TR5: Scalability Study of Ceph

Xing Lin
xinglin@cs.utah.edu
University of Utah

03/15/2013

1 Introduction

This document presents the results about the performance scalability of Ceph, running with AT&T virtual machines.

2 Experiment Setup

I did my experiments with 4 virtual machines, provided by a teammate from AT&T. Three virtual machines were used as data nodes and the forth was used as the client machine for benchmarking. The configuration of each VM is presented in Table 1.

CPU	8×2.7 GHz cores
Memory	32 GB
Virtual Disk	1×343 GB

Table 1: Virtual machine configuration

The configuration of the Ceph cluster is presented in Table 2. Table 3 presents the tools used in this measurement.

Attribute	Value
Number of osds	3
Journal	10 GB ramdisk
Ceph osd pool size (num of replicas)	3
Placement group number	100

Table 2: Ceph configurations

I only considered sequential workloads, because the performance of random workloads is not stable (which will be demonstrated in Section 3.5.). For sequential workloads, the default values of these parameters are presented in Table 4.

tool	version
OS	Ubuntu12.04
Ceph	Argoguant.0.56.3
fio	2.0.14

Table 3: Tools

block size	duration	directio	ioengine
4 MB	120 s	1	libaio

Table 4: Workload parameters

3 Results

3.1 Raw Disk Throughput

I did a simple measurement with dd (read/write 32 GB data in 4 MB blocks), to get the raw sequential read/write bandwidths of these virtual disks. The bandwidths for each virtual machine are presented in Table 5.

VM	4 MB Read	4 MB Write
vm1	193 MB/s	401 MB/s
vm2	224 MB/s	295 MB/s
vm3	216 MB/s	149 MB/s

Table 5: Raw bandwidths of the virtual disk for each virtual machine

Though the read bandwidths for these three virtual disks are quite similar, the write bandwidths are different quite significantly.

For scalability, I mainly looked into three aspects: IO depth, block size, and parallel reads/writes to many Rados block devices.

3.2 IO Depth

I first looked into how iodepth affects the throughput of a workload. The bandwidth was used as the throughput metric. The results are presented in Figure 1.

3.3 Block Size

In this measurement, I fixed iodepth=1 and varied the block size (not fixed at 4 MB for this experiment). The results are presented in Figure 2.

3.4 Parallel Writers/Readers

In this set of experiments, iodepth was fixed at 1 and the block size was fixed at 4 MB. The number of Rados block devices (RBD) that were concurrently read/written was varied. The results are presented in Figure 3.

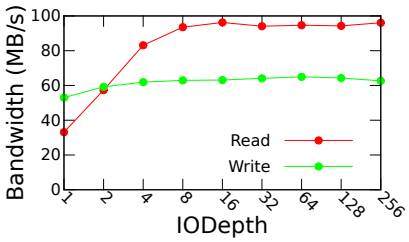


Figure 1: Bandwidths of sequential workloads, varying IO depth

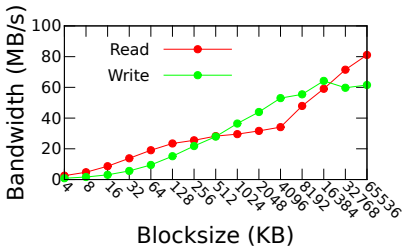


Figure 2: Bandwidths of sequential workloads, varying the block size

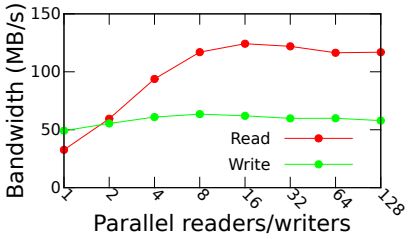
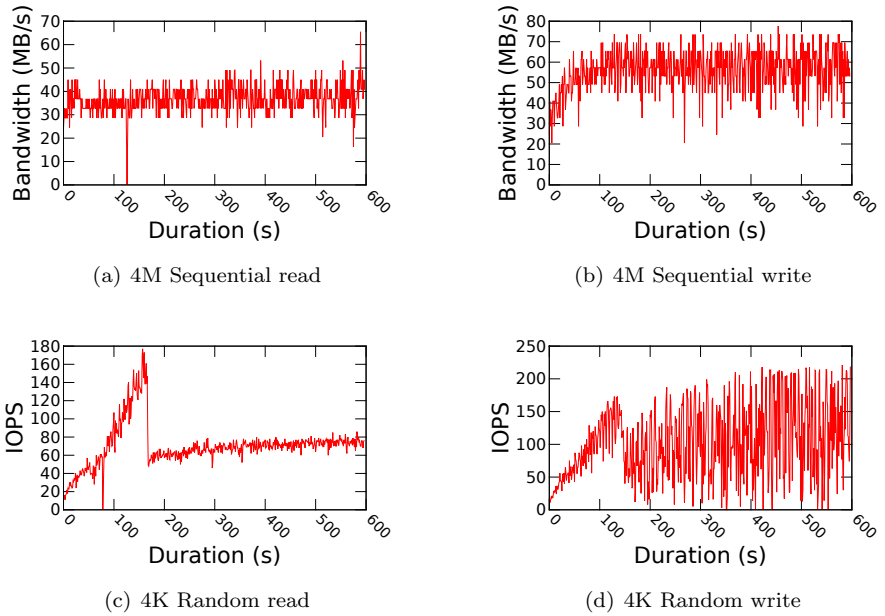


Figure 3: Aggregated throughputs of sequential workloads, when the number of parallel readers/writers is varied.

3.5 Stability Study of Workloads

Here I presented four figures to study the stability of different types of workloads. For sequential workloads, I presented the dynamic bandwidth while for random workloads, dynamic IOPS was presented. Basically, what we can see from these figures is that bandwidths of sequential workloads are more stable than IOPSs for random workloads (For sequential workloads, I also verified the stability for blocksize=4K and IODEPTH=32). IOPS of the random read workload becomes stable after the startup phase while for the random write workload, IOPS does not become stable at all.



(a) 4M Sequential read

(b) 4M Sequential write

(c) 4K Random read

(d) 4K Random write

Figure 4: Dynamical IOPSs of random workloads (IODepth=1).

4 Discussions

What we can see from Figure 1 is that bandwidths increase as iodepth is increased.

Figure 2 shows that the block size affects the bandwidth significantly. Ceph can **not** provide a high bandwidth for small block size workloads.

Figure 3 shows that as we increases the number of concurrent RBDs to read from, the aggregated bandwidth increases. That illustrates that Ceph is scalable for serving parallel reads. However, for parallel writes, the aggregated bandwidth does not change that much. One aspect that likely contributes to this difference is that for each write, Ceph will replicate it to all three virtual disks we are using. For reads, Ceph only needs to read from one disk.