HARDWARE RECOMMENDATIONS

Ceph was designed to run on commodity hardware, which makes building and maintaining petabyte-scale data clusters economically feasible. When planning out your cluster hardware, you will need to balance a number of considerations, including failure domains and potential performance issues. Hardware planning should include distributing Ceph daemons and other processes that use Ceph across many hosts. Generally, we recommend running Ceph daemons of a specific type on a host configured for that type of daemon. We recommend using other hosts for processes that utilize your data cluster (e.g., OpenStack, CloudStack, etc).

Inktank provides excellent premium support for hardware planning.

Tip: Check out the Ceph blog too. Articles like Ceph Write Throughput 1, Ceph Write Throughput 2, Argonaut v. Bobtail Performance Preview, Bobtail Performance - I/O Scheduler Comparison and others are an excellent source of information.

CPU

Ceph metadata servers dynamically redistribute their load, which is CPU intensive. So your metadata servers should have significant processing power (e.g., quad core or better CPUs). Ceph OSDs run the RADOS service, calculate data placement with CRUSH, replicate data, and maintain their own copy of the cluster map. Therefore, OSDs should have a reasonable amount of processing power (e.g., dual core processors). Monitors simply maintain a master copy of the cluster map, so they are not CPU intensive. You must also consider whether the host machine will run CPU-intensive processes in addition to Ceph daemons. For example, if your hosts will run computing VMs (e.g., OpenStack Nova), you will need to ensure that these other processes leave sufficient processing power for Ceph daemons. We recommend running additional CPU-intensive processes on separate hosts.

RAM

Metadata servers and monitors must be capable of serving their data quickly, so they should have plenty of RAM (e.g., 1GB of RAM per daemon instance). OSDs do not require as much RAM for regular operations (e.g., 200MB of RAM per daemon instance); however, during recovery they need significantly more RAM (e.g., 500MB-1GB). Generally, more RAM is better.

DATA STORAGE

Plan your data storage configuration carefully. There are significant cost and performance tradeoffs to consider when planning for data storage. Simultaneous OS operations, and simultaneous request for read and write operations from multiple daemons against a single drive can slow performance considerably. There are also file system limitations to consider: btrfs is not quite stable enough for production, but it has the ability to journal and write data simultaneously, whereas XFS and ext4 do not.

Important: Since Ceph has to write all data to the journal before it can send an ACK (for XFS and EXT4 at least), having the journals and OSD performance in balance is really important!

HARD DISK DRIVES

OSDs should have plenty of hard disk drive space for object data. We recommend a minimum hard disk drive size of 1 terabyte. Consider the cost-per-gigabyte advantage of larger disks. We recommend dividing the price of the hard disk drive by the number of gigabytes to arrive at a cost per gigabyte, because larger drives may have a significant impact on the cost-per-gigabyte. For example, a 1 terabyte hard disk priced at $75.00 has a cost of $0.07 per gigabyte (i.e., $75 / 1024 = 0.0732). By contrast, a 3 terabyte hard disk priced at $150.00 has a cost of $0.05 per gigabyte (i.e., $150 / 3072 = 0.0488). In the foregoing example, using the 1 terabyte disks would generally increase the cost per gigabyte by 40%–rendering your cluster substantially less cost efficient.

Tip: Running multiple OSDs on a single disk–irrespective of partitions–is **NOT** a good idea.

Tip: Running an OSD and a monitor or a metadata server on a single disk–irrespective of partitions–is **NOT** a good idea either.

Storage drives are subject to limitations on seek time, access time, read and write times, as well as total throughput. These physical limitations affect overall system performance–especially during recovery. We recommend using a dedicated drive for

the operating system and software, and one drive for each OSD daemon you run on the host. Most “slow OSD” issues arise due to running an operating system, multiple OSDs, and/or multiple journals on the same drive. Since the cost of troubleshooting performance issues on a small cluster likely exceeds the cost of the extra disk drives, you can accelerate your cluster design planning by avoiding the temptation to overtax the OSD storage drives.

You may run multiple OSDs per hard disk drive, but this will likely lead to resource contention and diminish the overall throughput. You may store a journal and object data on the same drive, but this may increase the time it takes to journal a write and ACK to the client. Ceph must write to the journal before it can ACK the write. The btrfs filesystem can write journal data and object data simultaneously, whereas XFS and ext4 cannot.

