



Hadoop @ Caltech CMS Tier2

June 30, 2009

Michael Thomas



What is Hadoop

Map-Reduce plus the HDFS filesystem implemented in java

Map-Reduce is a highly parallelized distributed computing system

HDFS is the distributed cluster filesystem

★ This is the feature that we are most interested in

Open source project hosted by Apache

Used by Yahoo for their search engine. Yahoo is a major contributor to the Apache Hadoop project.



HDFS



Distributed Cluster filesystem

Extremely scalable – Yahoo uses it for multi-PB storage

Easy to manage – few services and little hardware overhead

Files split into blocks and spread across multiple cluster datanodes

- ★ 64MB blocks default, configurable
- ★ Block-level decomposition avoids 'hot-file' access bottlenecks
- ★ Block-level decomposition means the loss of multiple data nodes will result in the loss of more files than file-level decomposition



HDFS Services



Namenode – Manages the filesystem namespace operations

- ★ File/directory creation/deletion
- ★ Block allocation/removal
- ★ Block locations

Datanode – Stores file blocks on one or more disk partitions

Secondary Namenode – Helper service for merging namespace changes

Services communicate through java RPC, with some functionality exposed through http interfaces



Namenode (NN)

Purpose is similar to dCache PNFS

Keeps track of entire fs image

- ★ The entire filesystem directory structure
- ★ The file block -> datanode mapping
- ★ Block replication level
- ★ ~1GB per 1e6 blocks recommended

Entire namespace is stored in memory, but persisted to disk

- ★ Block locations not persisted to disk
- ★ All namespace requests served from memory
- ★ Fsck across entire namespace is really fast



Namenode Journals

NN fs image is read from disk only once at startup.

Any changes to the namespace (mkdir, rm) are written to one or more journal files (local disk, NFS, ...)

Journal is periodically merged with the fs image

Merging can temporarily require extra memory to store two copies of fs image at once.



Secondary NN

The name is misleading... this is NOT a backup namenode or hot spare namenode. It does NOT respond to namespace requests.

Optional checkpoint server for offloading the NN journal -> fsimage merges

- **Download fs image from namenode (once)**
- **Periodically download journal from namenode**
- **Merge journal and fs image**
- **Uploaded merged fs image back to namenode**

Contents of merged fsimage can be manually copied to NN in case of namenode corruption or failure.



Datanode (DN)

Purpose is similar to dCache pool

Stores file block metadata and file block contents in one or more local disk partitions. Datanode scales well with # local partitions

- ★ Caltech is using one per local disk (2-4 per datanode)
- ★ Nebraska has 48 individual partitions on Sun Thumpers

Sends heartbeat to namenode every 3 seconds

Sends full block report to namenode every hour

Namenode uses report + heartbeats to keep track of which block replicas are still accessible



Client access

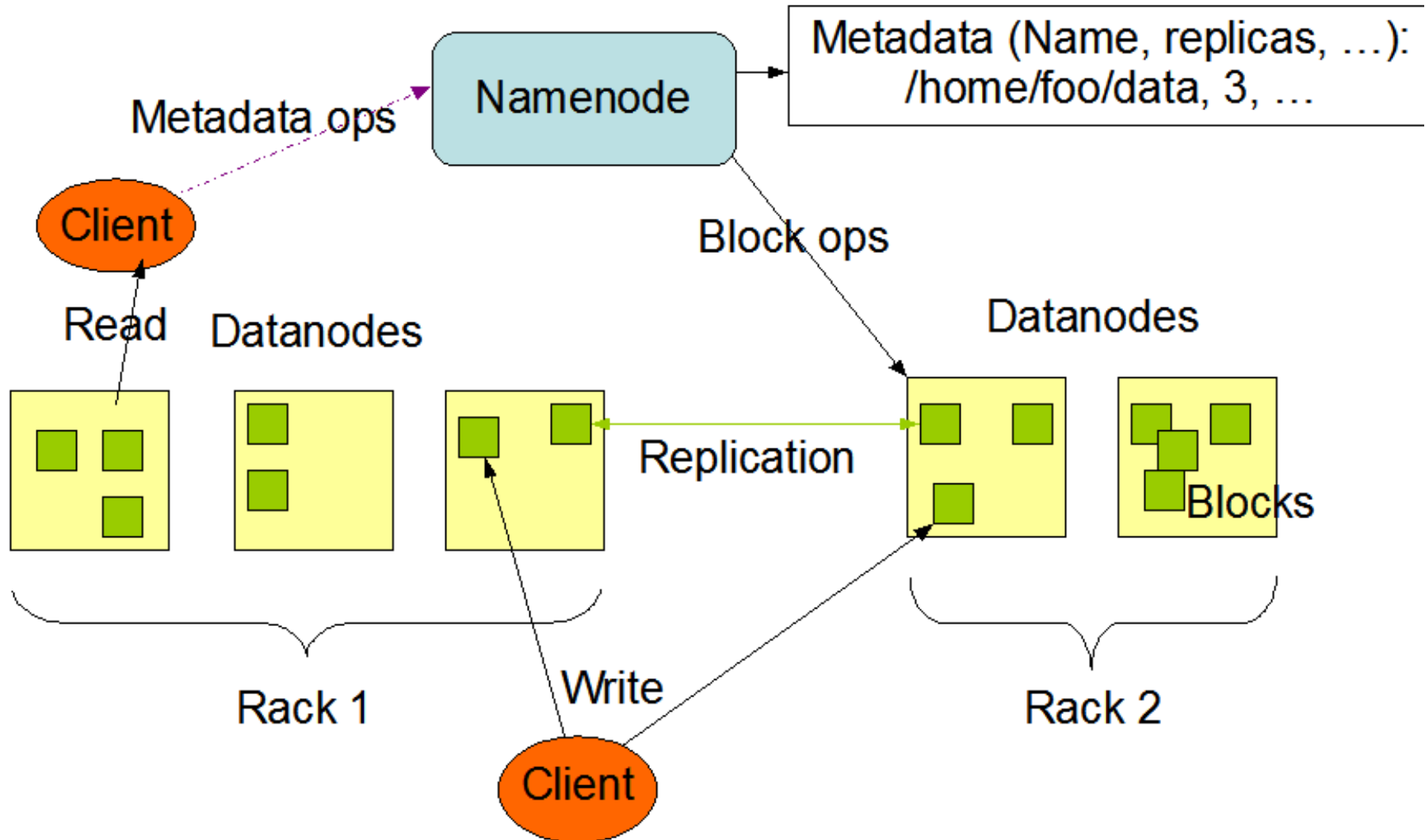
When a client requests a file, it first contacts the namenode for namespace information.

