cheap ‘n’ deep

THE HOLY-GRAIL OF STORAGE

Stéphane Thiell, Kilian Cavalotti - Stanford Research Computing

Lustre Administrators and Developers Workshop
Paris, France – Sep 20, 2016

Contents

Introduction

- Stanford University
- Stanford Research Computing Center (SRCC)
- Supported and shared computing resources

SRCC storage requirements

SRCC cheap ‘n’ deep storage: the Oak project

Google Drive: free ‘n’ deep storage?

Stanford University

FACTS AT A GLANCE

Stanford University

Stanford University is one of the world's leading research universities

- 2,153 faculty members
- 20 Nobel laureates
- 16,122 students (undergraduates and graduates)
- 5,500 externally sponsored research projects

- Seven Schools
  - Business
  - Earth, Energy & Environmental Sciences
  - Education
  - Engineering
  - Humanities and Sciences
  - Law
  - Medicine

- http://facts.stanford.edu/

Stanford Research Computing Center (SRCC)

OVERVIEW

Stanford Research Computing Facility (SRCF)

- Stanford building located on the SLAC campus
  - completed in the fall of 2013
  - production HPC services being offered since January 2014
  - can host 150 racks at 20kW
  - resilient power infrastructure (3 megawatts)

- Especially energy efficient
  - designed to host high performance computing equipment
  - cooled with ambient air fan systems for 90% of the year

- Network connectivity
  - 100 gigabit network linking the SRCF to campus backbone, the Internet, Internet2 and other national research networks
  - additional 100 gigabit to commodity Internet expected by year's end

- Three service models
  - Hosting
  - Supported clusters and servers
  - Shared computing resources

SRCC shared computing resources

Sherlock

- **Condo** cluster (850+ nodes, CPU and GPU)
- Open to the Stanford community as a resource to support **sponsored research**

Sherlock's storage spaces

- Isilon (NFS) for home directories
- Lustre "scratch" behind Inet routers
  - Dell servers, MD3x60 disk arrays
  - Lustre 2.7 (IEEL 3.0)
  - 3 PB

Farmshare

- General compute environment
  - Ubuntu Linux based
- Open to students, great for coursework and research-related computing prior to scaling up to Sherlock

Farmshare's storage spaces

- AFS
- Isilon (NFS)
- ZFS on Linux

SRCC supported computing/storage resources

XStream GPU cluster

- Cray CS-Storm
- 520 Nvidia K80s (1040 GPUs)
- National Science Foundation (NSF) Major Research Instrumentation (MRI) grant
- 20% of GPUh made available to XSEDE
- Main storage: Lustre Sonexion (1.4 PB)

Plus many smaller lab or departmental systems…

SCG's DDN GridScaler

SNI (ZFS on Linux)

SCG clusters

- The Stanford Genomics Clusters are available to members of the Genomics research community at Stanford
- Managed by the Genetics Bioinformatics Service Center (GBSC) through a charge-back model
- Main storage: 4.4 PB of GPFS (DDN GridScaler)

SNI virtualization system

- VDI based on KVM and Ceph with ZFS on Linux backend storage for the Stanford Neurosciences Institute

SRCC storage requirements

STANFORD RESEARCH COMPUTING CHEAP 'N' DEEP STORAGE

California live oak (Quercus agrifolia)

Source: wikipedia.org

Storage needs

Our researchers need:

- a capacity-oriented, longer-term storage space
- with immediate access to their data
- not intended for high duty cycle workloads
- easily accessible from anywhere
- ideally only pay for disks, at Fry's price of course…

Leading Innovation >>> 3TB P300 Series SATA 6Gb/s 3.5" Internal Hard Drive Retail Kit

- Ideal for Everyday Computing Needs
- High Quality and Reliability
- Data Protection Technology, NCQ Support

Regular Price $89.99

$20.99 PROMO CODE

$69*

7200RPM

PROMOCODE TOP PICK

SAVE 23%

*With Tuesday 6.21 Promo Code

MORE INFO

In-Store Only
Limit 1 per Customer
While Supplies Last
No Rainchecks
#8713721
Good Through
06.22.16

Use cases:

- HPC "campaign storage" for the duration of research projects
- Store and then easily access large datasets from experiments
  - › from an HPC cluster like Sherlock
  - › from desktops, laptops…
    - using NFS, CIFS, WebDAV, Globus…
    - through web applications like ownCloud, Jupyter…

Specifications

- Affordability
  - Cheap: that’s **< $100/TB**, for 3+ years, thus < **$33/TB/year**
  - **Condo model**: offer an affordable base expansion unit

- Scalability in a condo model
  - Condo model: start small, grow often, scale to petabytes
  - Focus on volume with no extension limit

- Diversity of access protocols/methods to enable:
  - standard HPC users
  - the growing number of "long tail of science" users

- Manageability and self-sustainability

Challenges

- "Low **price per TB**" ←/→ "Affordable increments"
  - contradictory goals

- Problems with most vendor solutions in our case
  - Not easily expandable, if at all
  - Expensive initial investment
  - Systems based on ultra-dense disk arrays (84 or 90 disks)
    - steep price for extensions
  - Closed-source management software

Introducting Oak

THE ONE-OF-A-KIND STORAGE SYSTEM

Major Oak, English Oak (Quercus robur)

Source: wikipedia.org

Introducing Oak

Main drivers

- Multiple expansion units
- Minimal infrastructure
- Affordable hardware components
- Maximum performance with given hardware

Solution

- Global Lustre filesystem
- Multi-protocol gateways
- Lightweight, open-source management software only

Introducing Oak – Main ideas

IO cells (OSS)

- Maximize the number of disks per server
- Use cheap mid-sized JBODs targeted for the Cloud market
- Build a design that:
  - fits a condo model
  - grows without downtime
  - is highly available

MD cells (MDS)

- Extensible and DNE ready
- Use traditional MDS and RAID-10 arrays

Main ideas (I/O cells)

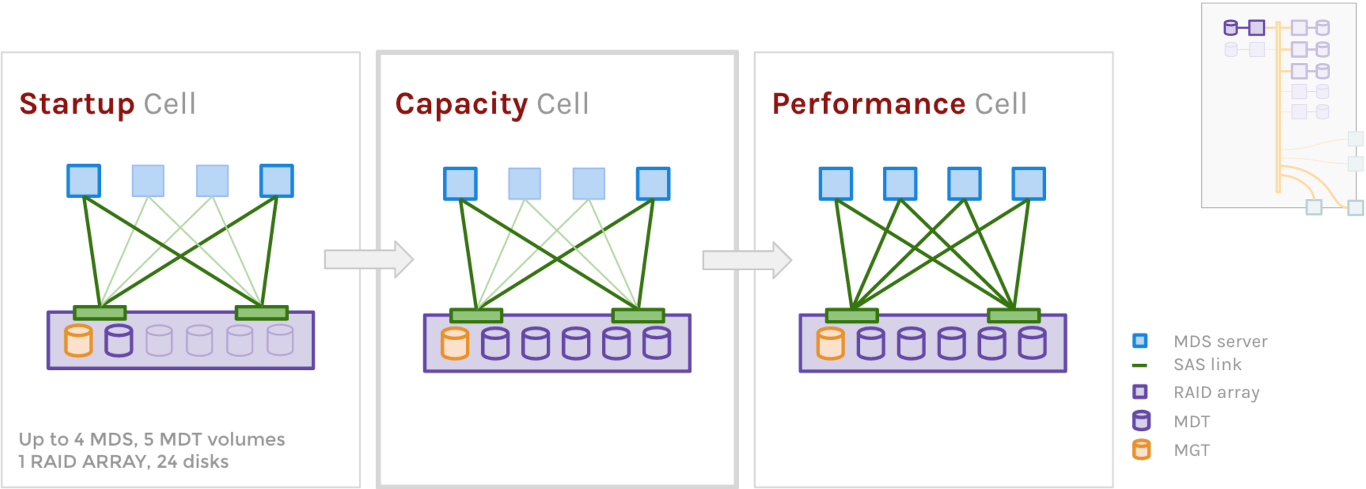
- Use mid-sized JBODs targeted for the Cloud market
  - 60 disks have a great price point
  - 6x 10-disk RAID volumes play well with 2 or 3 OSS

