SDSC Cloud Storage with OpenStack Swift

SDSC SAN DIEGO SUPERCOMPUTER CENTER

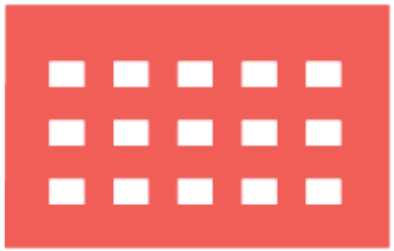
UC San Diego

openstack®

SWIFT

What is OpenStack Swift?

- Scalable Redundant Storage System
- Object Store
  - Key-value Pair (Key = URL; Value = File)
- Accessible via HTTP using a REST API
- No POSIX File System Access

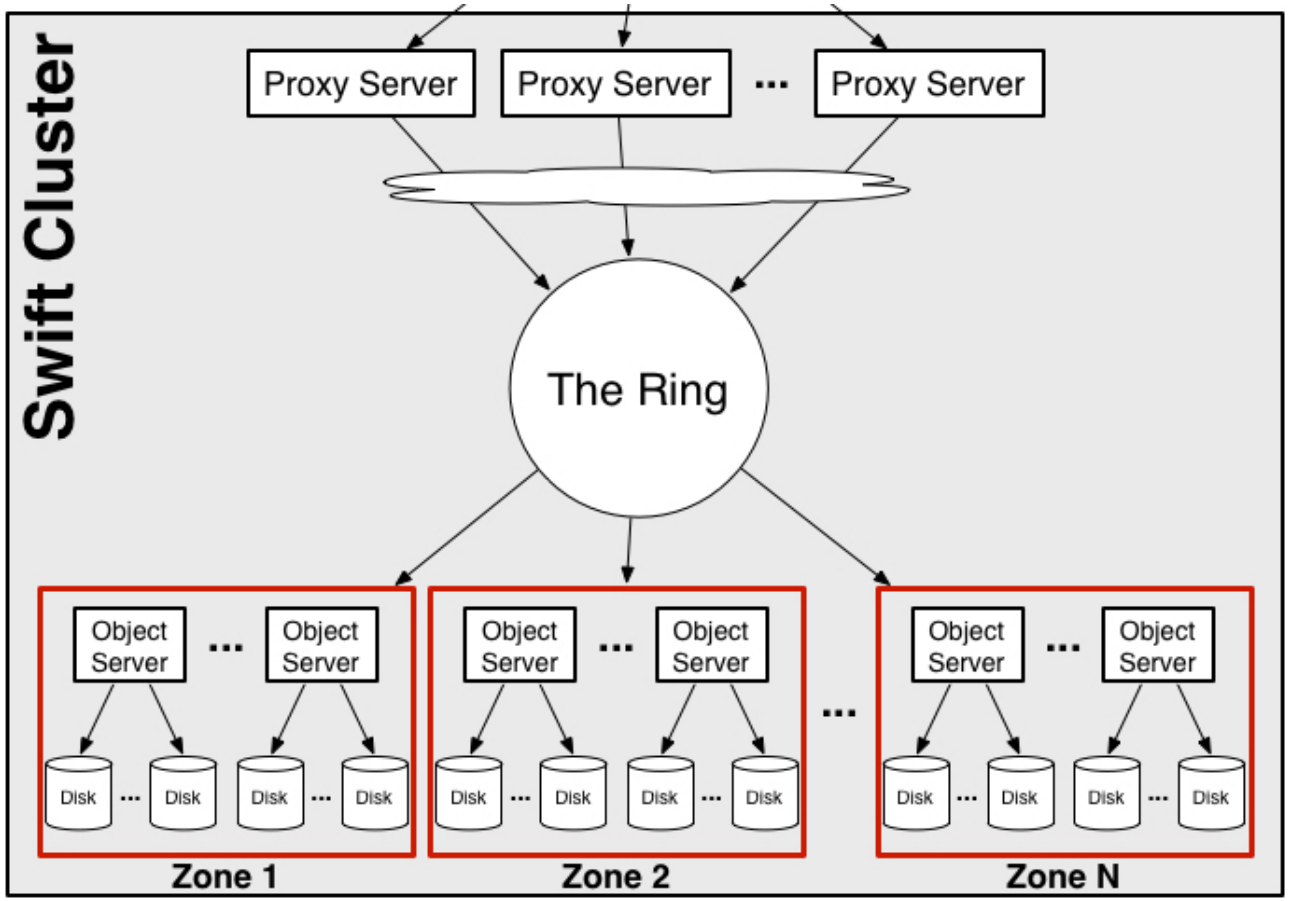


OBJECT
STORAGE

Components of OpenStack Swift

- **Proxy Server**
  - Authentication, Authorization
- **Account Server**
  - Auditor, Replicator, Reaper
- **Container Server**
  - Auditor, Replicator, Updater
- **Object Server**
  - Auditor, Replicator, Updater
- **The Ring**
  - Determines the locations for all Accounts, Containers, Objects

