Role of CEPH in the Facility and Distributed Storage

August 1, 2016

Shawn McKee

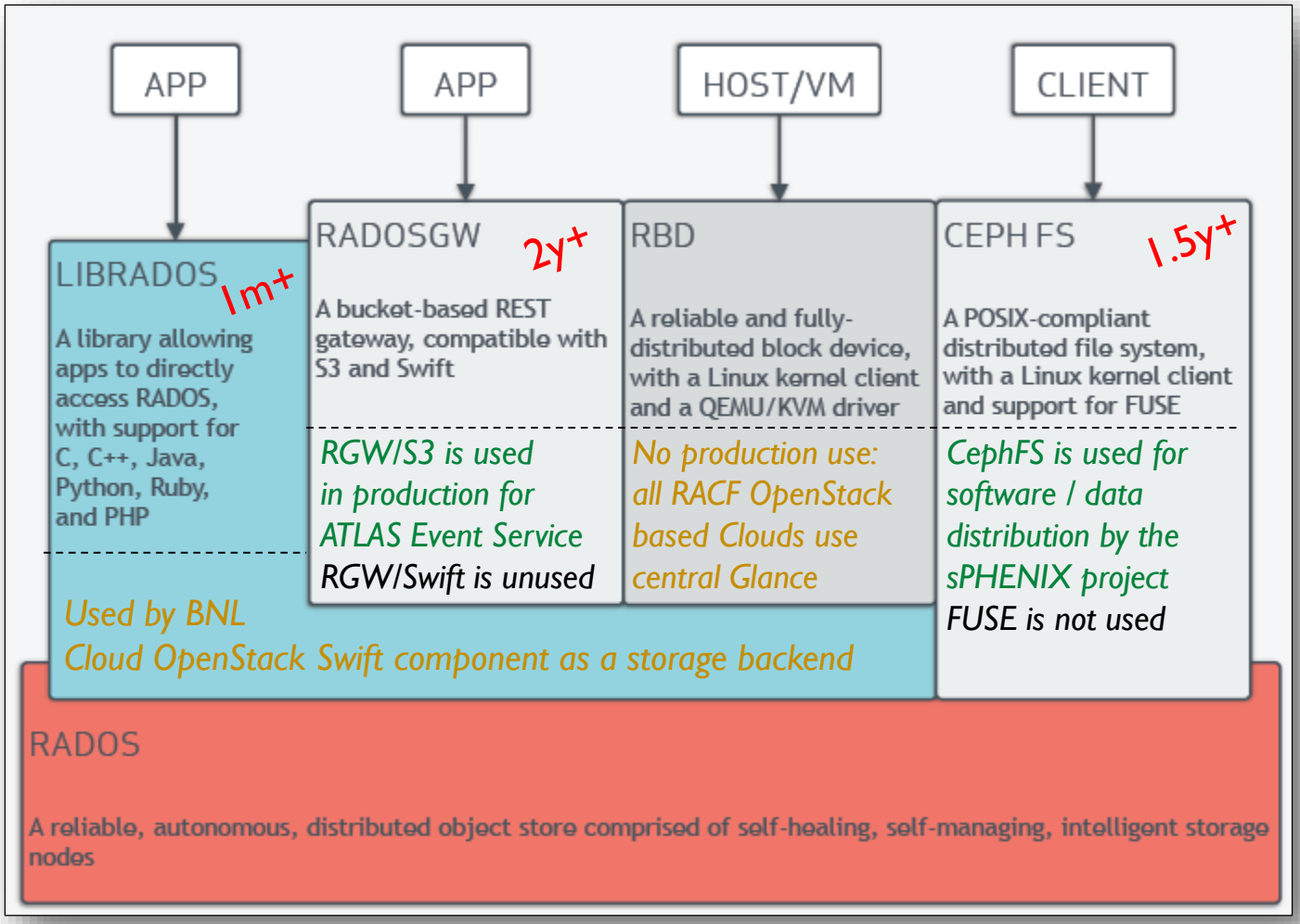
Ceph Features

- **Very active open source project**
- **Open source software components** allow straightforward customization of specific components to provide additional required functionality.
- Easily enables cloud infrastructures, e.g., OpenStack
- Can be built from inexpensive, commercial off-the-shelf (COTS) components.
- Enables storage-integrated **data lifecycle management** possibilities.
- Provides a **software-defined storage service**
