inktank

OpenNebula Techday

Ceph Introduction (2014-06-24) – Jean-Charles Lopez

Presentations Objectives

By the end of this presentation, you should be able to:

- Understand Ceph
  - Project history
  - Concepts
  - Versions
- Understand Ceph components
  - RADOSGW,
  - MONs, OSDs & MDSs
  - RBD,
  - CRUSH,
  - CephFS
  - Clients
- Know about the MIMO project

OpenNebula Techday

STORAGE LANDSCAPE & CEPH

Storage challenges today

- **Money**
  - More data, same budgets
  - Need a low cost per gigabyte
  - No vendor lock-in
- **Time**
  - Ease of administration
  - No manual data migration or load balancing
  - Seamless scaling – both expansion and contraction

Ceph - Foundations

- **A new philosophy**
  - Open Source
  - Community-focused equals strong, sustainable ecosystem
- **A new design**
  - Scalable (every component)
  - No single point of failure
  - Software-based runs on commodity hardware
  - Self-managing
  - Flexible
  - Unified

OpenNebula Techday

CEPH COMPONENTS

Ceph Storage Cluster Components

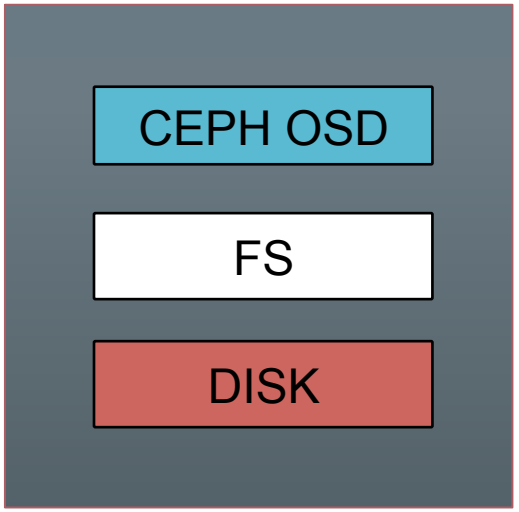
- **They include:**
  - OSDs (Object Storage Devices)
  - Monitors (Cluster Map management)
  - MDSs (Meta Data Servers)
  - RADOSGW (S3/Swift compatible Gateway)
  - CRUSH (Data placement algorithm)
  - Placement groups
  - Pools
  - Ceph journal

Ceph - Monitors

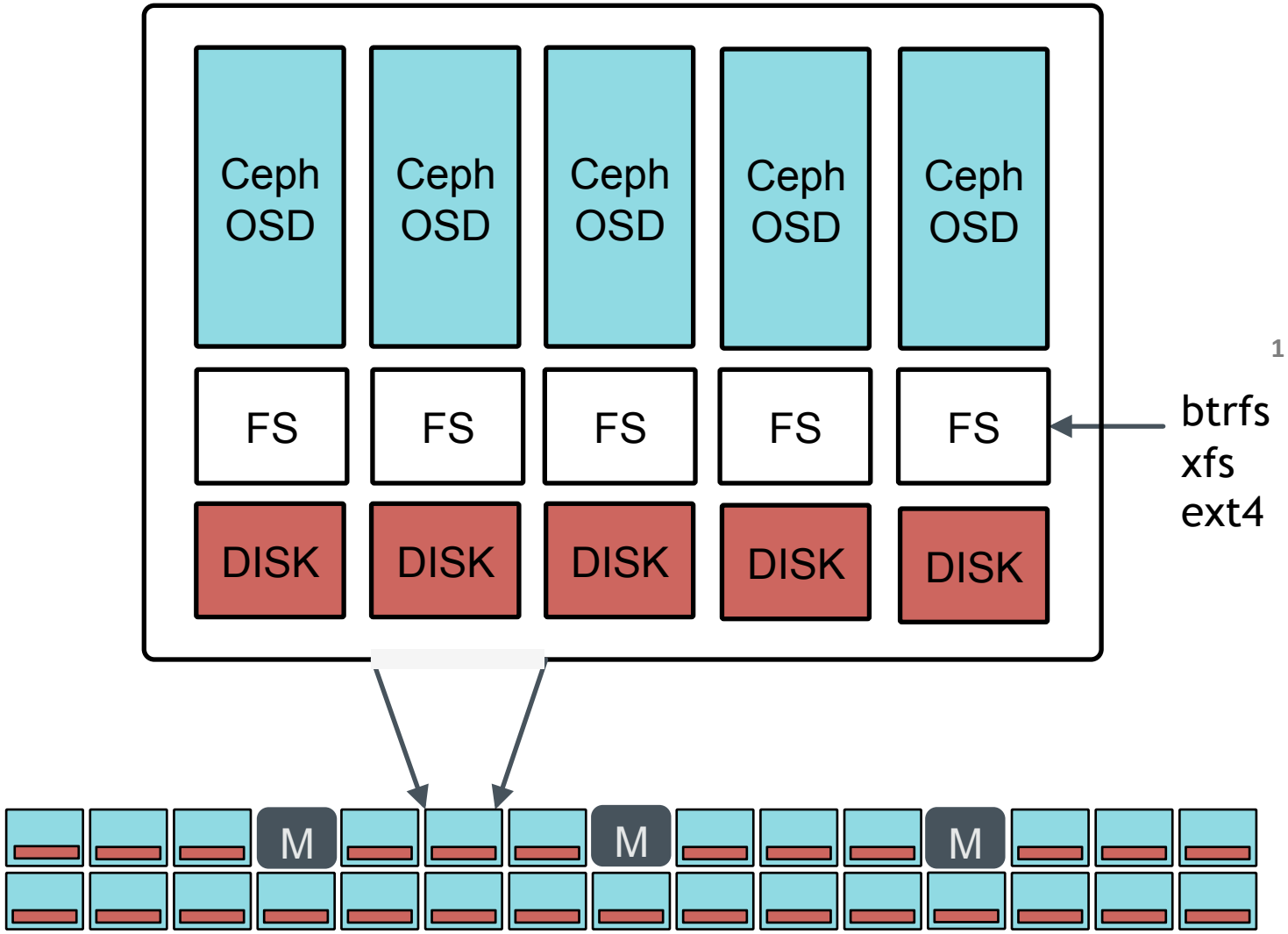
- **What do they do?**
  - They maintain the cluster map [1]
  - They provide consensus for distributed decision-making
- **What they do NOT do?**
  - They don't serve stored objects to clients
- **How many do we need?**
  - There must be an odd number of MONs [2]
  - 3 is the minimum number of MONs [3]

