Progress Report

NFSv4 extensions for performance and interoperability

Center for Information Technology Integration
October 19, 2007

Summary

This is a report on the status of CITI's progress in a project sponsored by EMC.

- The major goal of the project is to implement pNFS block layout capability in the Linux NFSv4.1 client; that work is progressing on schedule.
- CITI is implementing sessions in the Linux client and server. The portion of sessions required to support pNFS is substantially complete. The remainder is progressing on schedule.
- CITI is implementing directory delegation in the Linux client and server; that work is substantially complete.
- CITI is rebasing its block layout implementation to the latest Linux kernel and NFSv4.1 draft; that work is substantially complete.
- CITI is implementing pNFS features in the Python-based PyNFS server to aid prototyping and testing; that work is complete.
- CITI has installed and configured a Celerra system and is using that for its continuing development.

Task 1: pNFS block layout driver

☺ ***Task 1.1: I/O path***

This task is complete: the I/O path now uses the NFS page cache.

☹ ***Task 1.2: I/O path***

This task is incomplete: CITI has not yet eliminated the complex error handling, a vestige of the former I/O path that went directly to the block device cache.

☺ ***Task 1.3: GETDEVICELIST, GETDEVICEINFO, and LAYOUTGET update***

This task is complete: CITI succeeded in using the Linux kernel multi-device driver for complex topology management.

😐 ***Task 1.4: Client layout cache***

This task is progressing: CITI is working with Panasas engineer Benny Halevy to implement a common mechanism in the generic Linux pNFS client code for caching layout segments.

☹ ***Task 1.5: Client reboot recovery***

This task is incomplete: CITI has not yet extended the layout cache to respond to server reboot.

Task 2: PyNFS block layout server suite

☺ ***Task 2.1***

This task is complete: CITI has developed a suite of PyNFS NFSv4.1 pNFS block layout volume topology tests using OP_GETDEVICELIST and OP_GETDEVICEINFO that test the block layout client's ability to parse and manage complex topologies.

☺ ***Task 2.2***

This task is complete: CITI has developed a suite of PyNFS NFSv4.1 pNFS block layout volume tests using OP_LAYOUTGET, CB_LAYOUTRECALL, and OP_LAYOUTRETURN to exercise the layout cache management in the pNFS block layout client.

☺ ***Task 2.3***

This task is complete: CITI has developed a suite of PyNFS NFSv4.1 metadata server reboot recovery tests for the client block layout driver using OP_LAYRETURN with the reclaim flag set.

Task 3: Support latest draft

☺ CITI has implemented the following portions of the NFSv4.1 minor version draft:

- Support sessions operations EXCHANGE_ID, CREATE_SESSION, OP_SEQUENCE.
- Replace NFSv4.0 clientid and lock sequencing with sessionid and OP_SEQUENCE sequencing.
- Use OP_SEQUENCE sequencing (to prevent LAYOUTGET, LAYOUTRETURN races).
- Support the sessions callback infrastructure, and use it for CB_LAYOUTRECALL.
- Extend the pNFS client GETATTR processing to a 96-bit attribute bit mask and include all minor version pNFS attributes.

😐 CITI has rebased the block pNFS client implementation to the Linux 2.6.18.3 kernel, which is the latest kernel with the pNFS generic client.

😐 Task 3 is an ongoing effort, as the Linux kernel and the NFSv4.1 minor version are moving targets.

Task 4: Directory delegations

😐 This task is progressing.

CITI has implemented and tested directory delegations. To promote acceptance of directory delegations into the Linux kernel, CITI is identifying use cases that demonstrate the advantages of directory delegations.