Hadoop DFS for USLHC Facilities

Brian Bockelman

19 August 2009

Read while you listen

- HDFS design document: http://hadoop.apache.org/common/docs/current/hdfs_design.html
- OSG Hadoop documentation: https://twiki.grid.iu.edu/bin/view/Storage/Hadoop
- Google FS Paper: http://labs.google.com/papers/gfs.html

Hadoop

- Hadoop is an open source implementation of the MapReduce data processing paradigm.
- Biggest user is Yahoo! - 14PB of raw disk space.
- Apache-based project; Yahoo is the largest contributor
  - Development team of about 20.
  - External developers probably add up to 10-20.
- Commercial startup, Cloudera, supports HDFS.
  - Jeff Hammerbacher, chief scientist @ Cloudera, will be giving a colloquium at CERN on Friday.

HDFS

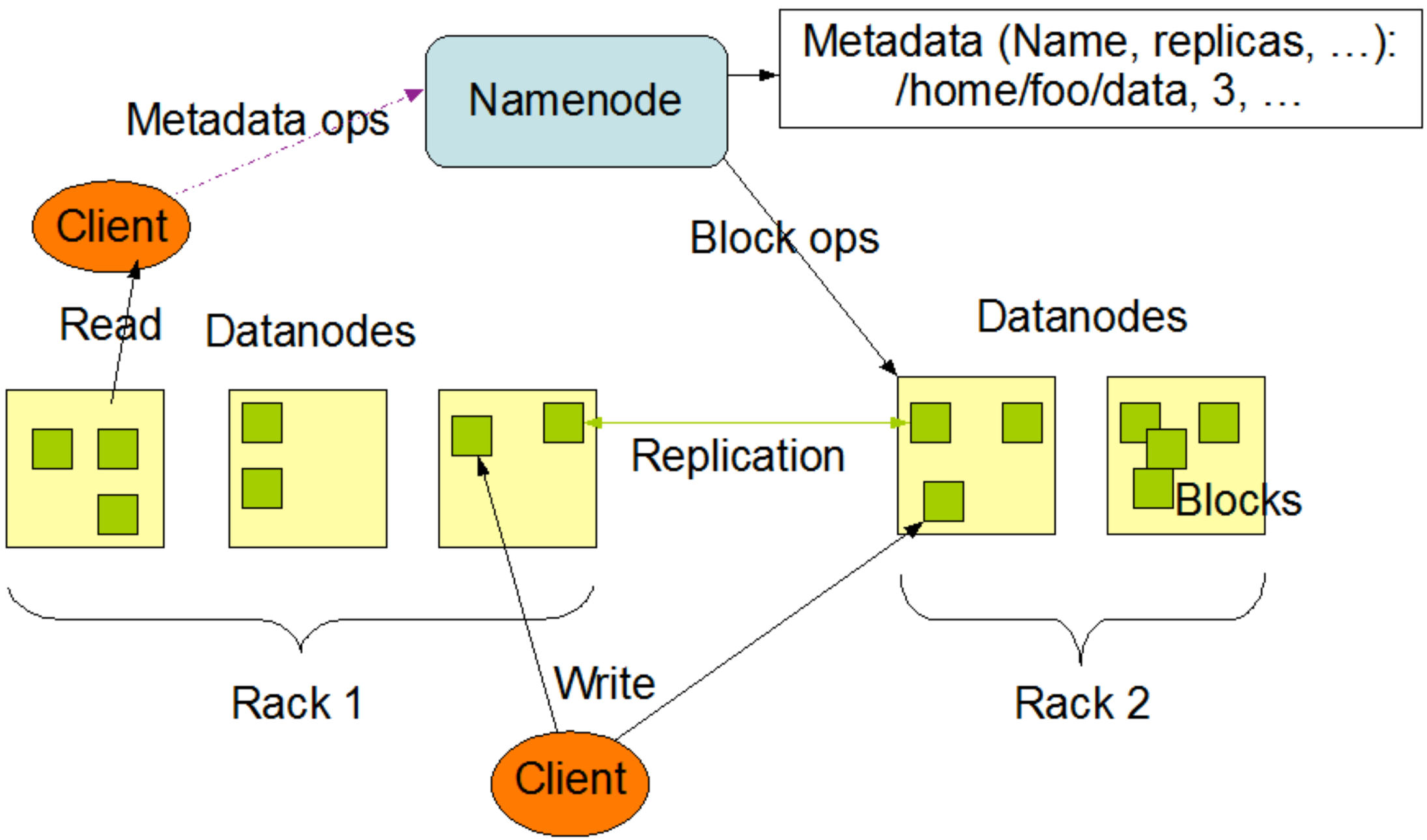
- One of the core pieces of Hadoop is the distributed file system - HDFS.
  - Central namenode
  - Many datanodes
  - Add-on components of Globus GridFTP and SRM.
  - Everything is designed for handling failure.