- Robust, reliable infrastructure constructed to minimize single points of failure.
- **High performance** achievable by integrating high-bandwidth network links, a large number of disk spindles, large server memory and SSDs.
- **Supports detailed monitoring** of the infrastructure/network topology.

Ceph in (US)ATLAS

- ATLAS has been exploring Ceph at CERN, RAL and BNL for a while.
- We already have quite a bit of experience with Ceph within the US facility
  - Both **BNL** and **MWT2** have been running Ceph for quite a while (1-2+ years)
  - **AGLT2** (via the NSF OSiRIS project) is working on enabling **ATLAS** as a client of OSiRIS(Ceph) this month
- In general Ceph has been shown to be a useful way to organize and enable storage
  - **Is this a way to go for our future needs?**

RACF: Current Use of Ceph Components



The BNL Cloud instance is the first user of our Ceph installations that utilizes the low level object store API of Ceph directly, and thus benefiting from the lowest API-driven overhead possible(still in pre-production phase).

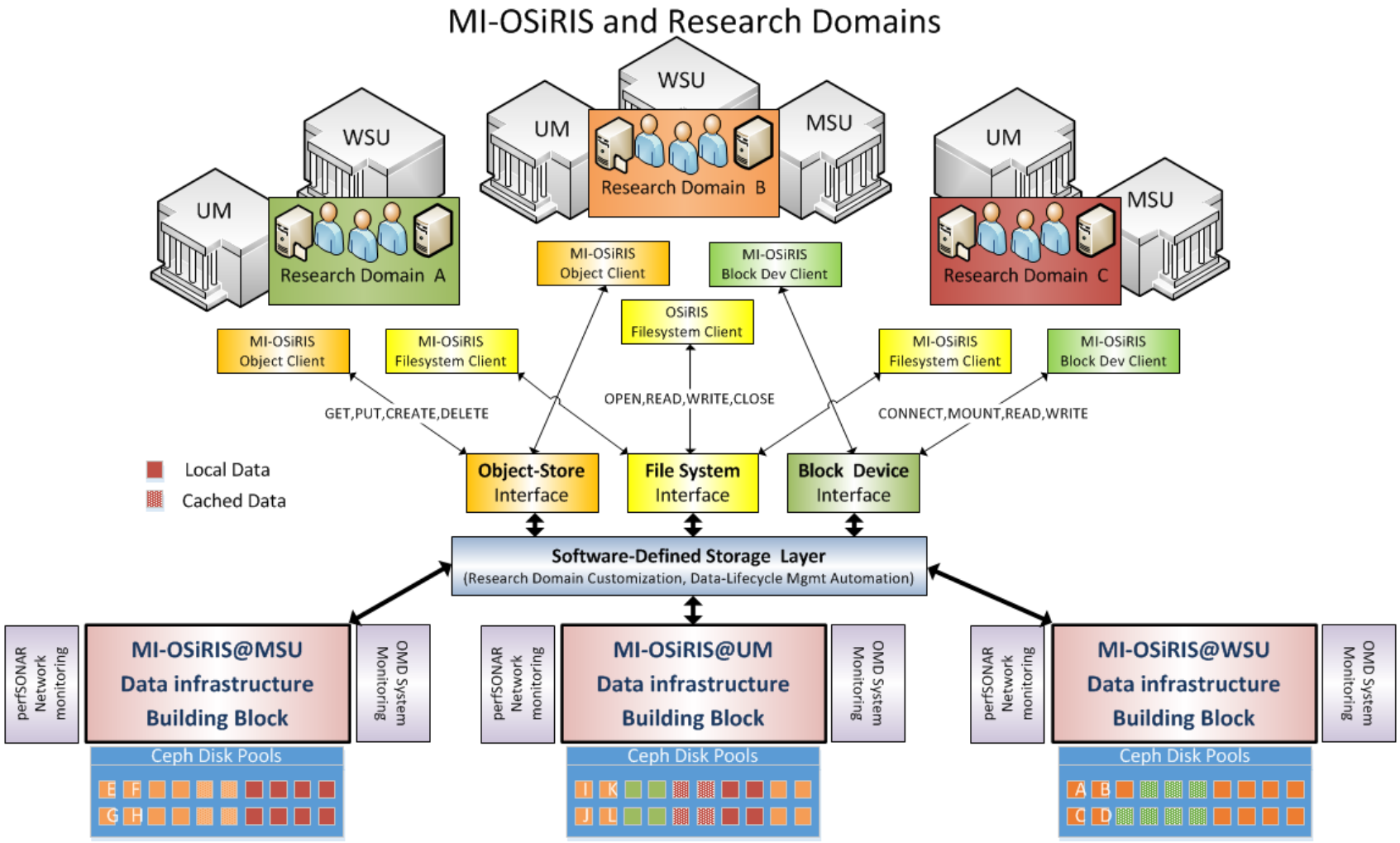
Why Ceph for our Facilities?

