Reference counting in pnfs

The are several inter-related caches. We have layouts which can reference multiple devices, each of which can reference multiple data servers. Each data server can be referenced by multiple devices. Each device can be referenced by multiple layouts. To keep all of this straight, we need to reference count.

struct pnfs_layout_hdr

The on-the-wire command LAYOUTGET corresponds to struct pnfs_layout_segment, usually referred to by the variable name lseg. Each nfs_inode may hold a pointer to a cache of these layout segments in nfsi->layout, of type struct pnfs_layout_hdr.

We reference the header for the inode pointing to it, across each outstanding RPC call that references it (LAYOUTGET, LAYOUTRETURN, LAYOUTCOMMIT), and for each lseg held within.

Each header is also (when non-empty) put on a list associated with struct nfs_client (cl_layouts). Being put on this list does not bump the reference count, as the layout is kept around by the lseg that keeps it in the list.

deviceid_cache

lsegs reference device ids, which are resolved per nfs_client and layout driver type. The device ids are held in a RCU cache (struct nfs4_deviceid_cache). The cache itself is referenced across each mount. The entries (struct nfs4_deviceid) themselves are held across the lifetime of each lseg referencing them.

RCU is used because the deviceid is basically a write once, read many data structure. The hlist size of 32 buckets needs better justification, but seems reasonable given that we can have multiple deviceid's per filesystem, and multiple filesystems per nfs_client.

The hash code is copied from the nfsd code base. A discussion of hashing and variations of this algorithm can be found here.

data server cache

file driver devices refer to data servers, which are kept in a module level cache. Its reference is held over the lifetime of the deviceid pointing to it.

lseg

lseg maintains an extra reference corresponding to the NFS_LSEG_VALID bit which holds it in the pnfs_layout_hdr's list. When the final lseg is removed from the pnfs_layout_hdr's list, the NFS_LAYOUT_DESTROYED bit is set, preventing any new lsegs from being added.

layout drivers

PNFS utilizes what is called layout drivers. The STD defines 4 basic layout types: "files", "objects", "blocks", and "flexfiles". For each of these types there is a layout-driver with a common function-vectors table which are called by the nfs-client pnfs-core to implement the different layout types.

Files-layout-driver code is in: fs/nfs/filelayout/.. directory Blocks-layout-driver code is in: fs/nfs/blocklayout/.. directory Flexfiles-layout-driver code is in: fs/nfs/flexfilelayout/.. directory

blocks-layout setup

TODO: Document the setup needs of the blocks layout driver