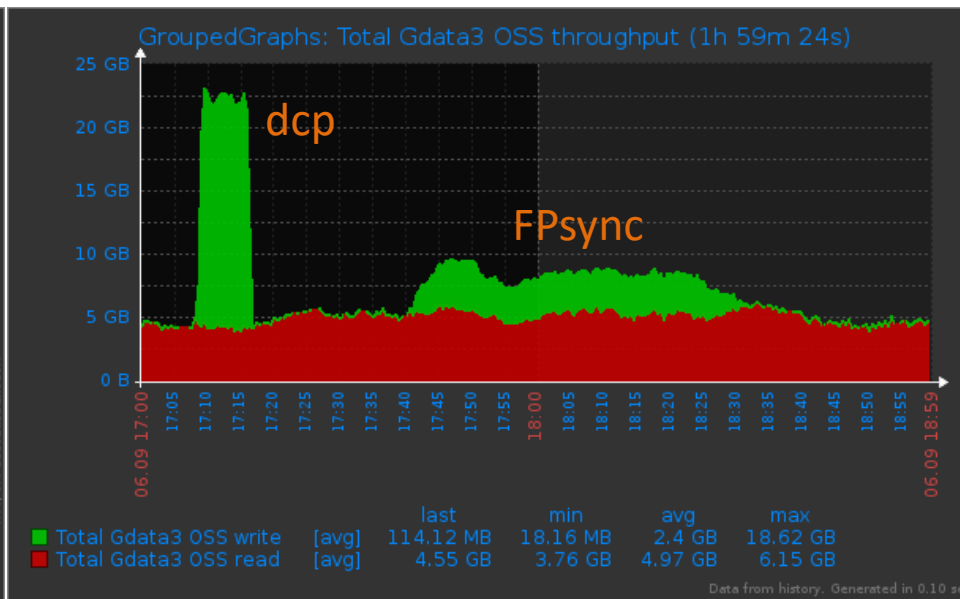
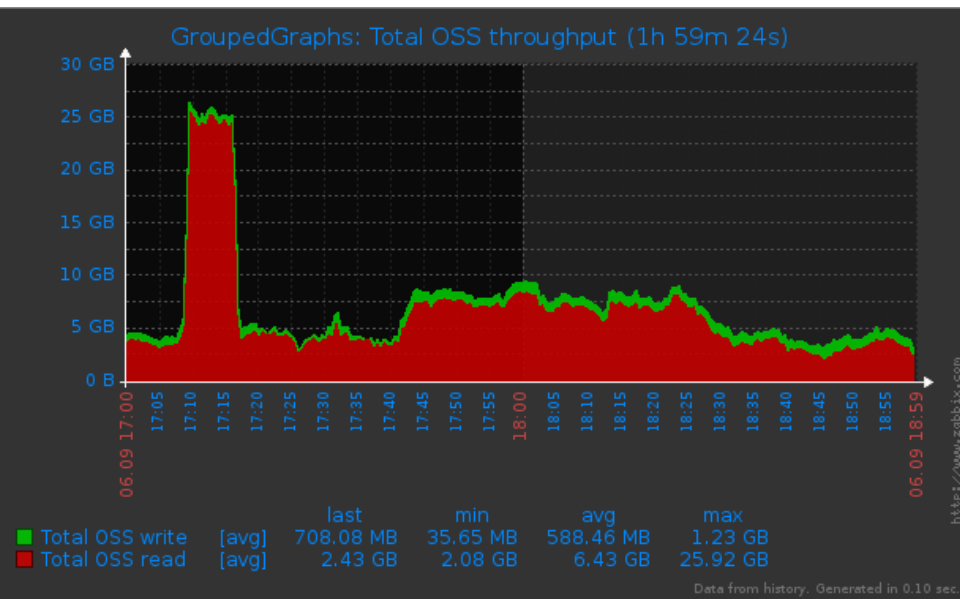
```
# module load dcp/1.0-NCI2
# mpirun --allow-run-as-root -np 256 -H
r10,r11,r12,r13,r14,r15,r16,r17,r18,r19,r20,r21,r22,r23,r24,r25
/apps/dcp/1.0-NCI2/bin/dcp2 -p /g/data1/fu2/exampledata/Bin1/Bin1A
/g/data3/fu2/exampletransfer/
```

— 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 – dcp vs fpsync, 16 nodes
- Aggregate OST throughput



Source Filesystem:
Gdata1 – 25.92GB/sec Read peak

Destination Filesystem:
Gdata3 – 18.62GB/sec Write peak

