Performance and Scalability Evaluation of the Ceph Parallel File System

Presented by Feiyi Wang

Co-authors: Mark Nelson (Inktank), Sarp Oral, Scotty Atchley, Sage Weil (Inktank), Bradley W. Settlemyer, Blake Caldwell, Jason Hill

U.S. DEPARTMENT OF ENERGY

OAK RIDGE NATIONAL LABORATORY
MANAGED BY UT-BATTELLE FOR THE DEPARTMENT OF ENERGY

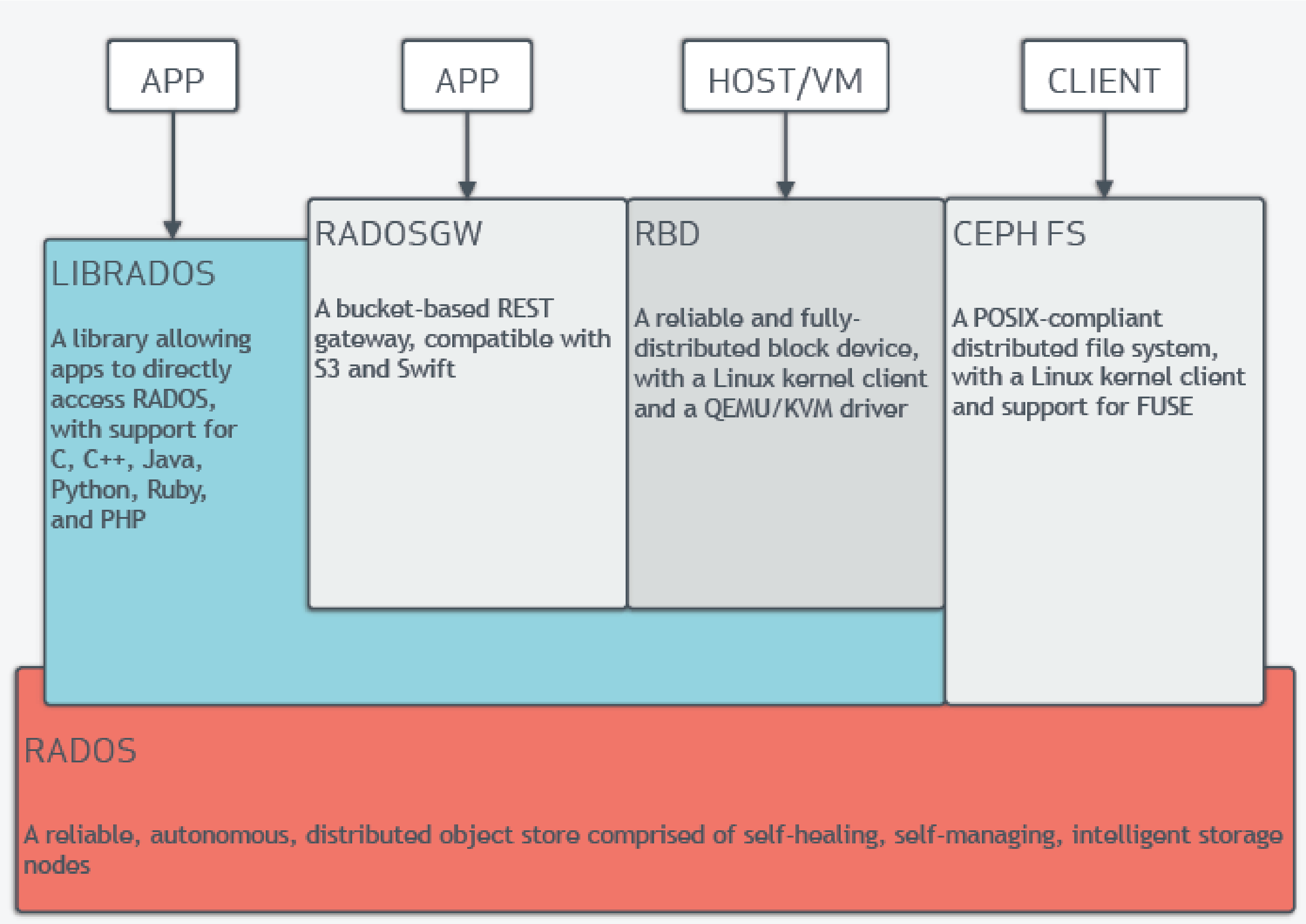
Introduction

- Oak Ridge Leadership Computing (OLCF)
  - Jaguar, served by Spider 1 (2008), 240 GB/s, 10 PB, serving more than 26,000 clients. 192 OSS and 1, 344 OSTs
  - Titan, to be served by Spider 2 (2013), 1TB/s, 32 PB (after RAID)
- Both Spider 1 and 2 are used for scratch I/O. HPSS is used for archival storage.
- New technology evaluation: Ceph for HPC?

