

OSiRIS Site Deployment

Leveraging puppet and foreman to build a distributed ceph cluster



Shawn McKee / Ben Meekhof

University of Michigan / ARC-TS

Michigan Institute for Computational Discovery and Engineering

Supercomputing - November 2016

What is OSiRIS?

OSiRIS combines a [multi-site Ceph cluster](#) with [SDN](#) and [AAA infrastructure](#) enabling scientific researchers to efficiently access data with federated institution credentials.

The current OSiRIS deployment spans Michigan State University, University of Michigan, and Wayne State University. Indiana University is also a part of OSiRIS working on SDN network management tools.

OSiRIS Goals

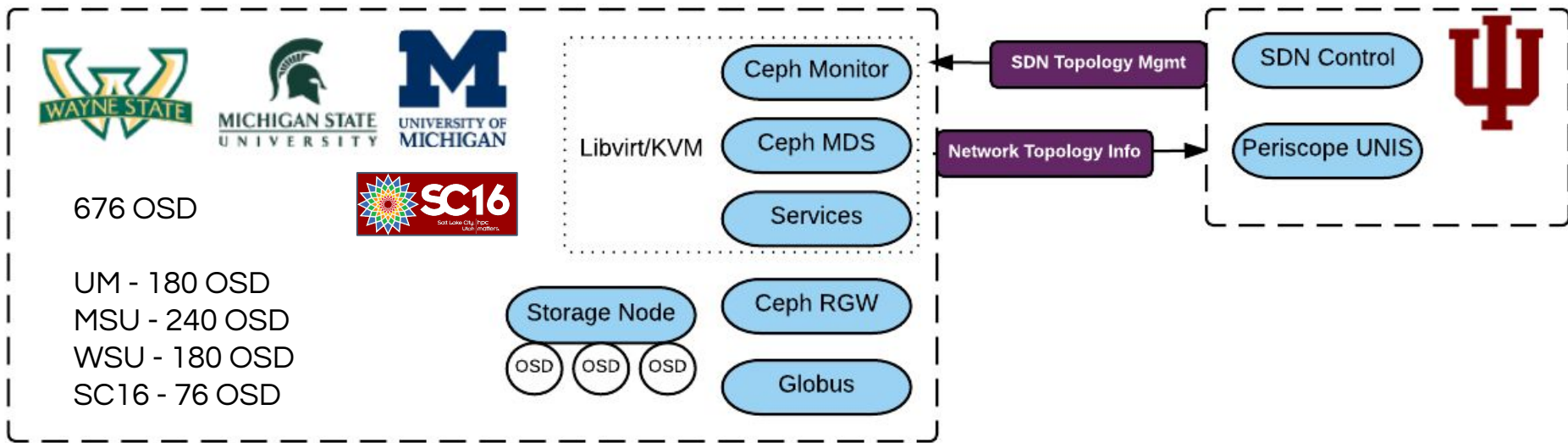


The OSiRIS project goal is enable scientists to collaborate on data easily and without building their own infrastructure.

We have a wide-range of science stakeholders who have data collaboration and data analysis challenges to address within, between and beyond our campuses.

