Deploying Software Defined Storage for the Enterprise with Ceph

Paul von Stamwitz

Fujitsu

Agenda

- Yet another attempt to define SDS
- Quick Overview of Ceph from a SDS perspective
- Why Ceph is a viable component in the SDDC
- Design considerations for the Enterprise
- Challenges in deploying Ceph in the Enterprise

What does SDS mean?

Separate Control Plane?

Policy Driven?

Commodity HW?

HW Abstraction?

Open API?

Marketing Buzzword?

SLA?

Open Source?

Pools?

Automation?

Scale Out?

Converged Storage?

What does SDS mean?

Separate Control Plane?

Simplicity

Policy Driven?

Commodity HW?

HW Abstraction?

Open API?

Efficiency

Marketing Buzzword?

SLA?

Open Source?

Pools?

Automation?

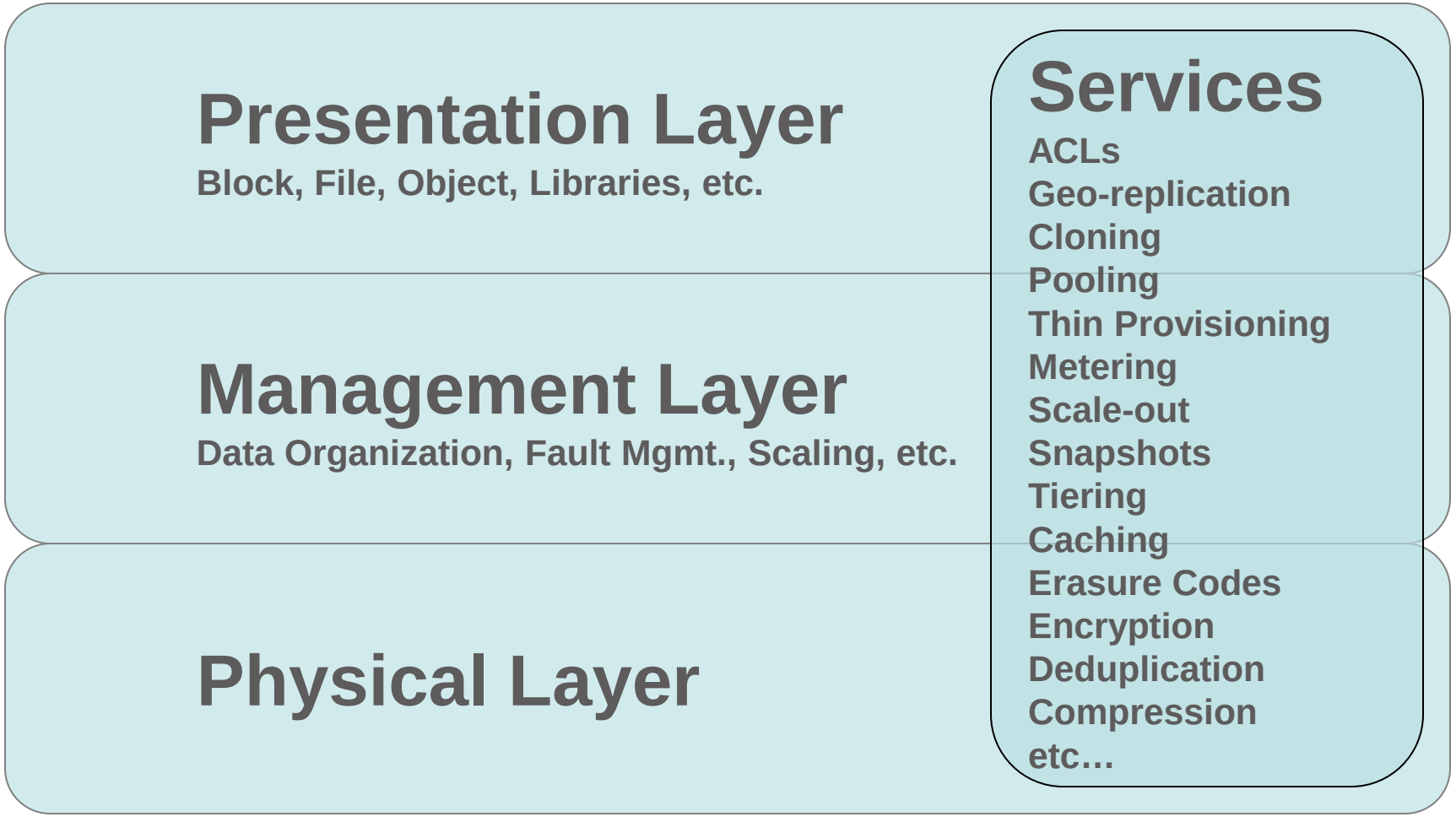
Scale Out?