Ceph Overview

- Ceph is a distributed storage system designed for scalability, reliability and performance.
- The system is based on a distributed object storage service called (RADOS).
- Data objects are distributed across Object Storage Devices (OSD), using CRUSH, a deterministic hashing function that allows flexible placement policies.
- CephFS builds distributed cache-coherent file system on top of RADOS.
- Ceph metadata servers store all metadata in RADOS objects; Ceph can adaptively adjust the distribution of namespace across a pool of metadata servers.

Ceph Architecture



LIBRADOS

A library allowing apps to directly access RADOS, with support for C, C++, Java, Python, Ruby, and PHP

RADOSGW

A bucket-based REST gateway, compatible with S3 and Swift

RBD

A reliable and fully-distributed block device, with a Linux kernel client and a QEMU/KVM driver

CEPH FS

A POSIX-compliant distributed file system, with a Linux kernel client and support for FUSE

RADOS

A reliable, autonomous, distributed object store comprised of self-healing, self-managing, intelligent storage nodes

Testbed Environment

- DDN SFA10K as storage backend
- SFA10K organizes disks into various RAID levels by two active-active RAID controllers; each RAID controller has two RAID processors; each RAID processor has a dual-port IB QDR cards.
- 200 SAS drives and 280 SATA drives in 10 disk enclosures.
- The storage rack is driven by 4 server hosts with IB QDR connections.

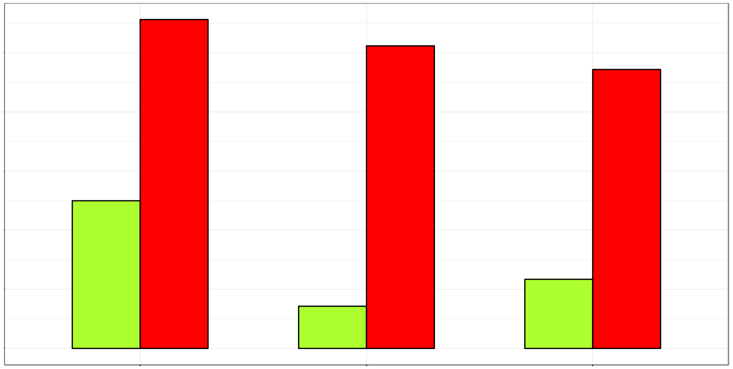
Test Methodology

- Our strategy is bottom up. Along I/O path, we establish first the expected theoretical performance, then the observed performance. After tuning efforts, we finally establish the baseline performance at that level.
- Generally, we expect performance loss as we move up; The degree of the loss is an indication of how well the system is engineered and balanced.
- Four key components:
  - Block devices
  - Local/back-end file system
  - Storage network
  - Parallel File system

Baseline Performance

- IB QDR theoretical maximum is around 3.2GB/s, in practice, we observed 3.0 GB/s. With 4 IB QDR connections, we are inline with DDN's theoretical maximum: 12 GB/s.
- Block-level:
  - Each LUN is a RAID 6 (8+2) array – 8 data disks and 2 parity disks
  - Write-back cache on has a major impact on SATA RAID group (288 MB vs. 955 MB/s), a minor impact on SAS RAID group (1.12 GB vs. 1.4 GB/s)
- Aggregate performance: we observe 11 GB/s for 28 SATA LUNs or 20 SAS LUNs
- We conclude that 11 GB/s as baseline performance number, and limitation comes from RAID controller performance.