HDFS Design

- HDFS decomposes files into multiple blocks, and then makes multiple replicas of each block.
  - Block decomposition means reads will be spread across several servers, even if *many* clients reading the same file

HDFS Design

HDFS Architecture



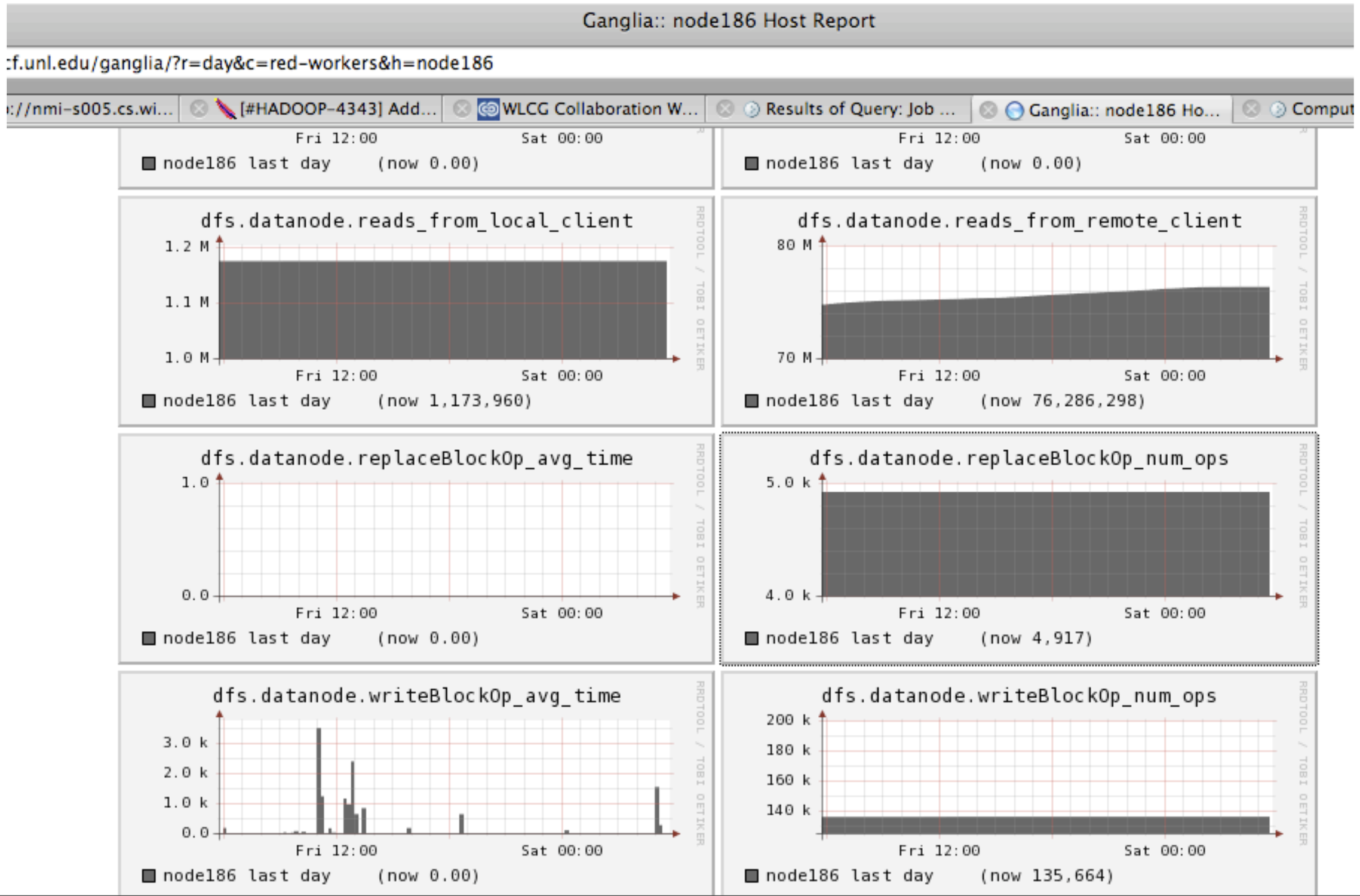
HDFS Datanode

- Similar to dCache pool
- Stores file block metadata and file block contents.
  - Can have multiple partitions per DN (we have up to 48)
- Sends heartbeat to namenode every 3s; used for synchronization and sending work messages.
  - Prevents “orphan files”
- Continuous background checksumming.