— 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 bin imbalance with fpsync if planning on a stage bulk data, then 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 – dcp, 16 nodes

- 50TB
- 500,000 x 100MB files

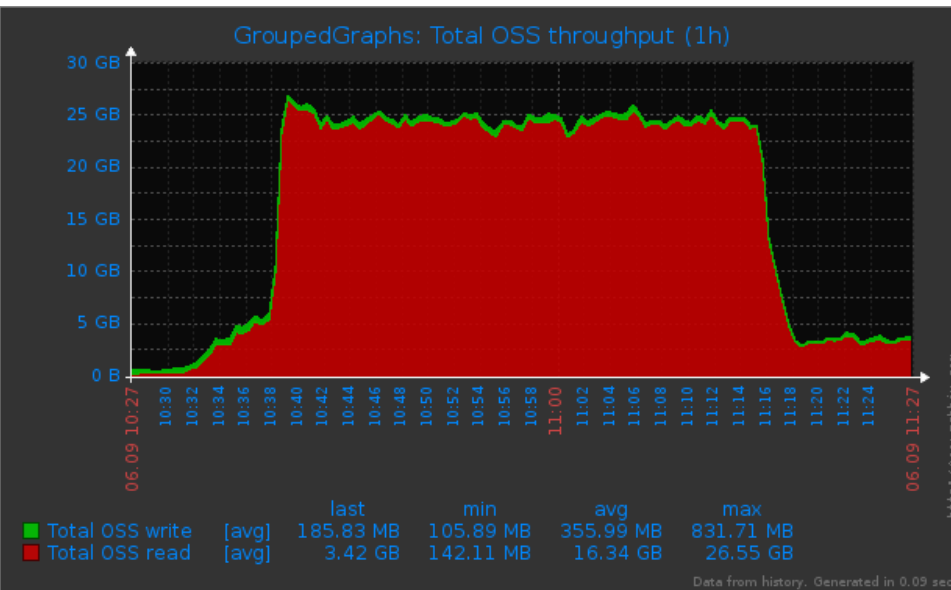
```
# module load dcp/1.0-NCI2
# mpirun --allow-run-as-root -np 256 -H
r10,r11,r12,r13,r14,r15,r16,r17,r18,r19,r20,r21,r22,r23,r24,r25
/apps/dcp/1.0-NCI2/bin/dcp2 -p /g/data1/fu2/exampledata/Bin1
/g/data3/fu2/exampletransfer/
```

— Large Scale Test – dcp, 16 nodes

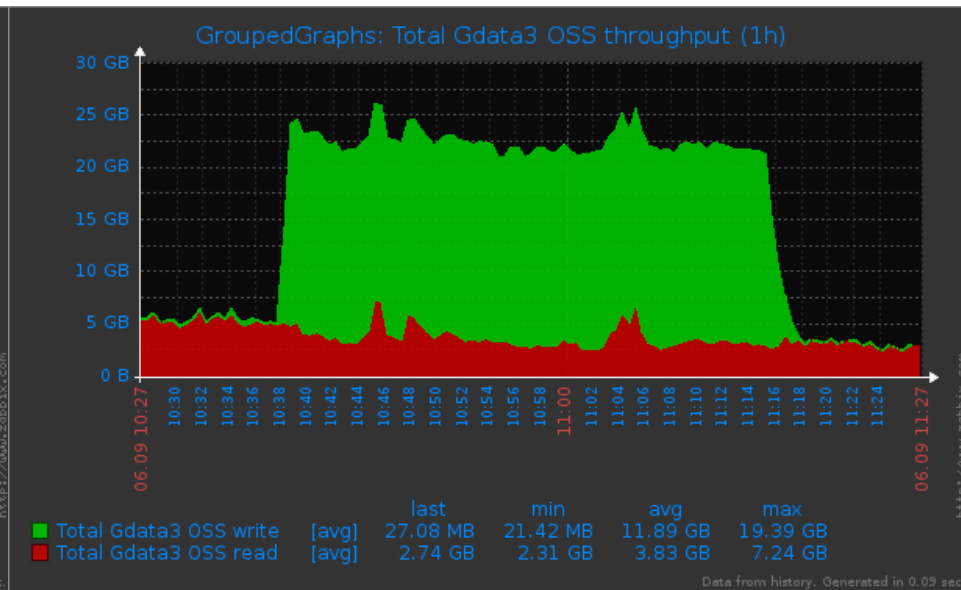
- 16 Nodes, 256 Processes
- 50T, 500000 files copied
- 41 minutes, 10 seconds.