RADOS Scaling (1)

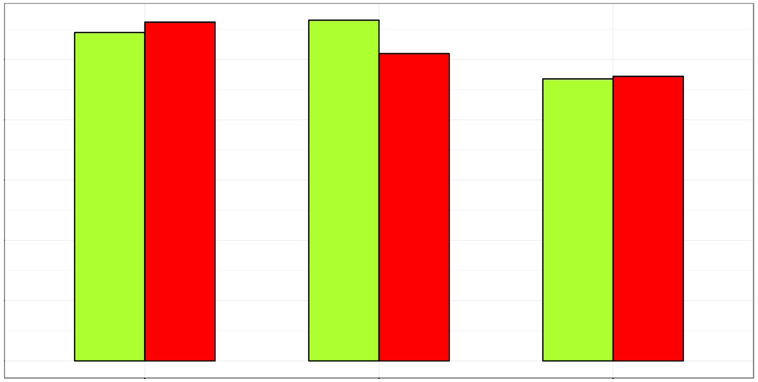


4 Servers, 4 Clients, 4MB I/O

We observed period of high performance followed by period of low performance or outright stalls across different backend file systems

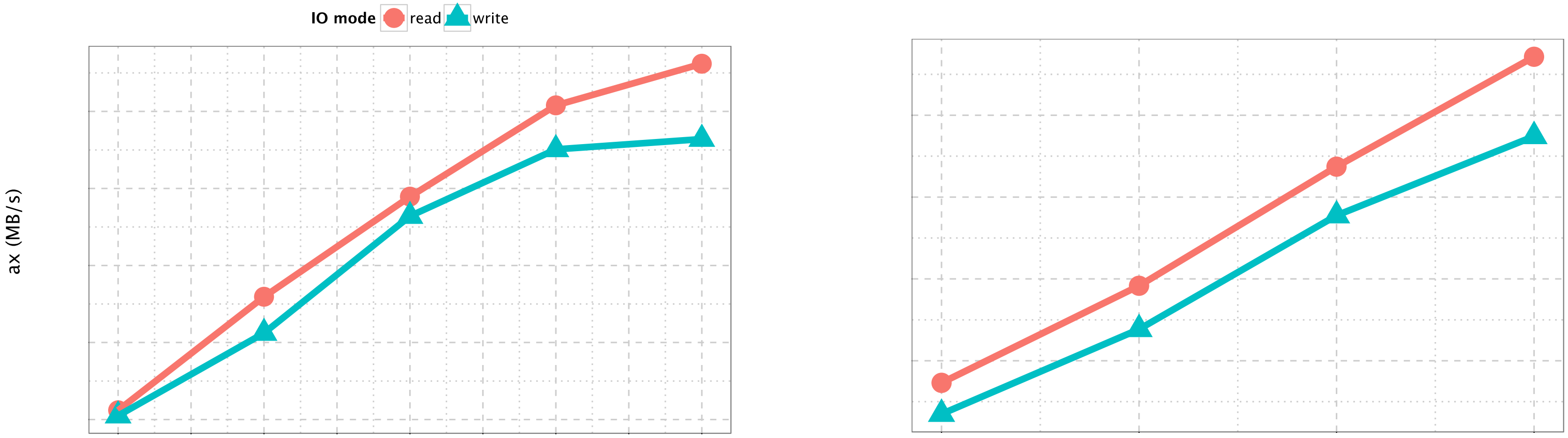
(1) TCP auto-tune enabled

Jim Schutt: “ … *unfortunate interaction between the number of OSDs/server, number of clients, TCP socket buffer autotuning, the policy throtter, and limits on the total memory used by TCP stack*”



(2) TCP auto-tune disabled

RADOS Scaling (2)



