NFSv4 for Parallel File Systems

November 2004

Task T1

Deliverable D1 (September 2004)

Analyze NFSv4 server state and determine those state elements that require support from the underlying file system.

Status Complete. Summary: the three types of state (share locks, byte-range locks and delegations) each need interfaces into the underlying file system.

Deliverable D2 (February 2005)

CITI will implement the VFS extensions and test them with a freely available parallel file system such as GFS.

Status GFS is up and running on our ASCI cluster with an NFSv4 server exporting GFS on one node. Baseline performance tests have been run.

The Linux cluster community has proposed extensions to the existing byte-range lock file system call to enable cluster/parallel file systems to run concurrent `lockd`s. CITI has reviewed and augmented the extensions to ensure that they satisfy the requirements enumerated in **T1D1**.

Task T2

Deliverable D1 (November 2004)

Complete the implementation of reboot recovery on the Linux NFSv4 server, as specified in RFC 3530, namely storing ClientID state and associated lease information to stable storage and retrieving it on reboot.

Status A design for storing, reaping, and retrieving clientIDs has been proposed and reviewed. Following `statd`, we propose creating a file in a recovery directory(s) with the clientID as the file name. Initial coding has started. We anticipate a prototype by mid-December.

Deliverable D2 (February 2005)

Invent a mechanism that allows us to "hand off" the valid ClientIDs from one server to another.

Status All NFSv4 servers will store valid clientIDs in the shared file space. ClientID uniqueness across all servers will be enforced. Any server will accept state reclaims from any clientID present in the shared file space enabling the 'hand off' of a clientID from one server to the other.

Administrative tasks to migrate a client:

- Turn on (toggle) the grace period for all NFSv4 servers exporting the shared file system
- Expire the client on the current NFSv4 server (remove the in-memory client state)
- Notify the client of the new NFSv4 server via **NFS4ERR_MOVED** and fs_locations
- Client performs normal reboot recovery. The new server will accept **OPEN**, **LOCK**, and **DELEGATION** reclaims from the client because it can find the clientID in the shared stable recovery storage.

N.B. Note the dependency on `FS_LOCATIONS` implementation.

Deliverable D3 (February 2005)

Implement a bookkeeping procedure that associates client state with NFSv4 server (logical) IP addresses.

Status No progress.

Task T3

Deliverable D1 (November 2004: prototype; February 2005: complete)

Prototype a consistent file handle solution for cluster file systems and work with Linux kernel maintainers to promote a solution compatible with PolyServe's requirements.

Status no progress

Task T4

Deliverable D1 (February 2005)

Develop a tool to construct `FS_LOCATIONS` referrals to construct name spaces and manage client migration.

Status no progress

Deliverable D2 (February 2005)

Develop a tool to cause a client to migrate from one server to another.

Status no progress

Task T5

Deliverable D1 (September 2004)

Complete the server-side implementation of named attributes, as specified in RFC 3530.

Status An initial server implementation of named attributes has been completed. See September status report.

Task T6

Deliverable D1 (February 2005)

Construct and maintain a test bed as described in the proposal.

Status awaiting equipment.

Task T7

Deliverable D1 (November 2004: progress report; February 2005: final report)

Report on project progress and deliverable status.

Status this document.