The namenode looks up the block locations for the requested files, and returns the datanodes that contain the requested blocks

The client contacts the datanodes directly to retrieve the file contents from the blocks on the datanodes



Hadoop Architecture





Native client

A native java client can be used to perform all file and management operations

All operations use native Hadoop java APIs



FUSE client

FUSE == Filesystem in Userspace

Presents a posix-like interface to arbitrary backend storage systems (ntfs, lustre, ssh)

HDFS fuse module provides posix interface to HDFS using the HDFS APIs. Allows the use of rm, mkdir, cat, and other standard filesystem commands on HDFS.

HDFS does not support non-sequential (random) writes

- ★ root TFile can't write directly to HDFS fuse, but not really necessary for CMS**

Random reads are ok



Gridftp/SRM clients

Gridftp could write to HDFS+FUSE with a single stream

Multiple streams will fail due to non-sequential writes

UNL developed a GridFTP dsi module to buffer multiple streams so that data can be written to HDFS sequentially

Bestman SRM can perform namespace operations by using FUSE

- ★ srmrm, srmls, srmmdir**
- ★ Treats hdfs as local posix filesystem**



Caltech Setup

Current Tier2 cluster runs RHEL4 with dCache. We did not want to disturb this working setup.

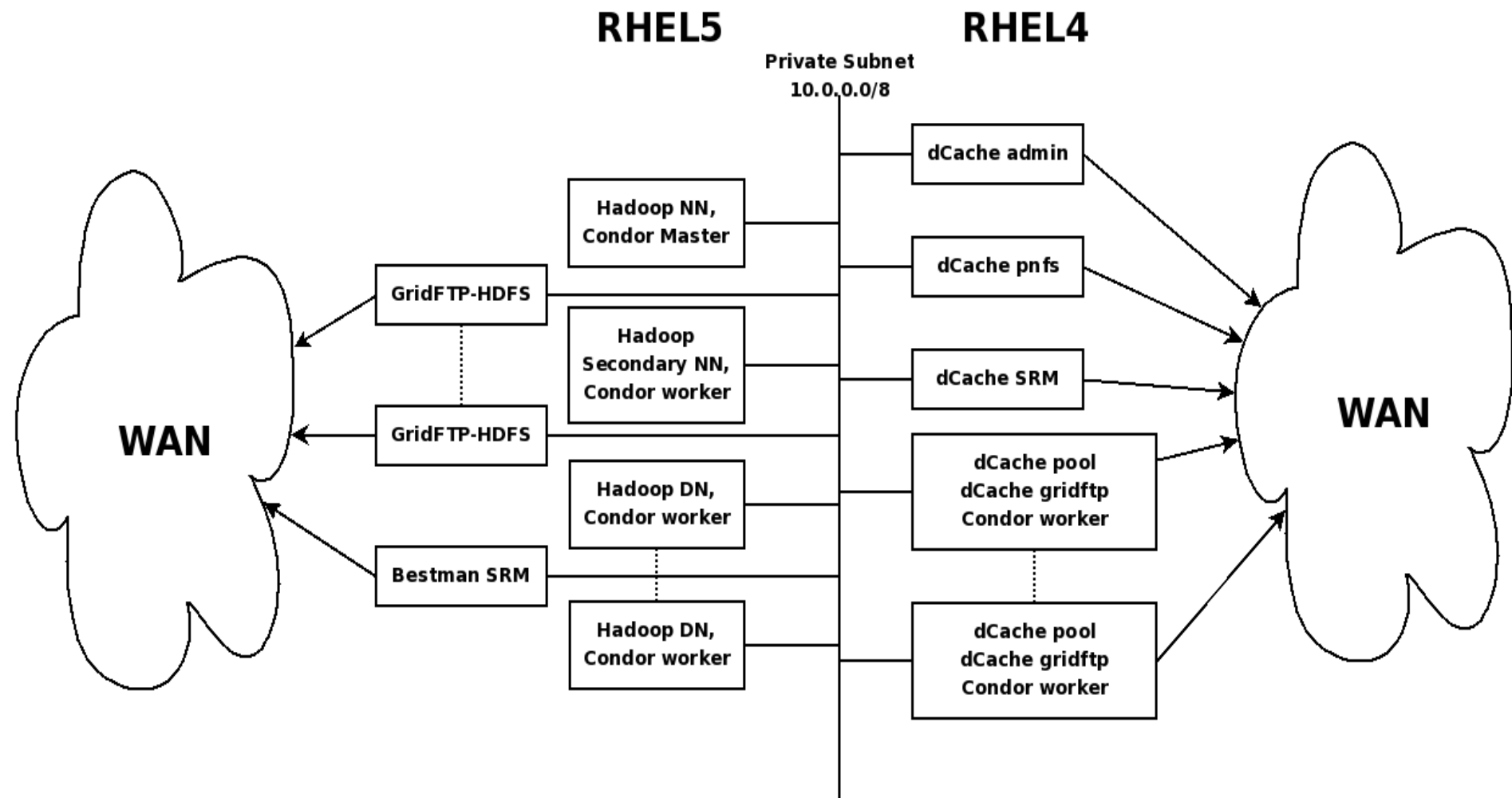
Recently acquired 64 additional nodes, installed with Rocks5/RHEL5. This is set up as a separate cluster with its own CE and SE. Avoids interfering with working RHEL4 cluster.