Datanodes

- Datanode performance can match hardware performance, esp. in terms of IOPS and bandwidth.
- Hadoop telemetry integrates with site's Ganglia.
- If a client process is on a datanode and the datanode holds the block it wants, no network transfer will occur.
  - Will also optimize for rack location.

Ganglia Graphs



HDFS Namenode

- The HDFS namenode is a highly scalable metadata server.
- Namenode persists namespace information and keeps the namespace and file location information in memory. Every namespace modification is written to a journal.
  - Hence, reads *never* cause a disk lookup on the NN; writes cause it to append to a file - no random I/O on the NN.
  - A secondary node merges the journal once every hour or 64MB of journal.
  - Requires about 1GB RAM per 1 million objects (object = 1 file, directory, or block). We have 1.9M objects (300k files) and 750MB RAM used as of writing.

Namenode Scalability

- Practically, you're limited to tens of millions of files in a single system.
- Yahoo! clocks their namenode at 50,000 "open" operations per second.
  - No limits to # of files opened.
- Can write about 5,000 files / second.
- In typical operation, namenode utilization is around 2-5%.

Namenode resilience

- You provide a list of destinations where the namenode should write its journal (recommended: 2 local devices, 1 NFS)
- Secondary NN always keeps the last few images.
- High availability / failover solution available.

Add-Ons

- FUSE provides a mountable POSIX FS.
  - Limits clients to supported HDFS semantics; basically, write-once-read-many (WORM)
- Globus GridFTP server + HDFS module.
- BeStMan SRM server.
- Xrootd integration
- HTTPS integration (using FUSE + Apache)

Caveats

- HDFS is only for use *in cluster*. Do not open up datanodes to WAN!
- Write semantics are limited (single-stream writing; flush and sync not yet supported).
- Rapidly maturing system - new feature releases will happen 2-3 times a year (new features often introduce new bugs, right?).
  - But Yahoo! has a dedicated QA team, a stringent unittest requirement, and test clusters larger than our T2

HDFS Sites on the Grid

- The largest sites are:
  - (All) 1 namenode, 1 secondary namenode
  - UCSD: 106 datanodes, 106 gridftp servers, 1 BestMan. 390 TB.
  - Caltech: 103 datanodes, 12 gridftp servers, 1 BeStMan. 340 TB.
  - Nebraska: 120 datanodes, 5 gridftp servers, 3 BeStMan. 370 TB.
    - All T2s will grow significantly in the next few months.
  - A handful of T3 sites.
  - All deploys (even tests) need 1 NN, 1 secondary NN, 1 SRM, >1 gridftp, >2 datanodes. The SRM and GridFTP can overlap.

Why HDFS?

- In the next few slides, we'll discuss why we think HDFS means:
  - Less management.
  - More reliability.
  - Better scalability.

Why HDFS?

- Why?
  - HDFS is solid software, amply funded by large commercial companies.
  - HDFS is built with the assumption your nodes will fail - it makes most recoveries automatically.
    - No admin intervention!
- Just finishing up with official USCMS review.

Why HDFS?

- This is (roughly) the install procedure for HDFS:
  - yum install hadoop-fuse
- To configure, edit /etc/sysconfig/hadoop (about 5 parameters), then run hadoop-firstboot.
  - Then, “service hadoop start”

Management

- The following tasks are trivial:
  - Integration of statistics with **Ganglia**.
  - **Decommissioning** hardware.
  - **Recovery** from hardware failure.
  - **Fsck**!
    - Checks the current knowledge of the filesystem and counts how many block replicas there are per file, and highlights any which are under-replicated.
  - **RPM** install (including Grid components).
  - Many of our “well-known” problems are not possible.
    - **Don’t need a separate admin toolkit**! (one gremlin)
  - Setting **quotas (***per directory***)**.
  - **Backups of namespace**.
  - **Balancer** is included.

Log files

- We centralize logfiles using syslog out of the box.
- Things that happen at WARNING level mean something failed, but no error went to user.
- ERROR level means system error.
- The above two mean a lot!

