OSiRIS: a distributed Ceph deployment using software defined networking for multi-institutional research

Shawn McKee¹, Ezra Kissel³, Benjeman Meekhof², Martin Swany³, Charles Miller⁴, Michael Gregorowicz⁵

¹ Physics Department, University of Michigan, Ann Arbor, USA
² Advanced Research Computing - Technology Services, University of Michigan, Ann Arbor, USA
³ Center for Research in Extreme Scale Technologies, Indiana University, Bloomington, USA
⁴ Institute for Cyber-Enabled Research, Michigan State University, East Lansing, USA
⁵ Computing and Information Technology Department, Wayne State University, Detroit, USA
E-mail: smckee@umich.edu

Abstract. We report on the first year of the OSiRIS project (NSF Award #1541335, UM, IU, MSU and WSU) which is targeting the creation of a distributed Ceph storage infrastructure coupled together with software-defined networking to provide high-performance access for well-connected locations on any participating campus. The project’s goal is to provide a single scalable, distributed storage infrastructure that allows researchers at each campus to read, write, manage and share data directly from their own computing locations. The NSF CC*DNI DIBBS program which funded OSiRIS is seeking solutions to the challenges of multi-institutional collaborations involving large amounts of data and we are exploring the creative use of Ceph and networking to address those challenges.

While OSiRIS will eventually be serving a broad range of science domains, its first adopter will be the LHC ATLAS detector project via the ATLAS Great Lakes Tier-2 (AGLT2) jointly located at the University of Michigan and Michigan State University. Part of our presentation will cover how ATLAS is using the OSiRIS infrastructure and our experiences integrating our first user community. The presentation will also review the motivations for and goals of the project, the technical details of the OSiRIS infrastructure, the challenges in providing such an infrastructure, and the technical choices made to address those challenges. We will conclude with our plans for the remaining 4 years of the project and our vision for what we hope to deliver by the project’s end.

1. Introduction

The OSiRIS project has successfully connected four campuses with a software defined networking and storage system that will allow the seamless sharing of large datasets. In the first year of the project, we established an infrastructure to deploy and manage project services, deployed 4.5 Petabytes of Ceph storage, began designing AAA (Authentication, Authorization, and Accounting) infrastructure bridging Ceph to authentication federations, and began to store data from our first science domain. The remainder of the paper will cover these accomplishments in more detail.
2. The OSiRIS Project

OSiRIS (Open Storage Research InfraStructure) is a collaboration of scientists, computer engineers and technicians, network and storage researchers and information science professionals from University of Michigan/ARC-TS (UM), Michigan State University/iCER (MSU), Wayne State University (WSU), and Indiana University (IU) (focusing on SDN and network topology).

We are one of 4 NSF “Campus Computing: Data, Networking, Innovation: Data Infrastructure Building Blocks” (CC*DNI DIBBs) projects funded in 2015. OSiRIS is prototyping and evaluating a software-defined storage infrastructure, initially for our primary Michigan research universities, designed to support many science domains. Our goal is to provide transparent, high-performance access to the same storage infrastructure from well-connected locations on any of our campuses. By providing a single data infrastructure that supports computational access in-place, we can meet many of the data-intensive and collaboration challenges faced by our research communities and enable them to easily undertake research collaborations beyond the border of their own universities.

A single scalable infrastructure is easier to build and maintain than isolated campus data silos. Data sharing, archiving, security, and life-cycle management can all be implemented under one infrastructure. At the same time, our architecture will allow the configuration for each research domain to be optimized for performance and resiliency (Figure 1).

3. Ceph in OSiRIS

Ceph is a distributed object storage system that gives us a robust open source platform to host multi-institutional science data. The core of Ceph is the Reliable Autonomic Distributed Object Store. RADOS is self healing, self manages replication, and has excellent scalability and performance[1]. RADOS supports multiple data interfaces including POSIX, S3 compatible
object storage, and kernel block devices. Ceph has sophisticated allocation mapping using the Controlled Replication Under Scalable Hashing (CRUSH) algorithm to allow us to customize data placement by use case and available resources[2].

Our Ceph deployment is distributed across sites at WSU, MSU, and UM. We have also had experience distributing the deployment to a site geographically farther away (Utah) with a slight loss of performance but retaining functionality[3]. Ceph allows us to choose the level of replication among these sites based on the needs of participating science domains. Typically our highest level of data resiliency would be provided by having one or more replicas at each site. Ceph also has options for creating Erasure Coded data pools which provide configurable redundancy similar to RAID. However, these pools cannot be used for all Ceph access methods at this point. Upcoming versions of Ceph will allow EC pools in more contexts[4]. EC pools can also be overlaid with a conventional replicated pool in a caching-mode which would allow an EC pool to back any given Ceph access protocol.

