Scaling Linux Storage to Petabytes

Sage A. Weil, Scott A. Brandt, and Carlos Maltzahn
University of California, Santa Cruz

We are developing a distributed POSIX-based filesystem called Ceph that is designed to provide excellent performance and reliability with seamlessly scaling from a handful to many thousands of storage nodes [1]. A Ceph file system consists of a potentially large number of storage nodes with a locally attached disk or RAID, a small number of metadata servers, and a small number of monitors that manage cluster configuration and state. The storage, metadata, and monitor components are each implemented as user-space processes for ease of administration and deployment.

We are in the process of implementing a Ceph client in the Linux kernel, and aim to have a working (if not feature-complete) version in early 2008. (Although a prototype client has already been implemented using FUSE, that approach is limited in terms of performance, consistency, and correctness.)

I am primarily interested in meeting and talking to other kernel developers at the workshop (although I can make a short presentation if there is interest). Some topics I am interested in discussing include:

- General expectations of any new network file system client in the kernel
- Dealing with inode/dentry cache coherency in clustered file systems.
- Memory management issues during writeback when near-OOM. (Ceph's IO and communications model is somewhat more involved than the usual client/server arrangement.)
- Issues preventing Ceph from using existing file systems for local object storage. Write barriers and direct io. Practicality of extending inotify to indicate when writes commit to disk. Status/future of libbtrfs, any special steps it takes to safely do its thing in user space.

[1] http://ceph.sf.net/