you can maximize the efficiency by using separate low latency SSDs or NVMe devices for the block database and WAL. Multiple block databases and WALs can share the same SSD or NVMe device, reducing the cost of the storage infrastructure.

Consider the impact of the following SSD specifications against the expected workload:

- Mean Time Between Failures (MTBF) for the number of supported writes
- IOPS capabilities
- Data transfer rate
- Bus/SSD couple capabilities

Warning

When an SSD or NVMe device that hosts journals fails, every OSD using it to host its journal also becomes unavailable. Consider this when deciding how many block databases or WALs to place on the same storage device.

Recommendations for Ceph RADOS Gateways

Workloads on a RADOS Gateway are often throughput-intensive. Audio and video materials being stored as objects can be large. However, the bucket index pool typically displays a more I/O-intensive workload pattern. Store the index pools on SSD devices.

The RADOS Gateway maintains one index per bucket. By default, Ceph stores this index in one RADOS object. When a bucket stores more than 100,000 objects, the index performance degrades because the single index object becomes a bottleneck.

Ceph can keep large indexes in multiple RADOS objects, or *shards*. Enable this feature by setting the `rgw_override_bucket_index_max_shards` parameter. The recommended value is the number of objects expected in a bucket divided by 100,000.

As the index grows, Ceph must regularly reshard the bucket. Red Hat Ceph Storage provides a bucket index automatic resharding feature. The `rgw_dynamic_resharding` parameter, set to true by default, controls this feature.

Recommendations for CephFS

The metadata pool, which holds the directory structure and other indexes, can become a CephFS bottleneck. To minimize this limitation, use SSD devices for the metadata pool.

Each MDS maintains a cache in memory for different kinds of items, such as inodes. Ceph limits the size of this cache with the `mds_cache_memory_limit` parameter. Its default value, expressed in absolute bytes, is equal to 4 GB.

Placement Group Algebra

The total number of PGs in a cluster can impact overall performance due to unnecessary CPU and RAM activity on some OSD nodes. Red Hat recommended validating PG allocation for each pool before putting a cluster into production. Also consider specific testing of the backfill and recovery impact on client I/O requests.

There are two important values:

- The overall number of PGs in the cluster
- The number of PGs for a specific pool

Use this formula to estimate how many PGs should be available for a single, specific pool:

*Total Placement Groups = (OSDs * 100) / Number of replicas*