4. OSiRIS Network Management Abstraction Layer (NMAL)
Another important part of the OSiRIS project is active network monitoring, management and network orchestration via the NMAL. Network topology and perfSONAR Periscope monitoring components deployed to hosts and switches ensure that our distributed system can optimize the network for performance and resiliency through SDN (Software Defined Networking) control of components.

Main components in NMAP include BLiPP, UNIS, and an SDN controller.

- BLiPP - Basic Lightweight Periscope Probe. BLiPP agents may reside in both the end hosts (monitoring end-to-end network status) and dedicated diagnose hosts inside networks.
- UNIS - Unified Network Information Store. The Periscope UNIS data store exposes a RESTful interface for information necessary to perform data logistics. The data store can hold measurements from BLiPP or network topology inferred through various agents.
- SDN Controller - Driven by information collected in UNIS, an SDN controller can dynamically modify network topologies to enable the best path between clients and data and between internal OSiRIS components (i.e., for Ceph replication).

5. OSiRIS Access Assertions
The OSiRIS approach to authentication is to use identity federations and avoid managing authentication accounts. Federation participants will use their local providers to verify identity and create OSiRIS Access Assertions (OAA), which are then used to create bearer tokens (OSiRIS Access Tokens; OATs) that can be used with a variety of OSiRIS interfaces or use cases. Virtual OSiRIS organizations should be able to self-organize and manage members and roles via OSiRIS services.

In detail, session and affiliation data are first pulled into OSiRIS from SAML2 (Security Assertion Markup Language) Assertions made by Identity Providers at federation participant organizations. Valid SAML2 sessions are combined with OSiRIS Access Assertions to create the aforementioned OAT bearer tokens (Figure 2)

Ultimately the OSiRIS project will provide a service based upon our access assertions and other existing software and services like InCommon[5], Shibboleth[6] and CoManage [7] to enable our science domain users to self-organize and control access to their own storage within OSiRIS.

6. Science Domain Engagements
Our project has started engaging with the ATLAS experiment to serve as a store of ATLAS physics events for compute jobs. ATLAS compute jobs will use the OSiRIS Ceph S3 gateway to read/write single events. S3 refers to the object storage access protocol used and popularized
Figure 2. OSiRIS Client Authentication with OAA

by Amazon Simple Storage Service. By reading only a single event at a time, ATLAS can leverage transient computing resources to run short jobs as available. Preliminary testing with ATLAS client code has been successful. At the time of that testing, we had only one S3 gateway online. We have since added two more for a total of three spread across our three main storage sites. At this point, we are coordinating heavier load-testing with ATLAS. To prevent resource domination we provision multiple S3 instances per host and isolate system resources for domain specific instances of the S3 gateway with Linux kernel "CGroups". It is yet to be determined if the ATLAS portion of three hardware nodes is a sufficient resource to satisfy ATLAS compute needs.

There are other known methods for accessing ATLAS resources in Ceph. The Dynafed project[8] from CERN includes plugins to access S3 endpoints such as ours via the ATLAS Dynafed namespace. The GridFTP plugin which was created and is used by the RAL ATLAS Tier-1[9] is also a potential interface. We will be exploring both these options for ATLAS.

The US Naval Research Lab is collaborating with researchers at UM to share their high-resolution ocean models with the broader community by storing them with OSiRIS. We currently provide space via CephFS and accessed through our transfer gateways via scp, FDT, and Globus. For collaborators geographically near OSiRIS sites we would also like to enable direct CephFS mounts once our authentication infrastructure is more fully enabled. There has been some discussion of using S3 type storage with object meta-data mapped to a URL for direct retrieval,
but at this time we are staying with a simpler direct-copy approach.

In the longer term, this is our road-map for science domain engagement:

- End Year 1: High-energy Physics, High-resolution Ocean Modeling (now ongoing)
- End Year 2: Biosocial Methods and Population Studies, Aquatic Bio-Geochemistry
- End Year 3: Neurodegenerative Disease Studies
- End Year 4: Statistical Genetics, Genomics and Bioinformatics
- Year 5: Remaining participants, New Science Domains

**Figure 3.** Resource Deployment with Foreman and Smart Proxies

**Figure 4.** Resource Deployment with Foreman and Smart Proxies
7. OSiRIS Deployment

As diagrammed in Figure 3, deployment and management of OSiRIS resources leverages existing open-source technologies and collaborative workflows. Infrastructure services are hosted as virtual machines on KVM/Libvirt systems. These are built on Foreman, which is lifecycle management tool for physical and virtual servers. Foreman is structured with a remote ‘smart proxy’ architecture allowing for a small proxy build server to be controlled at any remote site by one central instance. It is also capable of provisioning new VMs and can thus be used as a complete integrated solution to configure and deploy new VM instances at any of the OSiRIS sites (Figure 4).

