Ceph Distributed File System

Ceph is a distributed network file system designed to provide good performance, reliability, and scalability.

Basic features include:

 * POSIX semantics
 * Seamless scaling from 1 to many thousands of nodes
 * High availability and reliability. No single point of failure.
 * N-way replication of data across storage nodes
 * Fast recovery from node failures
 * Automatic rebalancing of data on node addition/removal
 * Easy deployment: most FS components are userspace daemons

Also,
 * Flexible snapshots (on any directory)
 * Recursive accounting (nested files, directories, bytes)

In contrast to cluster filesystems like GFS, OCFS2, and GPFS that rely on symmetric access by all clients to shared block devices, Ceph separates data and metadata management into independent server clusters, similar to Lustre. Unlike Lustre, however, metadata and storage nodes run entirely as user space daemons. Storage nodes utilize btrfs to store data objects, leveraging its advanced features (checksumming, metadata replication, etc.). File data is striped across storage nodes in large chunks to distribute workload and facilitate high throughputs. When storage nodes fail, data is re-replicated in a distributed fashion by the storage nodes themselves (with some minimal coordination from a cluster monitor), making the system extremely efficient and scalable.

Metadata servers effectively form a large, consistent, distributed in-memory cache above the file namespace that is extremely scalable, dynamically redistributes metadata in response to workload changes, and can tolerate arbitrary (well, non-Byzantine) node failures. The metadata server takes a somewhat unconventional approach to metadata storage to significantly improve performance for common workloads. In particular, inodes with only a single link are embedded in directories, allowing entire directories of dentries and inodes to be loaded into its cache with a single I/O operation. The contents of extremely large directories can be fragmented and managed by independent metadata servers, allowing scalable concurrent access.

The system offers automatic data rebalancing/migration when scaling from a small cluster of just a few nodes to many hundreds, without requiring an administrator carve the data set into static volumes or go through the tedious process of migrating data between servers. When the file system approaches full, new nodes can be easily added and things will "just work."

Ceph includes flexible snapshot mechanism that allows a user to create a snapshot on any subdirectory (and its nested contents) in the system. Snapshot creation and deletion are as simple as 'mkdir .snap/foo' and 'rmdir .snap/foo'.

Ceph also provides some recursive accounting on directories for nested files and bytes. That is, a ’getfattr -d foo’ on any directory in the system will reveal the total number of nested regular files and subdirectories, and a summation of all nested file sizes. This makes the identification of large disk space consumers relatively quick, as no ’du’ or similar recursive scan of the file system is required.

Mount Syntax
============

The basic mount syntax is:

```
 # mount -t ceph monip[:port][,monip2[:port]...]:/[subdir] mnt
```

You only need to specify a single monitor, as the client will get the full list when it connects. (However, if the monitor you specify happens to be down, the mount won’t succeed.) The port can be left off if the monitor is using the default. So if the monitor is at 1.2.3.4,

```
 # mount -t ceph 1.2.3.4:/ /mnt/ceph
```

is sufficient. If /sbin/mount.ceph is installed, a hostname can be used instead of an IP address.

Mount Options
=============

ip=A.B.C.D[:N]
: Specify the IP and/or port the client should bind to locally. There is normally not much reason to do this. If the IP is not specified, the client’s IP address is determined by looking at the address its connection to the monitor originates from.

wsize=X
: Specify the maximum write size in bytes. By default there is no maximum. Ceph will normally size writes based on the file stripe size.

rsize=X
: Specify the maximum readahead.

mount_timeout=X
: Specify the timeout value for mount (in seconds), in the case of a non-responsive Ceph file system. The default is 30 seconds.

rbytes
: When stat() is called on a directory, set st_size to ’rbytes’, the summation of file sizes over all files nested beneath that directory. This is the default.

norbytes

When stat() is called on a directory, set st_size to the number of entries in that directory.

nocrc

Disable CRC32C calculation for data writes. If set, the storage node must rely on TCP's error correction to detect data corruption in the data payload.

noasyncreaddir

Disable client's use its local cache to satisfy readdir requests. (This does not change correctness; the client uses cached metadata only when a lease or capability ensures it is valid.)

More Information

For more information on Ceph, see the home page at
http://ceph.newdream.net/

The Linux kernel client source tree is available at
git://ceph.newdream.net/git/ceph-client.git
git://git.kernel.org/pub/scm/linux/kernel/git/sage/ceph-client.git

and the source for the full system is at
git://ceph.newdream.net/git/ceph.git