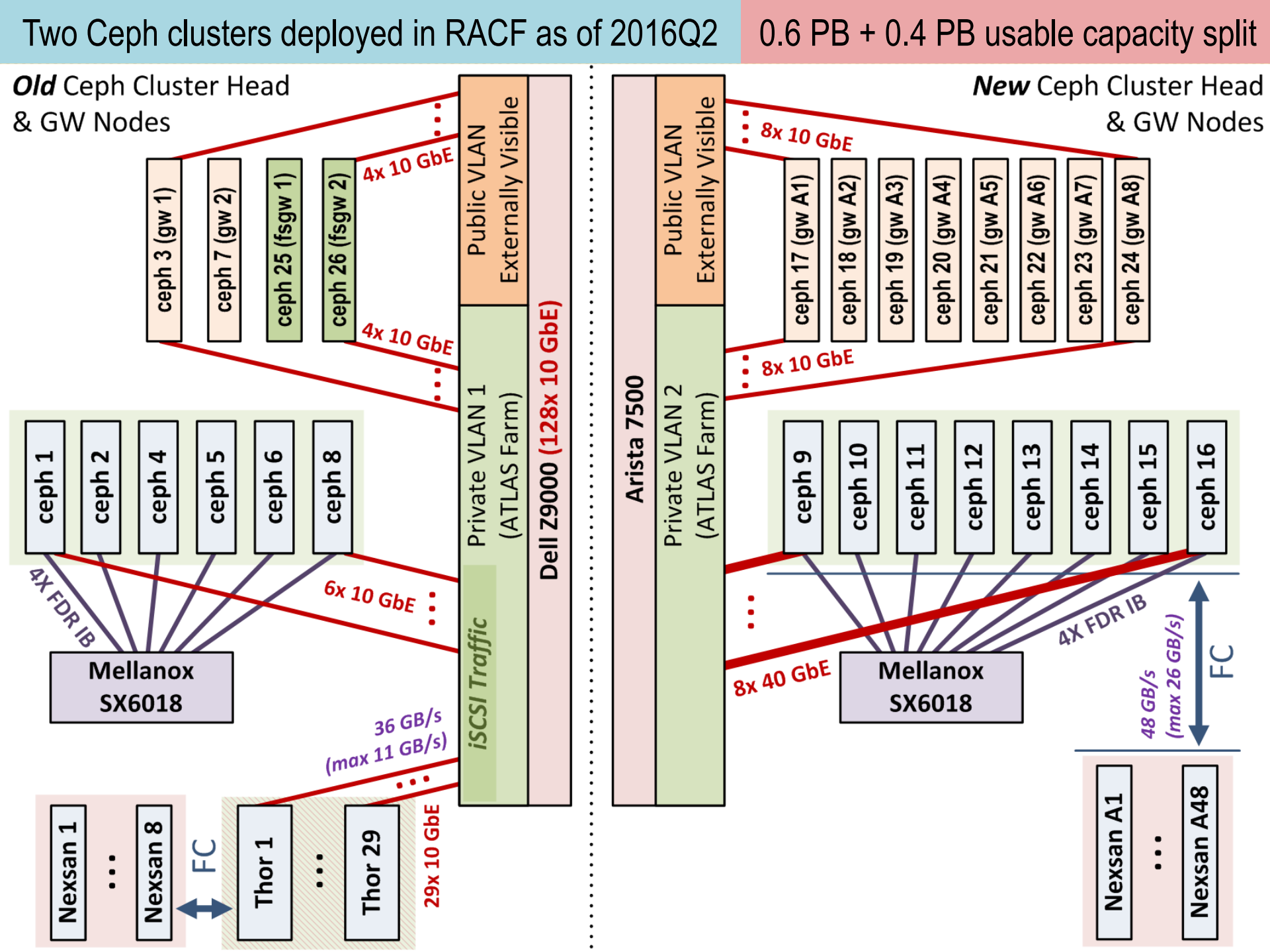
- Ceph gives us an Open Source platform to host our science data
  - Multiple interfaces between users and data are possible
  - Has aspects of Software Defined Storage built-in which give us options for future data lifecycle management automation
- The combination of self-healing and self-managing make it very attractive.
  - Ceph can handle rebalancing and hardware failure automatically
- Logical use-case is to source/sink data from the ATLAS Event Service
  - Object stores can map directly to events: <LFN>/<Event#>
- However we could also benefit in other use-cases
  - dCache over Ceph
  - Re-organizing our storage at Tier-N sites to leverage Ceph
- Ben Meekhof/UM-OSiRIS has a nice online presentation of the Ceph details at https://umich.app.box.com/s/f8ftr82smlbuf5x8r256hay7660soafk