FSCK example

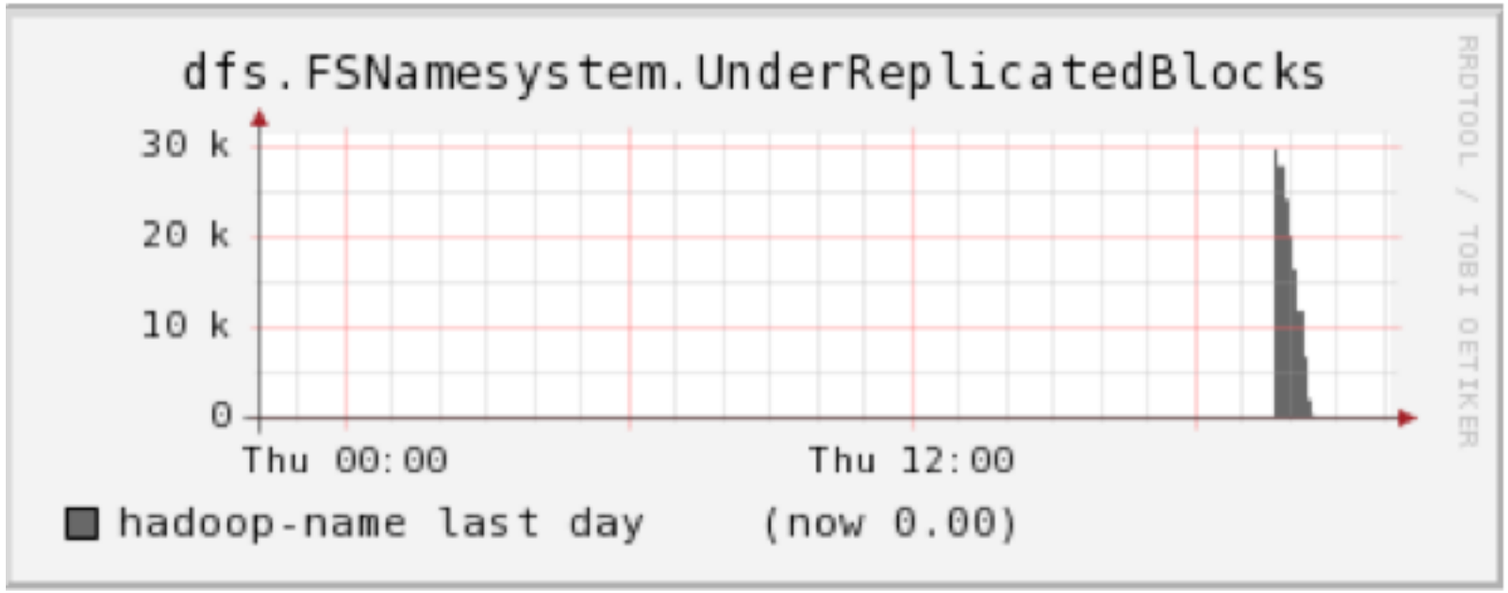
```
root@hadoop-name:~ — ssh — 107×33
.....................................................................................
.....................................................................................
.....................................................................................
.....................................................................................
.....................................................................................
.....................................................................................
.....................................................................................
.....................................................................................
.....................................................................................
.....................................................................Status: HEALTHY
 Total size:    72767054047268 B
 Total dirs:    2271
 Total files:   59765 (Files currently being written: 1)
 Total blocks (validated):      1053128 (avg. block size 69096115 B)
 Minimally replicated blocks:   1053128 (100.0 %)
 Over-replicated blocks:        3778 (0.3587408 %)
 Under-replicated blocks:       0 (0.0 %)
 Mis-replicated blocks:         0 (0.0 %)
 Default replication factor:    3
 Average block replication:     2.0923886
 Corrupt blocks:                0
 Missing replicas:              0 (0.0 %)
 Number of data-nodes:          113
 Number of racks:               1


The filesystem under path '/' is HEALTHY

real    0m7.753s
user    0m0.835s
sys     0m0.159s
[root@hadoop-name ~]#
```

Reliability

- Clients will automatically connect to a different datanode if one fails during a read.
- Blocks will automatically re-replicate -- and quickly! Often, we will recover from a loss in an hour.
  - Namenode controls this. We have it set to re-replicate if a node hasn't checked in for 10 minutes.
- All data is checksummed on read.