Single PhEDEx instance runs on the RHEL4 cluster, but each SE has its own SRM server.

Clusters share the same private subnet



Caltech Setup





Caltech Setup



- **Namenode runs on same system as Condor negotiator/collector**
 - ★ 8 cores, 16GB RAM
 - ★ System is very over-provisioned. Load never exceeds 1.0, JVM never exceeds 200MB
 - ★ Plenty of room for scaling to more blocks
- **Secondary NN runs on same system as condor batch worker**
 - ★ OOM twice
- **67 data nodes, 200TB available space**
 - ★ Includes 2 Sun Thumpers running Solaris
 - ★ Currently 82TB used
 - ★ All datanodes are also condor batch workers
- **Single Bestman SRM server using FUSE for file ops**
- **Two gridftp-hdfs servers with 2 x 10GbE (+2 more soon)**



Deployment history

T2_US_Nebraska first started investigating Hadoop last year. They performed a lot of R&D to get Hadoop to work in the CMS context

- **Two SEs in SAM**
- **Gridftp-hdfs DSI module**
- **Use of Bestman SRM**
- **Many internal Hadoop bug fixes and improvements**

Presented this work to the USCMS T2 community in February 2009



Caltech Deployment



Started using Hadoop in Feb. 2009 on a 4-node testbed

Created RPMS to greatly simplify the deployment across an entire cluster

Deployed Hadoop on new RHEL5 cluster of 64 nodes

Basic functionality worked out of the box, but performance was poor.

Attended a USCMS Tier2 hadoop workshop at UCSD in early March



Tier2 Hadoop Workshop

- **Held at UCSD in early March 2009**
- **Intended to help get interested USCMS Tier2 sites jump-start their hadoop installations**
- **Results:**
 - ★ Caltech, UCSD expanded their hadoop installations
 - ★ Wisconsin delayed deployment due to facility problems
 - ★ Bestman, GridFTP servers deployed
 - ★ Initial SRM stress tests performed
 - ★ UCSD <-> Caltech load tests started
 - ★ Hadoop SEs added to SAM
 - ★ Improved RPM packaging
 - ★ Better online documentation for CMS

<https://twiki.grid.iu.edu/bin/view/Storage/HdfsWorkshop>



Caltech Deployment



Migrated OSG RSV tests to Hadoop in mid-march

Migrated T1 -> Caltech load tests to Hadoop in early April

Attempted to move one /store/user/\$USER directory to hadoop in early April, but failed due to TFC problems

Added read-only http interface in mid may

Moved /store/data, /store/user in late May

Moved /store/unmerged in early June

Moved PhEDEx loadtest sources in mid-June



Hadoop monitoring



Nagios

- ★ `check_hadoop_health` – parses output of 'hadoop fsck'
- ★ `check_jmx` – blockverify failures, datanode space
- ★ `check_hadoop_checkpoint` – parses secondary nn logs to make sure checkpoints are occurring

Ganglia

- ★ Native integration with Hadoop
- ★ Many internal parameters

MonALISA

- ★ Collects Ganglia parameters

gridftpspy

Hadoop Chronicle

jconsole

hadoop native web pages




The Hadoop Chronicle - Mozilla Firefox 3.5 Beta 4

File Edit View History Bookmarks Tools Help

caltech.edu https://cms.hep.caltech.edu/hadoop/ Google

Most Visited EVO docs Caltech T2 Fedora

The Hadoop Chronicle



Selected or last chronicle

2009_06_26_08:30

=====

The Hadoop Chronicle | 46 % | Fri Jun.26.2009 08:30

=====

Global storage

Configured Capacity: 191813069336576 (174.45 TB)
Present Capacity: 191707600577536 (174.36 TB)
DFS Remaining: 102718547960182 (93.42 TB)
DFS Used: 88989052617354 (80.94 TB)
DFS Used%: 46.42%