- Link servers and disk arrays through SAS switches
  - connect **more disks** to **less servers**
  - keep number of required servers minimal
  - SAS cheaper than Fiber Channel

- Advantage of a dual-switch setup:
  - easy redundancy (HA) and SAS multipathing
  - no need to touch exiting servers or JBODs to grow (**no downtime**):
    - need more space? plug in a new JBOD
    - need more performance? plug in a new server
  - can still start with bare minimum configuration: 1 JBOD

Scalable cell designs

MD



Startup Cell

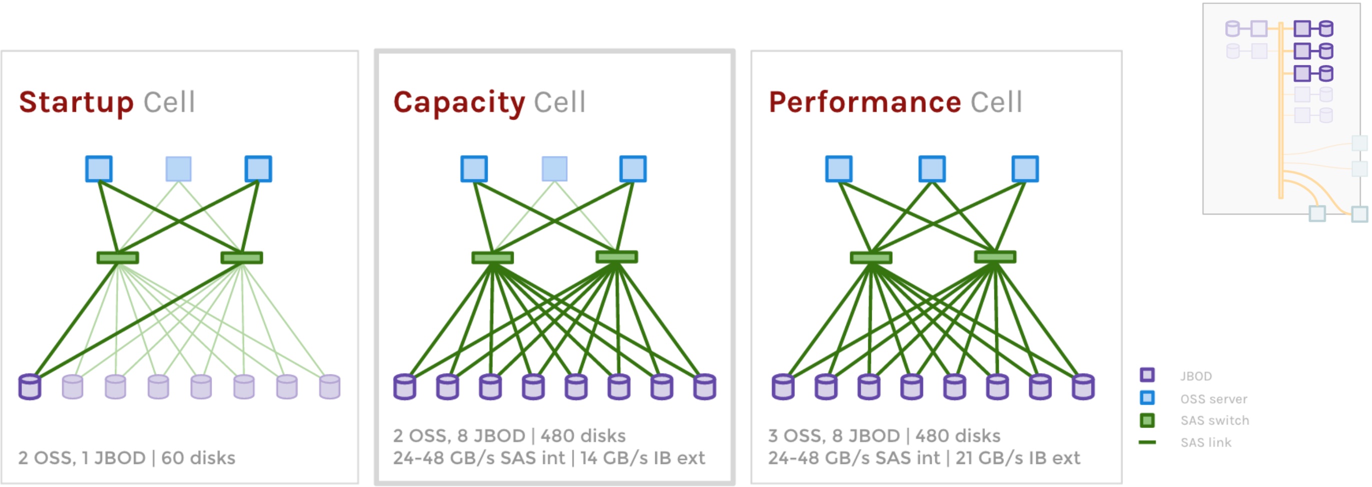
Up to 4 MDS, 5 MDT volumes
1 RAID ARRAY, 24 disks

Capacity Cell

Performance Cell

🟦 MDS server
— SAS link
🟪 RAID array
● MDT
● MGT

IO



Startup Cell

2 OSS, 1 JBOD | 60 disks

Capacity Cell

2 OSS, 8 JBOD | 480 disks
24-48 GB/s SAS int | 14 GB/s IB ext

Performance Cell

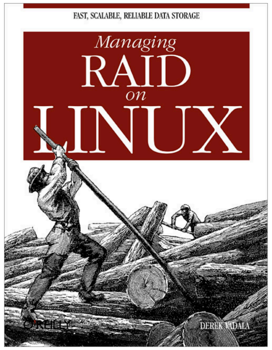
3 OSS, 8 JBOD | 480 disks
24-48 GB/s SAS int | 21 GB/s IB ext

🟪 JBOD
🟦 OSS server
🟩 SAS switch
— SAS link

Main ideas (IO cells cont'd)

- Interconnect IO cells with Infiniband
  - few ports required
  - allows for infinite expansion
  - fill up switch, extend the fabric if needed

- Use Linux software RAID (and ldiskfs)
  - cheaper than hardware RAID (as in free)
  - great RAID-6 performance on Haswell processors
  - memory management control
    - per md array *stripe_cache_size* dynamic setting
  - periodic background data check
  - target failover with multi-mount protection



MD cell hardware components

- Traditional but extensible MD cells
- Similar to our current /scratch config (has been very reliable)
- Startup MD cell up to 900M inodes

2 x MDS Dell R630 2 x Intel E5-2643 v3 128 GB RAM

- 1 x Avago SAS 9300-8e dual-port SAS 12Gb/s HBA
- 1 x Infiniband FDR

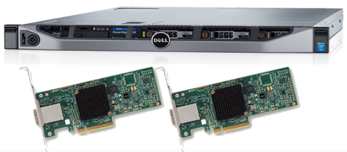
1 x Dell MD3420 2U 24-disk array w/ dual-controller

- 4 x SAS 12Gb/s ports per controller
  - up to 4 MDS in MD cell
- RAID-10 and snapshot support for DR

Dell 900GB 10K rpm SAS hard drives

- Best option (cost and IOPS)
- First MDT with 4 drives in RAID-10

IO cell hardware components



2 x OSS Dell R630 2 x Intel E5-2650v3 256 GB RAM

- 2 x Avago SAS 9300-8e dual-port SAS 12Gb/s HBA
- 1 x Infiniband FDR

2 x SAS Switch Astek A54812-SW (<$3,500 ea)

- 12 x SAS 12Gb/s ports (x4 each)
- Rackable
- Redundant power supplies
- LSI SAS expander chip
- Management port and tools

8 x 60-disk JBOD QCT QuantaVault JB4602 (<$4,000 ea)