Flexibility

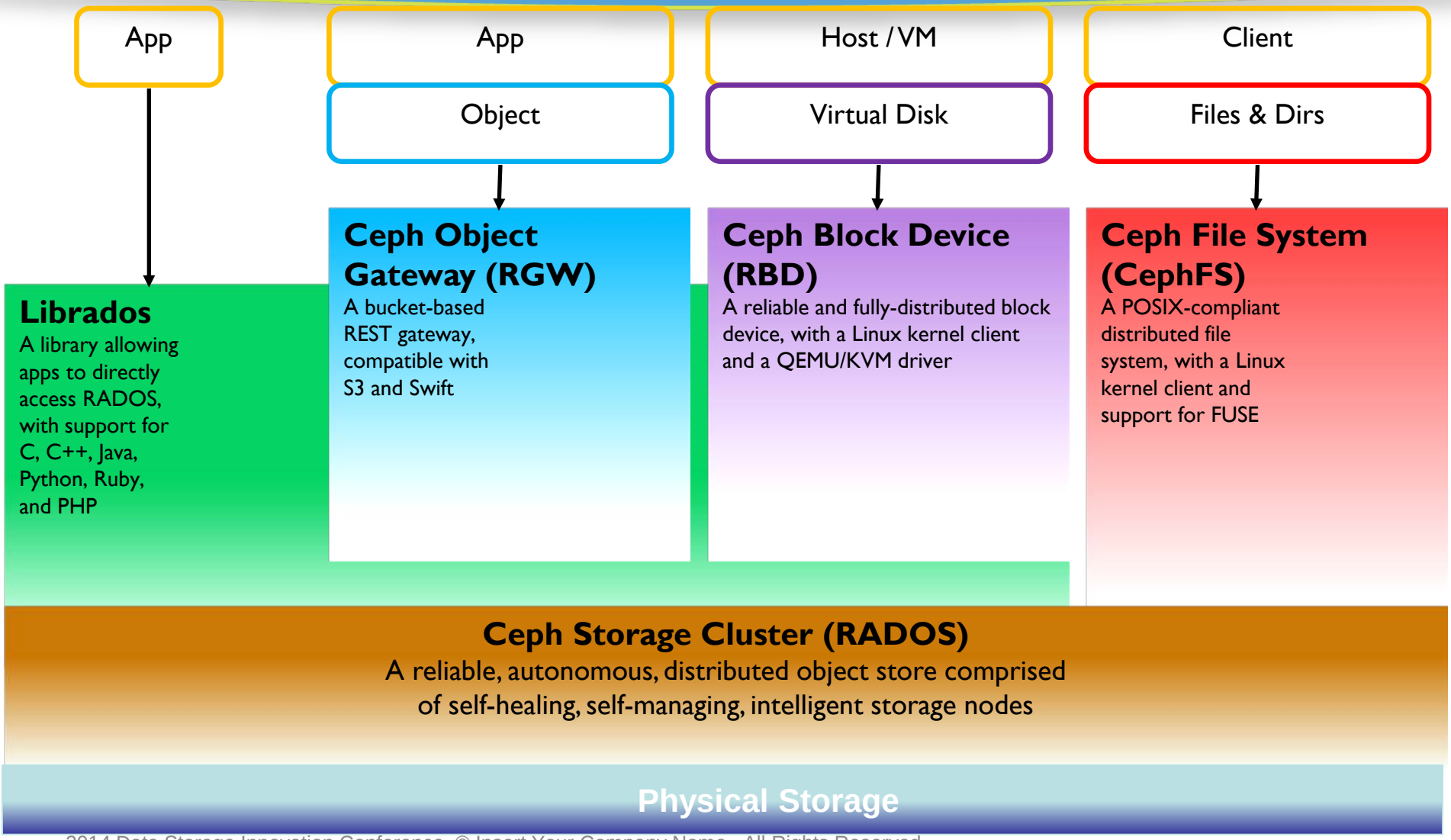
Converged Storage?