Components of OpenStack Swift

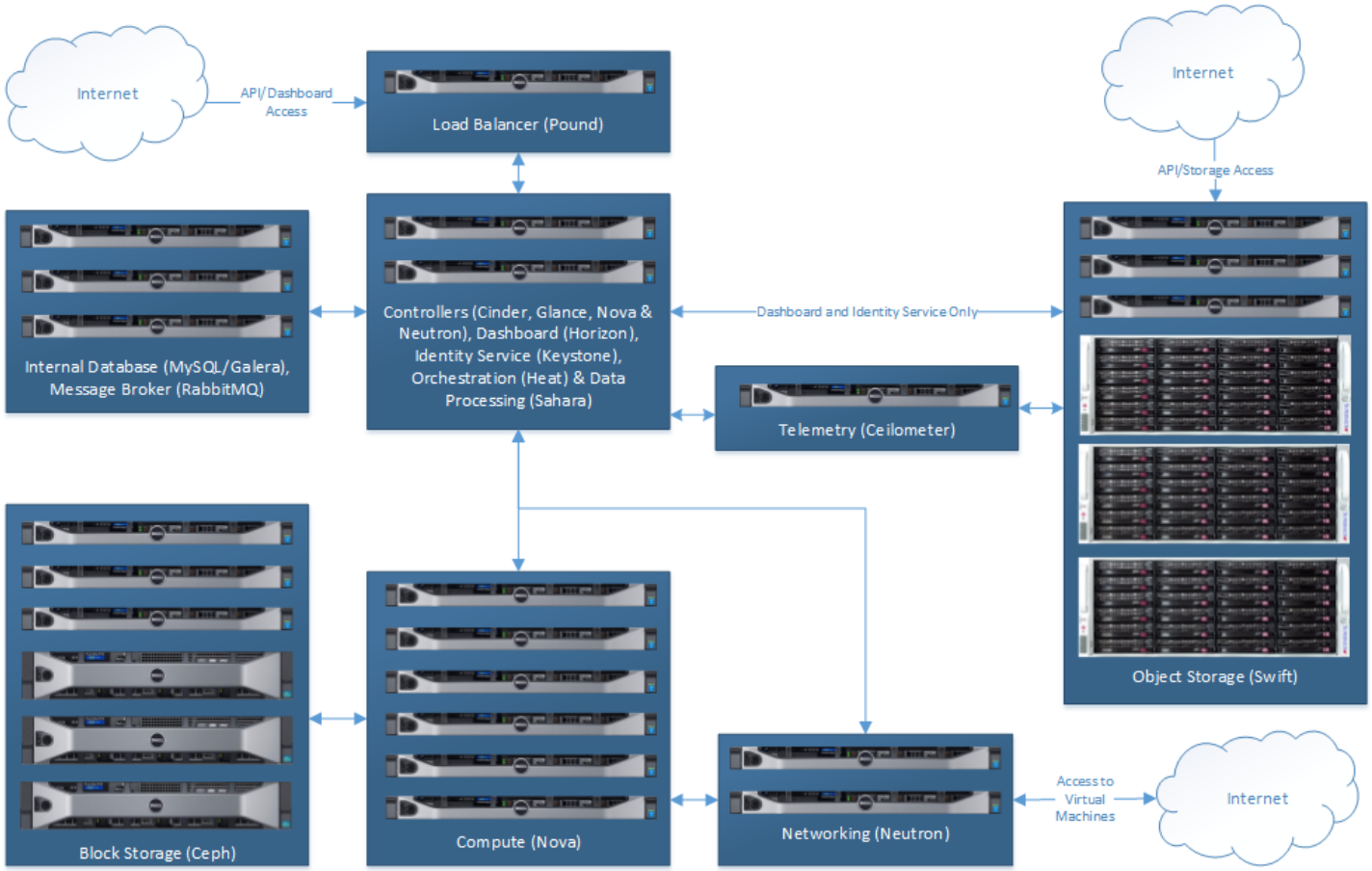


OpenStack Swift @ SDSC

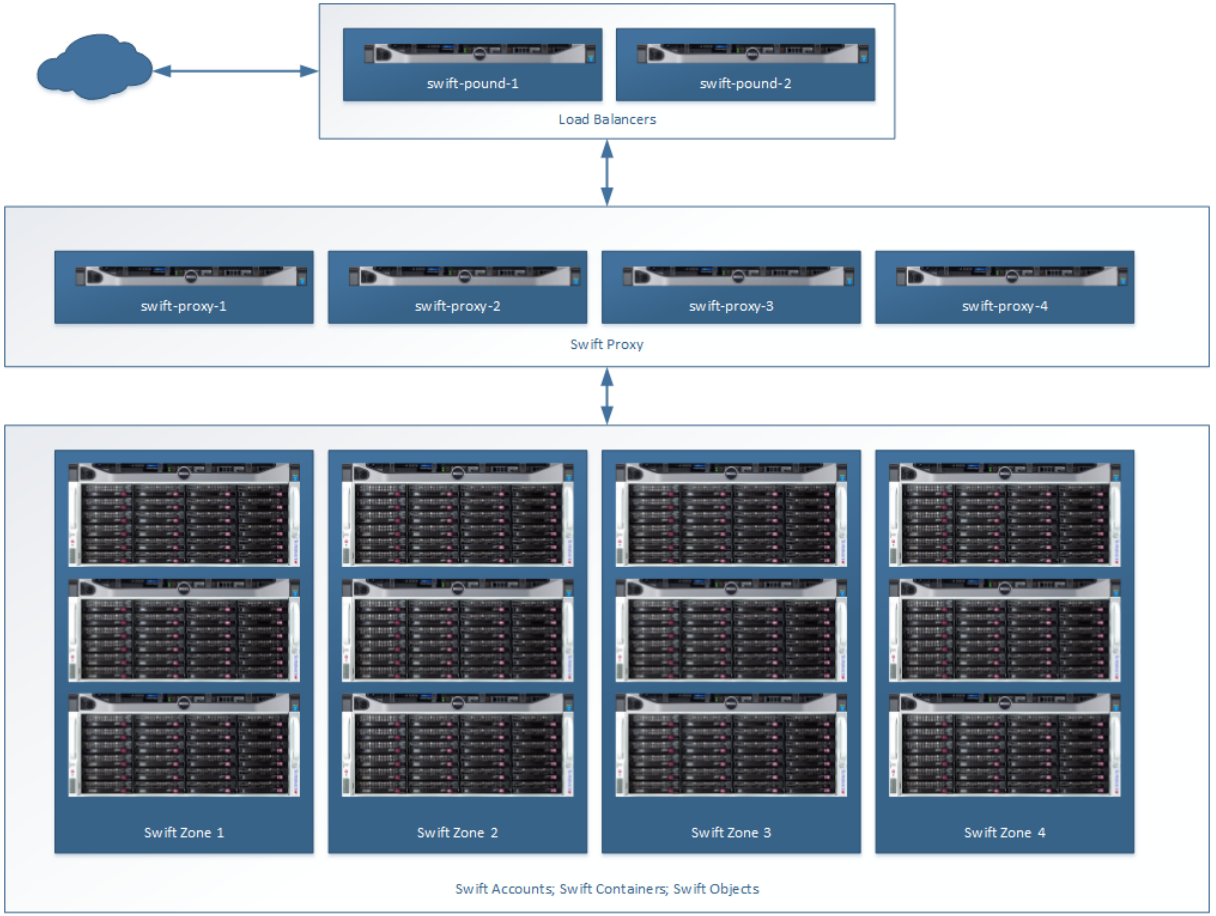
OpenStack Swift @ SDSC

- Running Swift since Sept 2011 (Diablo release)
- 5+ PB raw capacity
- 42 servers; 1000+ disks
- 100+ active projects
- 2 major architectural changes w/o downtime
- 2 outages in 5+ years (building power failures)
- Replaced tape archive