```
[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 – dcp, 16 nodes
- OSS Activity

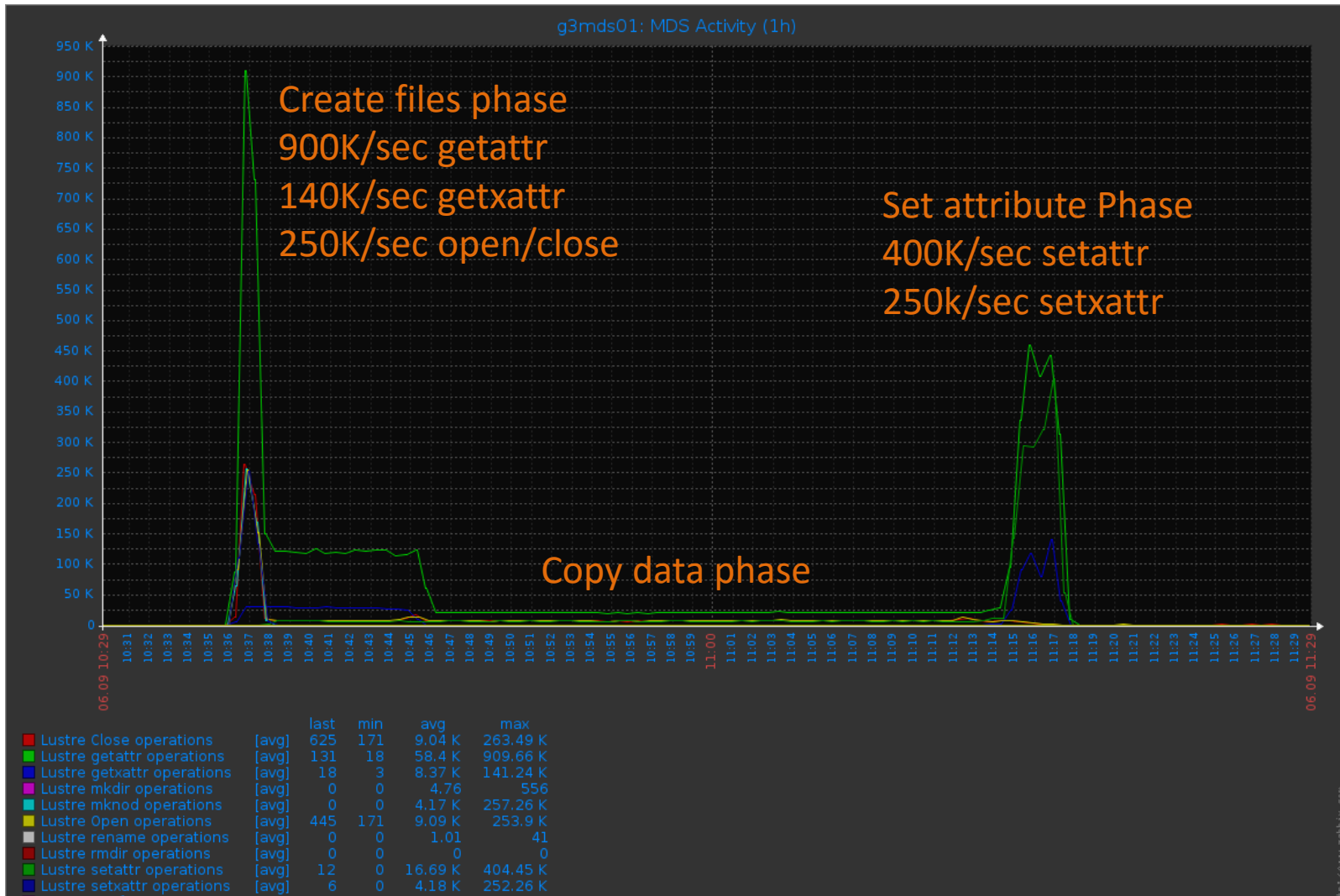


Source Filesystem:
Gdata1 – 26.55GB/sec Read peak



Destination Filesystem:
Gdata3 – 19.39GB/sec Write peak

Large Scale Test – MDS activity



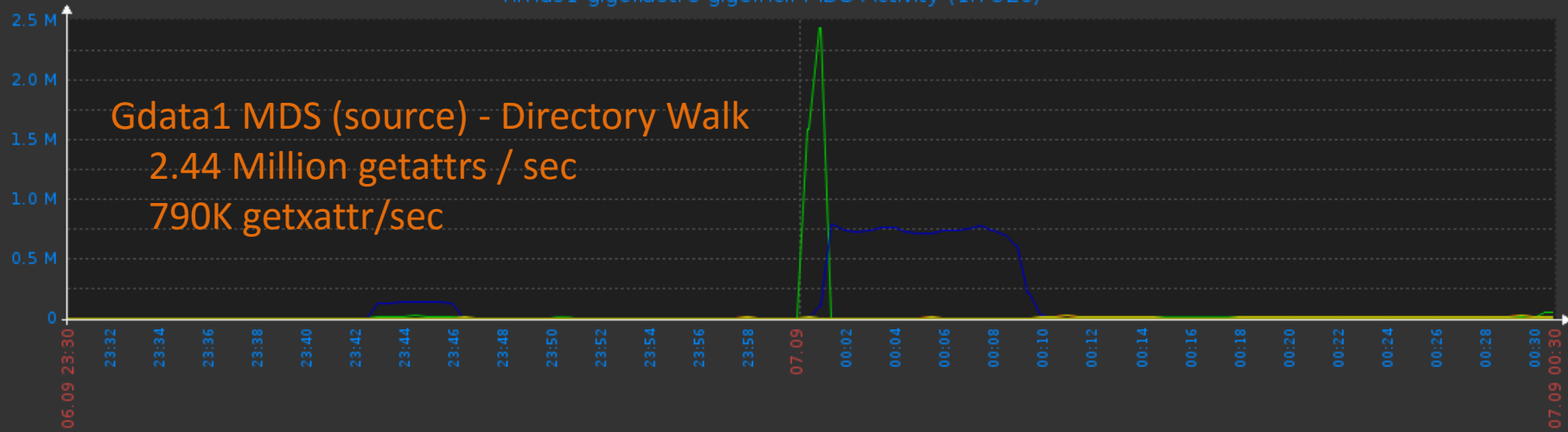
— XLarge Scale Test – dcp, 16 nodes

- 16 Nodes, 256 Processes
- 400T, 4 000 000 files copied
- 5 hours, 51 minutes.

```
[2016-09-07T00:00:25] [0] [../../../../src/dcp2/dcp2.c:1404] Preserving file attributes.
[2016-09-07T00:00:25] [0] [../../../../src/dcp2/handle_args.c:297] Walking /g/data1/fu2/exampledata
2016-09-07T00:00:36: Items walked 1130055 ...
2016-09-07T00:00:46: Items walked 2506291 ...
2016-09-07T00:00:56: Items walked 3923387 ...