Ceph Versions Deployed

- MWT2 has v0.94 (Hammer)
  - 3 PB (raw), storage BW ~32 GB/sec
  - Uses newer hardware and SSDs
  - Experience testing all Ceph interface types
  - https://indico.cern.ch/event/438205/contributions/2202280/attachments/1292586/1926587/ceph-mwt2.pdf
- BNL has v9.2.1 (Infernalis) on new cluster
  - 1658 TB (raw), storage BW 26 GB/s
  - Extensive testing for event service
  - 2+ years of experience
  - https://indico.cern.ch/event/438205/contributions/2203418/attachments/1292656/1926020/BNL_object_store_tim_2016.pdf
- AGLT2/OSiRIS has 10.2.2 (Jewel)
  - 1440 TB (raw), storage BW 7.5 GB/s
  - Extensive SSD use (tunable)
  - Good networking, new hardware

Logical View of OSiRIS




Two Ceph clusters deployed in RACF as of 2016Q2
0.6 PB + 0.4 PB usable capacity split
Old Ceph Cluster Head & GW Nodes
New Ceph Cluster Head & GW Nodes
ceph 3 (gw 1)
ceph 7 (gw 2)
ceph 25 (fsgw 1)
ceph 26 (fsgw 2)
4x 10 GbE
4x 10 GbE
Public VLAN Externally Visible
Private VLAN 1 (ATLAS Farm)
iSCSI Traffic
Dell Z9000 (128x 10 GbE)
Arista 7500
Public VLAN Externally Visible
Private VLAN 2 (ATLAS Farm)
8x 10 GbE
8x 10 GbE
ceph 17 (gw A1)
ceph 18 (gw A2)
ceph 19 (gw A3)
ceph 20 (gw A4)
ceph 21 (gw A5)
ceph 22 (gw A6)
ceph 23 (gw A7)
ceph 24 (gw A8)
ceph 1
ceph 2
ceph 4
ceph 5
ceph 6
ceph 8
4X FDR IB
6x 10 GbE
Mellanox SX6018
ceph 9
ceph 10
ceph 11
ceph 12
ceph 13
ceph 14
ceph 15
ceph 16
8x 40 GbE
4X FDR IB
Mellanox SX6018
48 GB/s (max 26 GB/s)
FC
36 GB/s (max 11 GB/s)
29x 10 GbE
Nexsan 1
Nexsan 8
FC
Thor 1
Thor 29
Nexsan A1
Nexsan A48

OSiRIS and ATLAS

- OSiRIS targets many science domain stakeholders
  - Year 1: High-energy Physics, High-resolution Ocean Modeling
  - Year 2-5: Biosocial Methods and Population Studies, Aquatic Bio-Geochemistry, Neurodegenerative Disease Studies, Statistical Genetics, Genomics and Bioinformatics, Remaining participants, New Science Domains
- ATLAS (using AGLT2: UM and MSU) will be the first OSiRIS Science Domain incorporated
- **Two use-cases:** 1) OpenStack customized VM storage and 2) PANDA event service
  1. We deploy customized worker nodes based upon workload
  2. **Use Ceph's object-store to provide a high-performance event service for HPC and Cloud resources as well as our Tier-2**
- Homepage http://www.osris.org/

Moving Ahead with OSiRIS/ATLAS

- How best to map ATLAS workflows and needs onto OSiRIS?
  - Event service "server" seems to be a logical target…regional service for nearby HPC or Cloud resources?
  - How does OSiRIS hardware/software scale?
- We would welcome input on integrating ATLAS workflows with OSiRIS capabilities
  - Start integration **this month**!!

Issues / Questions about Ceph

- Is object-store (S3 or Swift) capable of meeting ATLAS event service demands?
  - Load issues have been observed in BNL testing
- When limitations or bottlenecks are seen is it because of:
  - Ceph design and architecture
  - Hardware (old?, use of SSD, below min mem/cpu, disk/controller, network, etc)
  - OS/Kernel (4.4+ is much better
  - Ceph version?
- How to benefit beyond object-store access?

Program of Work?

- We still have a lot of testing and "using" of Ceph to better get a feel for all the issues and opportunities.
  - There is a "feeling" that Ceph will be beneficial but that needs to be quantified
- We should continue our efforts to enable Ceph as an option in our facilities
- **Suggestions, Questions ?**