SDSC Cloud Architectural Diagram



SDSC Cloud Storage Detail



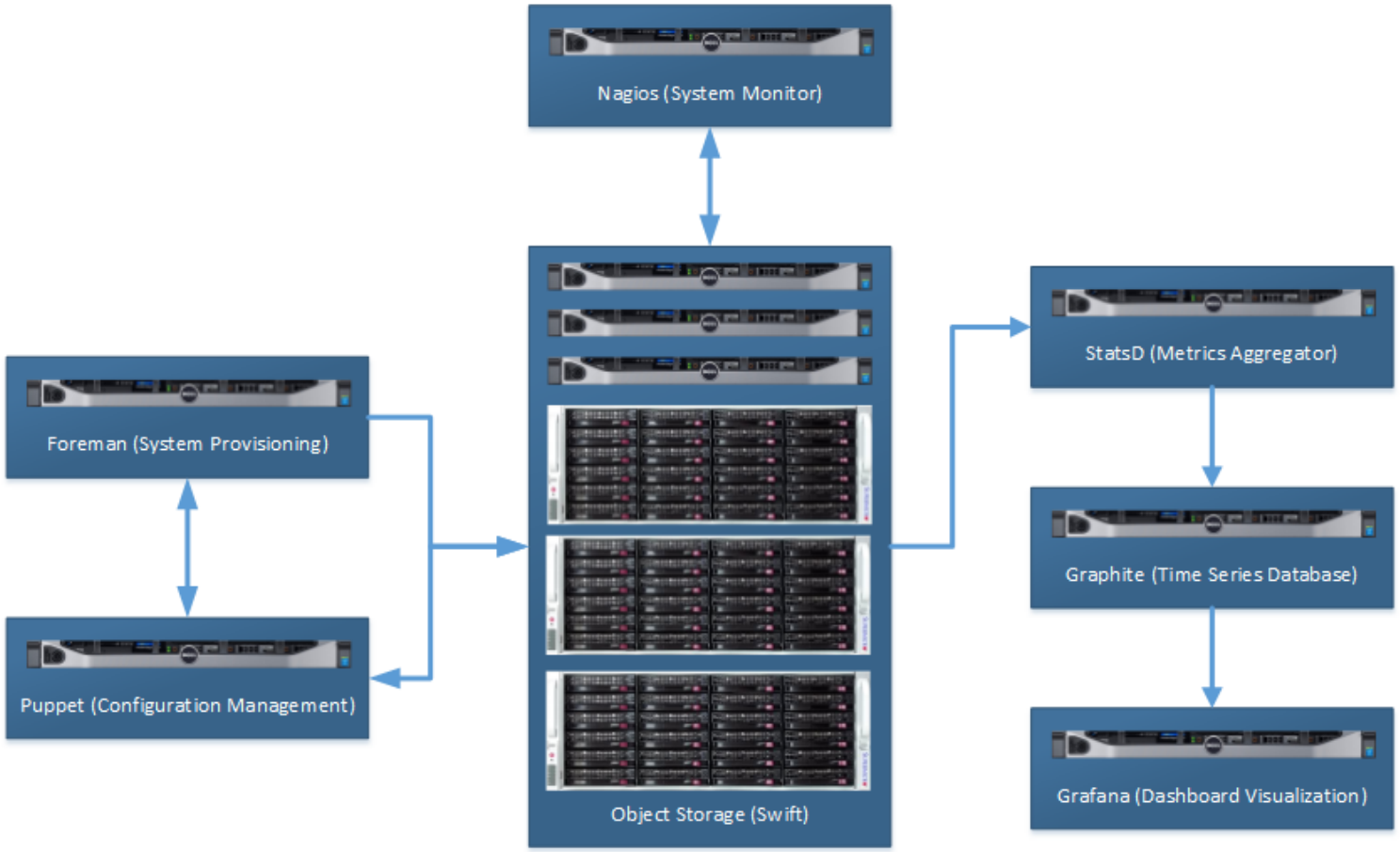
Hardware/Software Details

- **Dell R600 Series Head Nodes**
- **SuperMicro 45 Disk JBODs**
  - Moving to SuperMicro 90 Disk JBODs
  - SAS (3, 4 & 6 TB) for Object Data
  - SATA SSD (100 & 200 GB) for Accounts and Container Data
- **10 GbE Internal Networking**
- **CentOS 7**
- **OpenStack Swift Mitaka**
- **Pound for Load Balancers**
- **Keystone (v2 & v3) and Swauth (v1) Auth**
- **Horizon for Web Dashboard**