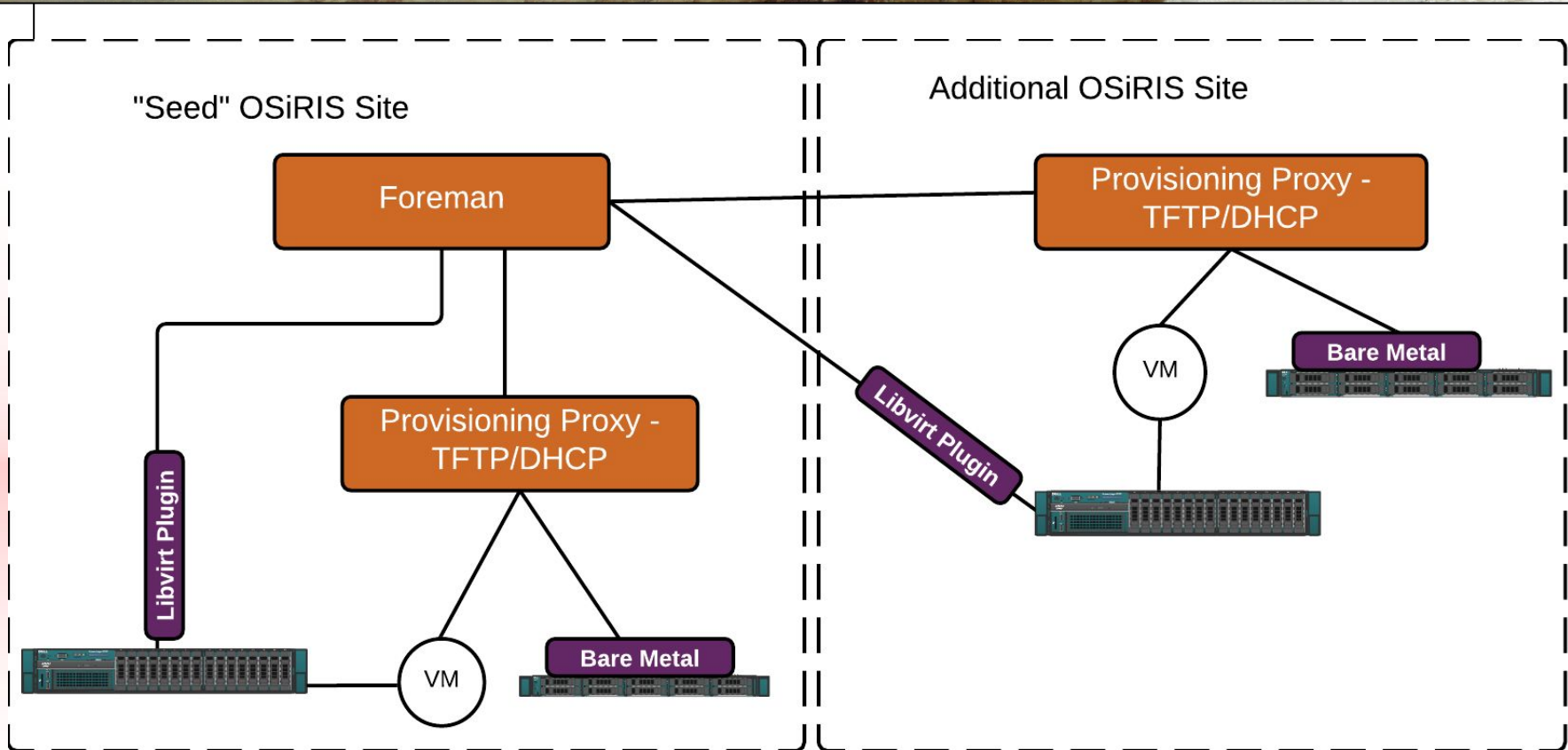
High-energy physics, High-Resolution Ocean Modeling, Degenerative Diseases, Biostatics and Bioinformatics, Population Studies, Genomics, Statistical Genetics and Aquatic Bio-Geochemistry