Ceph - OSD Daemon

- **Ceph Object Storage Device Daemon**
  - Intelligent Storage Servers [1]
  - Serve stored objects to clients
- **OSD is primary for some objects**
  - Responsible for replication
  - Responsible for coherency
  - Responsible for re-balancing
  - Responsible for recovery
- **OSD is secondary for some objects**
  - Under control of the primary
  - Capable of becoming primary
- **Supports extended object classes**
  - Atomic transactions
  - Synchronization and notifications
  - Send computation to the data



Ceph - Storage Node



1

btrfs
xfs
ext4

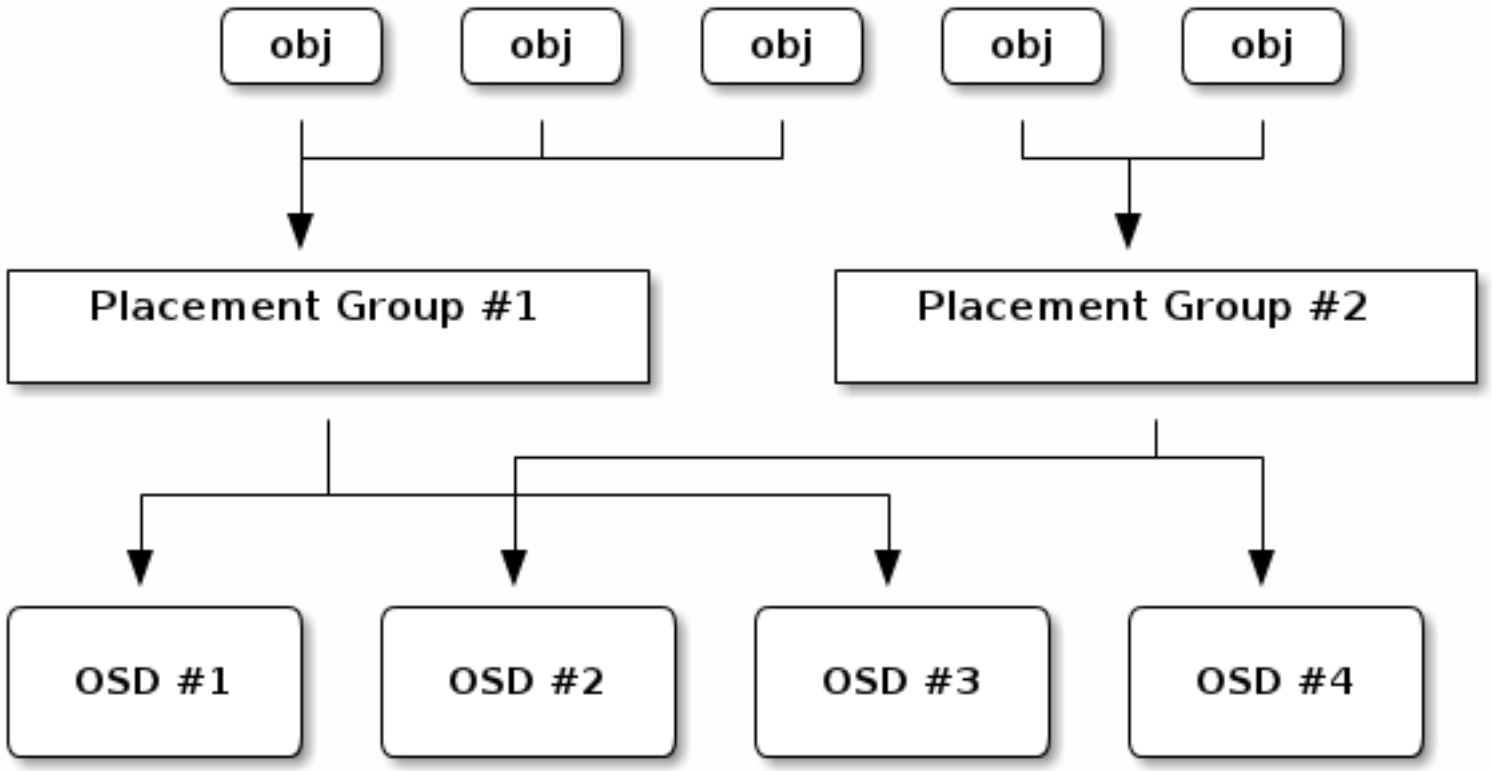
OpenNebula Techday

CEPH DATA PLACEMENT

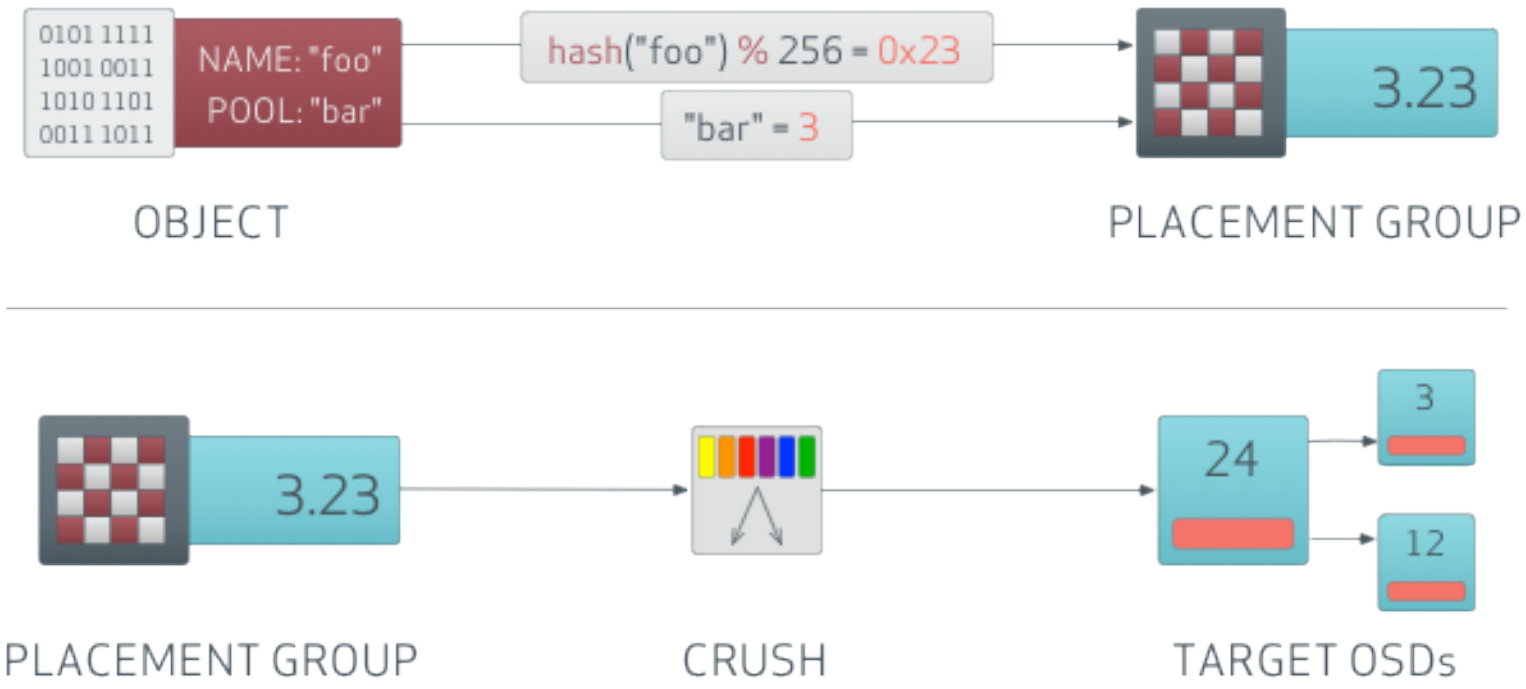
Ceph - CRUSH

- **CRUSH Algorithm**
  - Data is first split into a certain number of sections (PGs)
  - The number of placement groups (PGs) is configurable
  - Uses placement rules to determine target PG in the cluster [1]
  - This is a pseudo-random calculation [2]
  - New map is retrieved every time changes occur
  - New map is retrieved before any operation is performed
  - The CRUSH Map is maintained by the Monitors

Ceph - Placement Groups



Ceph - From Object To OSD



When calculating the location of an object, Ceph first determines the correct placement group from the object name, the total number of placement groups (256 in this example), and the destination pool. Once the placement group has been determined, CRUSH is used to calculate the target OSDs.