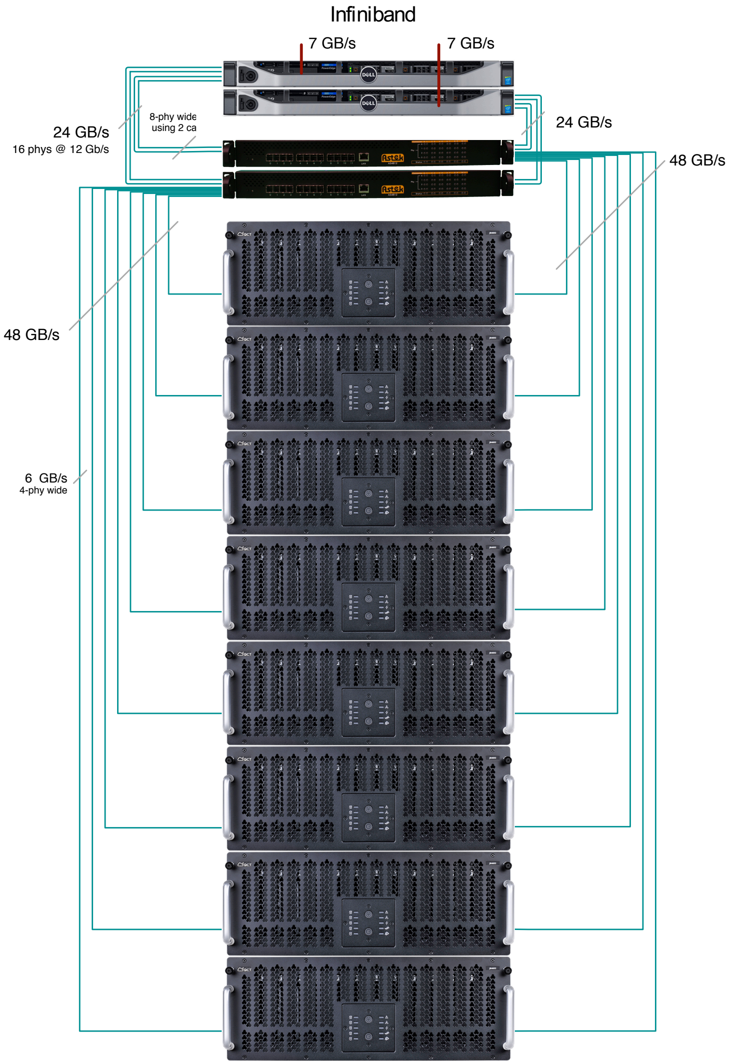
- SAS 12Gb/s ports
- LSI SAS expander chips
- Hot-swappable redundant SAS interface modules (SIM)

480 x Seagate Enterprise Capacity 3.5 v5 hard drive (<$400)

- Makara+ 8TB PMR native SAS 12Gb/s 256MB
- 5 years warranty
- Cheap! Same price for SATA or SAS version

IO cell SAS backend overview

- 8-phy SAS-3 wide ports (at 96 Gb/s) are created by using two HD Mini SAS cables
  - tested with Avago SAS 9300-8e HBAs and Astek SAS switches
  - all 24 SAS switch ports are used
- 48 GB/s SAS backend bandwidth
  - blocking factor of 2:1
  - found to be a good match for md/raid6 avx2x4 algorithm performance: ~22GB/s per server (2 x E5-2650v3)
- Redundant paths to each SAS drive



Stanford University

Oak: other hardware equipment

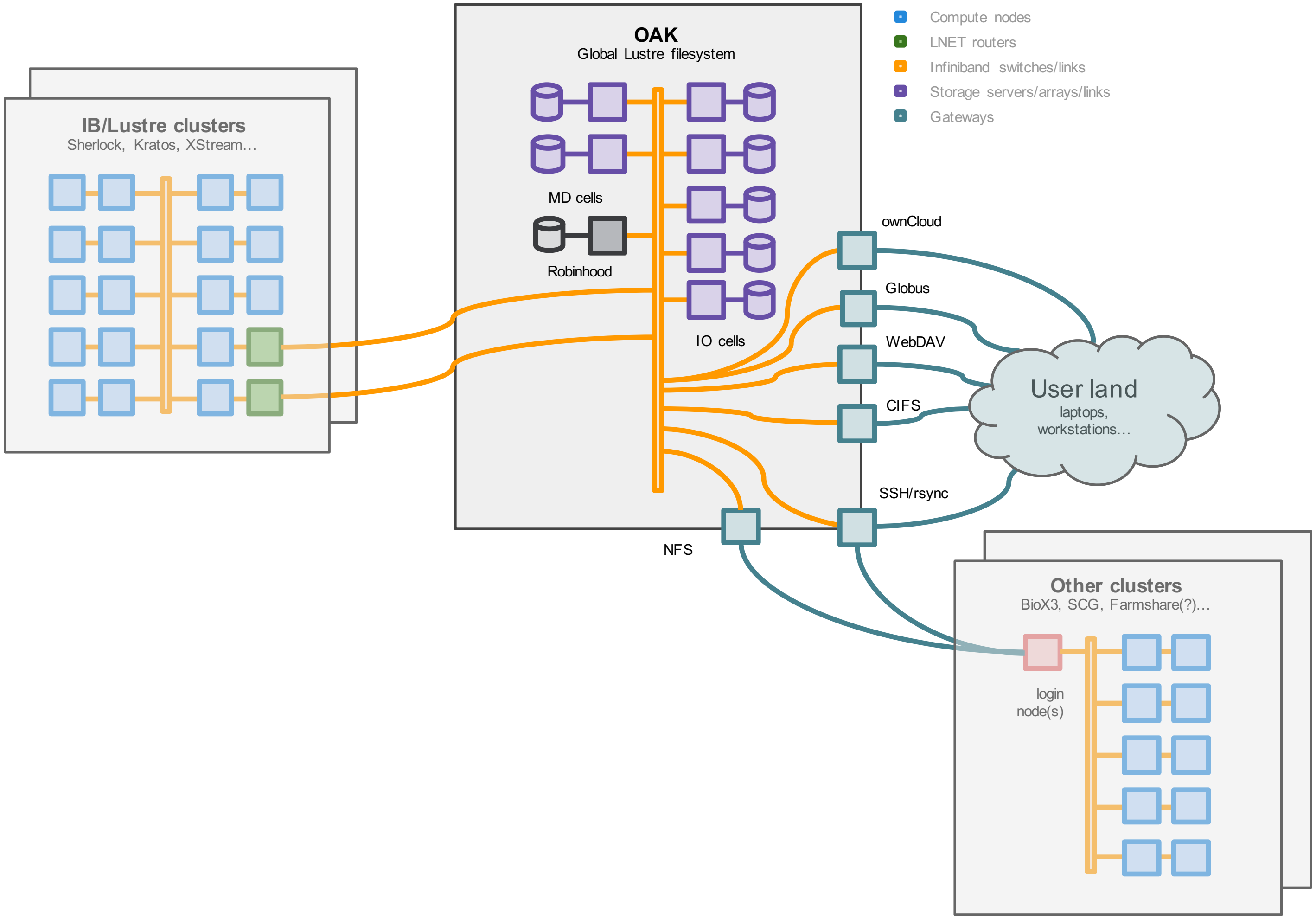
Gateways

- Dell R630 Haswell server
  - 2 x E5-2650 v3 2.3GHz 10C/20T 128 GB RAM
  - 10 gigabit external link
  - SR-IOV support on both 10G and IB