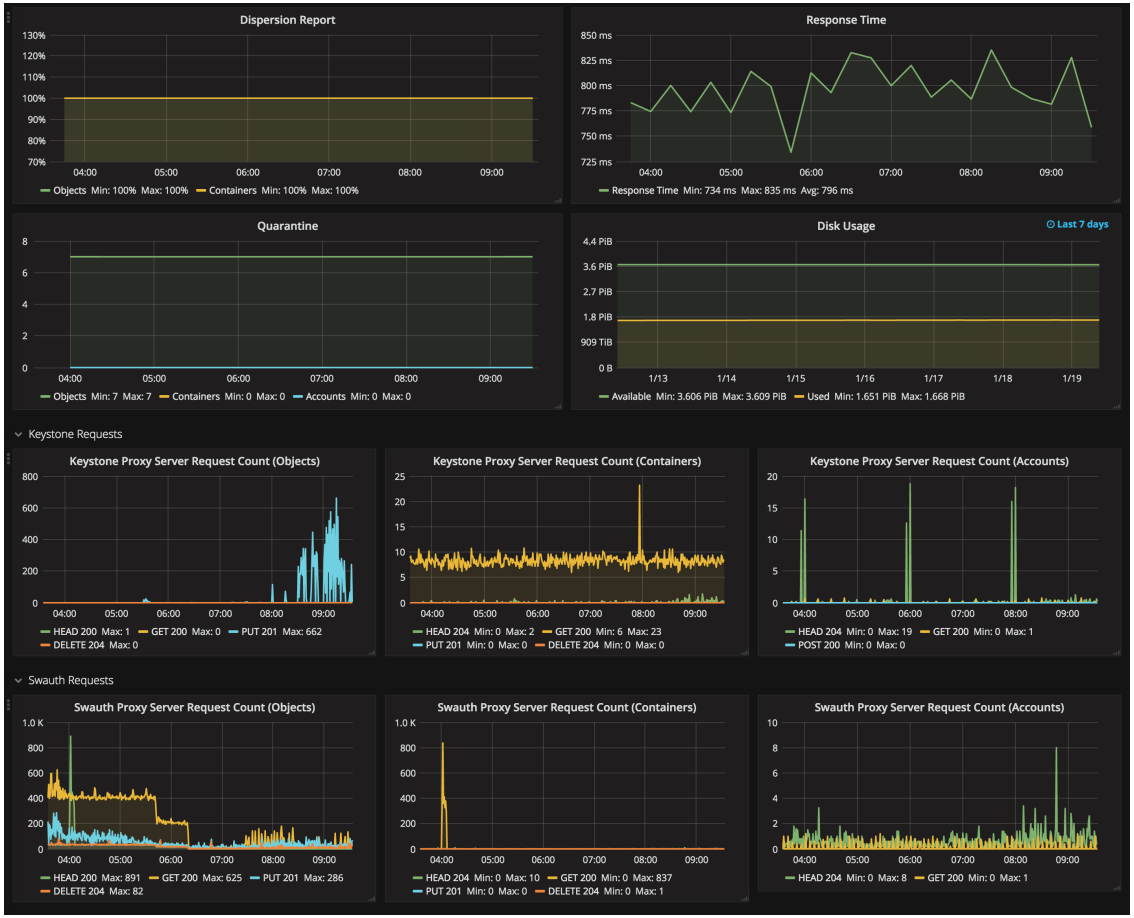
Supporting Services

- **The Foreman used for System Provisioning**
  - PXE boot
  - Kickstart
  - Keeps system facts
- **Puppet for Configuration Management**
  - Swift puppet modules from OpenStack
- **Nagios used for Monitoring**
- **Metrics**
  - StatsD (built into swift)
  - Graphite (database)
  - Grafana (visualization dashboard)

Supporting Services



Example Metrics Dashboard



Challenges (so far…)

- **Aging Hardware**
  - Most of the Hardware is 5+ Years Old
- **Overcoming early cost cutting design decisions**
  - 2 copy on raid 6 to 3 copy raid 0
- **CentOS 6 to 7 migration**
  - Python 2 to 3 coming soon…
- **Rapid Release Cycle**
  - Two releases per year/releases only supported for 1 year
  - Feature being deprecated (particularly with regard to auth)
- **Customer Expectations**
- **Lack of Tools for Monitoring**

Advantages

- **Cost Effective**
  - Open Source (No Software Licenses)
  - Inexpensive/Heterogeneous Hardware
- **Fully Redundant**
- **No Downtime Upgrades**
- **Easy Upgrades**
- **Once running, requires little day to day**
  - About 15% of 1 FTE

Disadvantages

- Most Users (in Research) Don't Know How to Use Object Storage
- No Support/Little Documentation