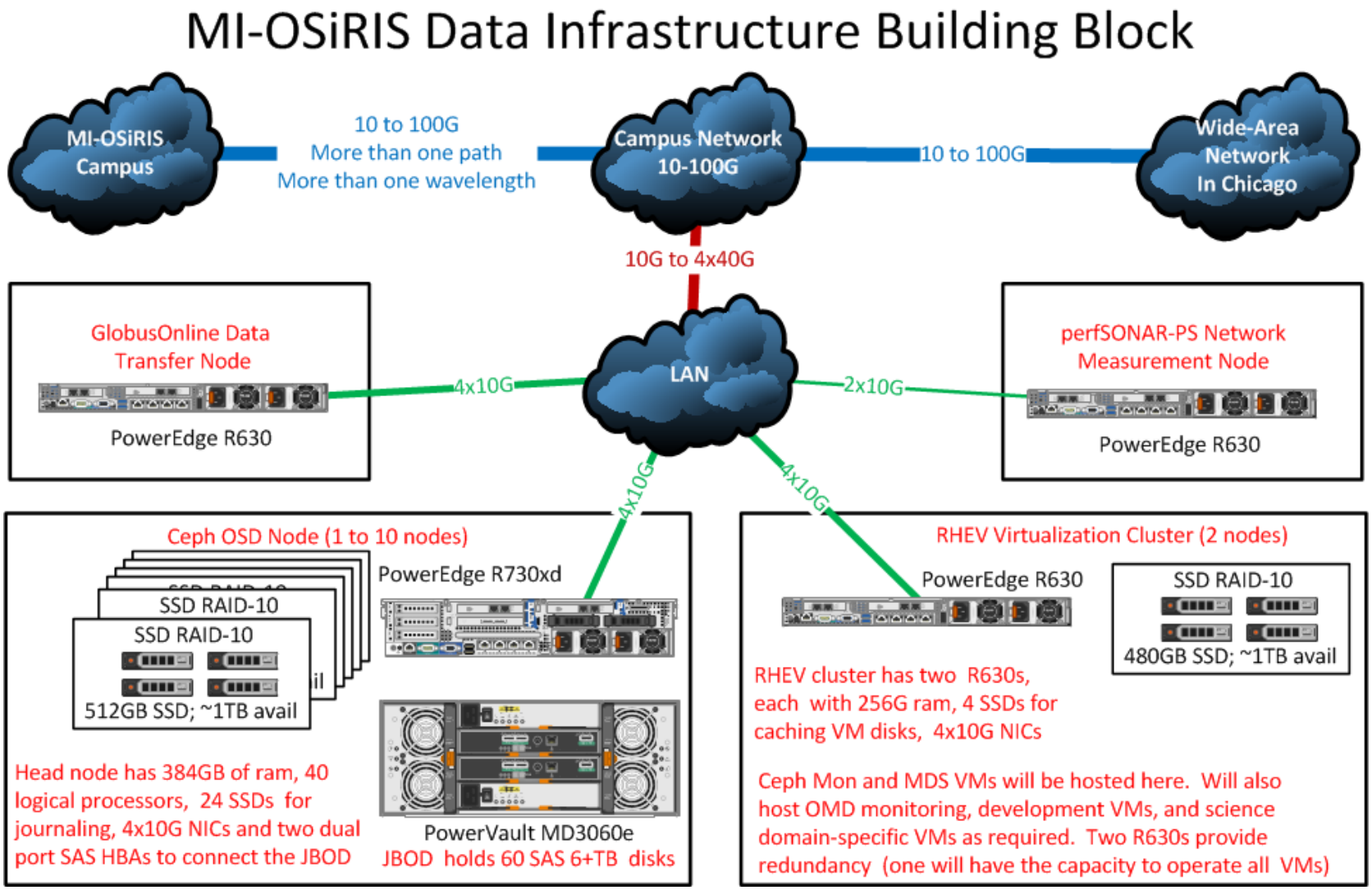
Slides for Additional Details

OSiRIS Links

- Homepage http://www.osris.org (couldn't get osiris.org ☹)
- GitHub: https://github.com/MI-OSiRIS
- OpenProject: https://oproj.aglt2.org/projects/os
- DokuWiki (uses Shibboleth): https://wiki.osris.org/doku.php?id=reference:start

- See OSiRIS technical talk tomorrow afternoon in OSG Technical Session

An OSiRIS Institutional Deployment



MWT2 Ceph Server Hardware

- **Two sets of servers in the current Ceph cluster for OSDs.**
  - Both are R730xd models.
- The first configuration has:
  - 2x Intel E5-2650v3 (40 hyperthreaded cores)
  - 96GB RAM
  - 12x 6TB Seagate 7200RPM disks in JBOD
  - Intel DC P3600 PCIe SSD 400GB
  - 2x 1TB Seagate 7200RPM disks in RAID 1 for the OS
  - single 10Gbps network
  - kernel 4.4.x everywhere
- The second, later configuration is similar:
  - 2x Intel E5-2650v3 (40 hyperthreaded cores)
  - 96GB RAM
  - 14x 8TB Seagate 7200RPM disks in JBOD
  - 2x 200GB mixed-use SSDs
  - 2x 1TB Seagate 7200RPM disks in RAID 1 for the OS
  - single 10Gbps network
  - kernel 4.4.x everywhere