Driven by Virtualization and the Cloud

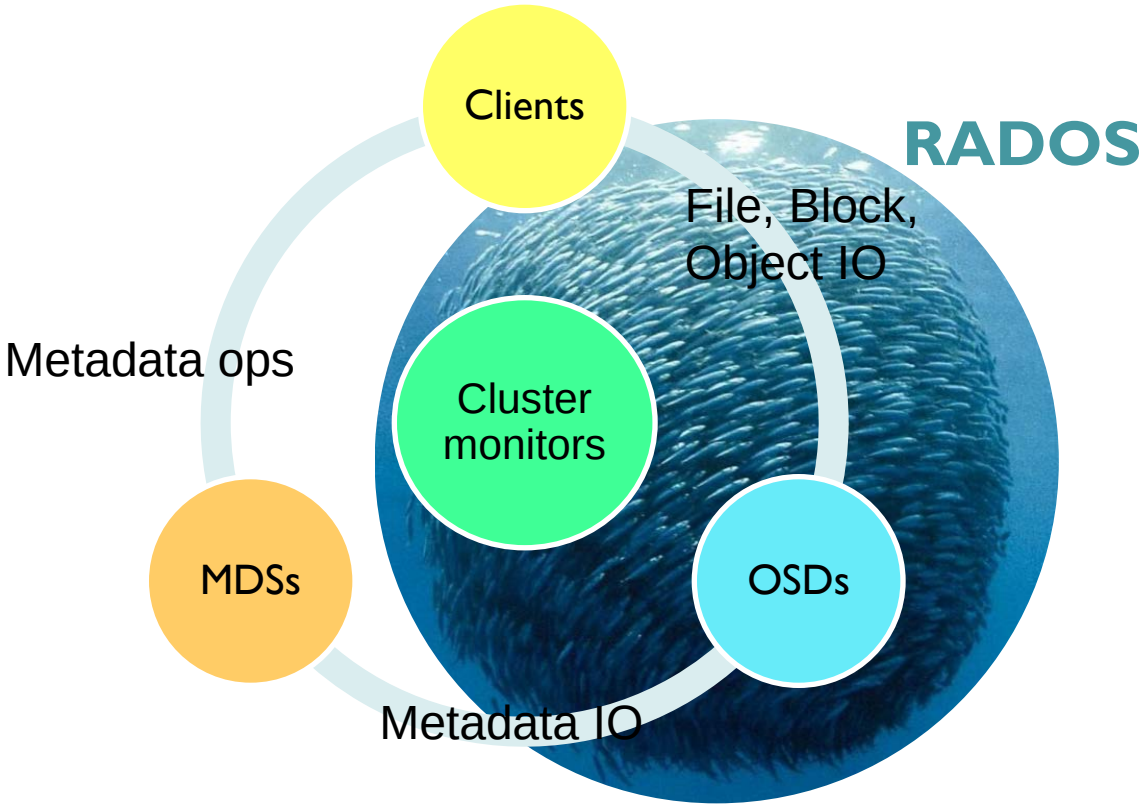
Simplified SDS Model



Ceph's Model

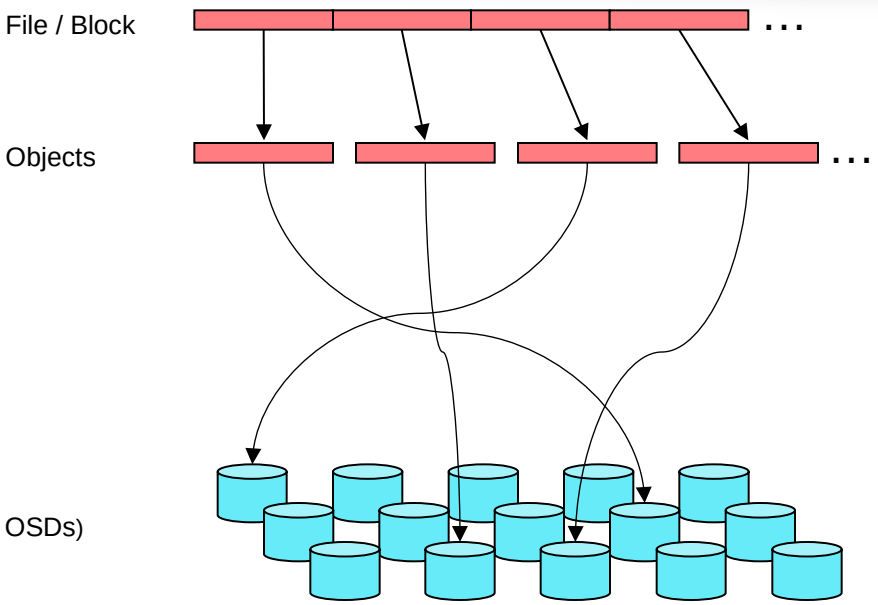


Ceph Components



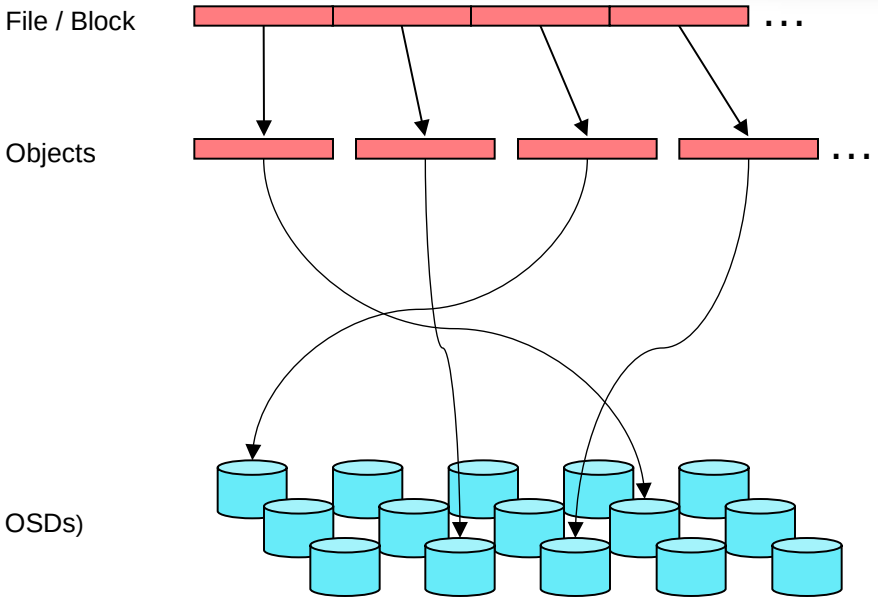
- Clients
  - Standard Interface to use the data (POSIX, Block Device, S3)
  - Transparent for Applications
- Object Storage Cluster (OSDs)
  - Stores all data and metadata
  - Organizes data into flexible-sized containers, called objects
- Monitors
  - Cluster membership
  - Authentication
  - Cluster state
  - Cluster map
- Metadata Server Cluster (MDSs)
  - Namespace Management
  - Metadata operations (open, stat, rename, …)
  - Ensure Security