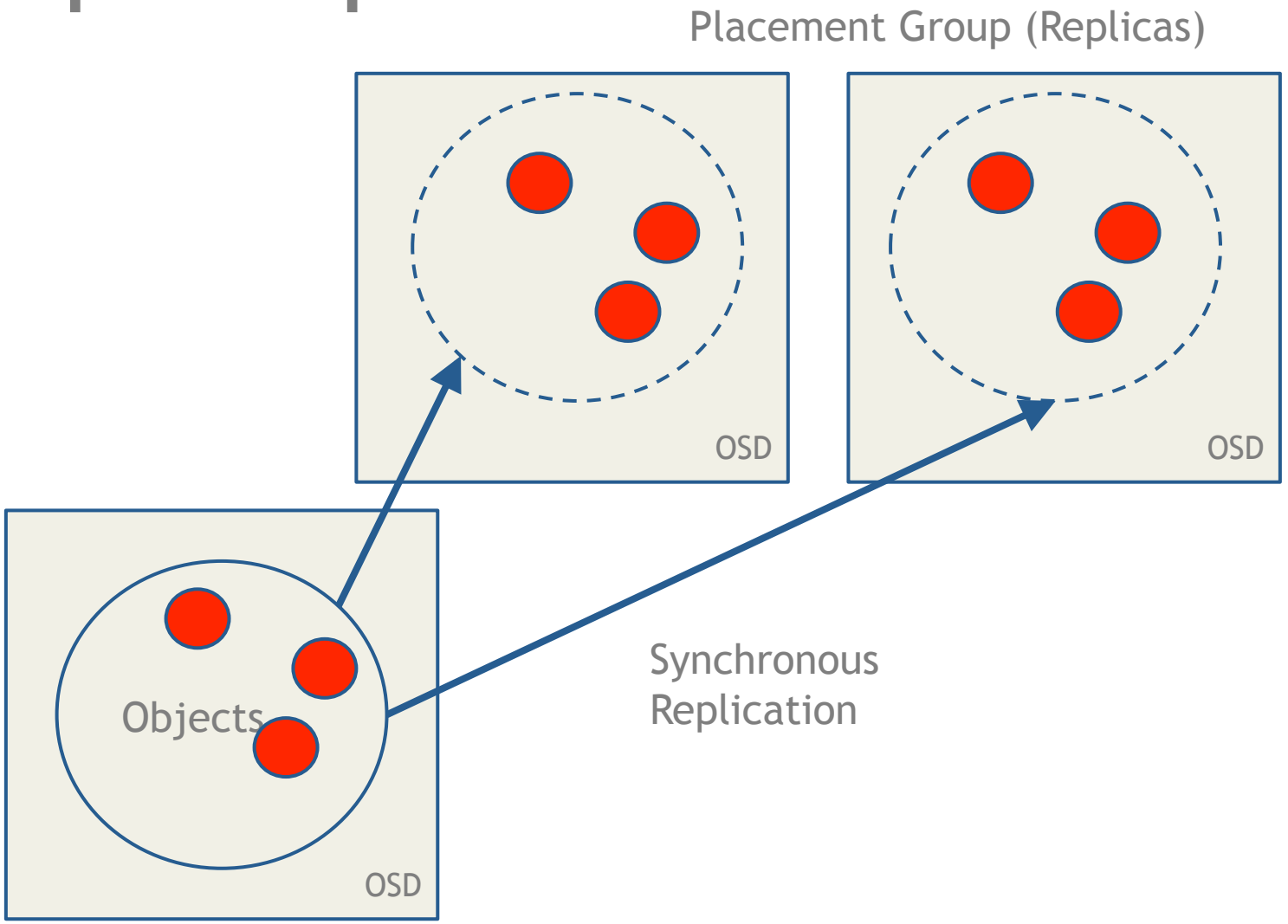
Ceph - Hierarchy In Action

```
ceph@daisy:~$sudo ceph osd tree
# id   weight  type name   up/down reweight
-5 0.03    root ssd
-4 0.03        host frank
2  0.009995            osd.2   up  1
4  0.009995            osd.4   up  1
8  0.009995            osd.8   up  1
-1 0.06    root default
-2 0.03        host daisy
0  0.009995            osd.0   up  1
3  0.009995            osd.3   up  1
6  0.009995            osd.6   up  1
-3 0.03        host eric
1  0.009995            osd.1   up  1
5  0.009995            osd.5   up  1
7  0.009995            osd.7   up  1
```

Ceph - Pools

- **Pools**
  - Will "split" a RADOS object store into numerous single parts
  - Are not a contiguous storage area in the OSDs
  - Are just a "name tag" for grouping objects
  - Have have their own set of permissions 1
    - Will let you choose which user can access which pool
    - Will let you choose if this access is read or write
    - So that not every user can access every pool

