Red Hat Ceph Storage 1.2.3 Hardware Guide

Hardware recommendations for Red Hat Ceph Storage v1.2.3.

Red Hat Customer Content Services

Red Hat Ceph Storage 1.2.3 Hardware Guide

Hardware recommendations for Red Hat Ceph Storage v1.2.3.

Legal Notice

Abstract

This document provides high level guidance on selecting hardware for use with Red Hat Ceph Storage.

Table of Contents

PREFACE 3

PART I. HOST RECOMMENDATIONS 4

CHAPTER 1. CPU 5

CHAPTER 2. RAM 6

CHAPTER 3. DATA STORAGE 7

3.1. IDENTICAL CONFIGURATIONS 7

3.2. JOURNALING 7

3.3. HARD DISK DRIVES 7

3.4. AVOID RAID 8

3.5. SOLID STATE DRIVES 8

3.6. CONTROLLERS 9

3.7. ADDITIONAL CONSIDERATIONS 9

PART II. NETWORKING RECOMMENDATIONS 10

CHAPTER 4. 10GBPS 11

CHAPTER 5. USE A CLUSTER/BACKSIDE NETWORK 12

CHAPTER 6. MULTI-RACK CONSIDERATIONS 13

CHAPTER 7. MULTIPLE DATACENTERS 14

CHAPTER 8. IPV6 15

CHAPTER 9. PORTS 16

CHAPTER 10. USE NTP 17

CHAPTER 11. IPMI 18

PART III. MINIMUM HARDWARE RECOMMENDATIONS 19

CHAPTER 12. PRODUCTION CLUSTERS 21

12.1. CALAMARI 21

12.2. MONITORS 21

12.3. OSDS 21

PREFACE

Ceph was designed to run on commodity hardware, which makes building and maintaining petabyte-to-exabyte 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.

PART I. HOST RECOMMENDATIONS

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).

How you select and configure a Ceph OSD host has a lot to do with how you intend to use the OSDs on it (e.g., for OpenStack volumes and images, for an S3 gateway, for a fast SSD pool or cache tier, etc.). See the Ceph’s Storage Stragegies Guide for details about defining storage strategies for your Ceph use case(s) and use these recommendations to help define your host requirements.

CHAPTER 1. CPU

Ceph OSDs run the storage cluster service, calculate data placement with CRUSH, replicate data, and maintain their own copy of the cluster map. Ceph OSDs that host erasure-coded pools will use more CPU than Ceph OSDs that host replicated pools. Therefore, OSDs should have a reasonable amount of processing power and should consider the storage strategy(ies) you intend to use. 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.

CHAPTER 2. RAM

Ceph 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., 500MB of RAM per daemon instance); however, during recovery they need significantly more RAM (e.g., ~1GB for 1TB of storage per daemon or more). Generally, more RAM is better (e.g., page caching).

CHAPTER 3. 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 requests for read and write operations from multiple daemons against a single drive can slow performance considerably.

Ceph can operate with heterogeneous systems. CRUSH supports weighting for different sized drives (e.g., 1TB, 3TB, etc), and primary affinity (the likeliness an OSD would be used as a primary) to address the performance issues introduced by dissimilar hardware in the same pool. However, using homogeneous configurations for the OSDs assigned to a pool is recommended.

3.1. IDENTICAL CONFIGURATIONS

We recommend creating pools and defining CRUSH hierarchies such that the OSD hardware within the pool is identical. That is:

- Same controller
- Same drive size
- Same RPMs
- Same seek times
- Same I/O
- Same network throughput
- Same journal configuration

Using the same hardware within a pool provides a consistent performance profile, simplifies provisioning and streamlines troubleshooting.

3.2. JOURNALING

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 journal and OSD performance in balance is really important!

3.3. 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. Also, the larger the storage drive capacity, the more memory per Ceph OSD Daemon you will need, especially during rebalancing, backfilling and recovery. Red Hat typically recommends a baseline of 16GB of RAM, with an additional 2GB of RAM per OSD.

Tip

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

Tip

Running an OSD and a monitor on a single disk—irrespective of partitions—is **NOT** a good idea.

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 Ceph 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 Ceph OSD Daemons 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. SSDs for operating system drives are preferred.

3.4. AVOID RAID

Ceph replicates or erasure codes objects. RAID is redundant and reduces available capacity, and therefore an unnecessary expense. A degraded RAID will have a negative impact on performance. If you have systems with RAID controllers, configure them for RAID 0 (JBOD).

3.5. 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 (e.g., journals or cache-tiers). 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 drive as the object data.

3.6. 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.

3.7. 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.

PART II. NETWORKING RECOMMENDATIONS

Carefully consider bandwidth requirements for your cluster network, be mindful of network link oversubscription, and segregate the intra-cluster traffic from the client-to-cluster traffic.

CHAPTER 4. 10GBPS

On smaller clusters, 1Gbps networks may be suitable for normal operating conditions, but not for heavy loads or failure recovery scenarios. In the case of a drive failure, 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. Remember that when an OSD fails, the cluster will recover by replicating the data it contained to other OSDs within the pool.

```
failed OSD(s)
-------------
 total OSDs
```

The failure of a larger domain such as a rack means that your cluster will utilize considerably more bandwidth. Administrators usually prefer that a cluster recovers as quickly as possible.

At a **minimum**, a single 10Gbps Ethernet link should be used for storage hardware. If your Ceph nodes have many drives each, add additional 10Gbps Ethernet links for connectivity and throughput.

CHAPTER 5. USE A CLUSTER/BACKSIDE NETWORK

Ceph supports a public (front-side) network and a cluster (back-side) network. The public network handles client traffic and communication with Ceph monitors. The cluster (back-side) network handles OSD heartbeats, replication, backfilling and recovery traffic. We recommend allocating bandwidth to the cluster (back-side) network such that it is a multiple of the front-side network using **`osd pool default size`** as the basis for your multiple. We also recommend running the public and cluster networks on separate NICs.

