Ceph: Software Defined Storage

UNIVERSITY OF OREGON

NSRC
Network Startup Resource Center

Why Ceph?

- Widely deployed and understood
  - Running at huge scale in many environments – e.g. DigitalOcean, CERN
- Many, many features
- Long history
  - Started in 2005 by Inktank
  - Red Hat purchased in 2014 (now IBM)
- Commercial distributions and support from multiple vendors

Features

- Block storage, File storage, Object storage
- Self healing, self balancing
- Very scalable
- Thin provisioning
- Cheap snapshotting and cloning
- Can be designed for high level of fault tolerance
- Off-site mirroring/replication natively supported

Downsides

- Steep learning curve
- Investment…
- Documentation is improving, but many advanced scenarios need some third party guidance
- Adding or removing nodes can cause a lot of data movement
  - Suggest starting with at least 5 nodes, so only ~20% of data moves
- Without proper operations and monitoring, it *can* fail

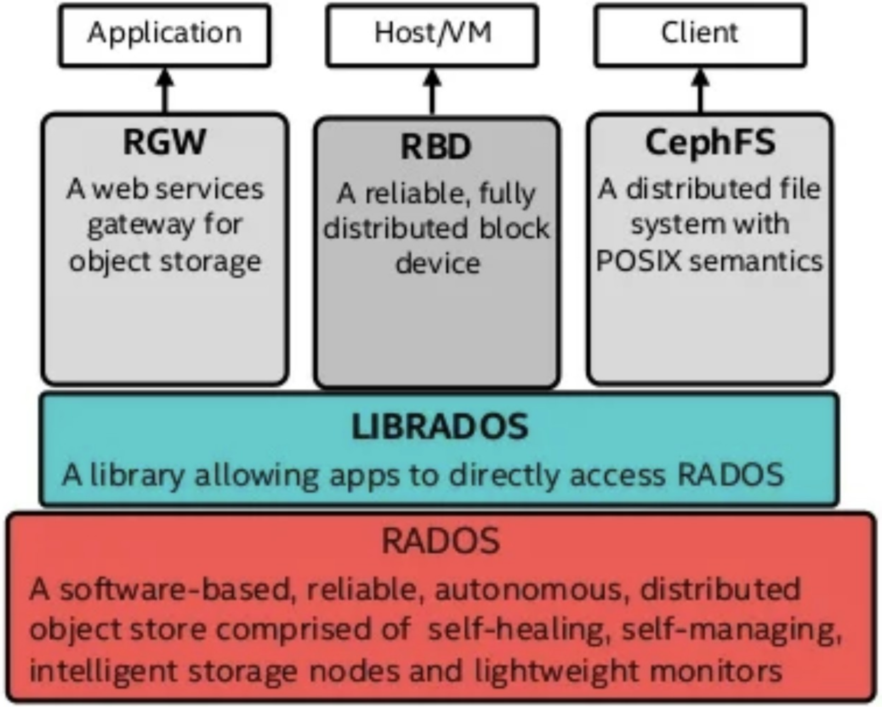
Main concepts

- RADOS - reliable autonomic distributed object store
- CRUSH map
- Placement groups - PGs
- MONs
- OSDs
- Pools

Components

- OSD daemon - stores data on one disk, on behalf of CEPH clients
- CEPH MONitor - maintains copy of the **cluster map**
- CEPH Manager (mgr) - info about PGs, metadata, collect stats
- Pools - logical partitions. Different pools for different types of data, replication policy, technology (SSD/HDD)
- Placement Group (pg) - a pool's data is spread across many PGs. PGs are then assigned to OSDs (by CRUSH)
- CRUSH is the algorithm and process for selecting data placement, failure domains, etc. based on rules.