Conventional Data Distribution: Consistent Hashing



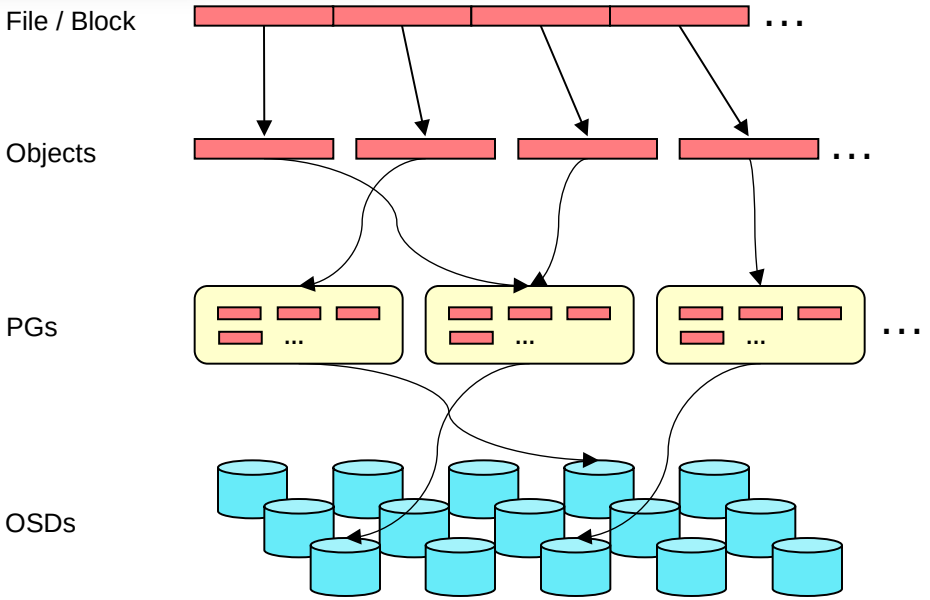
- Each object has unique hash ID
- Hash ranges assigned to physical nodes or disks
- Objects placed in appropriate “slot”
- Adding/removing disks changes has ranges

Ceph's Data Distribution: Placement Groups



Conventional

- Each object has unique hash ID
- Hash ranges assigned to physical nodes or disks
- Objects mapped to appropriate "slot"
- Adding/removing disks changes hash ranges

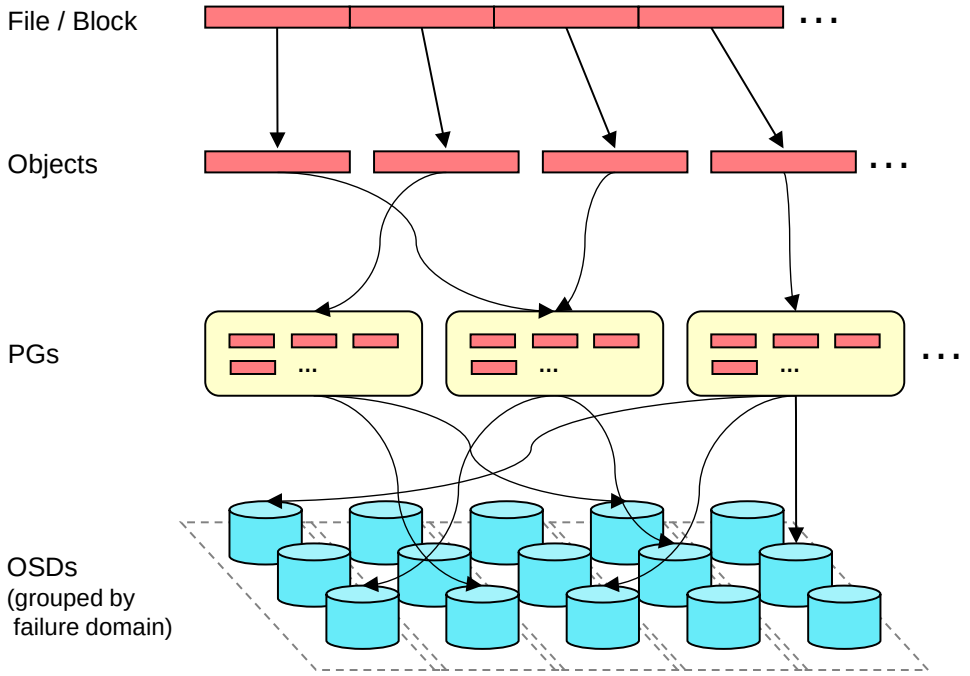


Ceph

- Hash ranges assigned to logical placement groups
- Decouples HW from hash ranges
- Objects mapped to Placement Group
- Changes in HW layer does not effect hash ranges

