# Linux Open Source Distributed Filesystem

## Ceph at SURFsara

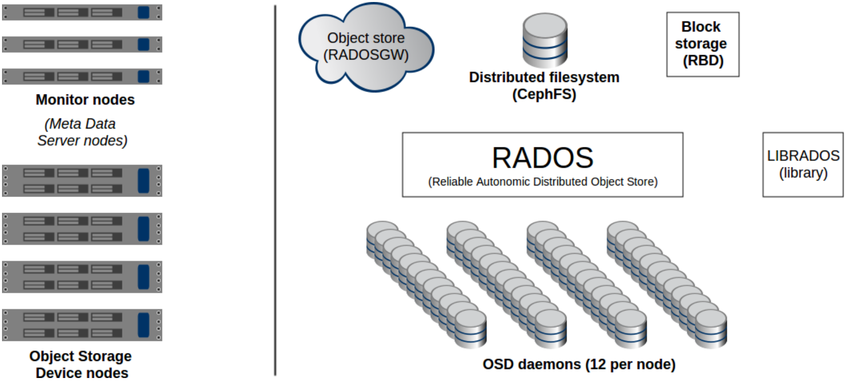
Remco van Vugt

July 2, 2013

# Agenda

- Ceph internal workings
  - Ceph components
  - CephFS
  - Ceph OSD
- Research project results
  - Stability
  - Performance
  - Scalability
  - Maintenance
  - Conclusion
- Questions

# Ceph components



**Monitor nodes**

*(Meta Data Server nodes)*

**Object Storage Device nodes**

Object store (RADOSGW)

**Distributed filesystem (CephFS)**

**Block storage (RBD)**

RADOS

(Reliable Autonomic Distributed Object Store)

LIBRADOS (library)

**OSD daemons (12 per node)**

# CephFS

- Fairly new, under heavy development
- POSIX compliant
- Can be mounted through FUSE in userspace, or by kernel driver

# CephFS (2)

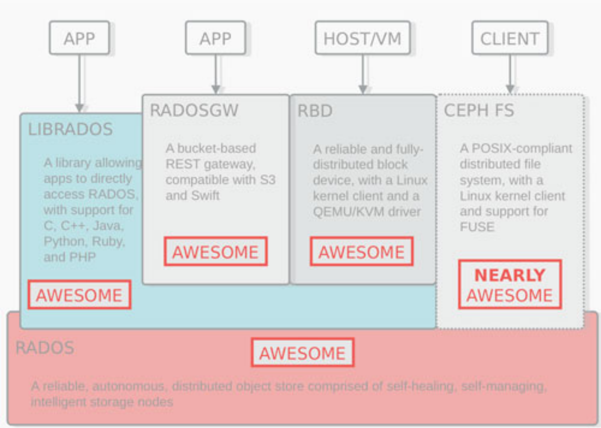

APP
APP
HOST/VM
CLIENT
LIBRADOS
A library allowing apps to directly access RADOS, with support for C, C++, Java, Python, Ruby, and PHP
AWESOME
RADOSGW
A bucket-based REST gateway, compatible with S3 and Swift
AWESOME
RBD
A reliable and fully-distributed block device, with a Linux kernel client and a QEMU/KVM driver
AWESOME
CEPH FS
A POSIX-compliant distributed file system, with a Linux kernel client and support for FUSE
NEARLY AWESOME
RADOS
AWESOME
A reliable, autonomous, distributed object store comprised of self-healing, self-managing, intelligent storage nodes

Figure: Ceph state of development

# CephFS (3)

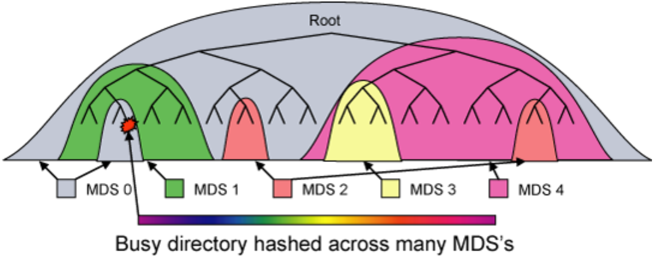
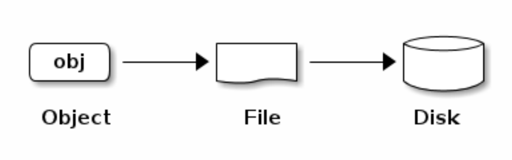