/store/ area

Path	Size(GB)	#Files	#Dirs
/store/PhEDEx_LoadTest07	667	262	668
/store/data	24337	33474	462
/store/mc	2353	2146	15
/store/unmerged	438	3416	172
/store/user	8461	25234	433

User area

Path	Size(GB)	#Files	#Dirs
/store/user/burt	0	2	1
/store/user/chiorbo	902	5341	127
/store/user/dkcira	0	17	13
/store/user/dorian	41	286	1
/store/user/hpi	2	6	21
/store/user/ligioi	0	2	1
/store/user/litvin	0	3	8
/store/user/oatramen	3178	1036	77
/store/user/ssekmen	0	4	4
/store/user/test	0	2	5
/store/user/tucker	0	17	6
/store/user/uscms0377	10	7	1
/store/user/uscms0755	614	2754	3
/store/user/vlitvin	3709	15754	153
/store/user/wart	1	3	1

System health

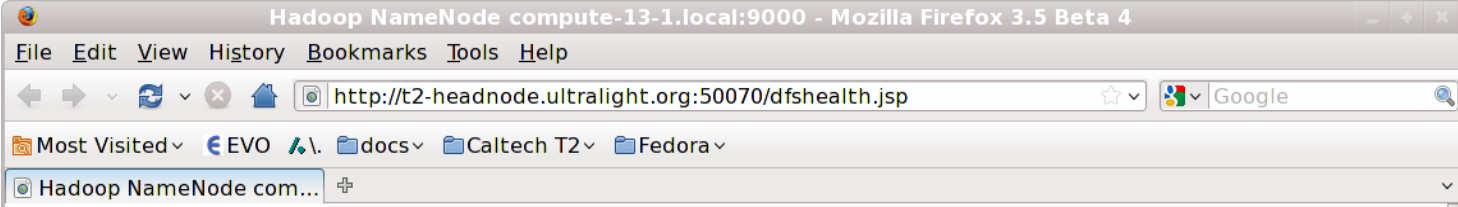
Total size: 38929932017766 B (Total open files size: 3087007744 B)
Total dirs: 1765
Total files: 64551 (Files currently being written: 2)
Total blocks (validated): 358503 (avg. block size 108590254 B) (Total open file blocks (not validated): 23)
Minimally replicated blocks: 358503 (100.0 %)
Over-replicated blocks: 63 (0.017573075 %)
Under-replicated blocks: 0 (0.0 %)
Mis-replicated blocks: 0 (0.0 %)
Default replication factor: 2
Average block replication: 2.349721
Corrupt blocks: 0
Missing replicas: 0 (0.0 %)

=====

All Chronicles

- 2009_06_26_08:30
- 2009_06_25_19:42
- 2009_06_25_08:30
- 2009_06_24_19:42
- 2009_06_24_08:30
- 2009_06_23_19:42
- 2009_06_23_08:30
- 2009_06_22_19:42
- 2009_06_22_08:30
- 2009_06_21_19:42
- 2009_06_21_08:30
- 2009_06_20_19:42
- 2009_06_20_08:30
- 2009_06_19_19:42
- 2009_06_19_08:30
- 2009_06_18_19:42
- 2009_06_18_08:30
- 2009_06_17_19:42
- 2009_06_17_08:30
- 2009_06_16_19:42
- 2009_06_16_08:30
- 2009_06_15_19:42
- 2009_06_15_08:30
- 2009_06_14_19:42
- 2009_06_14_08:30
- 2009_06_13_19:42
- 2009_06_13_08:30
- 2009_06_12_19:42
- 2009_06_12_08:30
- 2009_06_11_19:42
- 2009_06_11_08:30
- 2009_06_10_19:42
- 2009_06_10_08:30
- 2009_06_09_19:42
- 2009_06_09_08:30
- 2009_06_08_19:42
- 2009_06_08_08:30
- 2009_06_07_19:42
- 2009_06_07_08:30
- 2009_06_06_19:42
- 2009_06_06_08:30
- 2009_06_05_19:42
- 2009_06_05_08:30
- 2009_06_04_19:42
- 2009_06_04_08:30
- 2009_06_03_19:42
- 2009_06_03_08:30
- 2009_06_02_19:42
- 2009_06_02_08:30
- 2009_06_01_19:42
- 2009_06_01_08:30
- 2009_05_31_19:42
- 2009_05_31_18:55
- 2009_05_31_18:33
- 2009_05_31_17:52





NameNode 'compute-13-1.local:9000'

Started: Tue May 26 12:12:00 PDT 2009
Version: 0.19.2-dev, r748415
Compiled: Mon Mar 23 15:21:37 PDT 2009 by wart
Upgrades: There are no upgrades in progress.