[2016-09-07T00:00:57] [0] [../../../../src/dcp2/dcp2.c:194] Creating directories.
  level=4 min=0 max=1 sum=1 rate=173.096612/sec secs=0.005777
  level=5 min=0 max=6 sum=9 rate=350.658480/sec secs=0.025666
  level=6 min=0 max=20 sum=50 rate=555.353062/sec secs=0.090033
  level=7 min=0 max=71 sum=400 rate=779.842695/sec secs=0.512924
  level=8 min=0 max=252 sum=4000 rate=812.168556/sec secs=4.925086
  level=9 min=0 max=0 sum=0 rate=0.000000/sec secs=0.002318
[2016-09-07T00:01:03] [0] [../../../../src/dcp2/dcp2.c:363] Creating files.
  level=4 min=0 max=0 sum=0 rate=0.000000 secs=0.000256
  level=5 min=0 max=1 sum=1 rate=256.940946 secs=0.003892
  level=6 min=0 max=0 sum=0 rate=0.000000 secs=0.000047
  level=7 min=0 max=966 sum=10000 rate=2679.251208 secs=3.732386
  level=8 min=0 max=0 sum=0 rate=0.000000 secs=0.000206
  level=9 min=13787 max=24710 sum=4000000 rate=7596.528774 secs=526.556289
[2016-09-07T00:09:53] [0] [../../../../src/dcp2/dcp2.c:801] Copying data.
[2016-09-07T05:46:14] [0] [../../../../src/dcp2/dcp2.c:1165] Setting ownership, permissions, and timestamps.
[2016-09-07T05:52:25] [0] [../../../../src/dcp2/dcp2.c:1505] Syncing updates to disk.
[2016-09-07T05:52:26] [0] [../../../../src/dcp2/dcp2.c:124] Started: Sep-07-2016,00:00:25
[2016-09-07T05:52:26] [0] [../../../../src/dcp2/dcp2.c:125] Completed: Sep-07-2016,05:52:25