Figure: Dynamic subtree partitioning

# Ceph OSD

- Stores object data in flat files in underlying filesystem (XFS, BTRFS)
- Multiple OSDs on a single node (usually: one per disk)
- 'Intelligent daemon', handles replication, redundancy and consistency



# CRUSH

- Cluster map
- Object placement is calculated, instead of indexed
- Objects grouped into Placement Groups (PGs)
- Clients interact direct with OSDs

# Placement group

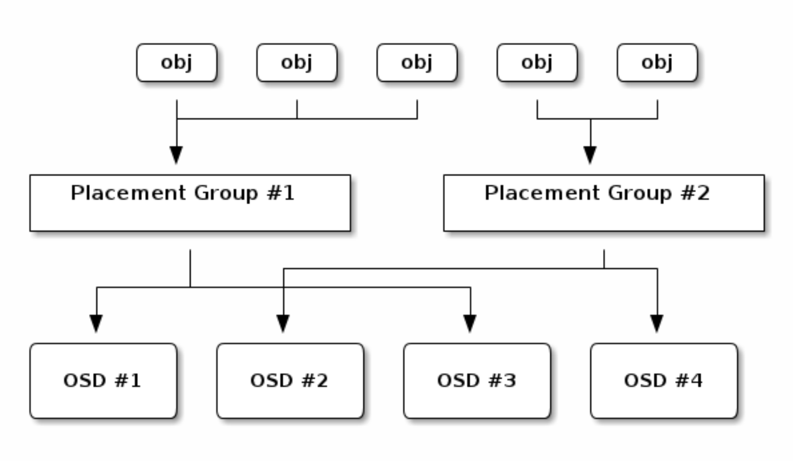


Figure: Placement groups

# Failure domains

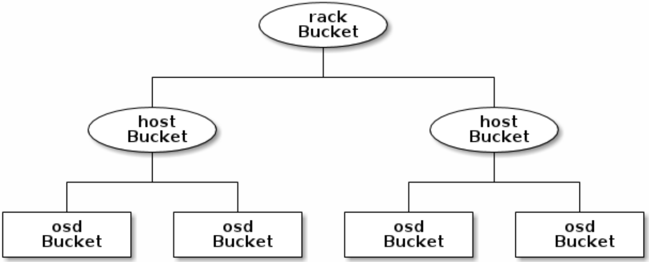


Figure: Crush algorithm

# Replication

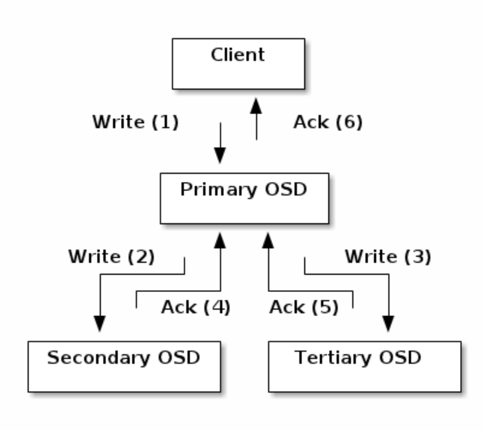


Figure: Replication

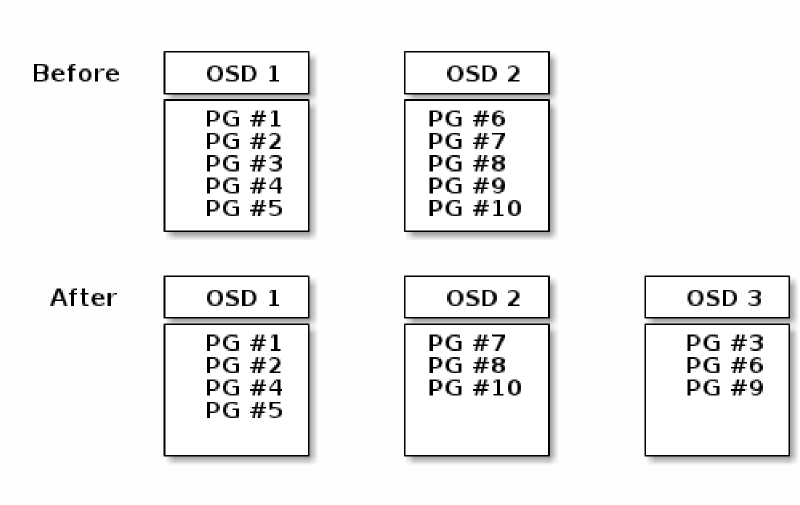
# Monitoring

- OSD use peering, and report about each other
- OSD either up or down
- OSD either in or out the cluster
- MON keeps overview, and distrubutes cluster map changes