Our Deployment

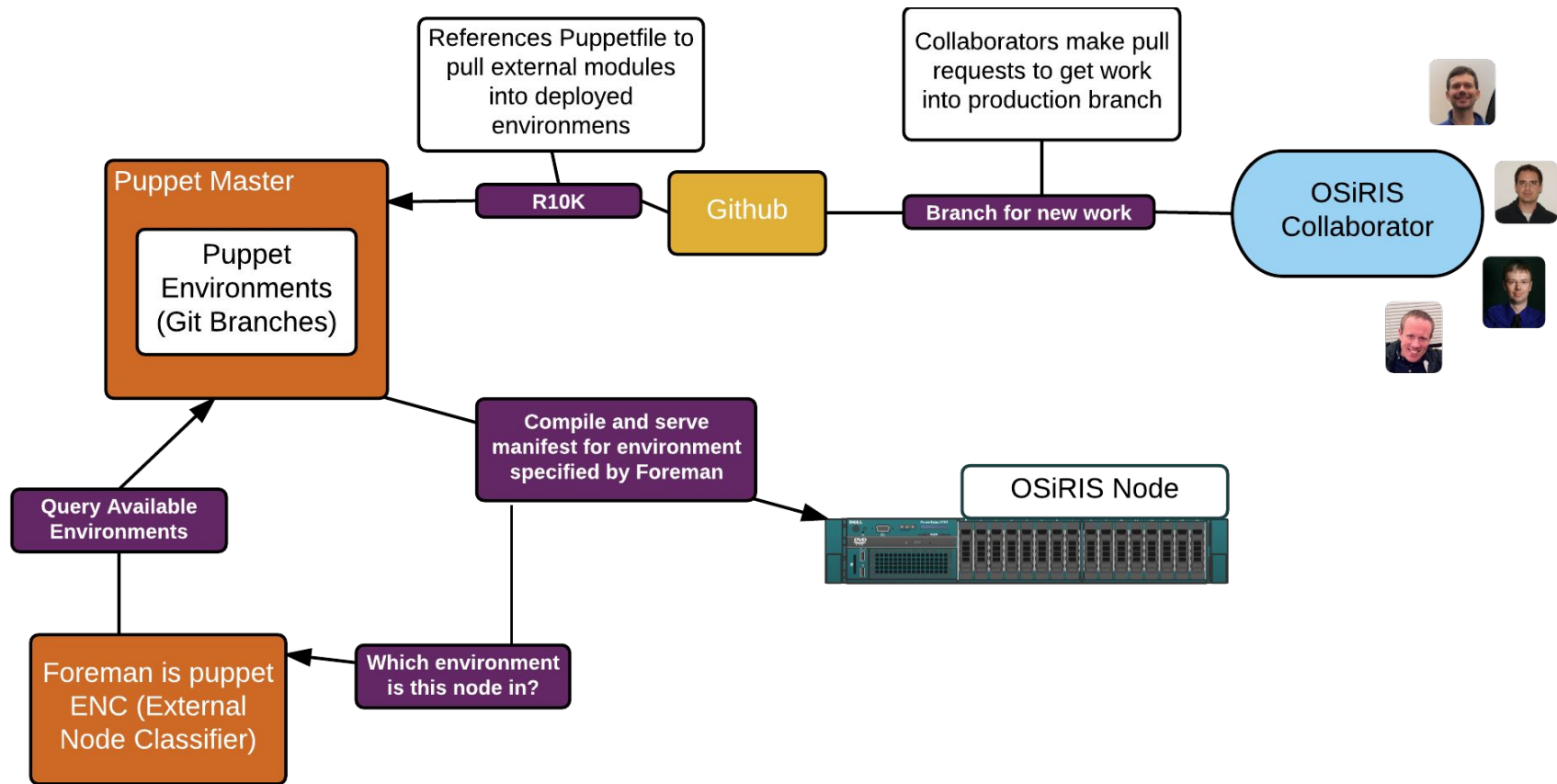


Our first site required manual steps to bring up VM host, and Foreman/Puppet installation. The rest, including Ceph components, is automated from there.

How we deploy



How we manage



How we organize

Entry point into puppet manifests (site.pp)

Derive \$site and \$role from site-role.osris.org

hieraprofile::include('classes')

site = um, msu, etc
role = stor, virt, omd, etc

Generally we don't directly include classes - instead we include 'profiles' that include classes

resolve or merge top to bottom

nodes/%{::clientcert}

sites/%{::site}

roles/%{::role}

os/%{::osfamily}/%{::operatingsystemmajrelease}

os/%{::osfamily}

common

Classes included on node (generally in 'role')

Class parameters or explicit lookups

Example lookup: ntp::servers

iu.yaml: ntp.indiana.edu

common.yaml: us.pool.ntp.org

iu-roleNN
ntp.indiana.edu

xx-roleNN
us.pool.ntp.org

Step 1: Define site specific information in site/sitename.yaml (hiera)

- Network information for provisioning (subnet info, dhcp ranges, etc)
- Ceph CRUSH location
- NTP, DNS, etc