MWT2 Ceph Configuration

- We have three monitors and MDS running on virtual machines. The monitors are fairly modest in their requirements (1-2 cores, few GB of RAM). However we use 64GB RAM VM for the MDS because the inodes are all cached in RAM. So, the more the better.
- For software, we're running the long-term support release Ceph 0.94.x on all nodes. We have plans to upgrade to Jewel once our new dCache storage is on hand. This will require the move to SL7, possibly on clients as well.
- We are configured in such a way that we only have a single network for both cluster and public traffic. We have attached all of our Ceph servers directly to our Juniper EX9200 and have had no noticeable issues with this configuration.
- Our SSDs are all configured to be journals. 1 NVMe per 12 disks or 1 SATA SSD per 7 disks has been totally OK for us.
- Our pools are laid out in the following way:
  - Erasure-coded pool for CephFS with 10 data chunks, 3 parity chunks (30% overhead on storage), jerasure plugin.
  - Replicated pool for CephFS with 3x replication, acting as a cache for the EC pool
  - Replicated pool for CephFS metadata, 3x replication.
  - Erasure-coded RADOSGW pool w/ 10 data chunks, 3 parity chunks, jerasure plugin.
- Due to performance problems, we split our CRUSH map into two sets of disks. We now have a dedicated pool of disks for CephFS cache, and a separate pool for erasure code and RBD disks. Further splitting would be even better.

CephFS at MWT2

- We've been using CephFS for a while now and have quite a bit of experience with it. Outside of the ATLAS facility, CephFS is the backing store for OSG Connect's Stash service. We have observed that the metadata server can be quite slow, since it seems to be effectively locked to a single CPU (I think due to a locking mechanism in their code). Once dynamic subtree partitioning is stable (either Kraken or the L-release?), this will be much, much better. A lot of small files is painful, a lot of files in a single directory is painful.

- EC + Cache tiering exacerbates this. We found the mix of the two on the same disks caused a LOT of blocked operations and very slow I/O. When pools get flushed from the Cache to the EC, they have to be read, re-chunked, parity data added, and re-written to the same set of disks. Slow, slow, slow. The aforementioned CRUSH map changes helped a lot here, but didn't completely solve the problem.

- We initially planned to use CephFS as a separate SE for MWT2, but then due to various logistical problems decided to use it to act for dCache pools. This works OK, but we had to greatly increase our timeouts for the XRootD doors since the data is quite slow to be read out of the EC pools once flushed from the cache. We also saw a number of hangs on dCache servers that were mounting CephFS, for various reasons.

- All clients mount kernel 4.4.x, as the vast majority of the CephFS client code is in the kernel.

- We hope that things improve in Jewel, and suspect multi-MDS will make things much more performant. Needless to say, building a distributed POSIX filesystem is a very ambitious and difficult task.

Rados Block Device at MWT2

- Initial testing had some issues w/ earlier kernels on clients. Once we upgraded to 4.4.x, most of the problems went away. Still see an occasional hang once in a great while.
- Works very well, I've been able to saturate a 10Gbps interface from RBD in synthetic benchmarks. The big problem is that it is VERY expensive in terms of space (3x replication). I have been very wary of 2x replication. In the 2x replication case, for example, If objects go inconsistent on disk for any number of reasons, there's no 3rd reference point to determine which version of the object is correct.
- It is possible to use EC+Cache for RBD, but in my experience it does not work well. In my testing, performance completely sunk whenever objects started getting flushed down into the EC pool. I did not test it further.
- We may continue to use RBD for cloudy things (currently investigating OpenStack, OpenNebula, etc where RBD could back VMs for HA), where 3x replication is not as painful.