# OSD fault recovery

- OSD down, I/O continues to secondary (or tertiary) OSD assigned to PG (active+degraded)
- OSD down longer than configured timeout, OSD is down and out (kicked out of the cluster)
- PG data is remapped to other OSD and re-replicated in the background
- PGs can be down if all copies are down

# Rebalancing



Before

OSD 1
PG #1
PG #2
PG #3
PG #4
PG #5

OSD 2
PG #6
PG #7
PG #8
PG #9
PG #10

After

OSD 1
PG #1
PG #2
PG #4
PG #5

OSD 2
PG #7
PG #8
PG #10

OSD 3
PG #3
PG #6
PG #9

# Research

## Research questions

- Research question
  - Is the current version of CephFS (0.61.3) production-ready for use as a distributed filesystem in a multi-petabyte environment, in terms of stability, scalability, performance and manageability?
- Sub questions
  - *Is Ceph, and an in particular the CephFS component, stable enough for production use at SURFsara?*
  - *What are the scaling limits in CephFS, in terms of capacity and performance?*
  - *Does Ceph(FS) meet the maintenance requirements for the environment at SURFsara?*

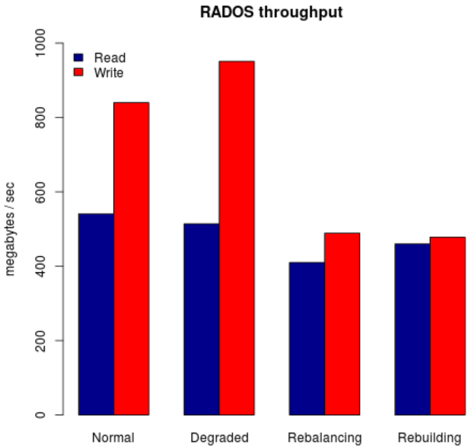
# Stability

- Various tests performed, including:
  - Cut power from OSD, MON and MDS nodes
  - Pull disks from OSD nodes (within failure domain)
  - Corrupt underlying storage files on OSD
  - Killed daemon processes
- No serious problems encountered, except for multi-mds
- Never encountered data loss

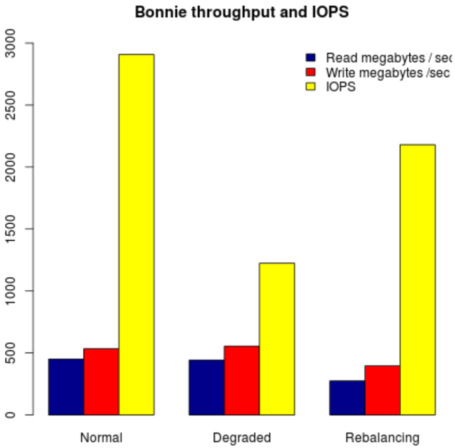
# Performance

- Benchmarked RADOS and CephFS
  - Bonnie++
  - RADOS bench
- Tested under various conditions:
  - Normal
  - Degraded
  - Rebuilding
  - Rebalancing

# RADOS Performance



# CephFS Performance



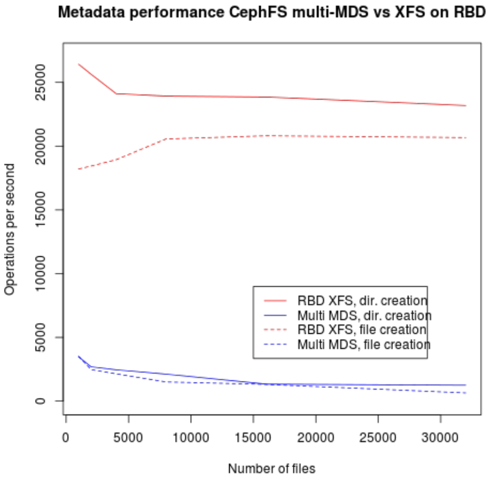
# CephFS MDS Scalability

- Tested metadata performance using mdtest
- Various POSIX operations, using 1000,2000,4000,8000 and 16000 files per directory
- Tested 1 and 3 MDS setup
- Tested single and multiple directories

# CephFS MDS Scalability (2)

- Results:
  - Did not multi-thread properly
  - Scaled over multiple MDS
  - Scaled over multiple directories
  - However...

