Center for Information Technology Integration

CITI–ARC Joint Project Status Report

November 3, 2008

This memorandum reports on the status of a joint project between IBM Almaden Research Center and CITI, University of Michigan under Agreement No. A0550295. CITI led or participated in all of the work reported work below, in coordination with IBM developers. The principal mechanisms for coordination are a weekly conference call among developers at CITI, IBM, Network Appliance, Panasas, Red Hat, and EMC; structured meetings on IRC, attended by CITI and Red Hat developers; and the pNFS, NFSv4, and Linux-NFS mailing lists.

GFS2-based MDS

At the May 2008 Connectathon, CITI demonstrated a prototype DS/MDS. Although it lacked a number of features (I/O was not validated, stateids were not checked, it lacked a control channel, and it had only one server, obviating parallel access), the prototype passed Connectathon tests. By the end of the Connectathon (with a lot of help from Dean Hildebrand and Marc Eshel), I/O from two servers worked as well.

Since May, CITI has focused on designing and implementing the pNFS control channel. We work closely with David Teigland at Red Hat, who "owns" gfs_controld (see discussion below) and some of the other GFS2 user space components. We also coordinate with Red Hat developers Christine Caulfield (Leeds, UK), Fabio Di Nitto (Copenhagen, DK), and Steven Whitehouse (who "owns" the GFS2 kernel).

Selecting the development target involved the following considerations.

- Cluster1 is mostly in-kernel, it is hard to maintain and debug.
- Cluster2, the current stable release of Red Hat's cluster suite in May, recast a lot of functionality in OpenAIS daemons.[1,2] cluster2 has extra layers of indirection between the daemons and the rest of the OpenAIS framework, which means it has drawbacks like longer call-chains and more complicated development.
- Cluster3, slated for Fedora 10, simplifies development by eliminating the openAIS glue layer. Fedora 10 also uses the newest version of OpenAIS, which is built to operate on the Corosync Cluster Engine.[3] Cluster3 can interoperate with cluster2 daemons or can run in cluster3-only mode. Fedora 10 is scheduled for release on November 25, 2008.

Teigland strongly recommended that new development go into cluster3; based on these considerations, we elected to develop in the cluster3 environment and to run in cluster3-only mode.

Structure of GFS2

GFS2 nodes use the OpenAIS CPG (Closed Process Group) protocol, a multicast-based virtual synchrony protocol, to maintain synchronization. The CPG protocol guarantees that each group member gets an identically ordered stream of events. Each member acts on events in the order they are received. To enforce consistent behavior, members do not act on their own events when they are sent, but when they are received from the group.

GFS2 uses a number of CPGs: one for GFS control (join, leave, mount, unmount, failure), another for I/O fencing, another for distributed lock management, etc. GFS control messages are generated and processed by a user space daemon, gfs_controld. The kernel portion of GFS2 makes upcalls to gfs_controld by posting uevents to a netlink socket. gfs_controld communicates to the kernel by manipulating kernel variables with sysfs.

We simplified the intranode communication by replacing the netlink socket and sysfs variables with an rpc_pipefs pipe and a single control channel message type that we use for upcalls, calls into the kernel, and pNFS messages between nodes. This is the same user↔kernel mechanism used by IDMAPD and GSSD. Red Hat developers have asked us to submit patches once the rest of the code is fleshed out.

[1] OpenAIS is an open source implementation of the SA Forum Application Interface Specification based upon extended virtual synchrony. The project currently implements APIs for application failover, application defined checkpointing, application eventing, extended virtual synchrony, and cluster membership.

[2] The Service Availability Forum is a consortium of industry-leading communications and computing companies working together to develop and publish high availability and management software interface specifications. The SA Forum then promotes and facilitates specification adoption by the industry.

[3] The Corosync Cluster Engine provides a cluster plug-in engine for third party cluster service developers.

GFS2 support for layout
Frank Filz implemented the following operations prior to Connectathon; CITI has tested and extended them since then.

layout_type NFSD asks GFS2 for a list of supported layout types. GFS2 replies with the file layout type.

layout_get NFSD requests layout information from GFS2. GFS2 constructs a layout and passes it back to NFSD. CITI added bookkeeping code to keep track of layouts on a per-file, per-fsid, and (soon) per-device basis so that GFS2 can cause NFSD to recall layouts.

CITI implemented the remaining pNFS functionality in GFS2:

layout_return GFS2 updates the bookkeeping mention in layout_get.

layout_commit Following GPFS practice, GFS2 layouts don't require layout commits when committing through the MDS, so this is a null function.

cb_layout_recall Code currently exists to allow GFS2 to call NFSD for per-file recalls. Once this is tested, we will complete the "per-fsid" and "all" modes.

GFS2 support for devices
CITI implemented the following operations for pNFS device support:

get_device_list GFS2 constructs a list of all eligible DS and returns this to NFSD as a single device. This was implemented and tested prior to Connectathon.

get_device_iter Because GFS2 uses a single device, this function is trivial.

cb_device_notify This code is used by GFS2 to tell NFSD about device changes (e.g., failure). CITI's implementation is under way but incomplete.

GFS2 support for pNFS control
We added the following pNFS operations to the GFS2 control CPG:

get_state Before a clientid is allowed to use a stateid for I/O, the DS validates the stateid with the MDS using a get_state call on the GFS control CPG. The DS caches the validation. This code is implemented and tested.

cb_change_state When a file is closed and its stateid is no longer valid, the MDS multicasts a change_state message to revoke the cached stateid validation at the DS. This code is implemented and tested.

get_verifier A DS tracks server reboots with the get_verifier RPC. The MDS responds with its boot time, which is cached by the DS. This code is implemented and partially tested.

.

GFS2-specific operations on the MDS are implemented by extending existing cluster suite configuration tools, which use a cluster-wide, XML-based configuration file called cluster.conf. We expand cluster.conf to specify pNFS parameters, such as denoting the MDS, discovering DS, and setting DS nodeids.

Status
CITI keeps a GFS2/MDS git repository at git.linux-nfs.org with the current kernel ("linux-2.6.git") and user ("cluster.git") code. Benny Halevy rebases upstream approximately weekly; we are tracking him at 2.6.27.