[Browse the filesystem](#)
[Namenode Logs](#)

Cluster Summary

66825 files and directories, 359972 blocks = 426797 total. Heap Size is 269.38 MB / 888.94 MB (30%)

Configured Capacity : 174.45 TB
DFS Used : 81.13 TB
Non DFS Used : 98.23 GB
DFS Remaining : 93.23 TB
DFS Used% : 46.51 %
DFS Remaining% : 53.44 %
Live Nodes : 65
Dead Nodes : 6

Live Datanodes : 65

Node	Last Contact	Admin State	Configured Capacity (TB)	Used (TB)	Non DFS Used (TB)	Remaining (TB)	Used (%)	Used (%)	Remaining (%)	Blocks
compute-11-11	2	In Service	1.61	0.75	0	0.86	46.87	<div><div></div></div>	53.13	7579
compute-11-12	1	In Service	1.61	0.82	0	0.79	50.93	<div><div></div></div>	49.07	8252
compute-11-9	1	In Service	1.61	0.8	0	0.81	49.54	<div><div></div></div>	50.46	7998
compute-14-10	0	In Service	1.61	0.82	0	0.79	50.87	<div><div></div></div>	49.13	8092
compute-14-11	2	In Service	1.61	0.82	0	0.79	51	<div><div></div></div>	49	8432
compute-14-12	1	In Service	1.61	0.81	0	0.8	50.17	<div><div></div></div>	49.83	8325
compute-14-13	0	In Service	1.61	0.83	0	0.78	51.36	<div><div></div></div>	48.64	8465
compute-14-14	2	In Service	1.61	0.81	0	0.8	50.26	<div><div></div></div>	49.74	8156
compute-14-15	1	In Service	1.61	0.83	0	0.78	51.27	<div><div></div></div>	48.73	8342
compute-14-16	2	In Service	1.61	0.79	0	0.82	49.2	<div><div></div></div>	50.8	8057
compute-14-17	1	In Service	1.38	0.71	0	0.67	51.51	<div><div></div></div>	48.49	6999
compute-14-18	0	In Service	1.61	0.82	0	0.79	51.21	<div><div></div></div>	48.79	8379
compute-14-19	2	In Service	1.61	0.8	0	0.81	49.97	<div><div></div></div>	50.03	8182