Architecture



Application

Host/VM

Client

RGW

A web services gateway for object storage

RBD

A reliable, fully distributed block device

CephFS

A distributed file system with POSIX semantics

LIBRADOS

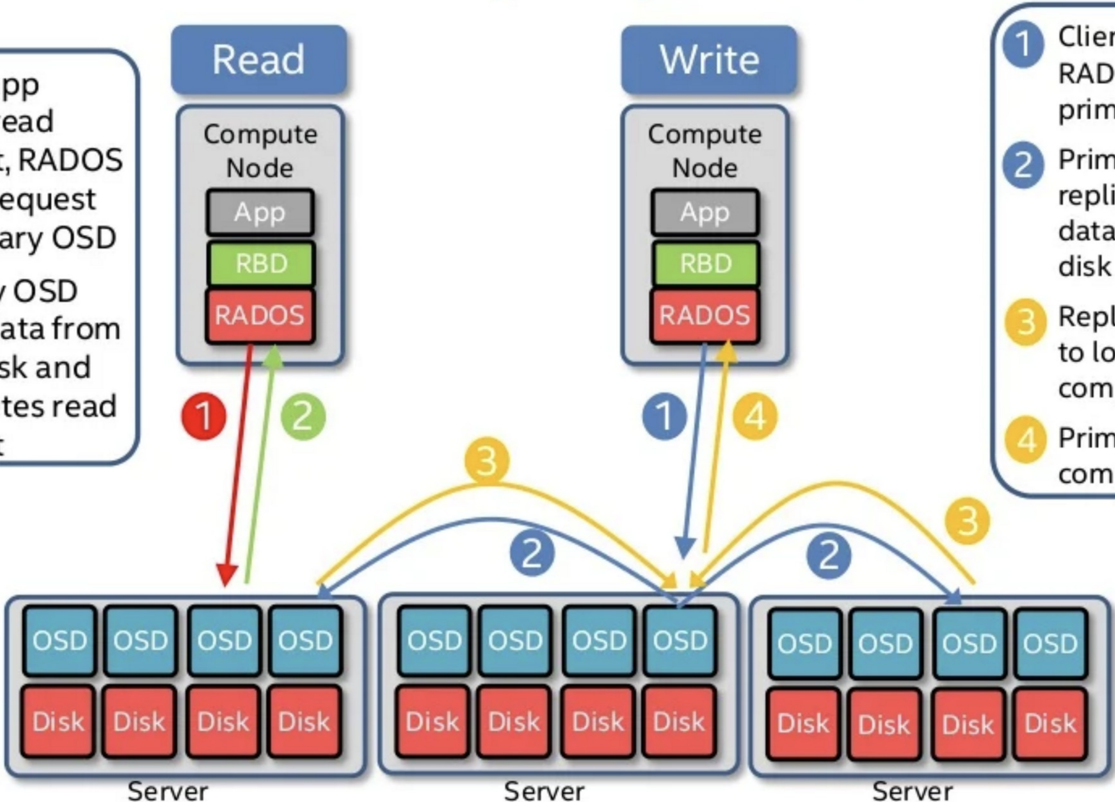
A library allowing apps to directly access RADOS

RADOS

A software-based, reliable, autonomous, distributed object store comprised of self-healing, self-managing, intelligent storage nodes and lightweight monitors

Object Store Daemon (OSD) Read and Write Flow

1. Client app issues read request, RADOS sends request to primary OSD
2. Primary OSD reads data from local disk and completes read request