S3 at MWT2

- Since the S3 interface is a lightweight shim ontop of Ceph's object interface, it's possible to use erasure coding directly without a cache tier here.
- We have 3 Ceph servers running Civetweb which are providing the S3 interface for MWT2's _ES and _LOGS endpoints. These have not been used heavily and still need to be stress tested.
- So far, though, outlook is good. Scaling is simply a matter of standing up more Civetweb instances.
- We're also using it for our VC3 project to serve software to pilots and have not seen any issues.

RACF Ceph Clusters: Building Blocks

First gen. head nodes, first and second gen. gateways

x18

Dell PowerEdge R420 (1U)

2x 1 TB HDDs in RAID-1 + 1 hot spare
50 GB RAM + 1x 250 GB SSD (up to 10 OSDs)
1x 40 GbE + 1x IPoIB/4X FDR IB (56 Gbps) – *Head nodes*
2x 10 GbE – *Gateways*

Second gen. head nodes

x8

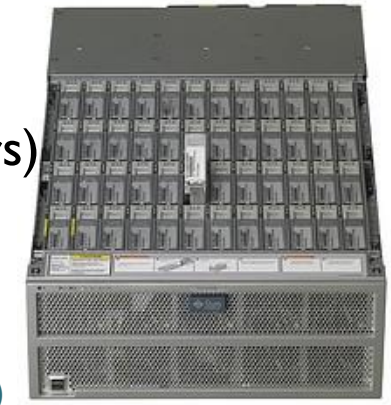
Dell PowerEdge R720XD (2U)

8x 4 TB HDDs in RAID-10 + 2 hot spares
128 GB RAM + 2x 250 GB SSDs (up to 24 OSDs)
1x 40 GbE + 1x IPoIB/4X FDR IB (56 Gbps) + 12x 4 Gbps FC ports

Storage backend (retired ATLAS dCache HW RAID disk arrays)

iSCSI export nodes

SUN Thor servers (Thors)



48x 1 TB HDDs under ZFS
8 GB RAM
1x 10 GbE
4x 4 Gbps FC (no longer used)

FC attached storage arrays

Nexsan SATABeast arrays (Thors)

40x 1 TB HDDs in HW RAID-6 + 2 hot spares
2x 4 Gbps FC (no longer used)

x56

BNL Ceph Details

- Current OSD count: 192 / 8 OSD hosts
- **Total raw capacity: 1658 TB**
- Replication schema used: **simple replication factor 3x** (erasure codes 4+3 layout tested, but found to be delivering significantly less performance).
- **Storage backend write throughput: 26 GB/s**, limited by RAID controller performance (48 FC attached disk arrays, 2k spinning drives behind 48 RAID controllers in total, most of the HDDs are of 1 TB capacity).
- All the storage used is still a retired BNL ATLAS dCache storage.
- Largest number of object stored in a single Ceph instance: >100M.
- Maximum parallel connections allowed through the Rados/S3 gateway subsystem (6x 10 GbE attached hosts): 24k, 4k hard limit per gateway host.
- Maximum write throughput observed with S3 traffic originating from local sources at BNL: 1.2 GB/s (limited by a 10 GbE interface at the source; object sizes used are 7 MB and larger).
- Maximum write throughput observed with S3 traffic over the WAN (originating from ANL in ~400 concurrent connections): 950 MB/s (through the pipe limited to 10 Gbps on the ANL side).
- Maximum throughput observed with local clients mounting CephFS: 8.7 GB/s (limited by the aggregate client host network connectivity).
- Maximum throughput observed via Swift/Ceph gateway serving the local OpenStack based BNL Cloud installation: 1.7 GB/s.