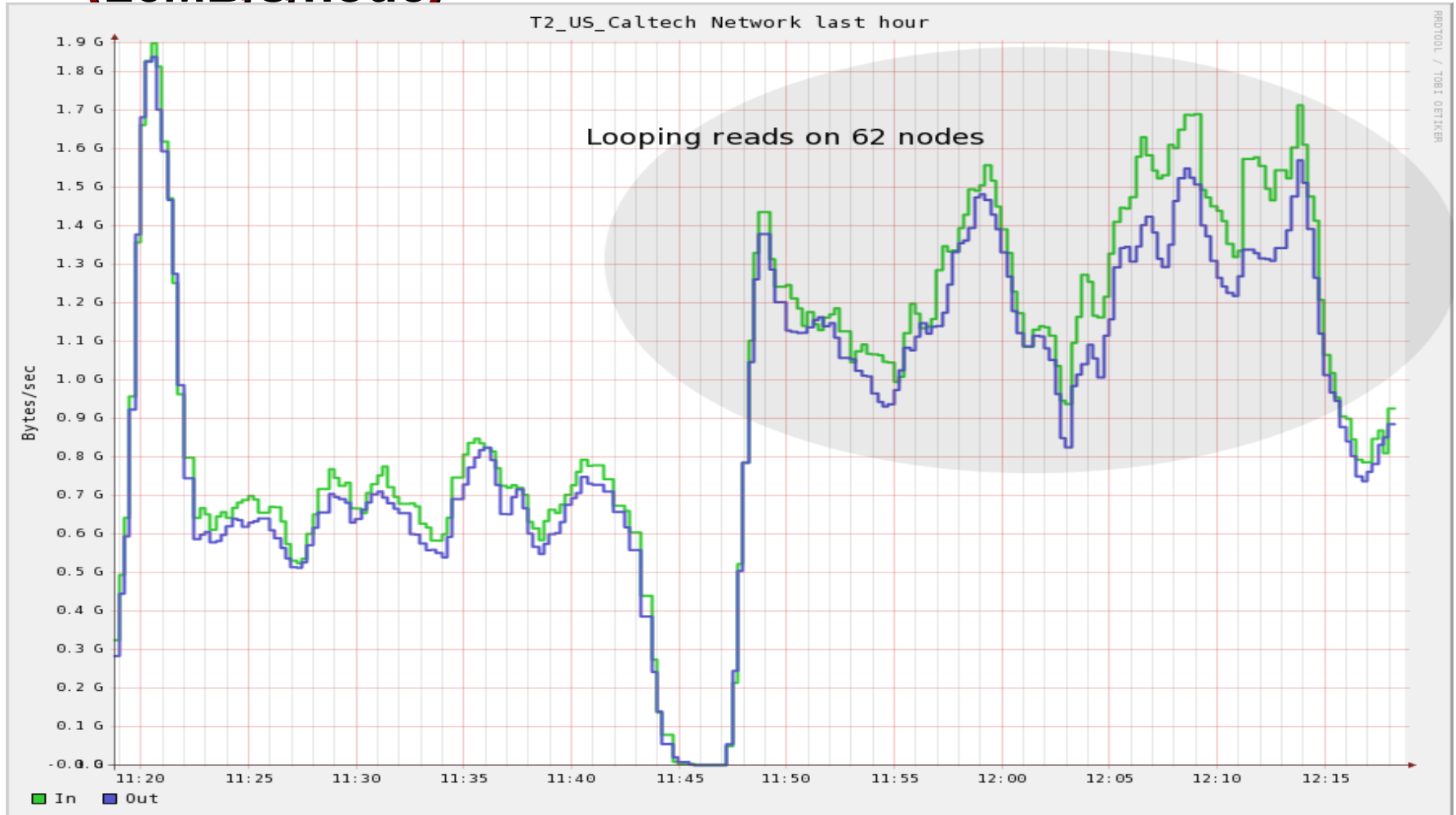
Current Successes

- **SAM tests passing**
- **All PhEDEx load tests passing**
- **RPMs provide easy installs, reinstalls**
 - ★ Now available for gridftp, too!
- **Bestman + GridFTP-HDFS have been stable**
- **Good Nagios coverage**
- **Expose fuse mount through apache**
 - ★ authenticated users with mod_ssl + SSLRequire
 - ★ Trivial for users to browse SE filesystem and download files
- **Great inter-node transfer rates (2GB/s aggregate)**
- **Adequate WAN transfer rates (300MB/s)**



Many Read processes

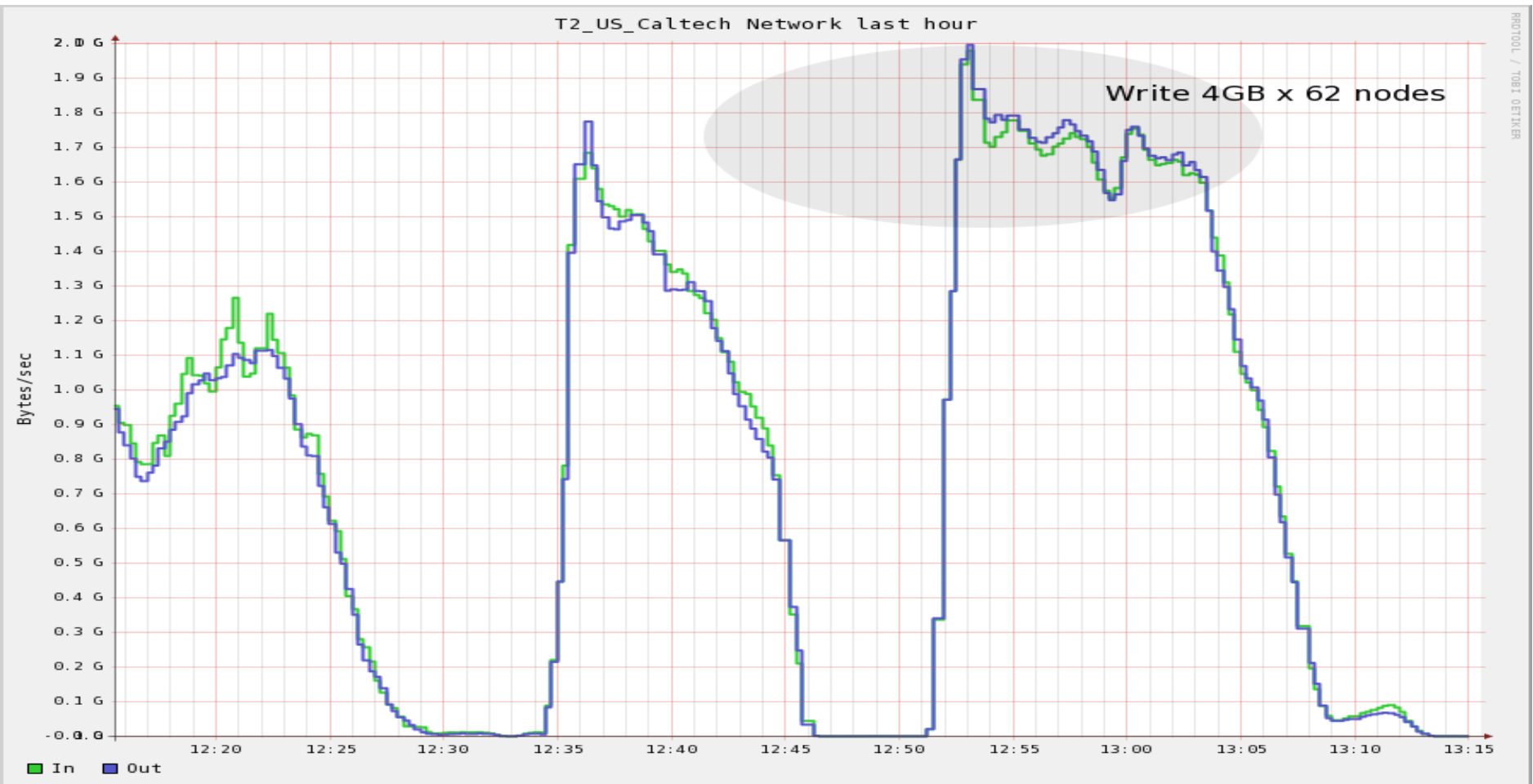
**Looping reads on 62 machines, one read per machine
(26MB/s/node)**