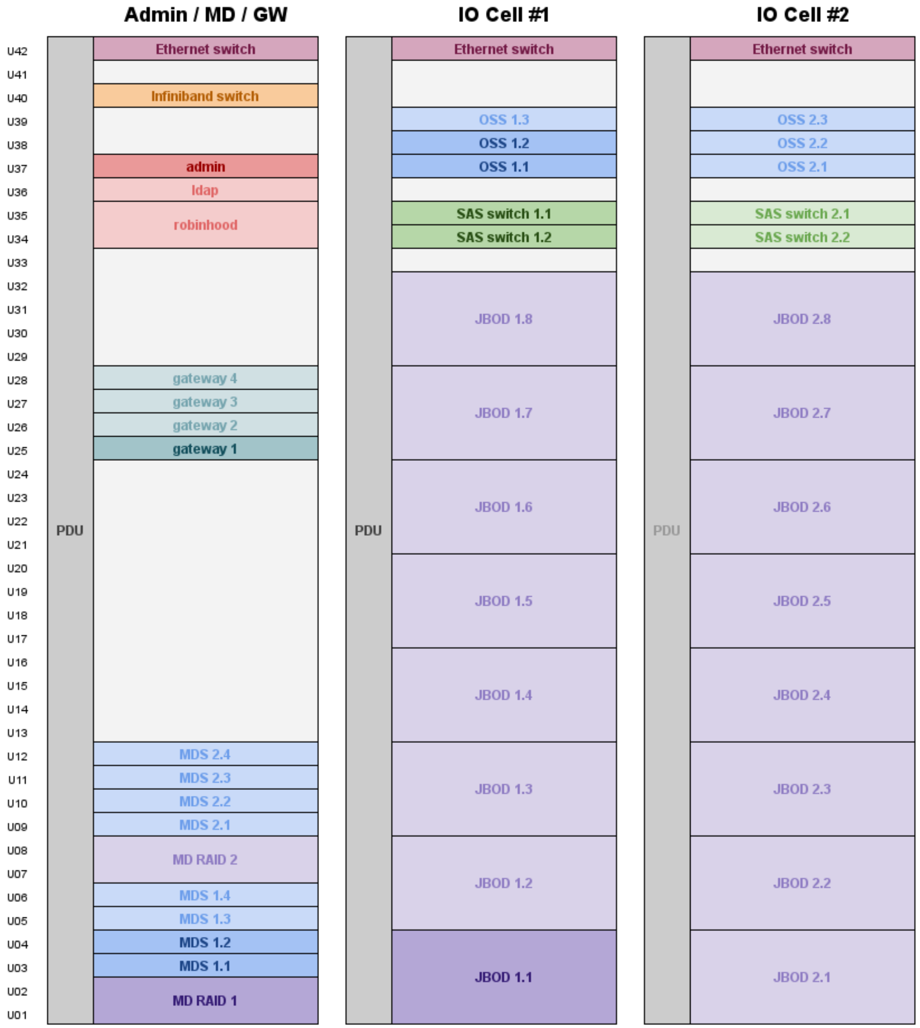
Robinhood server

- Dell R730 Haswell server
  - 2 x E5-2643 v3 3.4GHz 6C/12T 256 GB RAM
  - 4 x 200 GB SSD Intel S3610