[2016-09-07T05:52:26] [0] [../../../../src/dcp2/dcp2.c:126] Seconds: 21119.868
[2016-09-07T05:52:26] [0] [../../../../src/dcp2/dcp2.c:127] Items: 4014461
[2016-09-07T05:52:26] [0] [../../../../src/dcp2/dcp2.c:128] Directories: 4460
[2016-09-07T05:52:26] [0] [../../../../src/dcp2/dcp2.c:128] Files: 4010001
```

— XLarge Scale Test – MDS activity

nmuds1-gige.lustre-gige.nci: MDS Activity (1h 32s)

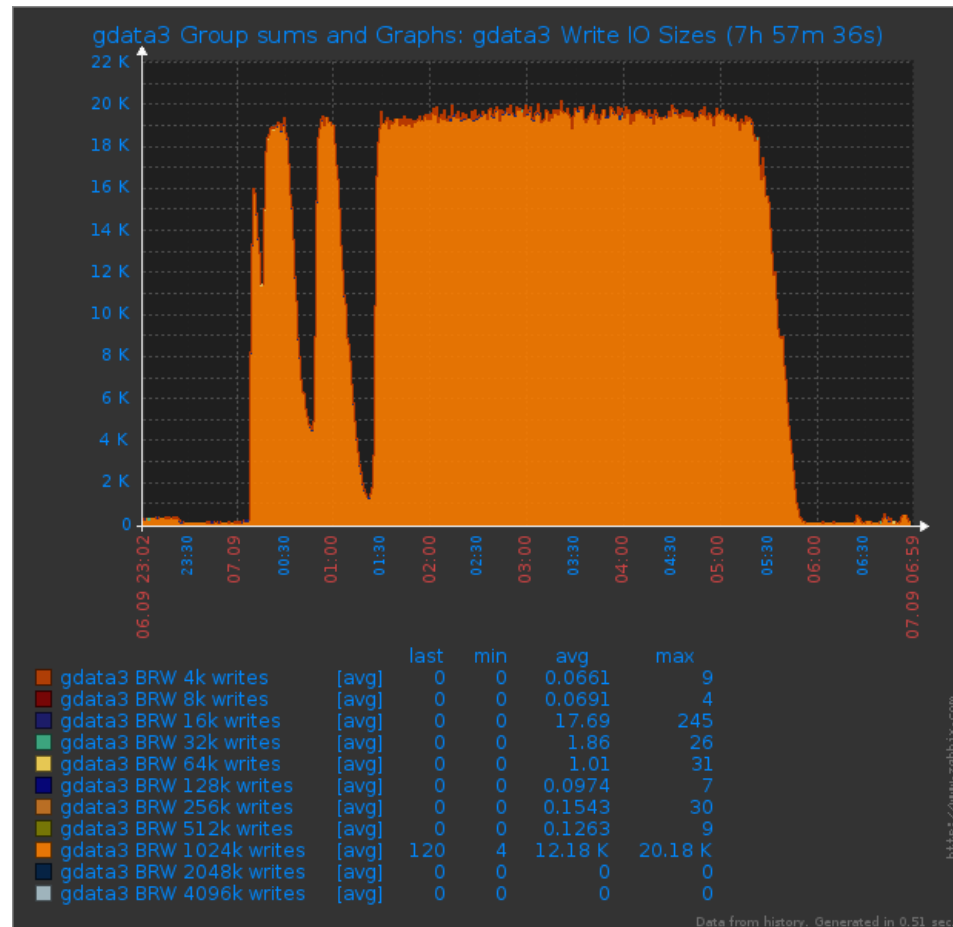
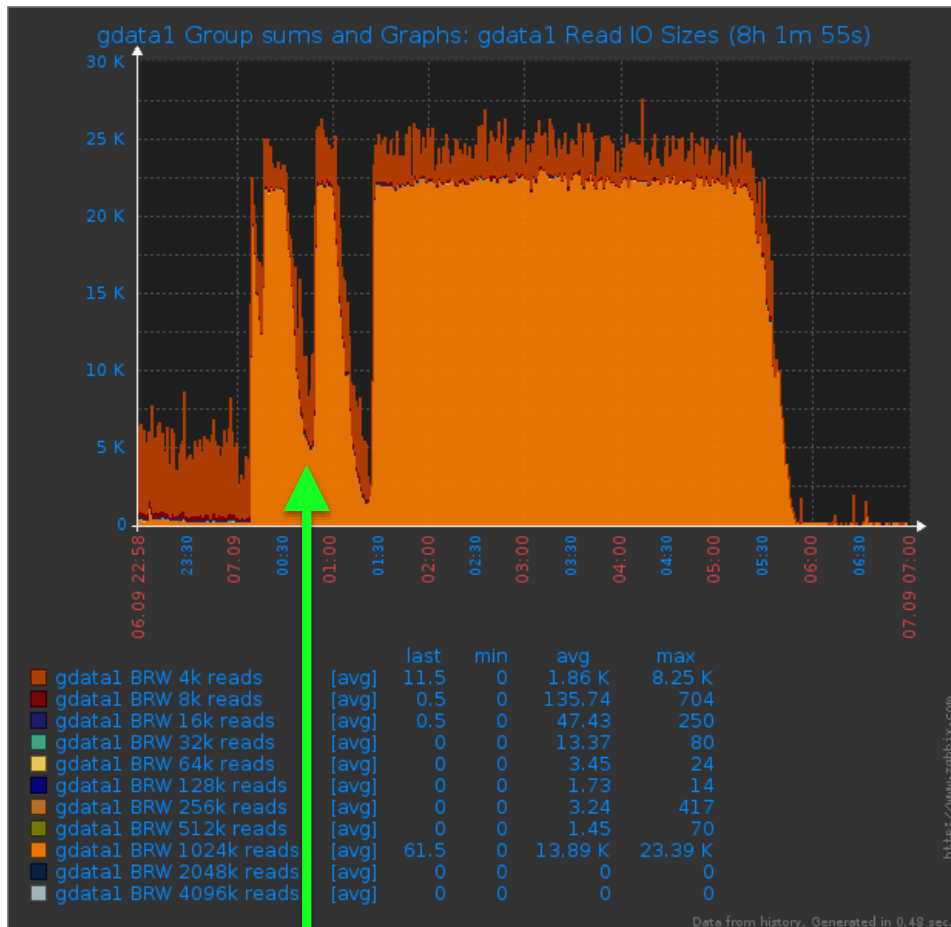


		last	min	avg	max
Lustre Close operations	[avg]	10.2 K	1.37 K	5.17 K	29.1 K
Lustre getattr operations	[avg]	51.52 K	203	36.74 K	2.44 M
Lustre getxattr operations	[avg]	399	0	107.15 K	792.73 K
Lustre mkdir operations	[avg]	0	0	3.55	78
Lustre mknod operations	[avg]	13	1	15.43	87
Lustre Open operations	[avg]	10.58 K	1.42 K	5.55 K	29.58 K
Lustre rename operations	[avg]	12	0	11.89	144
Lustre rmdir operations	[avg]	0	0	0.2562	7
Lustre setattr operations	[avg]	17	3	760.15	12.6 K
Lustre setxattr operations	[avg]	10	1	8.3	28
Lustre statfs operations	[avg]	14	10	15.86	27
Lustre sync operations	[avg]	51	13	67.31	566
Lustre unlink operations	[avg]	19	3	20.23	235

Data from history. Generated in 0.44 sec.

http://www.zabbix.com

Large Scale Test – dcp, 16 node - BRW Sizes



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 %



NCI Approach to Data Migration

Data Migration Process