Many parallel writes with fuse

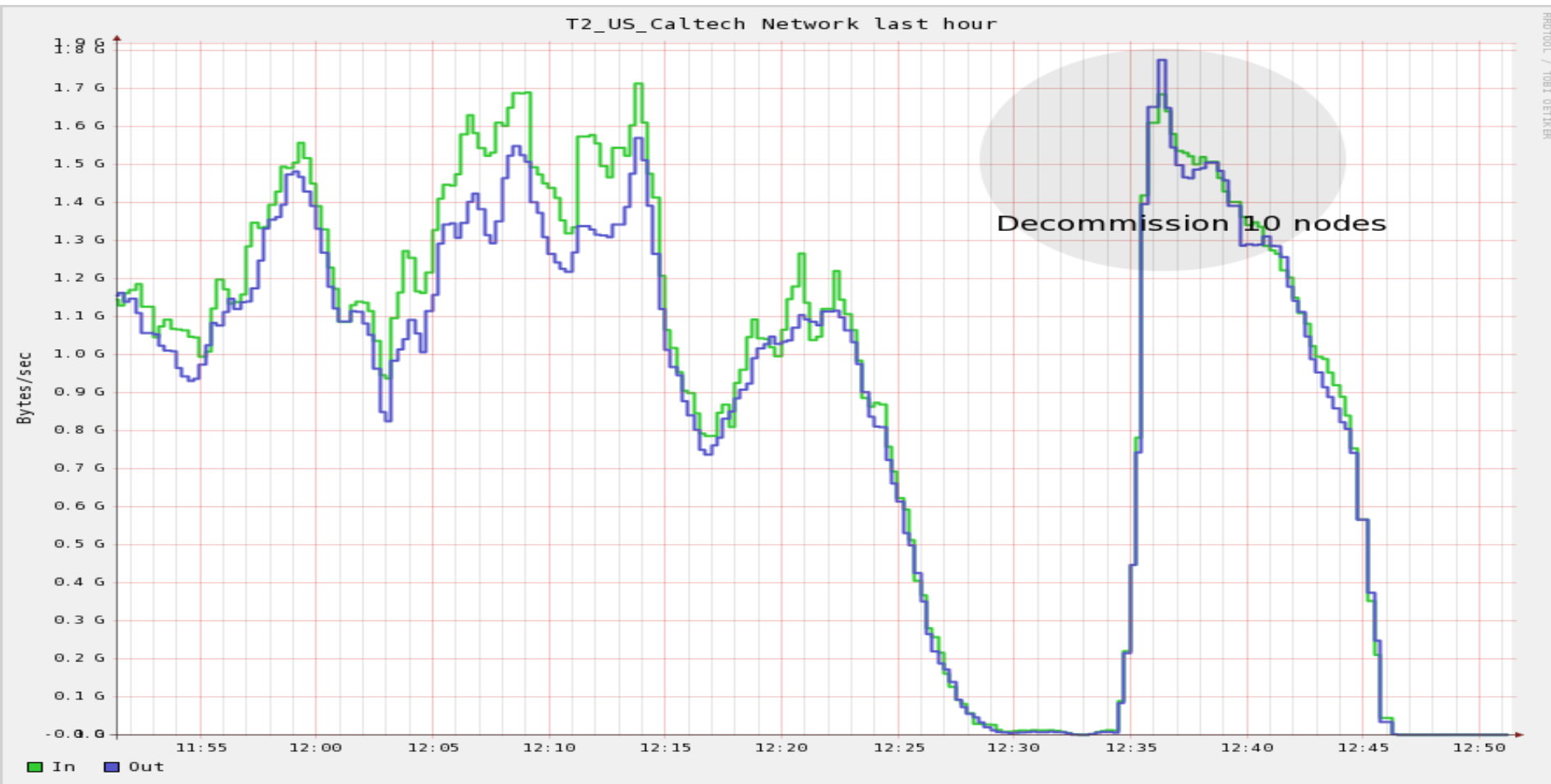
**Write 4GB file on 62 machines (dd+fuse) with 2x replication
(1.8GB/s) (30MB/s/node)**





Replicate by Decommission

Decommission 10 machines at once, resulting in the namenode issuing many replication tasks (1.7GB/s)