Global architecture

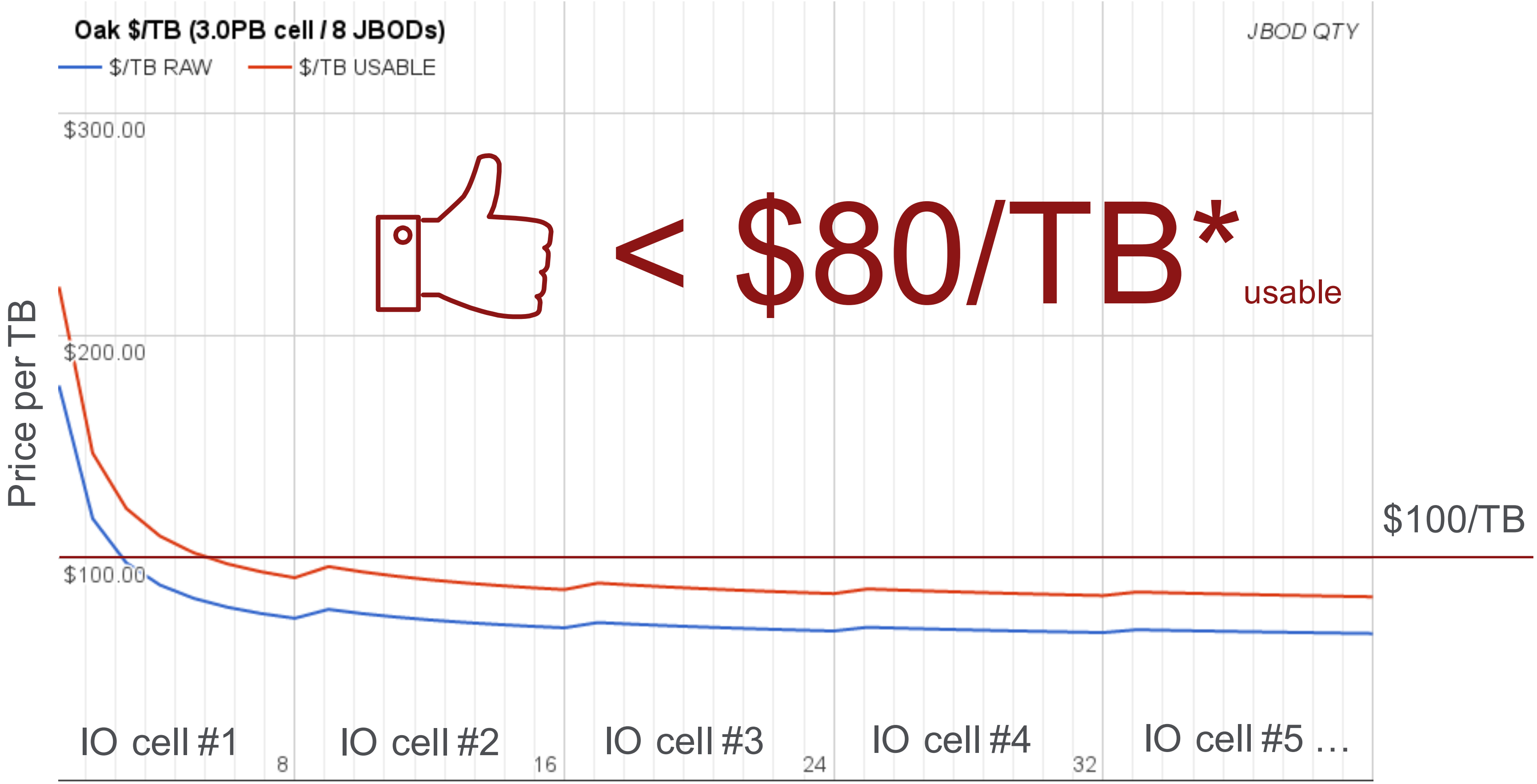


Rack layout



	Admin / MD / GW		IO Cell #1		IO Cell #2	
U42	PDU	Ethernet switch	PDU	Ethernet switch	PDU	Ethernet switch
U41						
U40		Infiniband switch				
U39				OSS 1.3		OSS 2.3
U38				OSS 1.2		OSS 2.2
U37		admin		OSS 1.1		OSS 2.1
U36		ldap				
U35		robinhood		SAS switch 1.1		SAS switch 2.1
U34				SAS switch 1.2		SAS switch 2.2
U33						
U32				JBOD 1.8		JBOD 2.8
U31						
U30						
U29						
U28		gateway 4		JBOD 1.7		JBOD 2.7
U27		gateway 3				
U26		gateway 2				
U25		gateway 1				
U24				JBOD 1.6		JBOD 2.6
U23						
U22						
U21						
U20				JBOD 1.5		JBOD 2.5
U19						
U18						
U17						
U16				JBOD 1.4		JBOD 2.4
U15						
U14						
U13						
U12		MDS 2.4		JBOD 1.3		JBOD 2.3
U11		MDS 2.3				
U10		MDS 2.2				
U09		MDS 2.1				
U08		MD RAID 2		JBOD 1.2		JBOD 2.2
U07						
U06		MDS 1.4				
U05		MDS 1.3				
U04		MDS 1.2		JBOD 1.1		JBOD 2.1
U03		MDS 1.1				
U02		MD RAID 1				
U01						

Predictable hardware cost



< $80/TB* usable

(15PB usable)

- Counting everything! Racks, cables, gateways…

***limitations may apply, prices subject to change**

Lustre-based solution for "cheap" storage systems?

Oak aims to demonstrate that a Lustre-based storage system can actually be affordable when coupled with software RAID and COTS hardware

- Oak's Lustre IO cell with Linux MD RAID facts
  - **3.75 PB raw**
  - **48 OSTs of 64 TB**
  - 2 x 256 GB of RAM = **136 MB of RAM per TB** of raw storage
    - by comparison, Ceph requires **1 GB of RAM per TB** of storage [1]
  - High number of disks per CPU socket: **120 disks per CPU socket**
    - *Yahoo!* has Ceph clusters, each has 54 nodes (2xCPU, 15x4TB HDD each) for a total capacity of 3.2 PB [2]: **7.5 disks per CPU socket**

- Is Lustre the perfect solution for cheap 'n' deep storage?
  - No because Lustre's administration complexity is still a *no go* for many…

[1] from http://docs.ceph.com/docs/jewel/start/hardware-recommendations/

[2] from http://www.nextplatform.com/2015/04/16/inside-the-ceph-exascale-storage-at-yahoo/

Oak’s software components

Operating system	CentOS 7.2
Cluster management	xCAT 2.12
Lustre	Lustre 2.7+ (TBD: Intel Foundation Edition for Education)
Lustre management	shine (master)
Policy Engine / FS monitoring	robinhood v3

Software development in progress

sasutils: Serial Attached SCSI (SAS) Linux utilities and Python library

- Display SAS fabric tree and provide aggregated view of devices
- sas_monitor daemon that generates change notifications
- Based on sysfs (and also sg3_utils and smp_utils)
- Support SES-2 Enclosure Nickname

- Available at https://github.com/stanford-rc/sasutils

```
$ sas_discover
oak-io1-s1
|--host19:  board:  SAS9300-8e  03-25656-02A  SV53345573,  product:  LSISAS3008,  bios:  04.00.00.00,  fw:  12.00.00.00
|  `---8x--expander-19:0  vendor:  ASTEK,  product:  Switch184,  rev:  0004
|          |---1x--end_device-19:0:0  vendor:  ASTEK,  model:  Switch184,  rev:  0004
|          `---4x--expander-19:1  vendor:  QCT,  product:  JB4602 SIM 0,  rev:  1100
|                  |---1x--end_device-19:1:10  vendor:  SEAGATE,  model:  ST8000NM0075,  rev:  E002  size  8.0TB
|                  |---1x--end_device-19:1:11  vendor:  SEAGATE,  model:  ST8000NM0075,  rev:  E002  size  8.0TB
...
```

Software development in progress (cont'd)

shine: tool to setup and manage Lustre file system(s) on a cluster

Add hooks in shine to easily assemble/stop MD arrays on target start/stop.

Other shine-related work: High Availability without Pacemaker

- shine already has target failover support (master branch)
- develop a centralized Lustre supervisor with simple policy rules to
  - check servers and possibly also some clients
  - fence non-responsive server in case of a real issue
  - trigger target failover using shine
  - generate notifications (eg. Email, Slack)

Google Drive

LUSTRE/HSM COPYTOOL

Oak Alley Plantation in Vacherie, Louisiana

Source: wikipedia.org (author: Emily Richardson)

Why Google Drive?

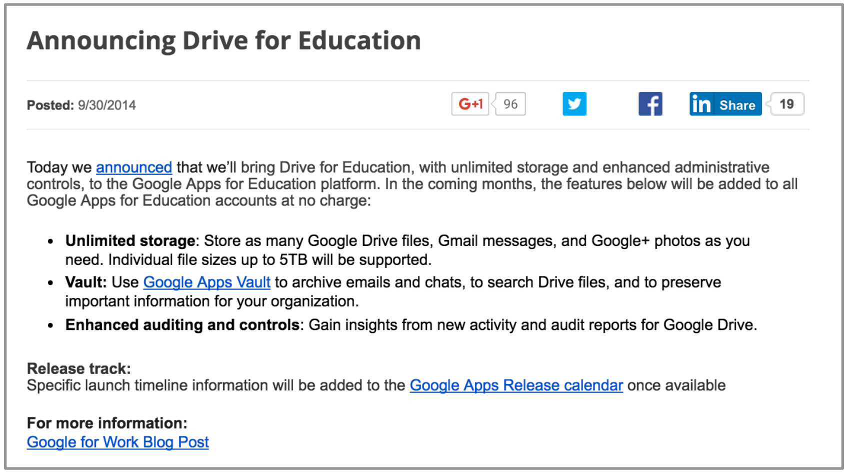
Cloud storage

- Google Drive is a file storage and synchronization service created by Google. It allows users to store files in the cloud, share files, and edit documents, spreadsheets, and presentations with collaborators.

Free and unlimited?

- "We'll bring Drive for Education, with unlimited storage and enhanced administrative controls, to the Google Apps for Education platform" - "Individual file sizes up to **5TB** will be supported"

http://googleappsupdates.blogspot.com/2014/09/announcing-drive-for-education.html



Announcing Drive for Education

Posted: 9/30/2014

Today we announced that we'll bring Drive for Education, with unlimited storage and enhanced administrative controls, to the Google Apps for Education platform. In the coming months, the features below will be added to all Google Apps for Education accounts at no charge:

- **Unlimited storage**: Store as many Google Drive files, Gmail messages, and Google+ photos as you need. Individual file sizes up to 5TB will be supported.
- **Vault:** Use Google Apps Vault to archive emails and chats, to search Drive files, and to preserve important information for your organization.
- **Enhanced auditing and controls**: Gain insights from new activity and audit reports for Google Drive.