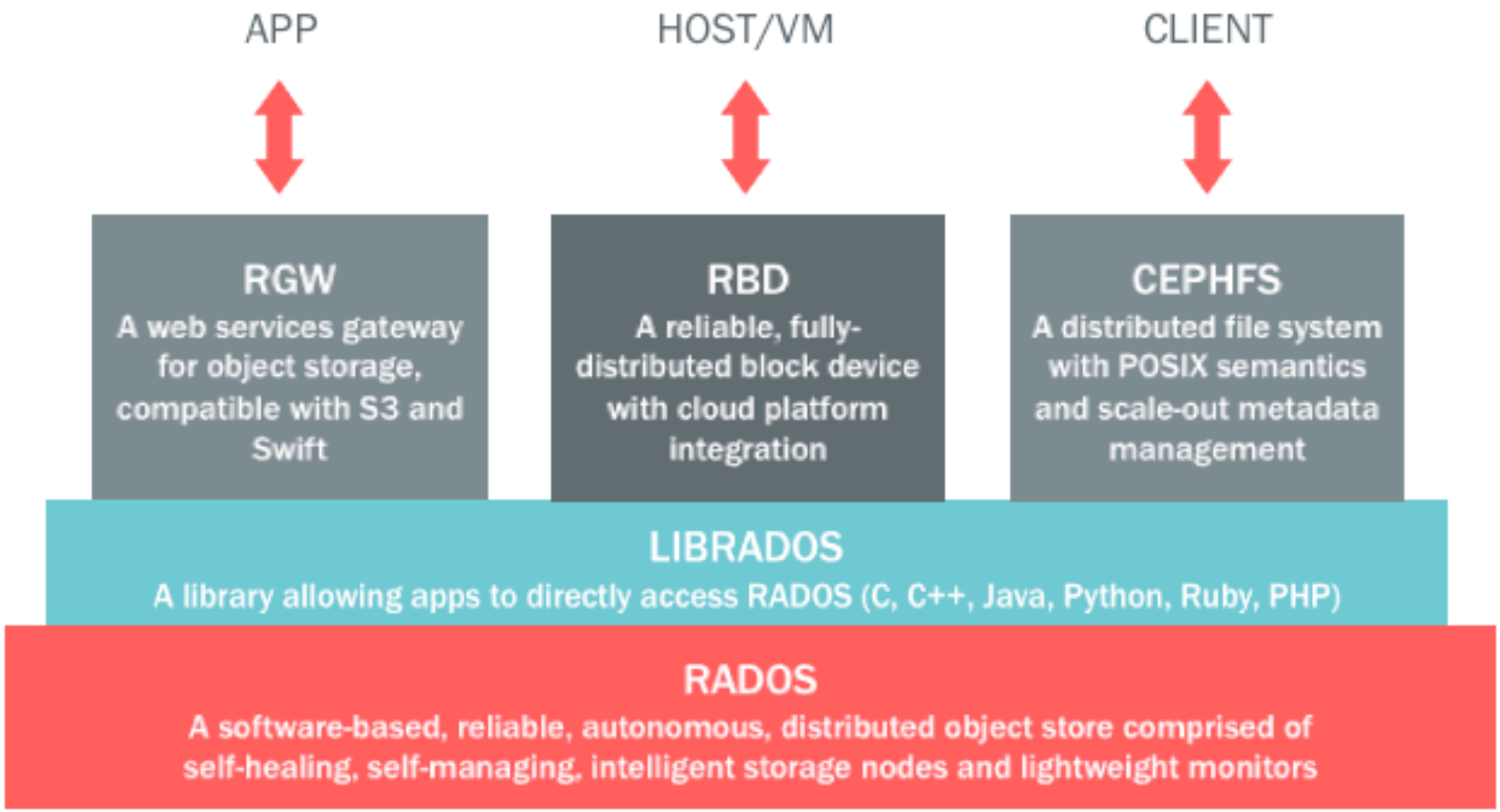
Ceph - Replciation



OpenNebula Techday

CEPH UNIFIED STORAGE

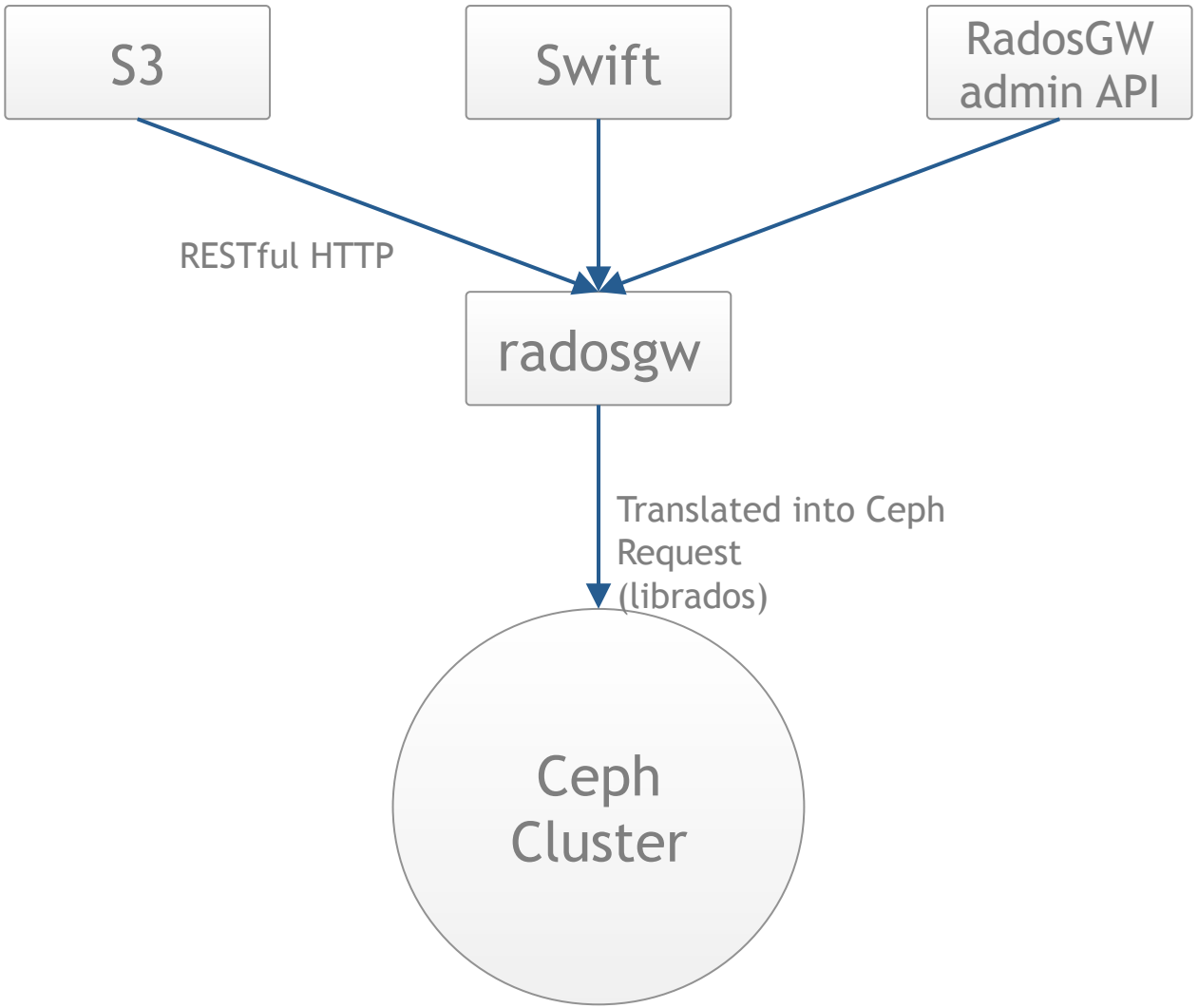
Ceph - Unified Storage



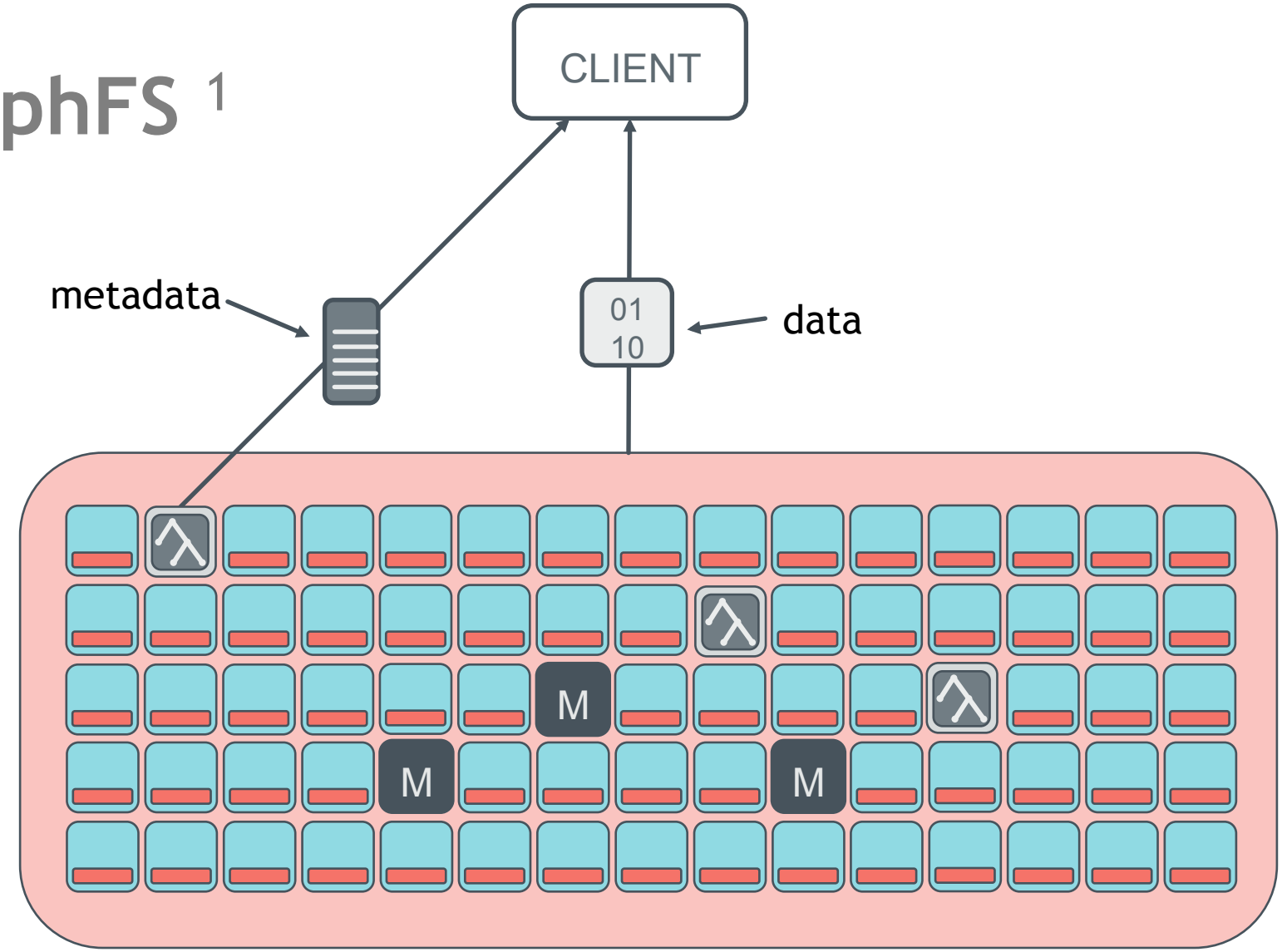
Ceph - LIBRADOS

- Ceph offers access to RADOS object store through