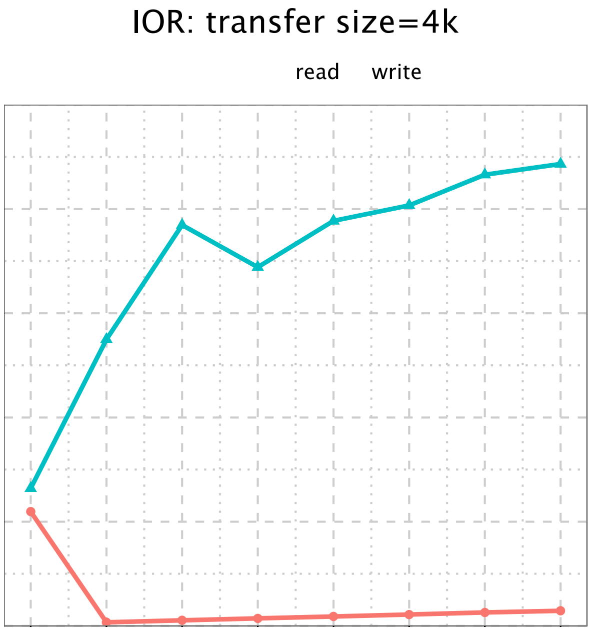
(a) Scaling OSD per server

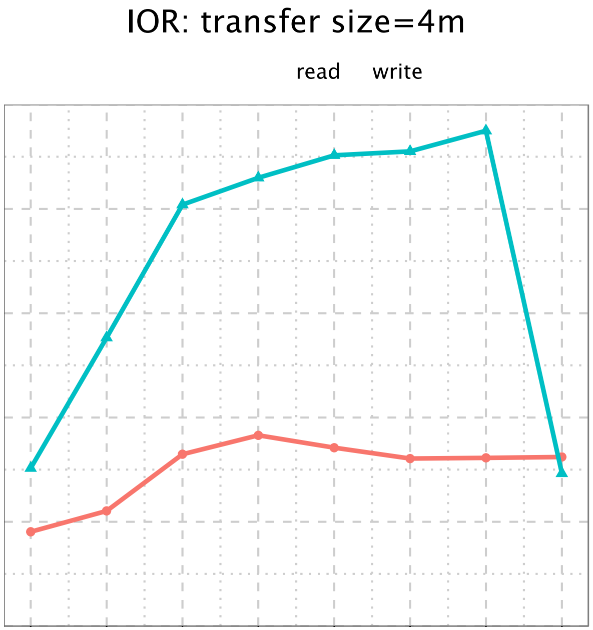
(b) Scaling OSD servers

(a) Through experimentation, we observed that number of concurrent operations
are critical to archive high throughput. The graph shows 32 concurrent 4MB objects in flight.
All tests are performed with replication set to 1.

(b) 4 OSD servers, each with 11 OSDs. The perfect scaling would give us aggregate read
at 6616MB/s and write at 5640 MB/s; We are observing a loss of 13.6% and 16.0% respectively.

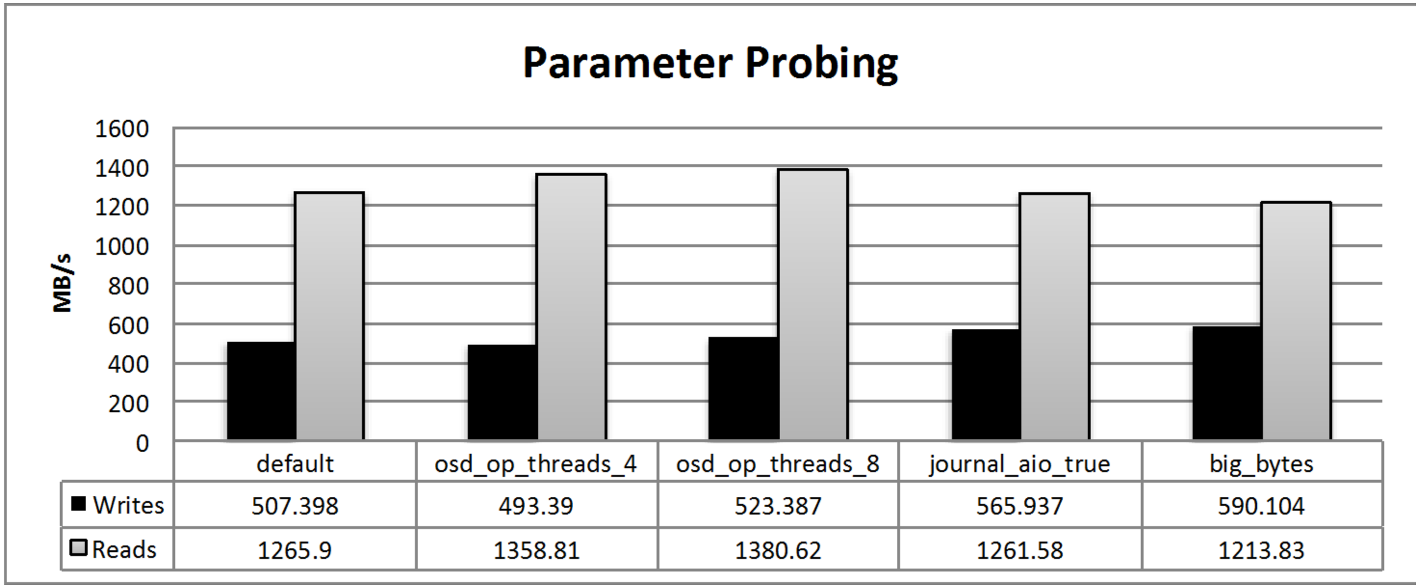
File System Level: A Different Story





Bottom line: though we have obtained reasonable performance at RADOS level, it did not translate into file system level performance, at all.