Release track:
Specific launch timeline information will be added to the Google Apps Release calendar once available

For more information:
Google for Work Blog Post

lhsmtool_cmd, a new Lustre/HSM agent

Generic command copytool that can be used with any backend.

Features

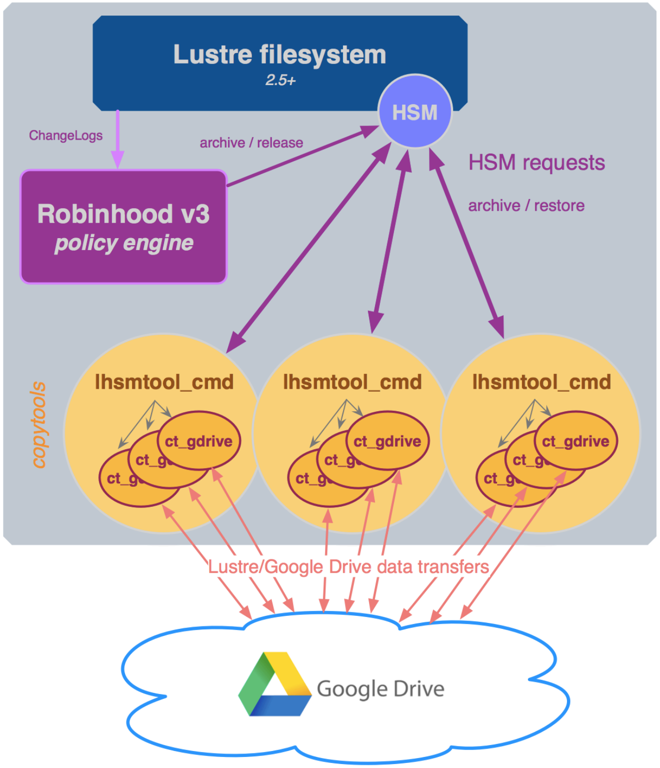
- comes with the Lustre/HSM magic but does NOT perform the copy itself
- spawns pre-defined sub-commands (copytools)
  - child commands inherit parent's file descriptors (basic Unix concept)
  - pass up seekable file descriptor suitable for Cloud API libraries
  - reports progress by sneaking into current file position ☺

Development

- open source, available in robinhood-tools
- created by Henri Doreau (CEA) – thank you, Chairman!
- lhsmtool_cmd is running at Stanford for more than 6 months now

Google Drive Copytool

Implemented because existing tools are not fast enough



ct_gdrive.py

- true *copytool* – copy data between Lustre and Google Drive
- based on the Google API Client Library for Python
- use Lustre FID as file name
- implement recommended **exponential backoff** strategy
- ct_gdrive is used for lhsmtool_cmd archive and restore commands

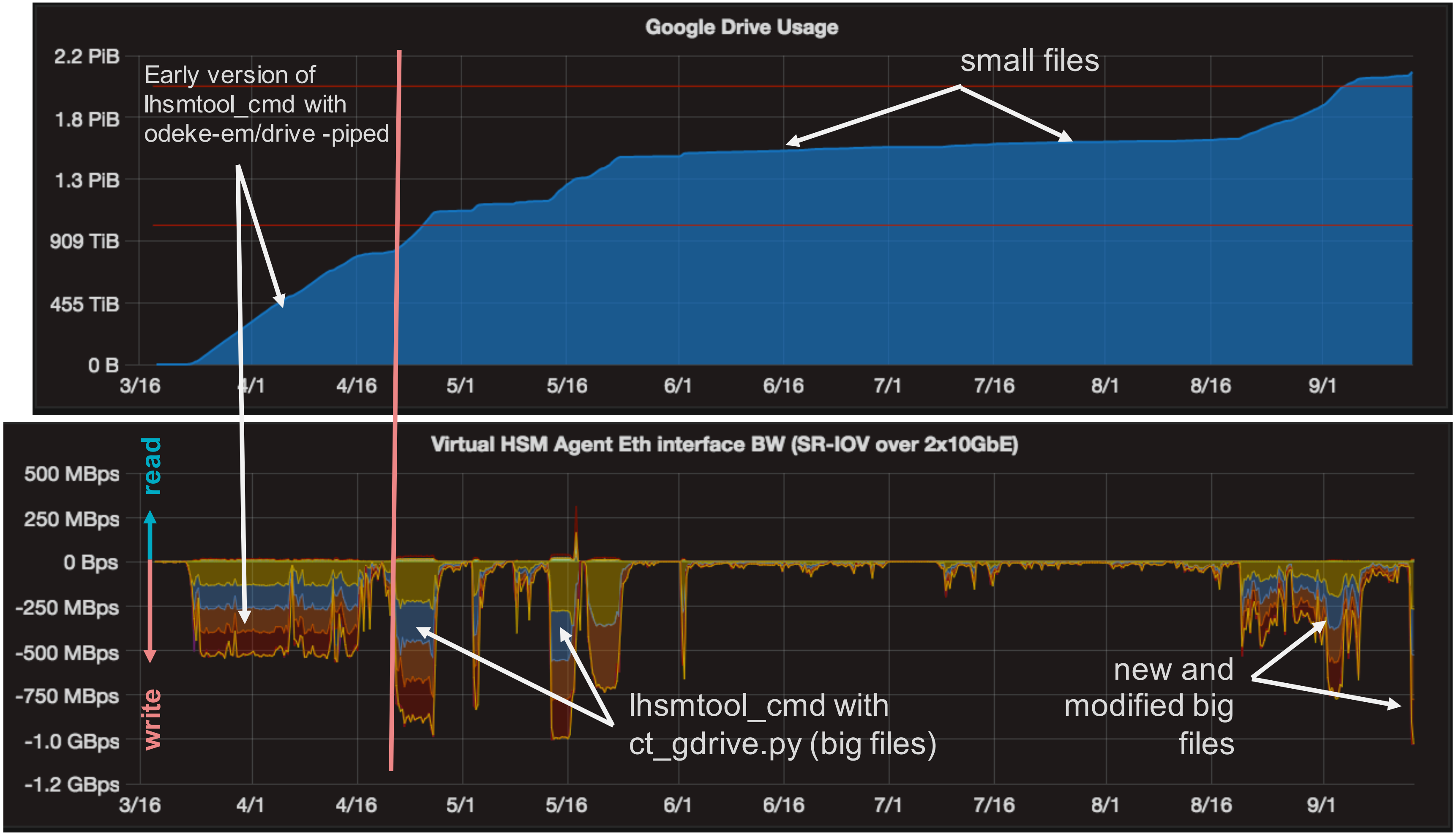
/etc/lhsm_cmd.conf:

```
[commands]
archive = /path/to/ct_gdrive.py --action=push --fd={fd} --fid={fid}  --logging_level=WARNING
        --lustre-root=/lustre  --gdrive-root=0B4bz2HUB5rZtallfYU03ABCDEFg  --creds-dir /path/to/creds/
restore = /path/to/ct_gdrive.py --action=pull --fd={fd} --fid={fid}  --logging_level=WARNING
        --lustre-root=/lustre  --gdrive-root=0B4bz2HUB5rZtallfYU03ABCDEFg  --creds-dir /path/to/creds
```

