Project Status

NFSv4 Extensions for Performance and Interoperability

Center for Information Technology Integration

This is a report on the status of CITI's EMC-funded pNFS development project as of November 21, 2008. Items marked in blue reflect change from the September 4, 2008 report.

Sessions in the generic Linux pNFS client

Task	Description	Status
S1	Session recovery.	This task is complete.
S2	Callback channel.	This task is complete.
S3	NFSv4.1 back channel security using machine credentials.	To provide for back channel security, we added support for machine credentials in the SETCLIENTID call. This makes it possible for the callback client to establish a secure channel to the corresponding principal on the callback server. Patches were committed to Linux 2.6.26-rc1. We are working on extending the RPC upcall mechanism so that the callback client can acquire appropriate credentials from gssd. Patches were posted to the linux-nfs mailing list and are under discussion.
S4	NFSv4.1 security using secret state verifiers.	We now have a working Python implementation to test against. Olga has begun preliminary work on EXCHANGE_ID to pass the required information to set up SSV contexts.

Other generic pNFS client issues

Task	Description	Status
C1	LAYOUTGET, LAYOUTRETURN, and CB_LAYOUTRECALL.	LAYOUTGET and LAYOUTRETURN are complete. We need to address a layering issue: the generic layer is unable to merge adjacent or overlapping layouts, so it sends more LAYOUTGET requests than it needs to. The block layer handles this under the covers, but it would be more efficient to merge them in the generic layer. We have a general framework and an untested draft implementation of CB_LAYOUTRECALL, with testing still to come.
C2	CB_RECALL_ANY, RECLAIM_COMPLETE, and CB_RECALLABLE_OBJ_AVAIL.	No progress to report. (So far, the NFSv4.1 development community is deferring work on these non-critical elements.)
C3	Integration of block layout requirements into generic client.	This task is under way and ongoing. The main pNFS branch now includes appropriate hooks for the block driver in the write path.
C4	Implement new NFSv4.1 draft 19–21 pNFS features and behavior.	Layout stateid is under active development in the NFSv4.1 development community, with Andy Adamson (NetApp) leading the way. Device notification is under active development in the pNFS development community, with Marc Eshel (IBM) leading the development activity. Draft rewrites have simplified this task considerably by eliminating the ADD operation. XDR formats have been worked out and we have an initial implementation of the generic client and server processing code.

Task	Description	Status
C5	Reboot recovery.	This task is nearly complete.

Block layout module

Task	Description	Status
B1	Rebase the implementation from block draft 3 to block draft 6.	We are at draft 9, which is in Last Call, so minimal further updates are expected, i.e., this task is complete.
B2	Extend the block layout implementation to support large server block sizes.	This task is complete.
B3	Block layout client implementation based on architectural review.	Thanks to Tang Haiying, we have functional user-space disk scanning code. Next steps are to extend the user space upcall handler to call `select()` instead of polling and to allow multiple records in an upcall.
B4	Support for complex volume topologies using the Linux device mapper (dm) needs to be reviewed to meet performance and quality requirements.	We have a working implementation that needs further testing. When we return to task B3, we will revisit this implementation.
B5	Extend the layout cache implementation to support at least two devices.	We have a working implementation that needs further testing. When we return to task B3, we will update this implementation.
B6	Extend the device mapper to support the asynchronous CB_NOTIFY_DEVICEID callback operation.	No progress to report. Block-specific device notification depends on generic device notification (Task C4). We will begin work on this task soon.
B7	The block layout client must implement a timed lease I/O fencing mechanism to insulate against network partition.	No progress to report

PyNFS

Task	Description	Status
P1	Update PyNFS client and server to support new protocol features in the latest drafts.	The PyNFS client and server now support the latest drafts (minorversion1 draft 26 and pnfs-block draft 9). PyNFS now supports SSV.
P2	Enhance the block server implementation to pass full Connectathon tests.	The PyNFS server passes all Connectathon NFSv4 and non-pNFS NFSv4.1 tests except for the large file test. We now have a prototype implementation of a "real" file system that supports read, write, and file creation.

Milestone summary

The following tasks were projected to be complete by the May 2008 Connectathon.

Task	Description	Status
S1	Session recovery	**Complete**
S2	Callback channel implementation	**Complete**
B1	Block layout draft 6	**Complete**
B2	Server block sizes greater then 4 KB	**Complete**
B3	Revisit block layout client implementation based on architectural review	**Under way**

The following tasks are projected to be complete by the Fall 2008 Bakeathon.

Task	Description	Status
S3	Back channel security using machine credentials	**Under way**
C1	LAYOUTGET, LAYOUTRETURN, and CB_LAYOUTRECALL	**Nearly complete**
C2	CB_RECALL_ANY, RECLAIM_COMPLETE, CB_RECALLABLE_OBJ_AVAIL	**No progress**
P1	PyNFS block client and server support latest drafts	**Complete**
P2	PyNFS block server passes full Connectathon tests, prototype file system.	**Nearly complete**

The following tasks are projected to be under way by the Fall 2008 Bakeathon.

Task	Description	Status
C3	Integration of block layout requirements into the generic client	**Under way**
C4	Draft 19–21 pNFS features and behavior. See Appendix for status.	**Under way**
B4	Complex volume topologies	**Testing**
B5	Copy-on-write	**Testing**

The remaining tasks are projected to be complete by the end of the project.

Task	Description	Status
S4	NFSv4.1 security using secret state verifiers	**Under way**
C5	Reboot recovery	**Nearly complete**
B6	CB_NOTIFY_DEVICEID	**No progress**
B7	Timed lease I/O fencing mechanism	**No progress**