Tuning Object Storage Cluster Performance

Objectives

After completing this section, you should be able to protect OSD and cluster hardware resources from over-utilization by controlling scrubbing, deep scrubbing, backfill, and recovery processes to balance CPU, RAM, and I/O requirements.

Maintaining OSD Performance

Good client performance requires utilizing your OSDs within their physical limits. To maintain OSD performance, evaluate these tuning opportunities:

- Tune the BlueStore back end used by OSDs to store objects on physical devices.
- Adjust the schedule for automatic data scrubbing and deep scrubbing.
- Adjust the schedule of asynchronous snapshot trimming (deleting removed snapshots).
- Control how quickly backfill and recovery operations occur when OSDs fail or are added or replaced.

Storing Data on Ceph BlueStore

The default back-end object store for OSD daemons is BlueStore. The following list describes some of the main features of using BlueStore:

Direct management of storage devices
: BlueStore consumes raw block devices or partitions. This simplifies the management of storage devices because no other abstraction layers, such as local file systems, are required.

Efficient copy-on-write
: The Ceph Block Device and Ceph File System snapshots rely on a copy-on-write clone mechanism that is implemented efficiently in BlueStore. This results in efficient I/O for regular snapshots and for erasure-coded pools that rely on cloning to implement efficient two-phase commits.

No large double writes
: BlueStore first writes any new data to unallocated space on a block device, and then commits a `RocksDB` transaction that updates the object metadata to reference the new region of the disk.

Multidevice support
: BlueStore can use multiple block devices for storing the data, metadata, and write-ahead log.

In BlueStore, the raw partition is managed in chunks of the size specified by the `bluestore_min_alloc_size` variable. The `bluestore_min_alloc_size` is set by default to 4,096, which is equivalent to 4 KB, for HDDs and SSDs. If the data to write in the raw partition is smaller than the chunk size, then it is filled with zeroes. This can lead to a waste of the unused space if the chunk size is not properly sized for your workload, such as for writing many small objects.

Red Hat recommends setting the `bluestore_min_alloc_size` variable to match the smallest common write to avoid wasting unused space. For example, if your client writes 4 KB objects frequently, then configure the settings on OSD nodes such as `bluestore_min_alloc_size`