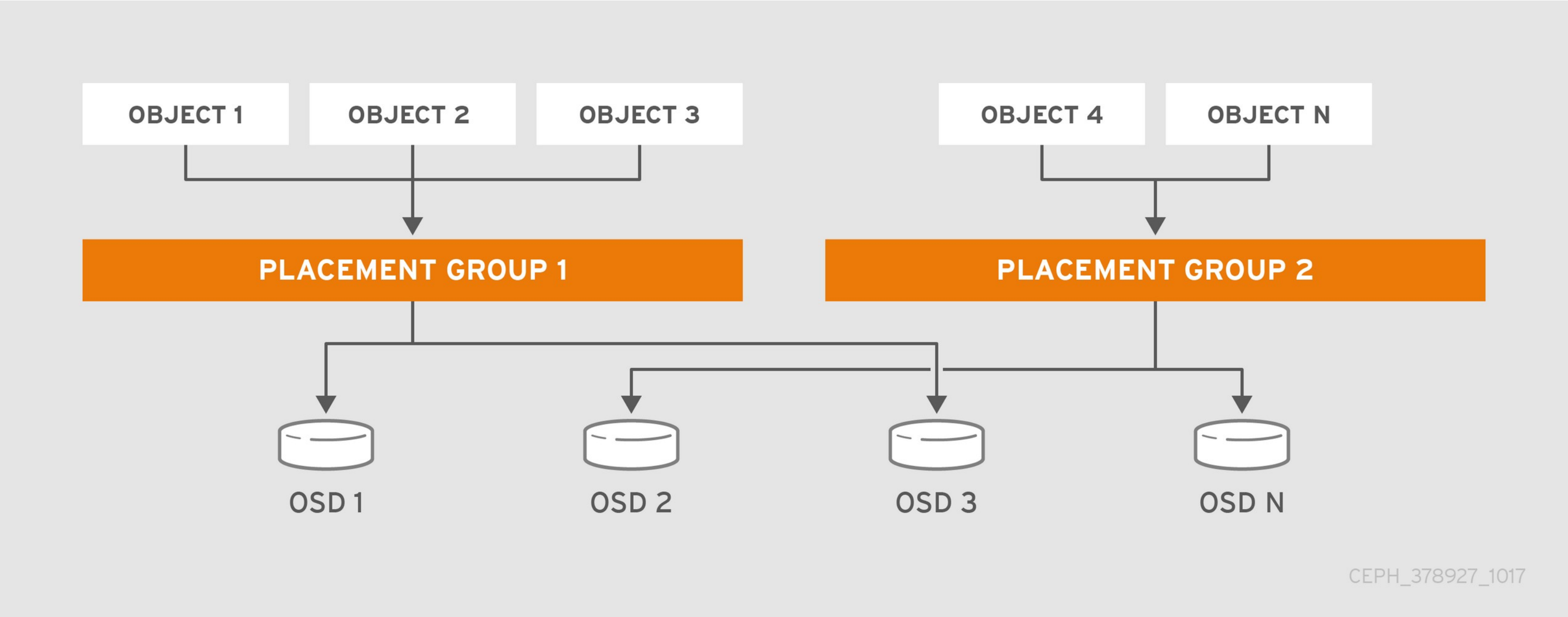


1. Client app writes data, RADOS sends data to primary OSD
2. Primary OSD identifies replica OSDs and sends data, writes data to local disk
3. Replica OSDs write data to local disk, signal completion to primary
4. Primary OSD signals completion to client app

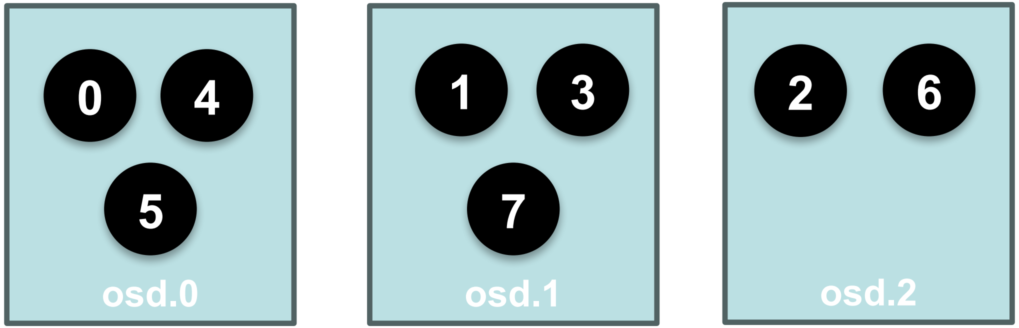
IDF15

https://www.slideshare.net/LarryCover/ceph-open-source-storage-software-optimizations-on-intel-architecture-for-cloud-workloads