Deploying a new site

- ▼ puppet
 - ▼ hieradata
 - ▶ nodes
 - ▶ os
 - ▶ roles
 - ▼ sites
 - apt.yaml
 - iu.yaml
 - msu.yaml
 - sc.yaml
 - um.yaml
 - wsu.yaml
 - common.eyaml
 - common.yaml
 - ▶ site
 - cluster-logfile-example
 - environment.conf

yaml file matching site from site-role.osris.org hostname

```
nameservers: [ '8.8.8.8', '8.8.4.4' ]
# this should work but maybe not in this version of hiera
# dhcp::nameservers: "%{alias('dnsclient::nameservers')}}"
4
5
6 dhcp::nameservers: [ '8.8.8.8', '8.8.4.4' ]
7
8 frontend_dhcp_range: "192.41.233.197 192.41.233.222"
9
10 frontend_network: '192.41.233.192'
11 frontend_gateway: '192.41.233.193'
12
13 frontend_netmask: '255.255.255.224'
14 frontend_masklen: '27'
15
16 syslocation: 'Supercomputing, SLC, Utah'
17
18 cephx::crush:
19   building: 'salt-palace'
20   member: "%{::site}"
21   rack: 'crate-2'
22
```

Site specific info such as dhcp for provisioning, ns, default osd crush location

Step 2:

- Create a new host in Foreman for the site virtualization host
- Export bootable image
- Install virtualization host, puppet configures necessary packages/services
- Register compute resource in Foreman

Deploying a new site

Interface em1

Type: Interface

MAC address: 24:6e:96:29:8a:98

Identifier: em1

DNS name: sc-virt01

Domain: osris.org

Subnet: sc16 (192.41.233.192/27)

IP address: 192.41.233.197
[Suggest new](#)

Managed Should this interface be managed via DHCP and DNS smart proxy and should it be configured during provisioning?

Primary The Primary interface is used for constructing the FQDN of the host

Provision The Provisioning interface is used for TFTP of PXELinux (or SSH for image-based hosts)

Compute Resources

Filter ...

[New Compute Resource](#)

Name	Type	
msu-virt01	Libvirt	<input type="button" value="Edit"/> <input type="button" value="v"/>
sc-virt01	Libvirt	<input type="button" value="Edit"/> <input type="button" value="v"/>
um-virt01	Libvirt	<input type="button" value="Edit"/> <input type="button" value="v"/>
wsu-virt01	Libvirt	<input type="button" value="Edit"/> <input type="button" value="v"/>

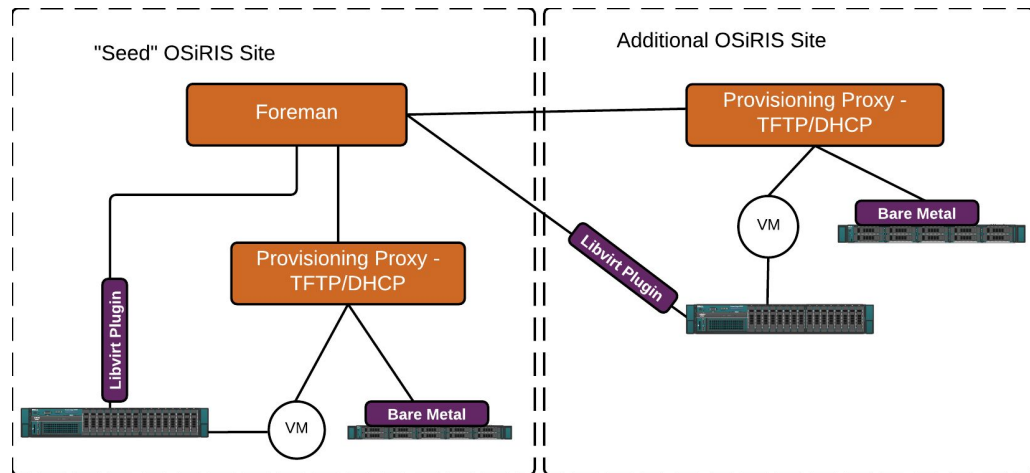
After build we can define as a compute resource in Foreman

Define host network interface, build by exporting boot image from Foreman

Deploying a new site

Step 3:

- Download VM template for provisioning proxy
- Run VM, configure network
- run puppet to complete configuration and register with master Foreman instance