Ceph best practices dictate that you should run operating systems, OSD data and OSD journals on separate drives.

SOLID STATE DRIVES

One opportunity for performance improvement is to use solid-state drives (SSDs) to reduce random access time and read latency while accelerating throughput. SSDs often cost more than 10x as much per gigabyte when compared to a hard disk drive, but SSDs often exhibit access times that are at least 100x faster than a hard disk drive.

SSDs do not have moving mechanical parts so they aren’t necessarily subject to the same types of limitations as hard disk drives. SSDs do have significant limitations though. When evaluating SSDs, it is important to consider the performance of sequential reads and writes. An SSD that has 400MB/s sequential write throughput may have much better performance than an SSD with 120MB/s of sequential write throughput when storing multiple journals for multiple OSDs.

Important: We recommend exploring the use of SSDs to improve performance. However, before making a significant investment in SSDs, we **strongly recommend** both reviewing the performance metrics of an SSD and testing the SSD in a test configuration to gauge performance.

Since SSDs have no moving mechanical parts, it makes sense to use them in the areas of Ceph that do not use a lot of storage space. Relatively inexpensive SSDs may appeal to your sense of economy. Use caution. Acceptable IOPS are not enough when selecting an SSD for use with Ceph. There are a few important performance considerations for journals and SSDs:

- **Write-intensive semantics:** Journaling involves write-intensive semantics, so you should ensure that the SSD you choose to deploy will perform equal to or better than a hard disk drive when writing data. Inexpensive SSDs may introduce write latency even as they accelerate access time, because sometimes high performance hard drives can write as fast or faster than some of the more economical SSDs available on the market!
- **Sequential Writes:** When you store multiple journals on an SSD you must consider the sequential write limitations of the SSD too, since they may be handling requests to write to multiple OSD journals simultaneously.
- **Partition Alignment:** A common problem with SSD performance is that people like to partition drives as a best practice, but they often overlook proper partition alignment with SSDs, which can cause SSDs to transfer data much more slowly. Ensure that SSD partitions are properly aligned.

While SSDs are cost prohibitive for object storage, OSDs may see a significant performance improvement by storing an OSD’s journal on an SSD and the OSD’s object data on a separate hard disk drive. The `osd journal` configuration setting defaults to `/var/lib/ceph/osd/$cluster-$id/journal`. You can mount this path to an SSD or to an SSD partition so that it is not merely a file on the same disk as the object data.

One way Ceph accelerates CephFS filesystem performance is to segregate the storage of CephFS metadata from the storage of the CephFS file contents. Ceph provides a default `metadata` pool for CephFS metadata. You will never have to create a pool for CephFS metadata, but you can create a CRUSH map hierarchy for your CephFS metadata pool that points only to a host’s SSD storage media. See Mapping Pools to Different Types of OSDs for details.

CONTROLLERS

Disk controllers also have a significant impact on write throughput. Carefully, consider your selection of disk controllers to ensure that they do not create a performance bottleneck.

Tip: The Ceph blog is often an excellent source of information on Ceph performance issues. See Ceph Write Throughput 1 and Ceph Write Throughput 2 for additional details.

ADDITIONAL CONSIDERATIONS

You may run multiple OSDs per host, but you should ensure that the sum of the total throughput of your OSD hard disks doesn’t exceed the network bandwidth required to service a client’s need to read or write data. You should also consider what percentage of the overall data the cluster stores on each host. If the percentage on a particular host is large and the host fails,

it can lead to problems such as exceeding the `full ratio`, which causes Ceph to halt operations as a safety precaution that prevents data loss.

When you run multiple OSDs per host, you also need to ensure that the kernel is up to date. See OS Recommendations for notes on `glibc` and `syncfs(2)` to ensure that your hardware performs as expected when running multiple OSDs per host.