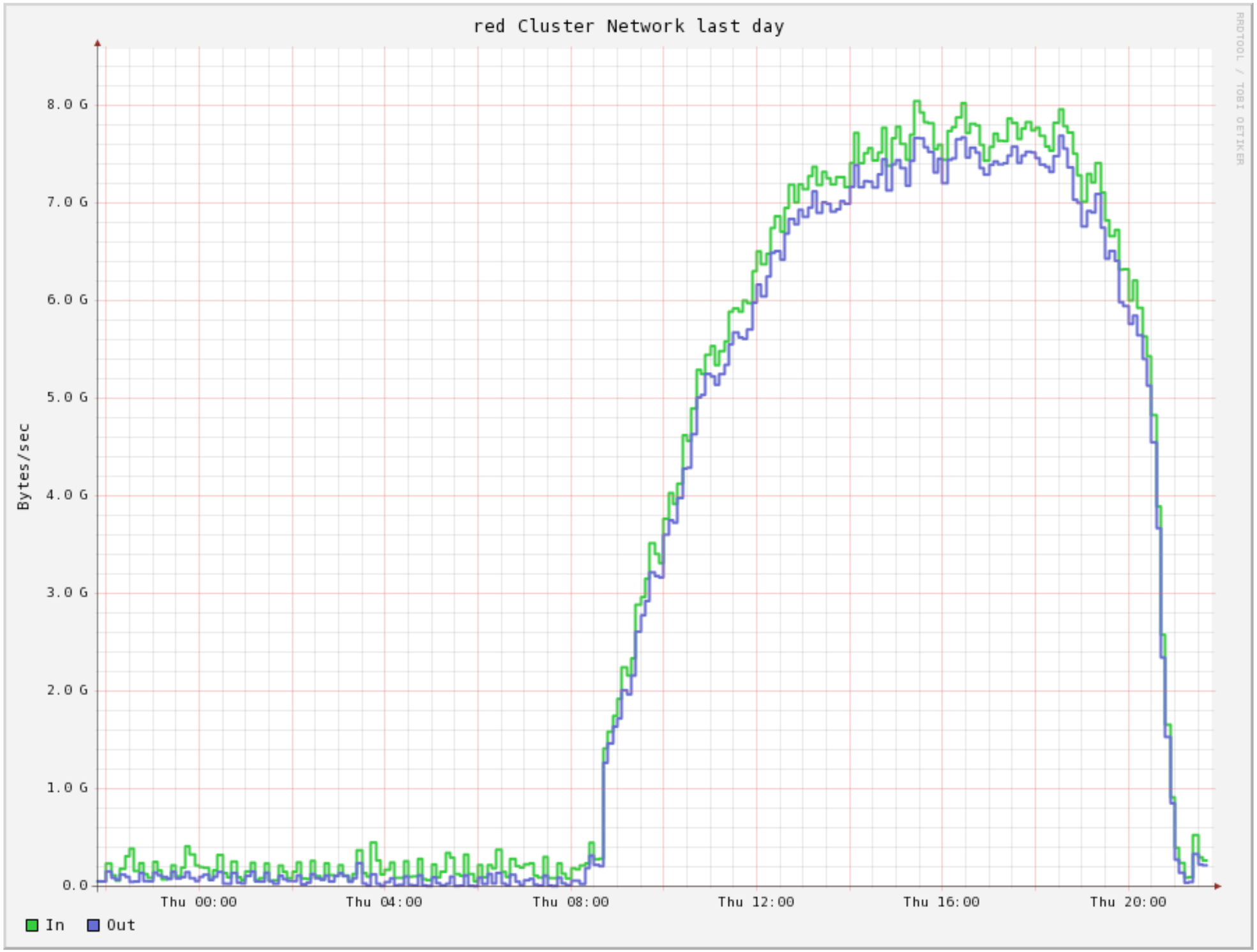
Reliability

- Data node failures (on read or write) do not result in client failures!