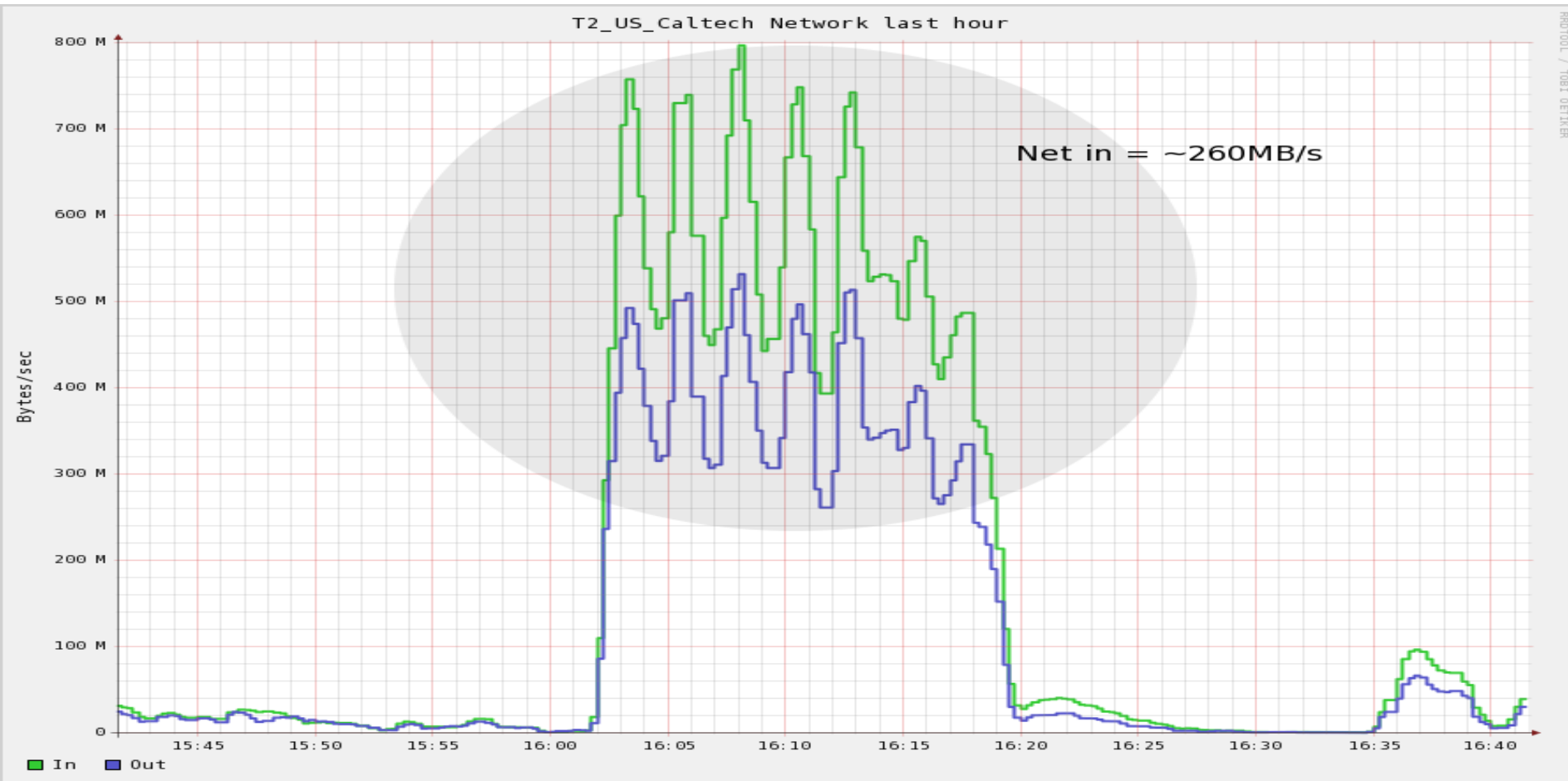




UCSD -> Caltech load tests



2 x 10GbE GridFTP servers, 260MB/s





Not without problems...

- **OSG RSV tests required patch to remove “:” from filenames. This is not a valid character in hadoop filenames. (resolved in upcoming OSG 1.2)**
- **Bestman dropped VOMS FQAN for non-delegated proxies, caused improper user mappings and filesystem permission failures for SAM, PhEDEx (resolved)**
- **Bestman error messages incompatible with lcg-utils (resolved)**
- **TFC not so “t” anymore due to multiple SEs**
- **Datanode/Namenode version mismatches (improved)**
- **Initial performance was poor (400MB/s aggregate) due to cluster switch configuration (resolved)**



Not without more problems...



FUSE was not so stable

- ✱ Boundary condition error for files with a specific size crashed fuse (resolved)
- ✱ df sometimes not showing fuse mount space (resolved)
- ✱ Lazy java garbage collection resulted in hitting ulimit for open files (resolved with larger ulimit)

GridFTP servers crashing

- ✱ Excessive memory usage for large files (resolved)
- ✱ temp file not configurable (resolved)
- ✱ Unstable NIC driver (ongoing)

Running two CEs and SEs requires extra care so that both CEs can access both SEs

- ✱ Some private network configuration issues (resolved)
- ✱ Lots of TFC wrangling (ongoing)



Next steps

Update benchmarks to show that HDFS satisfies the CMS SE technology requirements

More WAN transfer tests and tuning

- ★ FDT + HDFS integration underway
- ★ Install additional gridftp servers
- ★ Resolve gridftp NIC driver issues, hardware stability

Migrate additional data to Hadoop

- ★ All of /store/mc
- ★ Non-CMS storage areas

Further packaging improvements



Overall Impressions

Management of HDFS is simple relative to other SE options

Performance has been more than adequate

Scaled from 4 nodes to 64 nodes with minimal problems

~50% of our initial problems were related to Hadoop, the other 50% were Bestman, TFC, PhEDEx agent, or caused by running multiple SEs

We are currently committed to using Hadoop for most of our CMS data