Deploying a new site

Smart Proxies

Filter ... x Search

New Smart Proxy Documentation

Name	URL	Features	
msu-prov-be.osris.org	https://msu-prov-be.osris.org:8443	Templates, TFTP, and DHCP	Import subnets
sc-prov.osris.org	https://sc-prov.osris.org:8443	Templates, TFTP, and DHCP	Import subnets
um-puppet.osris.org	https://um-puppet.osris.org:8443	TFTP, Puppet, Puppet CA, and DHCP	Certificates
wsu-prov-be.osris.org	https://wsu-prov-be.osris.org:8443	Templates, TFTP, and DHCP	Import subnets

Puppet triggers provisioning host to register itself as a 'smart proxy' in foreman (auth info propagated in configuration)

Smart proxy can provide kickstart templates, tftp, dhcp to local network at site

Deploying a new site OSD

In hiera:

- Define the OSD devices used for storage block(s)
- Define the network interfaces to collect stats to Influx/Grafana (collectd-ethstat)
- Define OSD id to collect stats (collectd-ceph)

Deploying a new site OSD

The screenshot shows a Puppet configuration editor window titled "sc-stor01.osris.org.yaml — puppet". On the left, a sidebar lists folders: puppet, hieradata, and nodes. Under nodes, several .yaml files are listed, with "sc-stor01.osris.org.yaml" selected. The main editor area shows the following code:

```
1 collectd::plugin::ethstat::interfaces: [ 'p5p1', 'p5p2', 'p6p1', 'p6p2' ]
2 collectd::plugin::ceph::daemons: [ 'ceph-osd.171', 'ceph-osd.172', 'ceph-osd.173'
```

A callout box points to the ceph configuration section of the code, which includes:

```
30 # three of these specify test cluster (input
37
38 ceph::osd:
39   '/dev/mapper/mpatha':
40     journal: '/dev/nvme0n1'
41     cluster: 'test'
42   '/dev/mapper/mpathb':
43     journal: '/dev/nvme0n1'
44   '/dev/mapper/mpathc':
45     journal: '/dev/nvme0n1'
46   '/dev/mapper/mpathd':
47     journal: '/dev/nvme0n1'
48   '/dev/mapper/mpathe':
49     journal: '/dev/nvme0n1'
50   '/dev/mapper/mpathf':
51     journal: '/dev/nvme0n1'
```

Interfaces and collectd-ceph daemons in yaml matching hostname

Most of our storage nodes identical, define ceph osd devices at role level (for now)

Deploying a new site



From this point we're ready to build new storage blocks, monitor, mds, grafana, omd, etc.

All of the above automated with puppet, and with Foreman groups defining appropriate partitions or data volumes

