Bug 5921

February 11, 2008

1 Requirements:

Rebooted Lustre clients must be efficiently removed from the lists of connected clients maintained by the OSTs and MDTs.

2 Functional specification:

All clients should ping all servers (already completed on HEAD). In LLNL'scase, because all servers are recoverable, all servers are already being pinged.

If an MDT or OST detects that it has not received any traffic on an exportfor some period of time (some % of the timeout value), the client is immediately evicted.

2.1 An extra use case:

As an added benefit of having clients immediately evicted, we can eliminate the 2-second timeout for initial AST reply. This was originally added to quickly evict large numbers of rebooted clients holding the same read lock.

3 Logic description:

We define the eviction threshold to be some percentage of the timeout value. Exports should be kept in a new, ordered list. Exports at the top of this list are the soonest to be evicted. We add a new timer which will fire when this top-most export needs to be evicted. This timer should probably be rounded up to the nearest second, to avoid needless storms of timer firings and resetting.

Any time an RPC arrives for an export, it is moved to the end of the list, and the timer is adjusted to fire when the now-topmost export needs to be evicted.

When the timer fires, we evict all clients whose exports have not received an RPC within the eviction threshold.

As a further refinement, we could consider evicting only if we are still receiving traffic from any client. This would prevent evictions in the case that the entire network has collapsed (ie, switch failure), or if all of the server threads are hung (in which case a reboot would allow successful recovery).

4 State management:

The timer will fire in IRQ context and must not sleep; the work must be immediately handed off to a worker thread to carry out the actual eviction. This is all very similar to the current lock-timer eviction mechanism to handle clients who do not send cancellations in time.

The last-heard-from value in the export, the new export list, and the new timer are all shared between RPC-handling threads, the thread that handles the timer IRQ, and the new worker thread.

- Disk format: no changes
- Configuration: no changes

5 Wire protocol:

No changes that affect protocol compatibility, however I propose that we take Alex's change and begin to ping every (timeout / 10) seconds, instead of every (timeout) seconds, if there is not other traffic on the export. This will promote more aggressive eviction of dead clients. Clients would also be given a longer grace period in which to respond to the initial AST RPC.

6 Key API changes:

We will likely add new APIs for adding to and reordering exports on this new list and timer. Existing APIs will be minimally affected.

7 Scalability and performance:

I don't see any serious concerns here. Maintaining an ordered list is very cheap in this case, because it starts ordered, and the most expensive thing we ever do is move something to the end of the list.

8 Recovery:

The likelihood of a successful recovery is dramatically improved by proactively removing records of dead clients, which are currently responsible for the illusion of "double failures" in many cases.

The likelihood of evicting a slightly-slow client is also reduced, by giving it more than 2 seconds to reply to ASTs.

Alternatives:

Instead of a timer and ordered list, we could scan periodically for exports which have not received traffic. We could ping clients "in reverse", and

use those pings to detect dead clients. Neither of these alternatives is particularly attractive, in my opinion.