Available at https://github.com/stanford-rc/ct_gdrive

A Journey with Lustre/HSM to Google Drive

Archiving Sherlock's /scratch



Early version of lhsmtool_cmd with odeke-em/drive -piped

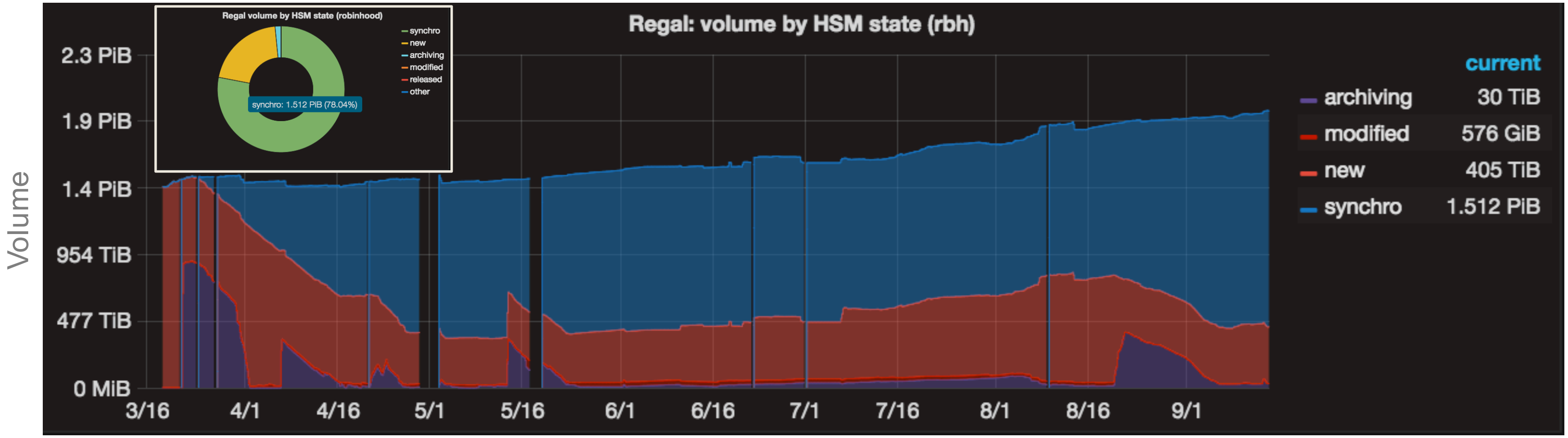
small files

lhsmtool_cmd with ct_gdrive.py (big files)

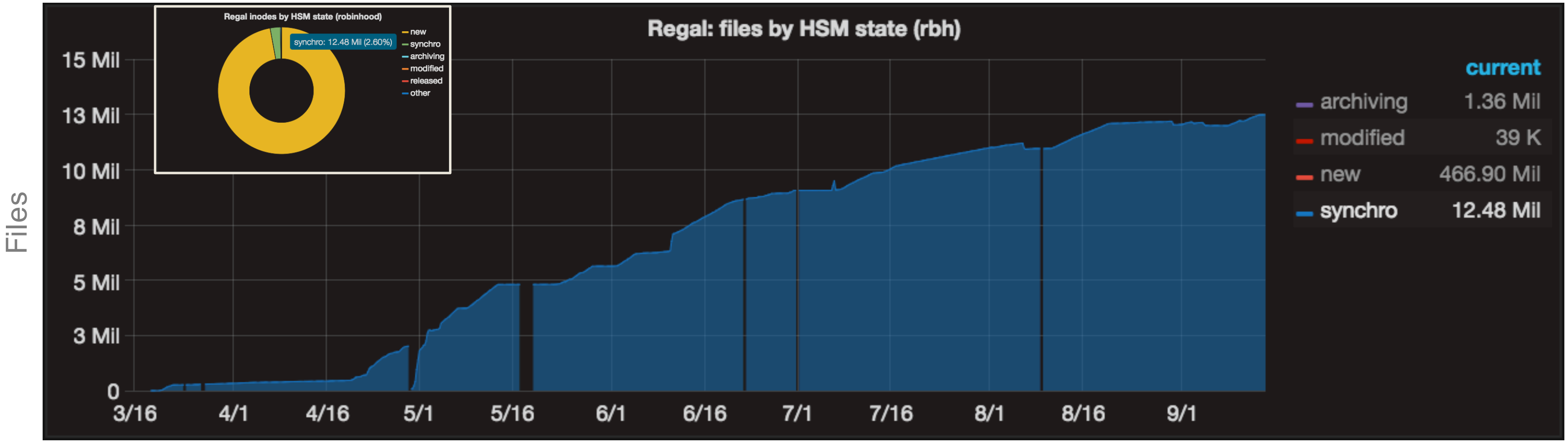
new and modified big files

A Journey with Lustre/HSM to Google Drive (robinhood)

Robinhood Grafana graphs



	current
archiving	30 TiB
modified	576 GiB
new	405 TiB
synchro	1.512 PiB



	current
archiving	1.36 Mil
modified	39 K
new	466.90 Mil
synchro	12.48 Mil

Stanford University

ct_gdrive experimentation: feedback and challenges

Google Drive's interesting features

- file versioning: older versions of files are kept at least 30 days
- search by description (ct_gdrive adds the original file path to the file description)

Google Drive's main limitation: small files

- effective QPS (Query Per Second) **per account** is very low (~3/sec eff.)