Dynamic and Scalable



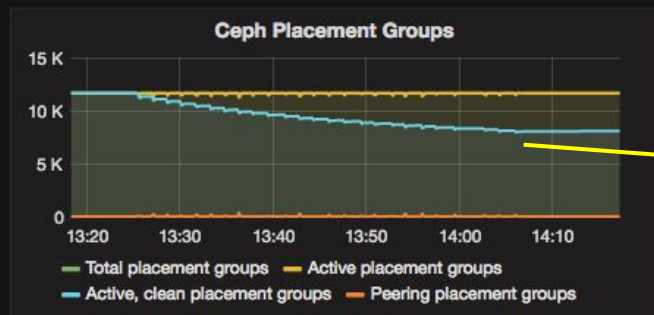
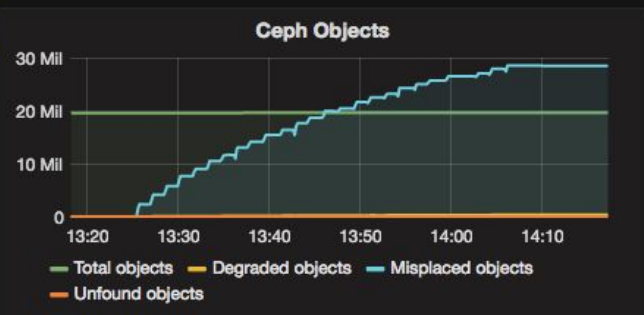
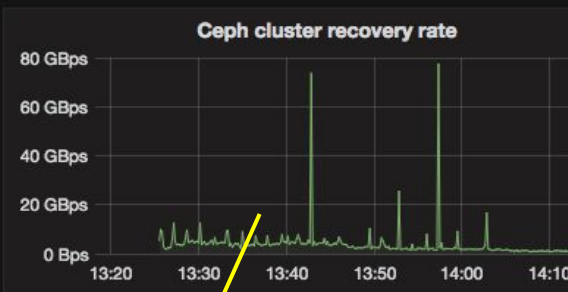
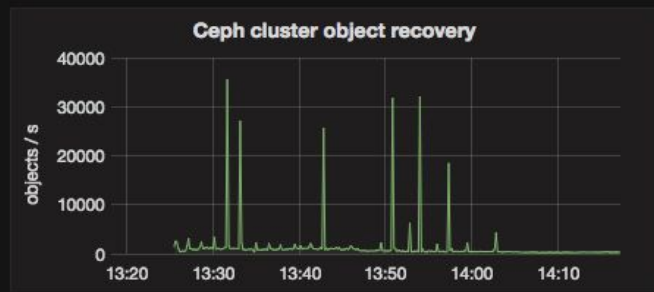
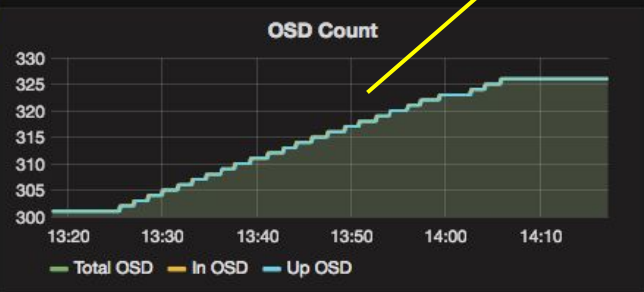
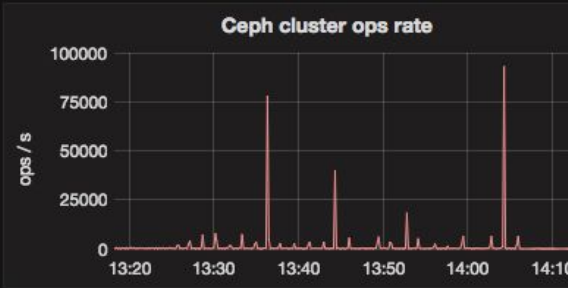
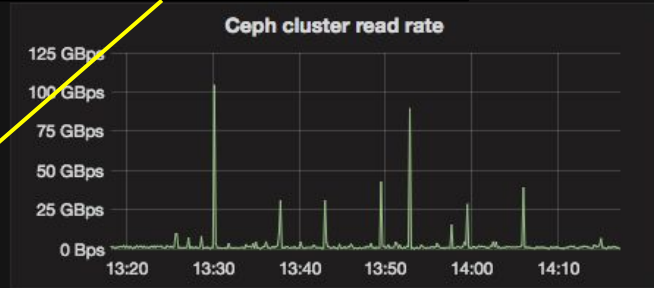
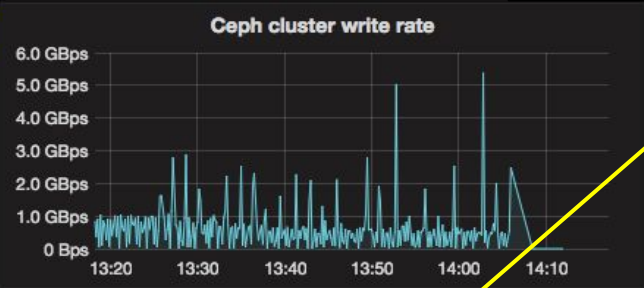
While OSD are initializing and coming online we have a client data transfer ongoing

You can see the impact on the transfer and the progress of the OSD addition on our monitoring dashboard



cluster: ceph Retention Policy: default

OSD Count climbing as puppet agent uses ceph-disk to init new



Cluster moving data replicas to new OSD

Ongoing during our talk is a demo of live data movement leveraging the Data Logistics Toolkit created at Indiana University.

This demo showcases the movement of USGS earthsat data from capture to storage not only in of the main OSiRIS Ceph cluster but also a dynamic OSiRIS Ceph cluster deployment built at Cloudlab.

Activity can be seen on the Periscope dashboard

<http://dev.crest.iu.edu/map/>



Questions?

Questions or comments?