# CephFS MDS Scalability (3)



# Ceph OSD Scalability

- Two options for scaling:
  - Horizontal: adding more OSD nodes
  - Vertical: adding more disks to OSD nodes
- But how far can we scale..?

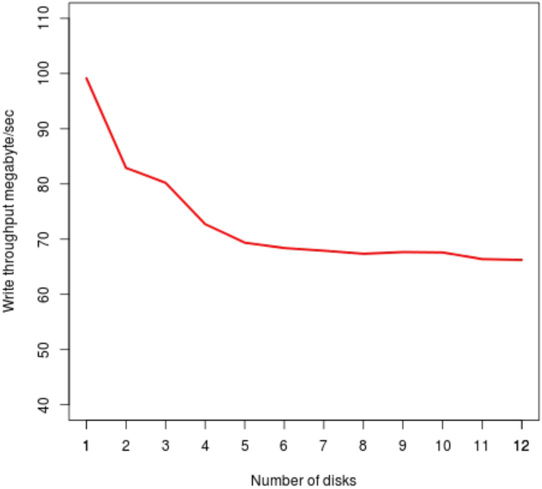
# Scaling horizontal

| Number of OSDs | PGs | MB /sec | max (MB /sec) | Overhead % |
|---|---|---|---|---|
| 24 | 1200 | 586 | 768 | 24 |
| 36 | 1800 | 908 | 1152 | 22 |
| 48 | 2400 | 1267 | 1500 | 16 |

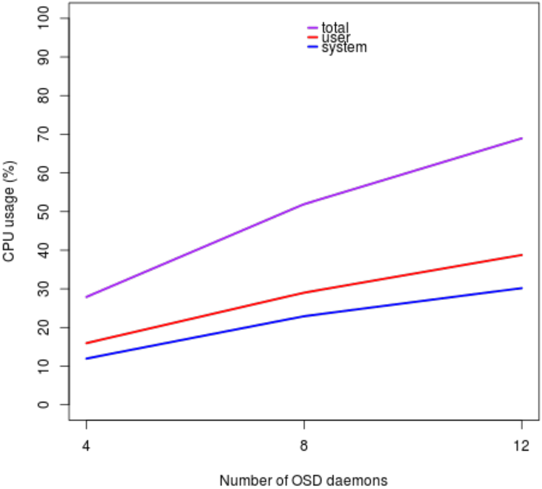
# Scaling vertical

- OSD scaling
  - Add more disks, possibly using external SAS enclosures
  - But, each disk adds overhead (CPU, I/O subsystem)

# Scaling vertical (2)



# Scaling vertical (3)



# Scaling OSDs

- Scaling horizontal seems no problem
- Scaling vertical has it's limits
  - Possibly tunable
  - Jumbo frames?

# Maintenance

- Built in tools sufficient
- Deployment
- Crowbar
- Chef
- Ceph deploy
- Configuration
- Puppet

# Research (2)

- Research question
  - Is the current version of CephFS (0.61.3) production-ready for use as a distributed filesystem in a multi-petabyte environment, in terms of stability, scalability, performance and manageability?
- Sub questions
  - *Is Ceph, and an in particular the CephFS component, stable enough for production use at SURFsara?*
  - *What are the scaling limits in CephFS, in terms of capacity and performance?*
  - *Does Ceph(FS) meet the maintenance requirements for the environment at SURFsara?*

# Conclusion

- Ceph is stable and scalable
  - RADOS storage backend
  - Possibly: RBD and object storage, but outside scope
- However: CephFS is not yet production ready
  - Scaling is a problem
  - MDS failover was not smooth
  - Multi-MDS not yet stable
  - Let alone directory sharding
- However: developer attention back on CephFS

# Conclusion (2)

- Maintenance
  - Extensive tooling available
  - Integration into existing toolset possible
  - Self-healing, low maintenance possible

# Questions?