NETWORKS

We recommend that each host have at least two 1Gbps network interface controllers (NICs). Since most commodity hard disk drives have a throughput of approximately 100MB/second, your NICs should be able to handle the traffic for the OSD disks on your host. We recommend a minimum of two NICs to account for a public (front-side) network and a cluster (back-side) network. A cluster network (preferably not connected to the internet) handles the additional load for data replication and helps stop denial of service attacks that prevent the cluster from achieving `active + clean` states for placement groups as OSDs replicate data across the cluster. Consider starting with a 10Gbps network in your racks. Replicating 1TB of data across a 1Gbps network takes 3 hours, and 3TBs (a typical drive configuration) takes 9 hours. By contrast, with a 10Gbps network, the replication times would be 20 minutes and 1 hour respectively. In a petabyte-scale cluster, failure of an OSD disk should be an expectation, not an exception. System administrators will appreciate PGs recovering from a `degraded` state to an `active + clean` state as rapidly as possible, with price / performance tradeoffs taken into consideration. Additionally, some deployment tools (e.g., Dell's Crowbar) deploy with five different networks, but employ VLANs to make hardware and network cabling more manageable. VLANs using 802.1q protocol require VLAN-capable NICs and Switches. The added hardware expense may be offset by the operational cost savings for network setup and maintenance. When using VLANs to handle VM traffic between between the cluster and compute stacks (e.g., OpenStack, CloudStack, etc.), it is also worth considering using 10G Ethernet. Top-of-rack routers for each network also need to be able to communicate with spine routers that have even faster throughput–e.g., 40Gbps to 100Gbps.

Your server hardware should have a Baseboard Management Controller (BMC). Administration and deployment tools may also use BMCs extensively, so consider the cost/benefit tradeoff of an out-of-band network for administration. Hypervisor SSH access, VM image uploads, OS image installs, management sockets, etc. can impose significant loads on a network. Running three networks may seem like overkill, but each traffic path represents a potential capacity, throughput and/or performance bottleneck that you should carefully consider before deploying a large scale data cluster.

FAILURE DOMAINS

A failure domain is any failure that prevents access to one or more OSDs. That could be a stopped daemon on a host; a hard disk failure, an OS crash, a malfunctioning NIC, a failed power supply, a network outage, a power outage, and so forth. When planning out your hardware needs, you must balance the temptation to reduce costs by placing too many responsibilities into too few failure domains, and the added costs of isolating every potential failure domain.

MINIMUM HARDWARE RECOMMENDATIONS

Ceph can run on inexpensive commodity hardware. Small production clusters and development clusters can run successfully with modest hardware.

Process	Criteria	Minimum Recommended
`ceph-osd`	Processor	1x 64-bit AMD-64/i386 dual-core
	RAM	500 MB per daemon
	Volume Storage	1x Disk per daemon
	Network	2x 1GB Ethernet NICs
`ceph-mon`	Processor	1x 64-bit AMD-64/i386
	RAM	1 GB per daemon
	Disk Space	10 GB per daemon
	Network	2x 1GB Ethernet NICs
`ceph-mds`	Processor	1x 64-bit AMD-64/i386 quad-core
	RAM	1 GB minimum per daemon
	Disk Space	1 MB per daemon
	Network	2x 1GB Ethernet NICs

Tip: If you are running an OSD with a single disk, create a partition for your volume storage that is separate from the partition containing the OS. Generally, we recommend separate disks for the OS and the volume storage.

PRODUCTION CLUSTER EXAMPLE

Production clusters for petabyte scale data storage may also use commodity hardware, but should have considerably more memory, processing power and data storage to account for heavy traffic loads.

A recent (2012) Ceph cluster project is using two fairly robust hardware configurations for Ceph OSDs, and a lighter configuration for monitors.

Configuration	Criteria	Minimum Recommended
Dell PE R510	Processor	2x 64-bit quad-core Xeon CPUs
	RAM	16 GB
	Volume Storage	8x 2TB drives. 1 OS, 7 Storage
	Client Network	2x 1GB Ethernet NICs
	OSD Network	2x 1GB Ethernet NICs
	Mgmt. Network	2x 1GB Ethernet NICs
Dell PE R515	Processor	1x hex-core Opteron CPU
	RAM	16 GB
	Volume Storage	12x 3TB drives. Storage
	OS Storage	1x 500GB drive. Operating System.
	Client Network	2x 1GB Ethernet NICs
	OSD Network	2x 1GB Ethernet NICs
	Mgmt. Network	2x 1GB Ethernet NICs