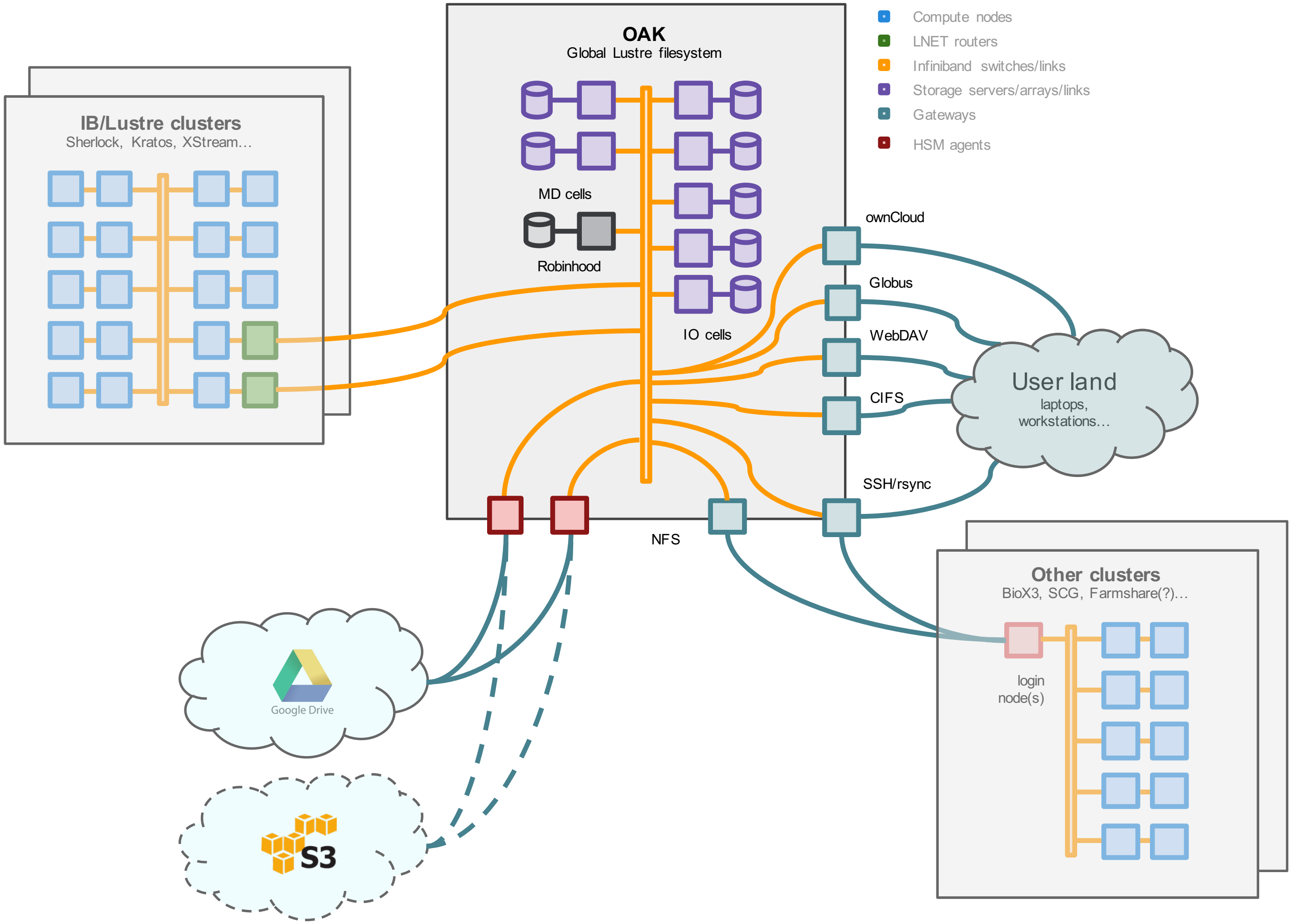
Lustre/HSM's max_requests

- max_requests doesn't play well with QPS limits
- for best performance, set max_requests by file class (robinhood) being archived
- a hsm/**max_requests_per_second** would work in our case
- or delegate this to a user-space process having the global view?
  - cf. discussion on https://jira.hpdd.intel.com/browse/LU-7920

Next Steps?

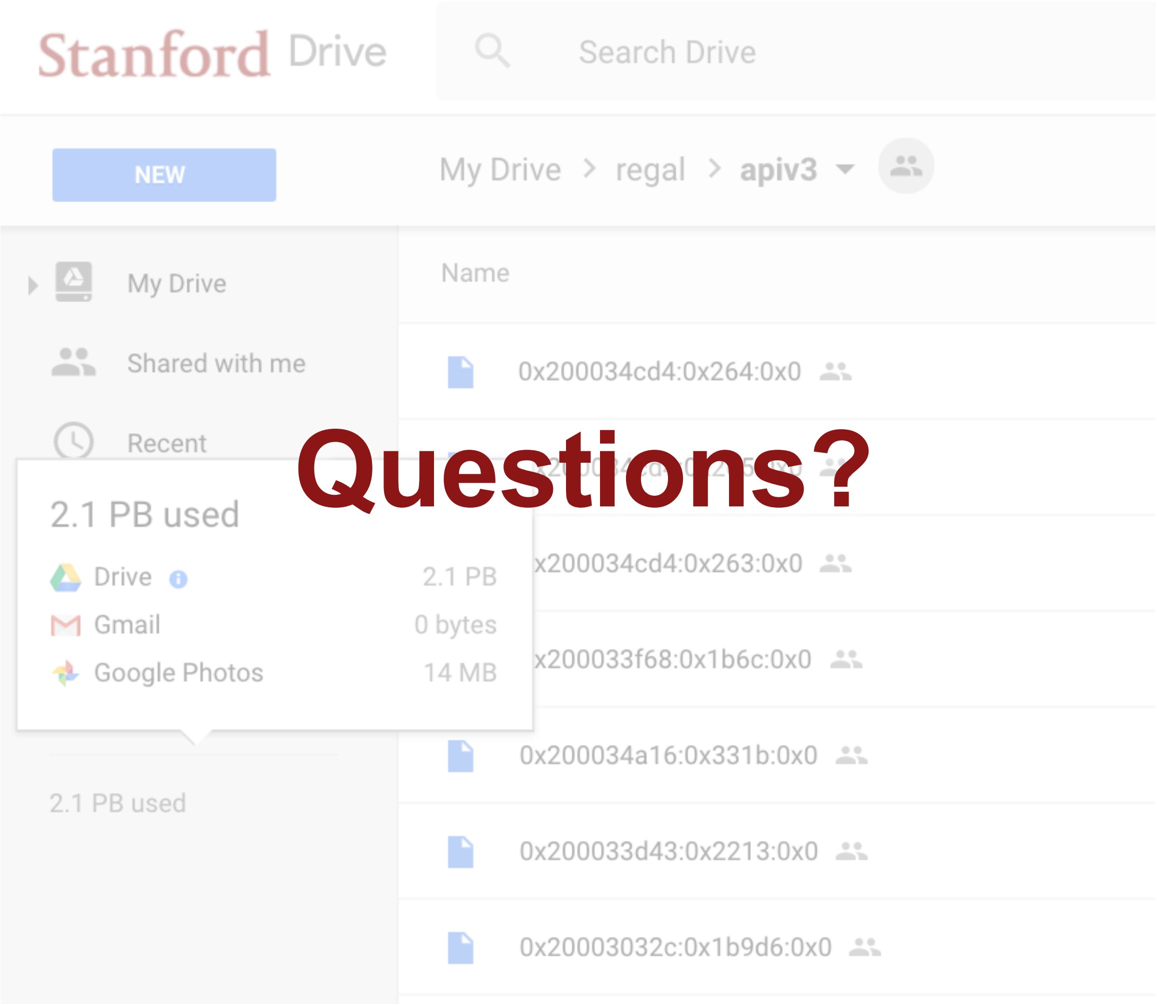
- store files in users' accounts (make ct_gdrive look for user credentials)
- per-user cloud backend selection
  - Google Drive, Google Cloud Storage, S3, etc.

Future HSM-to-the-Cloud architecture with Oak



🟦 Compute nodes
🟩 LNET routers
🟧 Infiniband switches/links
🟪 Storage servers/arrays/links
▨ Gateways
🟥 HSM agents

Questions?



Drive	2.1 PB
Gmail	0 bytes
Google Photos	14 MB

Name

0x200034cd4:0x264:0x0

x200034cd4:0x263:0x0

x200033f68:0x1b6c:0x0

0x200034a16:0x331b:0x0

0x200033d43:0x2213:0x0

0x20003032c:0x1b9d6:0x0