- No Service for Distributing the Ring
- No Tools for Adding New Servers to the Ring
- Poor Usage Reporting (OpenStack Telemetry)

Script to Add a New Server to an Existing Cluster

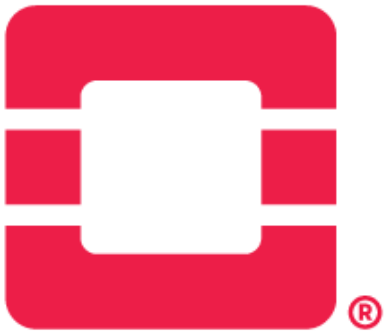
1. Check the Object Dispersion (swift-dispersion-report)
2. IF 99.9% objects found ELSE wait 4 hours & go to #1
3. Increase the Object Weight (swift-ring-builder)
4. Rebalance Ring (swift-ring-builder)
5. Commit Ring to SVN server
6. SSH to each Storage and Proxy Node
7. Checkout new Ring
8. Verify the MD5 Sum of all Rings (swift-recon)
9. Wait 4 Hours
10. Go to #1

OpenStack Telemetry

- **Ceilometer is not Mature**
  - Maturity: 1 of 8 (according to OS project navigator)
  - The 1 point is for having install documentation
- **Our Experience with Ceilometer**
  - Stops working frequently
  - Requests are slow
  - Upgrades break things
- **OpenStack Telemetry is now 4 services**
  - Ceilometer, Aodh, Gnocchi & Panko

OpenStack Swift vs Ceph

- **Swift Pros**
  - Less Hardware Needed
  - Simple
  - Updates are Easier



- **Ceph Pros**
  - Better Monitoring Tools
  - Better Deployment Tools
  - LTS Releases
  - Don’t have to Distribute the Ring
- **Ceph Cons**
  - Not 100% API Compatible with Swift

Kevin Coakley
kcoakley@sdsc.edu