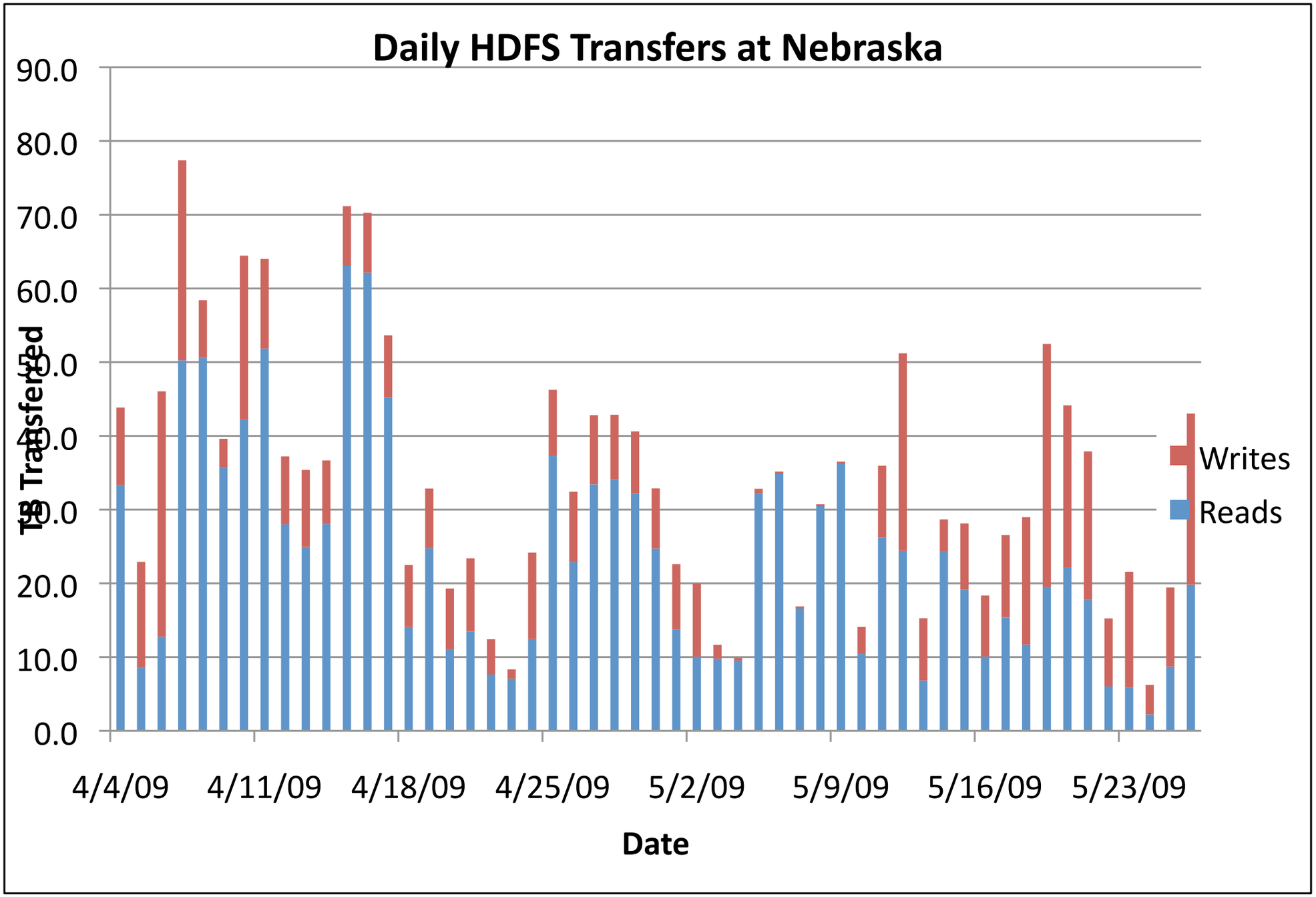
Scalability

- Next few slides show our favorite scalability plots from Nebraska

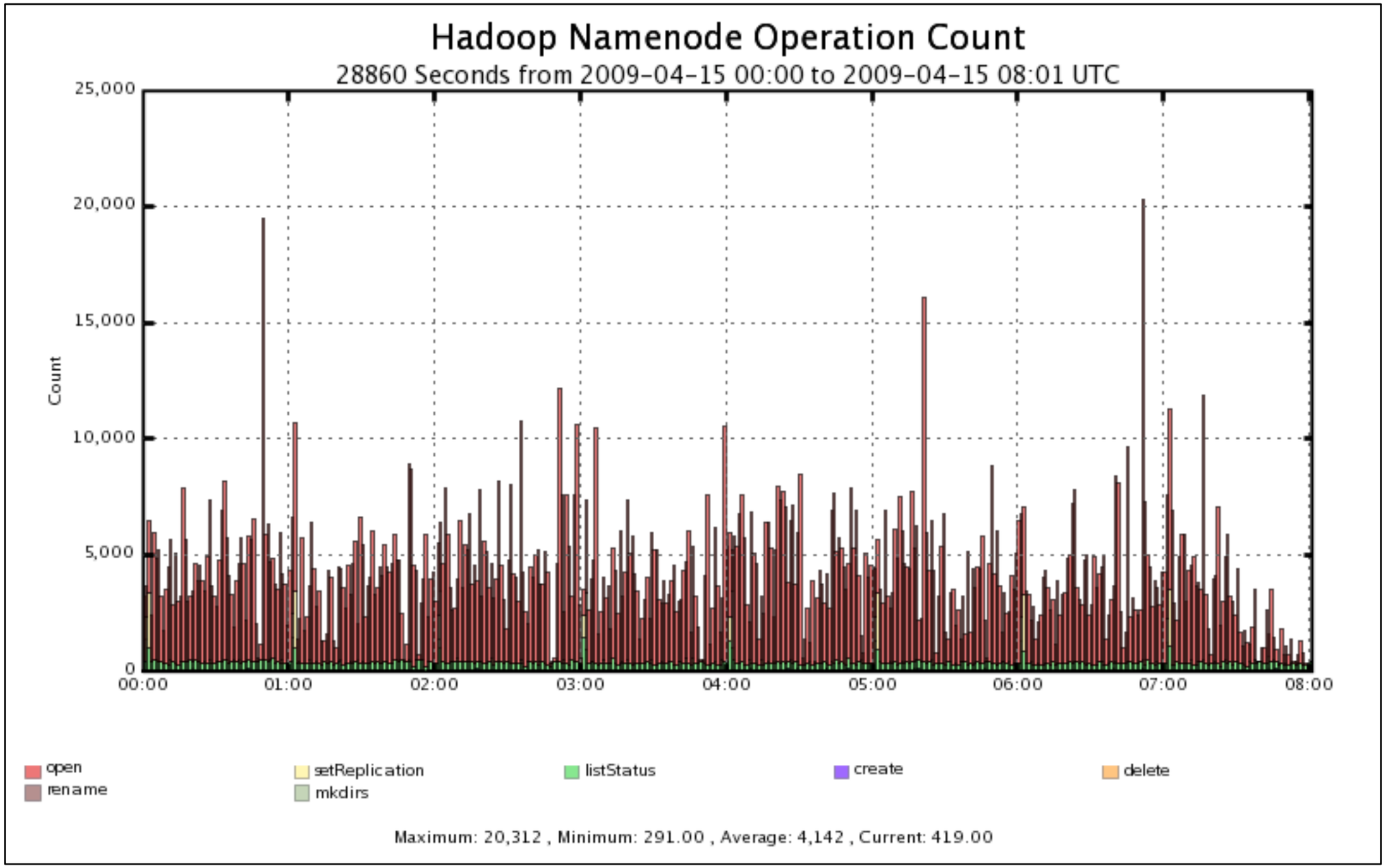
(CMSSW) Performance



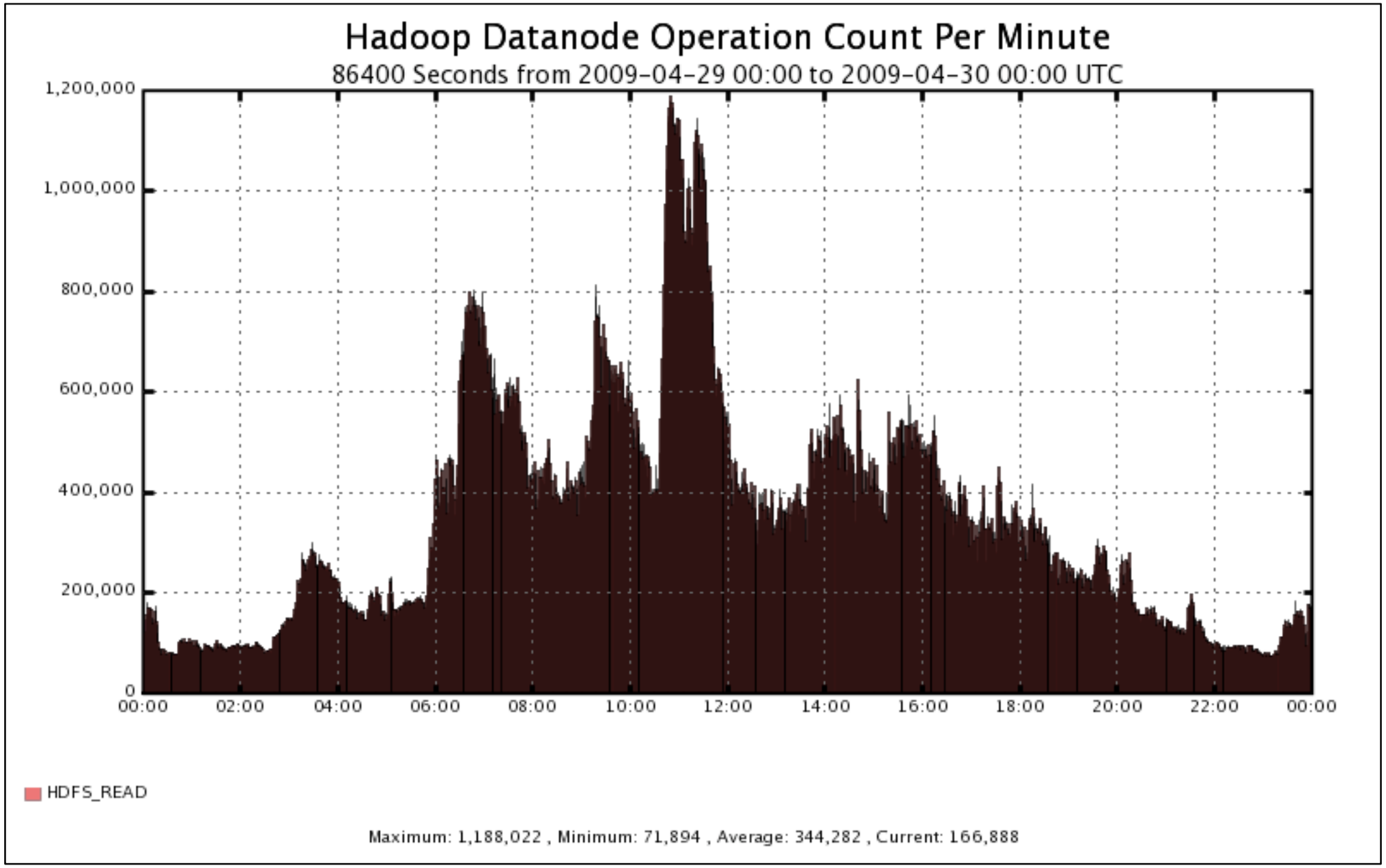
TB moved / day


Daily HDFS Transfers at Nebraska
90.0
80.0
70.0
60.0
50.0
40.0
30.0
20.0
10.0
0.0
TB Transferred
4/4/09
4/11/09
4/18/09
4/25/09
5/2/09
5/9/09
5/16/09
5/23/09
Date
Writes
Reads

Namenode Ops



I/O Ops per Second



Performance Stats

- We've clocked:
  - The filesystem at 80Gbps.
  - 23 Gbps for 300 CMSSW processes analyzing a *single file* @ 2 replicas (we picked a fake workflow to pump up the per-job rate).