  - A C API (librados.h)
  - A C++ API (librados.hpp)
- Documented and simple API.
- Bindings for various languages.
- Doesn't implement striping [1]

Ceph - RADOS Gateway

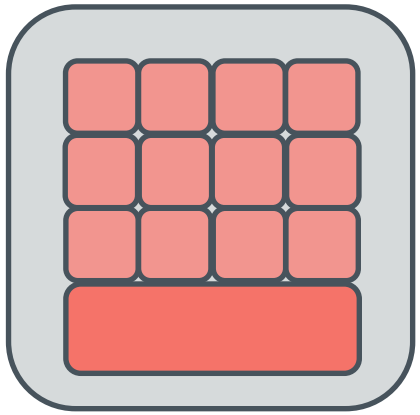


CephFS [1]



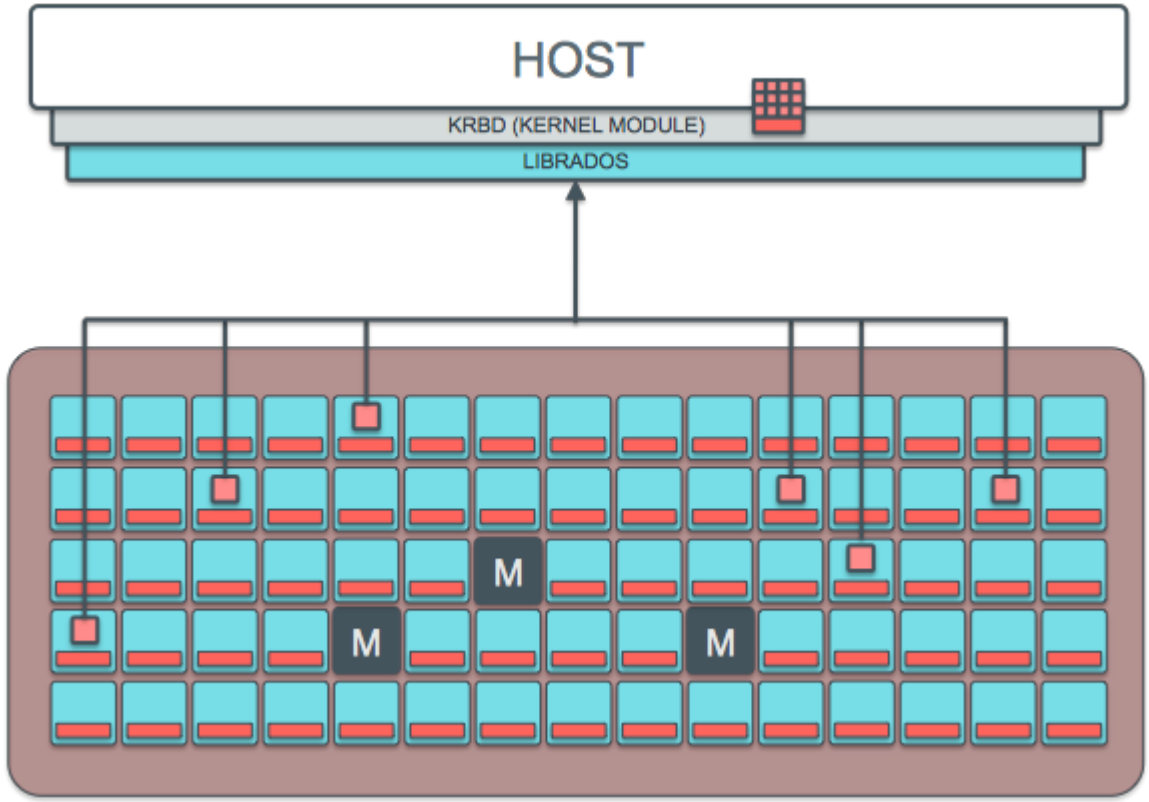
The Ceph File System (CephFS) is a parallel file system that provides a massively scalable, single-hierarchy, shared disk. At the current time, Ceph FS is not recommended for production environments.
Ceph FS should be not be mounted on a host that is a node in the Ceph Object Store.