CHAPTER 6. MULTI-RACK CONSIDERATIONS

If you are building a cluster consisting of multiple racks (common for large clusters), consider utilizing as much network bandwidth between switches in a "fat tree" design for optimal performance. A typical 10Gbps Ethernet switch has 48 10Gbps ports and four 40Gbps ports. If you only use one one 40Gbps port for connectivity, you can only connect 4 servers at full speed (i.e., 10gbps x 4). Use your 40Gbps ports for maximum throughput. If you have unused 10G ports, you can aggregate them (with QSFP+ to 4x SFP+ cables) into more 40G ports to connect to other racks and to spine routers.

For network optimization, we recommend a jumbo frame for a better CPU/bandwidth ratio. We also recommend a non-blocking network switch back-plane.

CHAPTER 7. MULTIPLE DATACENTERS

You may deploy a Ceph cluster across geographic regions; however, this is NOT RECOMMENDED UNLESS you use a dedicated network connection between datacenters. Ceph prefers consistency and acknowledges writes synchronously. Using the internet (packet-switched with many hops) between geographically separate datacenters will introduce significant write latency.

You may specify multiple IP addresses and subnets for your public and cluster networks in your Ceph configuration file. For example:

```
public network {ip-address}/{netmask} [, {ip-address}/{netmask}]
cluster network {ip-address}/{netmask} [, {ip-address}/{netmask}]
```

Ensure that the IP addresses/subnets within the public network can reach each other, and the IP addresses/subnets within the cluster network can reach each other. We recommend keeping the cluster network separate from the public network and not connected to the internet to prevent DDOS attacks from crippling heartbeats, replication, backfilling and recovery.

CHAPTER 8. IPV6

You may use IPv6 addresses; however, you must enable daemons to bind to them first. For example, you may enable IPv6 in your Ceph configuration file:

```
ms bind ipv6 = true
```

CHAPTER 9. PORTS

Monitors use port 6789 by default. Ensure you have the port open for each monitor host. Each Ceph OSD Daemon on a Ceph Node may use up to three ports, beginning at port 6800:

1. One for talking to clients and monitors.
2. One for sending data to other OSDs (replication, backfill and recovery).
3. One for heartbeating.

You need to open at least three ports per OSD beginning at port 6800 on a Ceph node to ensure that the OSDs can peer. The port for talking to monitors and clients must be open on the public (front-side) network. The ports for sending data to other OSDs and heartbeating must be open on the cluster (back-side) network.

If you want to use a different port range than 6800:7100 for Ceph daemons, you must adjust the following settings in your Ceph configuration file:

```
ms bind port min = {min-port-num}
ms bind port max = {max-port-num}
```

Ceph monitors bind on port 6789 by default. If you want to use a different port number than 6789, you may specify the the IP address and port in your Ceph configuration. For example:

```
[mon.monname]
host = {hostname}
mon addr = {ip-address}:{port}
```

CHAPTER 10. USE NTP

We further recommend installing NTP on your Ceph nodes—especially Ceph monitor nodes. Without clock synchronization, clock drift may prevent monitors from agreeing on the state of the cluster, which means that clients lose access to data until the quorum is re-established and the monitors agree on the state of the cluster.

CHAPTER 11. IPMI

As a best practice, we also recommend a 1GbE copper interface for an IPMI network.

PART III. MINIMUM HARDWARE RECOMMENDATIONS

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

Process	Criteria	Minimum Recommended
`calamari`	Processor	1x 64-bit x86-64 quad-core
	RAM	4 GB minimum per instance
	Disk Space	10 GB per instance
	Network	2x 1GB Ethernet NICs
`ceph-osd`	Processor	1x 64-bit x86-64
	RAM	~1GB for 1TB of storage per daemon
	Volume Storage	1x storage drive per daemon
	Journal	1x SSD partition per daemon (optional)
	Network	2x 1GB Ethernet NICs
`ceph-mon`	Processor	1x 64-bit x86-64
	RAM	1 GB per daemon
	Disk Space	10 GB 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.

CHAPTER 12. PRODUCTION CLUSTERS

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.

12.1. CALAMARI

The administration server hardware requirements vary with the size of your cluster. A minimum recommended hardware configuration for a Calamari server includes at least 4GB of RAM, a dual core CPU on x86_64 architecture and enough network throughput to handle communication with Ceph hosts. The hardware requirements scale linearly with the number of Ceph servers, so if you intend to run a fairly large cluster, ensure that you have enough RAM, processing power and network throughput for your administration node.

12.2. MONITORS

While it is tempting to run monitors on a single node, you SHOULD NOT do this in production. Monitors are relatively lightweight, but their purpose is to ensure high availability. Monitors are also **`fsync`** intensive. Consider using a high performance SSD for monitor data. You SHOULD NOT commingle monitor daemons and OSD daemons in production systems.

It is also a good idea to ensure that monitors have back-up power. In the event that monitor power fails, and the monitor cluster cannot achieve a quorum, you will lose access to data until the monitors are back up and achieve a quorum.

Remember, monitors are light weight, but they are mission critical to ensure high availability.

12.3. OSDS

Ensure that your network interface, controllers and drive throughput don’t leave bottlenecks—e.g., fast drives, but networks too slow to accommodate them. SSDs are typically used for journals and fast pools. Where the use of SSDs is write intensive (e.g., journals), make sure you select high performance SSDs.

When using Ceph as a backend for OpenStack volumes and images, we recommend a host bus adapter with SAS drives (10-15k rpms) and enterprise-grade SSDs using the same controller for journals.