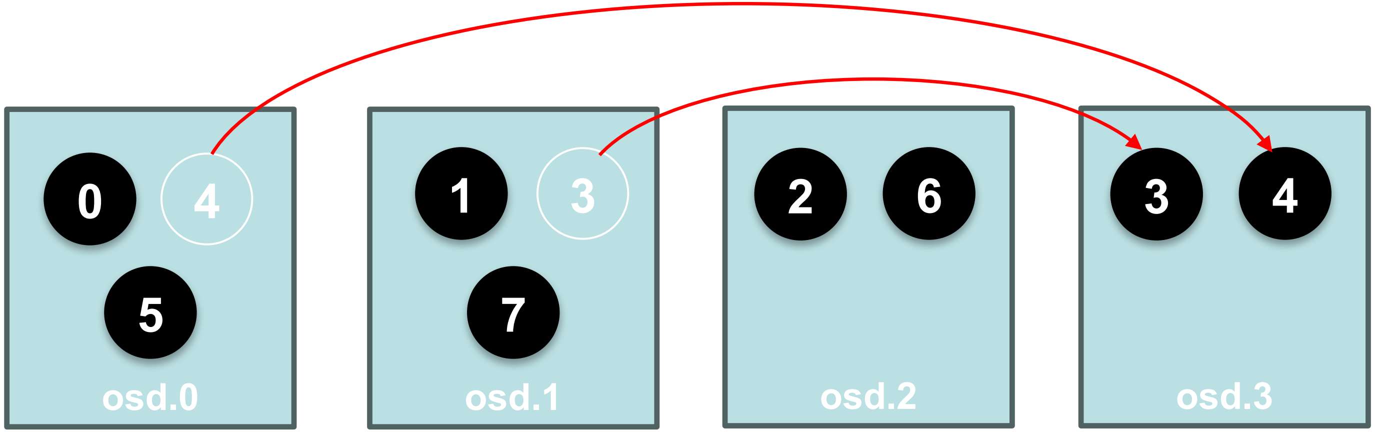
PGs, OSDs


OBJECT 1
OBJECT 2
OBJECT 3
OBJECT 4
OBJECT N
PLACEMENT GROUP 1
PLACEMENT GROUP 2
OSD 1
OSD 2
OSD 3
OSD N
CEPH_378927_1017

Assigning PGs to OSDs (cluster map)



Rebalancing - new OSD added



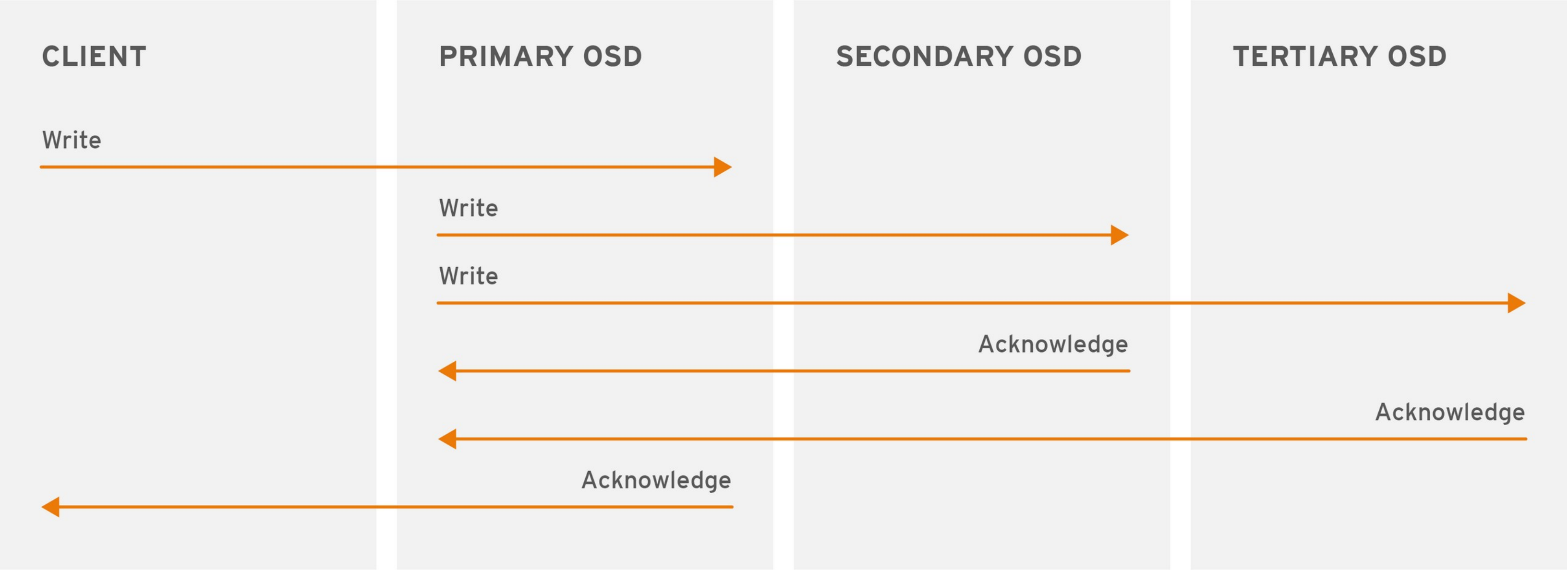
Algorithm moves minimum number of PGs

You need significantly *more* PGs in a pool than the number of OSDs in your cluster, to achieve good balance
(typical starting value: 128)

Where to place a new object?