Ceph – RADOS Block Device



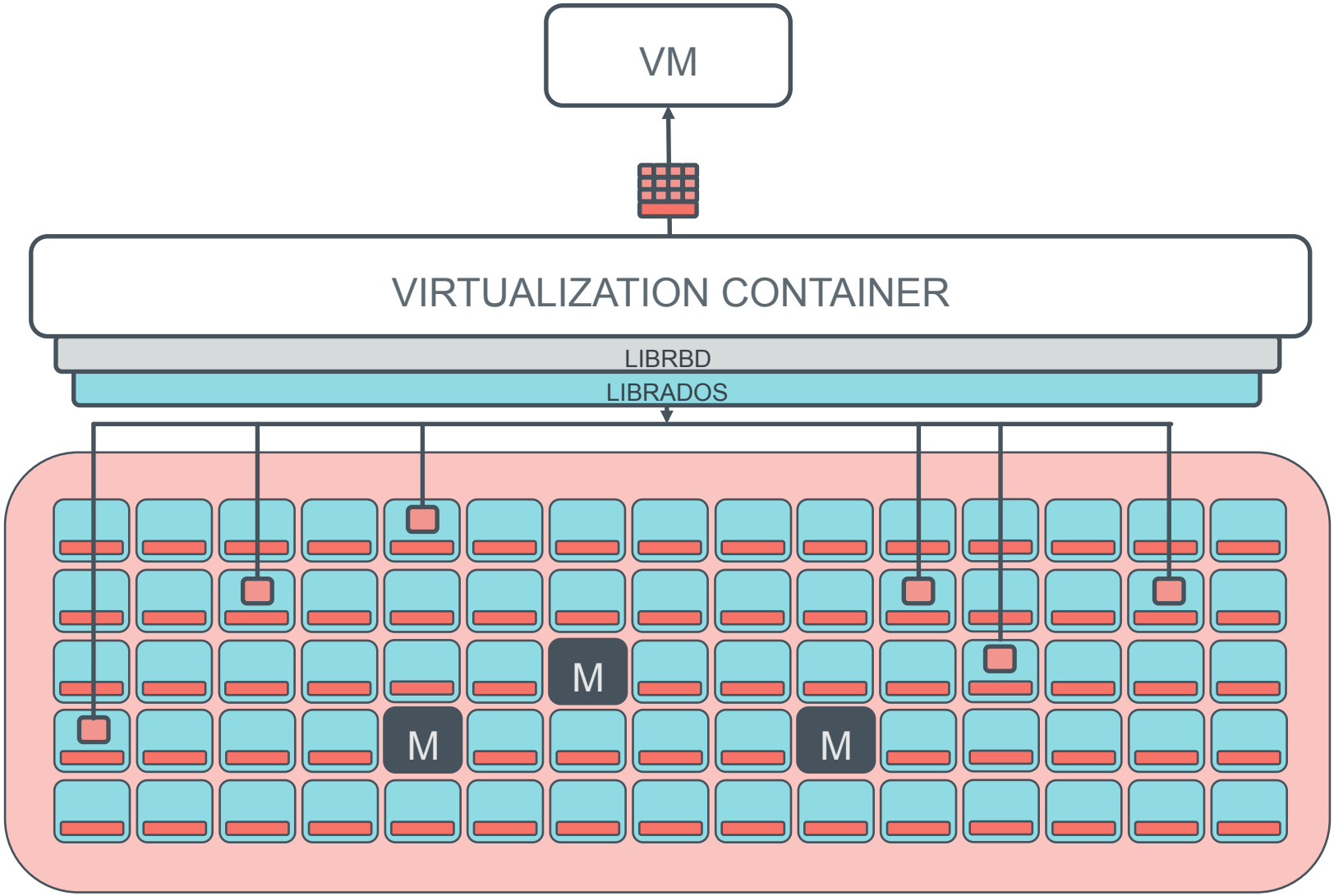
- Block-storage interfaces are the most common way to store data to disk
- Allows for storage of virtual disks in the Ceph Object Store
- Allows decoupling of VMs and containers
- High performance is gained from striping the data across the cluster
- Boot support in QEMU, KVM, and OpenStack (Cinder)
- Mount support in the Linux kernel

Ceph RBD – Native Mount



Machines (even those running on bare metal) can mount an RBD image using native Linux kernel drivers