  - SRM endpoints at ~200Hz (these SRMs are stateless; load-balancing is trivial). Done using GUMS auth.
  - fsck takes <10s.
  - Decommissioning a pool <1hr.
  - Namenode restart in about 60s.
  - WAN transfers peak at 9Gbps, sustain 5Gbps.
  - 18,400 metadata ops / sec from the namenode.

HadoopViz

- This is a 3D application put together by a grad student.
  - One "drop" per data transfer, going from the source node to the destination node - each node is a square.
  - http://www.youtube.com/watch?v=qoBoEzOkeDQ
  - Can do live demo if laptop adapter is working...
- We also have Gratia probes, but they aren't as fun.

Skunkworks.

- What R&D is going on at grid sites?
  - Xrootd protocol integration works; xrootd clustering integration doesn't.
  - GridFTP server speedups & refinements.
  - Improved logging: get relevant data on the right file *on the right server*.
  - Packaging, packaging, packaging! Right now, you can "yum install hadoop".
  - Caltech has been working on FDT integration throughout the summer.

Usage on OSG

- Three Tier 2 sites use HDFS at scale: Caltech, Nebraska, UCSD.
  - Wisconsin is on the fence.
- Two Tier 3 sites: U Minnesota, UC Davis
- One non-LHC site: LIGO Caltech ITB

Contact Us!

- The HDFS community is strong both outside and inside the OSG.
- Community support is available at osg-hadoop@opensciencegrid.org
- "Official" support coming soon from OSG (FY2010).

Questions?

- Questions?
- Comments?
- Fears?