Ceph's Data Distribution: CRUSH

- Placement Groups mapped to sets of OSDs
  - 100's PGs per node
  - Adding/removing OSDs results in moving few PGs
- Pseudo-random, statistically uniform distribution
  - Fast: No central lookup
- Rule Based (per pool)
- Topology Aware
  - Replica can span failure domains
  - Example: 3 replicas, on same row, using different racks
- Weighted distribution
  - Weights can be assigned and changed dynamically



Ceph Protocols and Services

- **Ceph Block Device (RBD)**
  - A reliable and fully-distributed block device, with a Linux kernel client and a QEMU/KVM driver
- **Ceph Object Gateway (RGW)**
  - A bucket-based REST gateway compatible with S3 and Swift
- **Ceph File System (CephFS)**
  - A POSIX-compliant distributed filesystem
- **Libraries**
  - Integrations with Web Services
  - Integrations into Linux, Openstack, Cloudstack, KVM, XEN, and more…

- **Geo-replication**
  - Async replication for objects
  - Diff-based backup for block
- **Block-based VM Cloning**
- **Pooling**
- **Thin Provisioning**
- **Metering**
  - Performance stats
- **Scale-out**
- **Snapshots**
- **Tiering and Caching**
  - Cache Pool
- **Client-side caching for block**
- **Write Journaling**
  - For performance & consistency
- **Erasure Codes**
- **Encryption via Linux dm-crypt**
- **Compression/Dedup via btrfs**

Comprehensive Implementation of Unified Storage

Design Considerations for Enterprise Private Cloud

- Storage Efficiency
- Non-Disruptive Operations
- Integrated Data Protection: “Set it and forget it”
- Seamless Scaling – For capacity, performance, or both
- Unified Storage, include legacy interconnects
- Automatic Storage Tiering
- Service Automation and Analytics
- Continuous Data Protection
- Embedded Data Security
- Secure Multi-Tenancy
- Cost Efficient

Challenges for Ceph (1)

- Performance
  - 10Gbe with 10k rpm SAS works well
  - SSD pools and faster networks are not scaling as expected.
    - Ceph code contributes too much latency – there is room for improvement

http://snia.org/sites/default/files2/SDC2013/presentations/Revisions/DieterKasper_Transforming_PCIe-SSDs_Storage_v8.pdf

- Rebalancing
  - Minimal impact for adds and single disk loss.
  - Node loss is a concern
    - Favors small capacity nodes to minimize impact of rebalancing
- Legacy Interconnects
  - iSCSI and NFS
    - Integration with LIO and Ganesha
    - Enable Ceph storage with VMware

Challenges for Ceph (2)

- Automatic Storage Tiering
  - Multiple ways to handle this
    - Cache Pool, Hybrid Drives, etc
    - Can work in parallel or together
  - Balance performance requirements with SSD efficiency
- Data Protection
  - 3 levels
    - Non-disruptive failover between 2 main datacenters (Stretch Cluster)
    - Off-site replication (RGW Geo-replication)
    - Backup (RBD incremental backup)
- Security
  - Currently, Ceph assumes clients are trusted
    - Stronger Authorization, i.e. Kerberos
    - Enforcement of POSIX ACLs

There's more work to do, but Ceph is…

Simple

- Self-Managing and Self-Healing
- Peer-to-peer Storage Nodes

Efficient

- One System to Scale
- Easier Capacity Management

Flexible

- Adapt to Changing Demands
- 100% Open Source