HARD DISK AND FILE SYSTEM RECOMMENDATIONS

HARD DRIVE PREP

Ceph aims for data safety, which means that when the Ceph Client receives notice that data was written to a storage drive, that data was actually written to the storage drive. For old kernels (<2.6.33), disable the write cache if the journal is on a raw drive. Newer kernels should work fine.

Use `hdparm` to disable write caching on the hard disk:

```
sudo hdparm -W 0 /dev/hda 0
```

In production environments, we recommend running a Ceph OSD Daemon with separate drives for the operating system and the data. If you run data and an operating system on a single disk, we recommend creating a separate partition for your data.

FILESYSTEMS

Ceph OSD Daemons rely heavily upon the stability and performance of the underlying filesystem.

Note: We currently recommend XFS for production deployments. We recommend `btrfs` for testing, development, and any non-critical deployments. We believe that `btrfs` has the correct feature set and roadmap to serve Ceph in the long-term, but XFS and `ext4` provide the necessary stability for today's deployments. `btrfs` development is proceeding rapidly: users should be comfortable installing the latest released upstream kernels and be able to track development activity for critical bug fixes.

Ceph OSD Daemons depend on the Extended Attributes (XATTRs) of the underlying file system for various forms of internal object state and metadata. The underlying filesystem must provide sufficient capacity for XATTRs. `btrfs` does not bound the total xattr metadata stored with a file. XFS has a relatively large limit (64 KB) that most deployments won't encounter, but the `ext4` is too small to be usable.

You should always add the following line to the `[osd]` section of your `ceph.conf` file for `ext4` filesystems; you can optionally use it for `btrfs` and XFS.:

```
filestore xattr use omap = true
```

FILESYSTEM BACKGROUND INFO

The XFS and `btrfs` file systems provide numerous advantages in highly scaled data storage environments when compared to `ext3` and `ext4`. Both XFS and `btrfs` are journaling file systems, which means that they are more robust when recovering from crashes, power outages, etc. These filesystems journal all of the changes they will make before performing writes.

XFS was developed for Silicon Graphics, and is a mature and stable filesystem. By contrast, `btrfs` is a relatively new file system that aims to address the long-standing wishes of system administrators working with large scale data storage environments. `btrfs` has some unique features and advantages compared to other Linux filesystems.

`btrfs` is a copy-on-write filesystem. It supports file creation timestamps and checksums that verify metadata integrity, so it can detect bad copies of data and fix them with the good copies. The copy-on-write capability means that `btrfs` can support snapshots that are writable. `btrfs` supports transparent compression and other features.

`btrfs` also incorporates multi-device management into the file system, which enables you to support heterogeneous disk storage infrastructure, data allocation policies. The community also aims to provide `fsck`, deduplication, and data encryption support in the future. This compelling list of features makes `btrfs` the ideal choice for Ceph clusters.