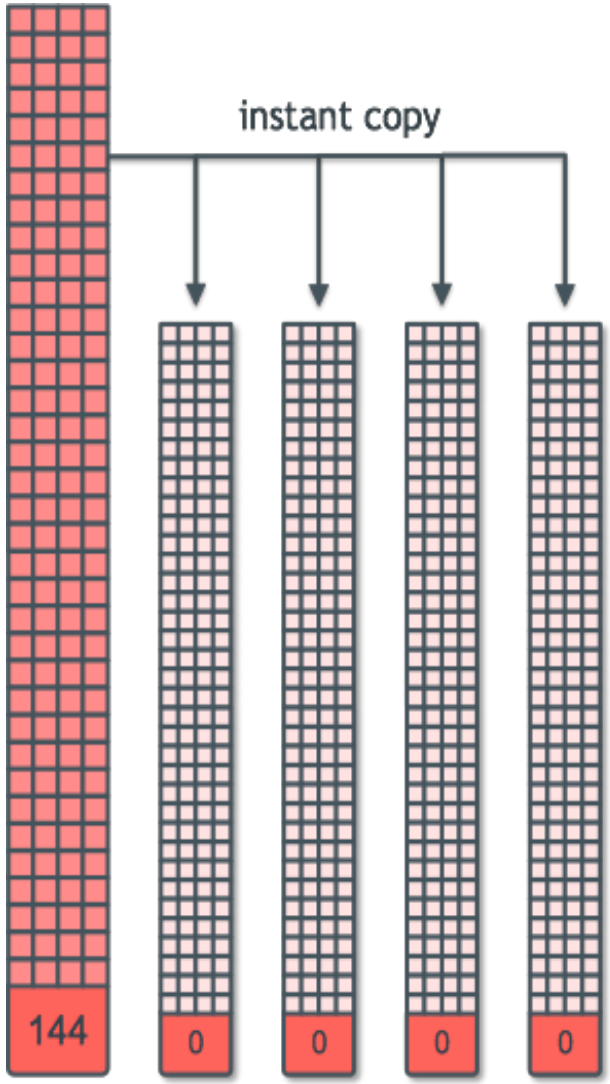
Ceph RBD - Virtual Machines



OpenNebula Techday

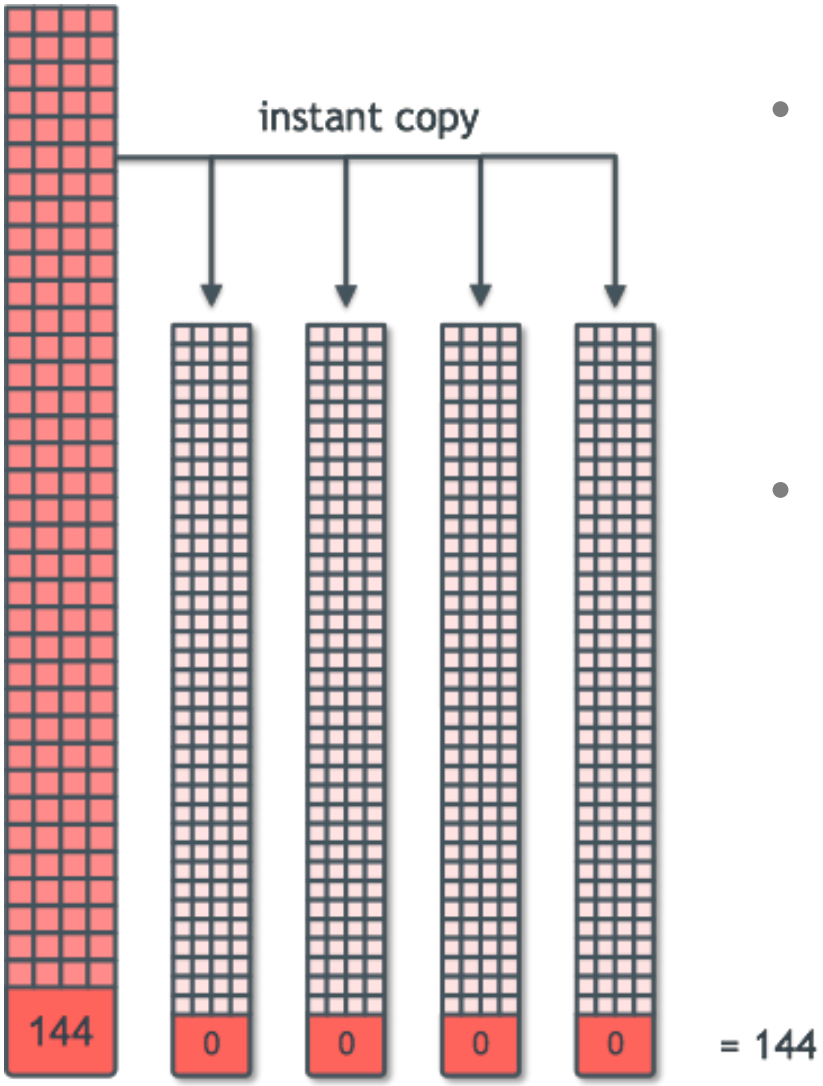
CEPH RBD FEATURES

Snapshots



- **How they work?**
  - Snapshots are instantly created
  - Read only copy of the data at the time of snapshot creation.
  - No space is consumed until the original data changes.
  - The snapshot never changes
  - Incremental RBD snap supported[1]
  - Data is read from the original data

Clones



- **Create a Clone by** [1]
  - create snapshot
  - protect snapshot
  - clone snapshot
- **A clone behaves exactly like any other Ceph block device image**
  - Read from [2]
  - write to
  - Clone [3]
  - Resize

inktank

OpenNebula Techday

Ceph Introduction (2014-06-24) - Thank You