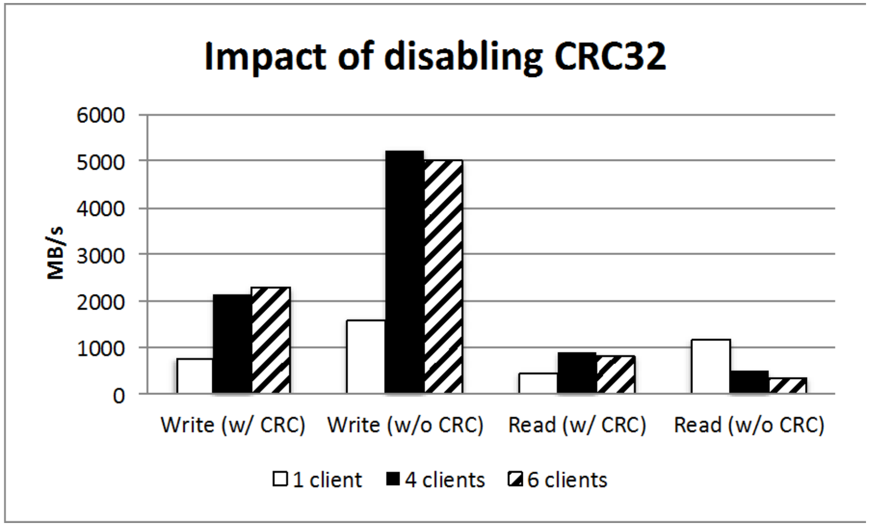
Improving RADOS


Parameter Probing

	default	osd_op_threads_4	osd_op_threads_8	journal_aio_true	big_bytes
■ Writes	507.398	493.39	523.387	565.937	590.104
□ Reads	1265.9	1358.81	1380.62	1261.58	1213.83

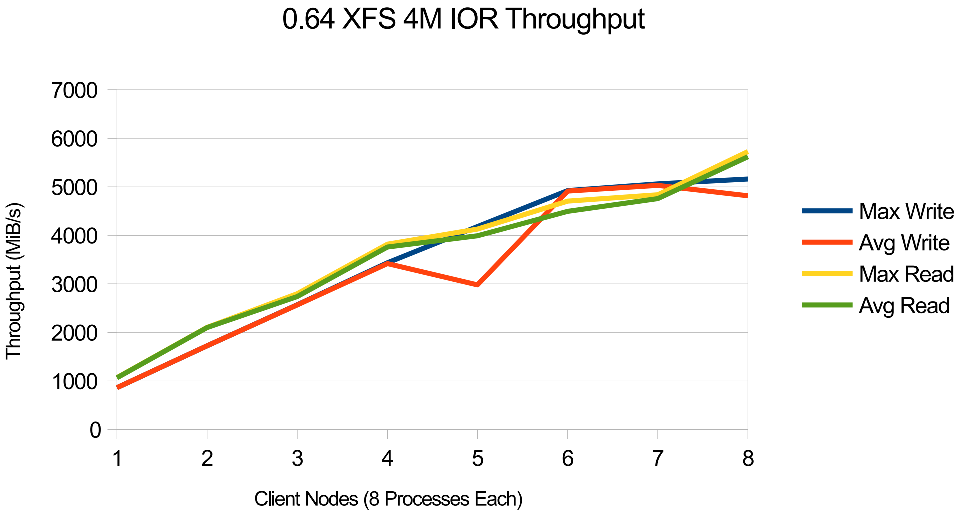
- osd_op_threads, 7.3% and 9% improvement
- journal_aio, 11.5% and 16.3% improvement
- Other probed parameters: no tangible and repeatable impacts

Improving Ceph File System Performance



We observed significant performance impact due to client side CRC32. More so on write then read.

Inktank has since implemented SSE4 instruction based CRC32 for Intel CPU.



To improvement IOR scaling performance:

(1) Increase read-ahead cache on client side
(2) Inktank investigated heavy lock contention during parallel compaction in Linux memory manager. A bug in kernel 3.5

Summary

- Ceph is still under rapid development, and our results shows that. In between versions, large performance swings. Comparing to CephFS, RADOS is much more stable.
- Through tuning efforts, we are able to observe Ceph perform at about 70% of raw hardware capacity at RADOS level and 62% at file system level.
- Ceph performs "metadata + data" journaling, which maybe fine for host system with locally attached disks, but hurts in SFA10K-alike hardware, where block devices are exposed through IB over SRP protocol.