- The client inputs the pool name and the object ID. For example, pool = liverpool and object-id = john.
  - CRUSH takes the object ID and hashes it.
  - CRUSH calculates the hash modulo of the number of PGs to get a PG ID. For example, 58.
  - CRUSH calculates the primary OSD corresponding to the PG ID.
  - The client gets the pool ID given the pool name. For example, the pool "liverpool" is pool number 4.
  - The client prepends the pool ID to the PG ID. For example, 4.58.
  - The client performs an object operation such as write, read, or delete by communicating directly with the Primary OSD in the Acting Set.

OSD replication



CLIENT

PRIMARY OSD

SECONDARY OSD

TERTIARY OSD

Write

Write

Write

Acknowledge

Acknowledge

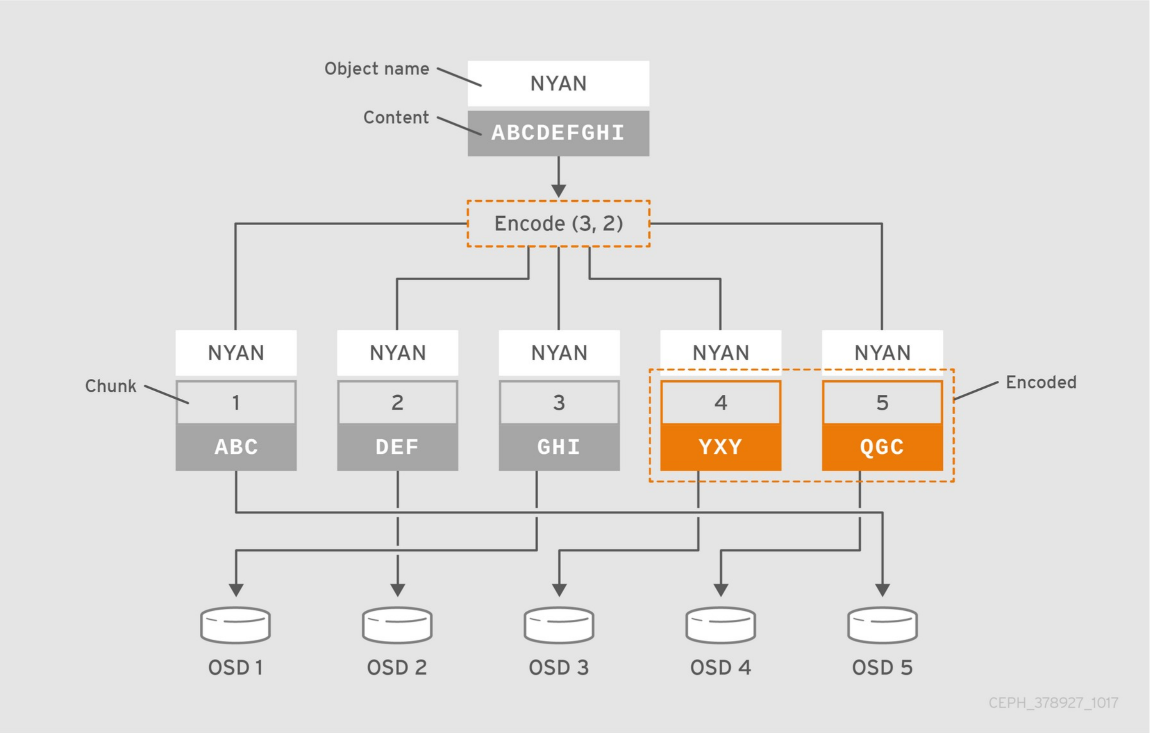
Acknowledge

CEPH_378927_1017

OSD states

- "Up" / "Down" = working / not working
- "In" / "Out" = enabled for use / not enabled for use
  - When a drive goes "Down", it's automatically made "Out"
  - You can manually set a drive to "Out", e.g. for maintenance
- Note that once a drive is Out, Ceph starts moving data around the cluster to ensure the target number of replicas is maintained
  - You can disable this during maintenance using "noout"

Erasure Coding Pools



Proxmox and Ceph

- Proxmox has Ceph integrated
  - It can talk to Ceph storage
  - It can run as a Ceph storage cluster itself
- VMs and storage can run on the same hosts
  - a.k.a. "Hyperconverged"
  - This can make your cluster smaller and cheaper, but it can also make performance problems hard to diagnose
  - During healing, large amounts of RAM may be used by OSDs
    - allow 4GiB per OSD, *in addition* to VM memory
  - If you have large quantities of file or object data (other than VM images) it may be better to build a dedicated storage cluster for that

Ceph Lab

UNIVERSITY OF OREGON

NSRC
Network Startup Resource Center