— NCI approach for Project level data Migration

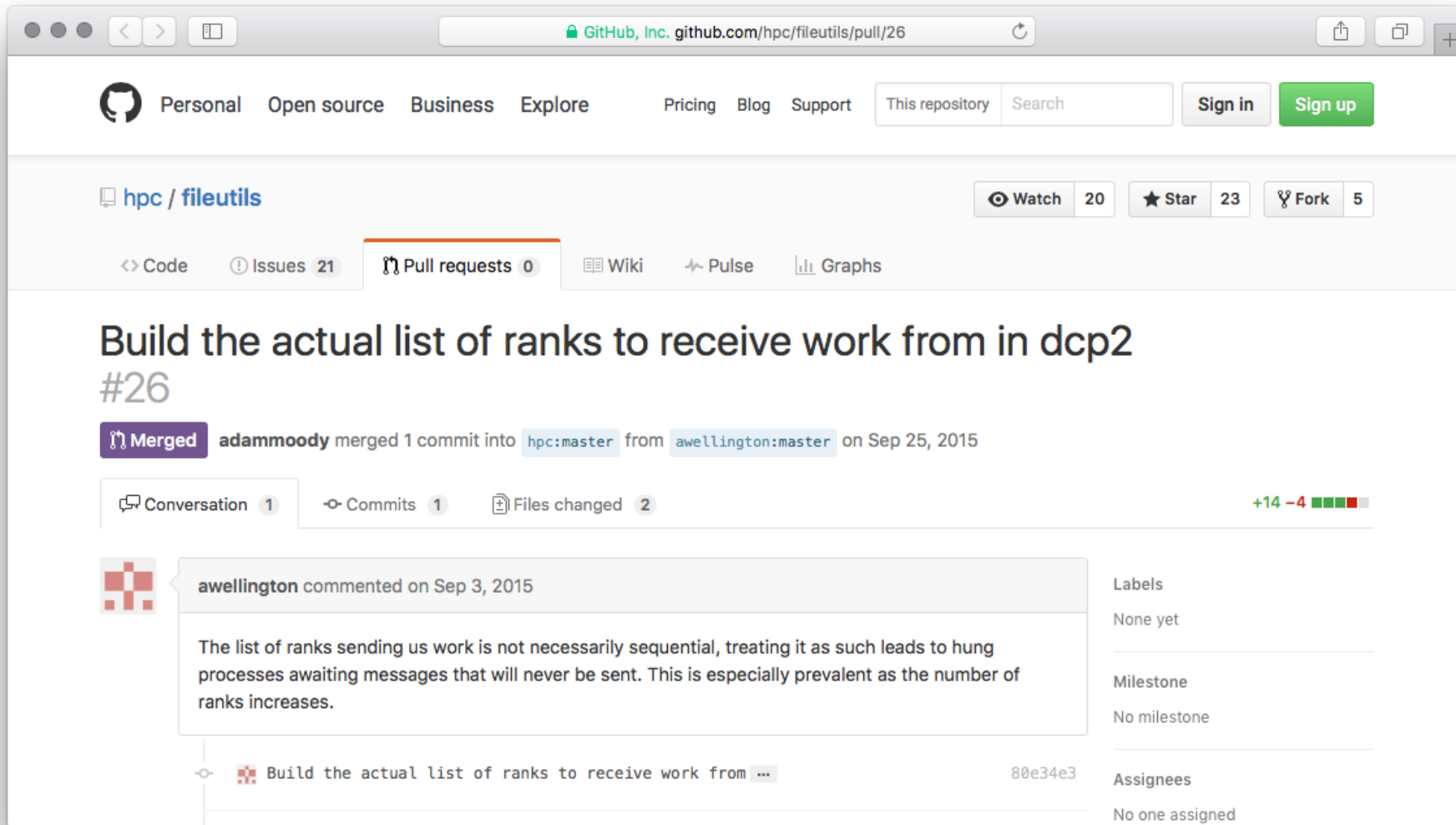
- Set Quota on Destination Filesystem
- Stop NFS exports for project directory being relocated
- Move project into restricted access migration directory
 - Root only accessible directory with obvious name
 - `drwx----- 5 root root 4096 Mar 10 02:53 migration_in_progress_g1_to_g3`
 - `mv /g/data1/projectID /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 custom mpi md5sum tool.
- Correct any mismatched files – create a filelist with full paths, split the list, use rsync. Typically none or very few files.
- Move data out of restricted target directory on destination filesystem
- Re-establish NFS exports

— Data Migration in Practice - example

- September 2015
- 2.4PB Project, migrated from Gdata1 to Gdata3

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

— Encountered a bug in dcp2, fix committed upstream



The screenshot shows a GitHub pull request page for the repository 'hpc/fileutils'. The pull request title is 'Build the actual list of ranks to receive work from in dcp2 #26'. It was merged by 'adammoody' on Sep 25, 2015, from the 'awellington:master' branch to the 'hpc:master' branch. The page shows 20 watches, 23 stars, and 5 forks. A comment from 'awellington' dated Sep 3, 2015, describes a bug: '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.' The commit hash is 80e34e3. The right sidebar shows 'Labels: None yet', 'Milestone: No milestone', and 'Assignees: No one assigned'.

GitHub, Inc. github.com/hpc/fileutils/pull/26

Personal Open source Business Explore Pricing Blog Support This repository Search Sign in Sign up

hpc / fileutils Watch 20 Star 23 Fork 5

Code Issues 21 Pull requests 0 Wiki Pulse Graphs

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

Conversation 1 Commits 1 Files changed 2 +14 -4

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

— **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

— **Improve scalability of Validation Processes**

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

— **Build dedicated Migration /Filesystem Load Test Cluster**

- Reuse 44x OSSes from gdata1 when decommissioned
- 12C Sandy Bridge Xeon, 256GB RAM, FDR Interconnect



Questions ?



Providing Australian researchers with
world-class computing services

NCI Contacts

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

Address:

NCI, Building 143, Ward Road The
Australian National
University Canberra ACT 0200



Australian Government
Department of Education



Australian
National
University



Australian Government
Bureau of Meteorology



Australian Government
Geoscience Australia



Australian Government
Australian Research Council



nci.org.au



[@NCInews](https://twitter.com/NCInews)