The well-known Puppet tool was our choice for configuration management in OSiRIS. Puppet’s yaml-based hiera database for configuration data simplifies storing parameters specific to site, node type, node role, etc. In fact, any arbitrary hierarchy based on any arbitrary node data can be used to organize parameters used for configuration. Puppet also supports a tool called ‘r10k’ which is designed to automatically map Git branches into isolated Puppet configuration environments. Using this tool and shared Git repository, we have a workflow that enables project admins to collaborate on configuration enhancements and changes without disrupting the stability of our production config. The r10k/git/puppet workflow is a common best practice[10].

8. Next Steps

An important next phase for OSiRIS is the full integration of OSiRIS Access Assertions into our Ceph and NMAL systems. This is a requirement to on-board future science domains and allow users in those domains to access OSiRIS with federated credentials as well as allow domains to self-manage collaboration.

As regards ATLAS and OSiRIS, the next few months will see full load-testing of our systems when confronted with a large number of ATLAS jobs using us as an event service. We would also like to see integration with the ATLAS Dynafed.

To reach it’s goals, the project faces a number of interesting challenges that need to be addressed in the next one to two years:

- Building up a tool-kit of client interface options spanning the range of laptop to cluster systems.
- Implementing software-defined networking (SDN) orchestration of both science-user and OSiRIS infrastructure network connectivity.
- Enabling science domain specific detailed metrics to track, manage and optimize use of OSiRIS.
- Developing automated data life-cycle meta-data creation for users of OSiRIS.
- Integrating authorization assertions made by science domains into Ceph and the NMAL components at a fine-grained level.
- Creating an OSiRIS authorization service which leverages institutional identities and allows science domains to self-organize and control use of their OSiRIS resources.
- Securing and hardening the OSiRIS infrastructure.

We would like to note that the project has made significant progress on the last two bullet points through its ongoing engagement with the Center for Trusted Scientific Cyberinfrastructure[11]. While the initial engagement should complete by April 2017, the recommendations, feedback and security evaluation of OSiRIS will have provided a sound basis on which to complete those items during the rest of 2017.
9. Conclusions
The OSiRIS project goal is enable scientists to collaborate on data easily and without building their own infrastructure. Scientists should be able to use our infrastructure by leveraging their existing institutional identities for authentication and self-management of resources. We aim not only to provide a scalable shared storage infrastructure, but to enable the most efficient use of that infrastructure with active network management via our NMAL layer. Users of OSiRIS should be able to get science done with their data instead of becoming bogged down in the details of data management and access.

Acknowledgments
We gratefully acknowledge the National Science Foundation (grant 1541335) which supported the work in this paper.

References
[1] Weil S A, Brandt S A, Miller E L, Long D D E and Maltzahn C 2006 Ceph: a scalable, high-performance distributed file system (USENIX Association) pp 307–320 URL http://www.ssrc.ucsc.edu/papers/weil-osdi06.pdf
[2] Weil S A, Brandt S A, Miller E L, Long D D E and Maltzahn C 2006 Crush: controlled, scalable, decentralized placement of replicated data 122 (ACM New York) URL http://www.ssrc.ucsc.edu/papers/weil-sc06.pdf
[3] OSiRIS 2016 Osiris at supercomputing 2016 URL http://www.osris.org/article/2016/11/18/osiris-at-supercomputing-2016
[4] Weil S 2016 What’s new in jewel and beyond URL http://www.slideshare.net/sageweil1/whats-new-and-jewel-and-beyond
[5] Internet2 URL https://www.incommon.org/
[6] Cantor S e a 2005 Shibboleth architecture: Protocols and profiles URL https://wiki.shibboleth.net/confluence/download/attachments/2162702/internet2-mace-shibboleth-arch-protocols-200509.pdf
[7] Olshansky S 2017 Comanage URL https://spaces.internet2.edu/display/CDmanage
[8] CERN 2016 Dynafed - the dynamic federation project URL http://1cgdm.web.cern.ch/dynafed-dynamic-federation-project
[9] Adams J, Canning B, Dewhurst A, Johnson I, Packer A and Vasilakakos G 2016 Building a large scale object store for the ral tier 1 URL http://indico.cern.ch/event/505613/contributions/2230932/attachments/1346825/2039423/Oral-v2-556.pdf
[10] Larizza G 2014 Building a functional puppet workflow part 3: Dynamic environments with r10k URL http://garylarizza.com/blog/2014/02/18/puppet-workflow-part-3/
[11] 2017 Center for trusted scientific cyberinfrastructure URL http://trustedci.org/