

---

Subject: [PATCH v3] NFSd: fix locking in nfsd\_forget\_delegations()  
Posted by Stanislav Kinsbursky on Tue, 03 Jul 2012 13:01:48 GMT  
[View Forum Message](#) <> [Reply to Message](#)

---

v3: rebased on 3.5-rc3 kernel

v2: dl\_recall\_lru list is used for delegations collect because it's modified both in unhash\_delegation() and nfsd\_break\_one\_deleg().

This patch adds recall\_lock hold to nfsd\_forget\_delegations() to protect nfsd\_process\_n\_delegations() call.

Also, looks like it would be better to collect delegations to some local on-stack list, and then unhash collected list. This split allows to simplify locking, because delegation traversing is protected by recall\_lock, when delegation unhash is protected by client\_mutex.

Signed-off-by: Stanislav Kinsbursky <[skinsbursky@parallels.com](mailto:skinsbursky@parallels.com)>

---

fs/nfsd/nfs4state.c | 21 ++++++-----  
1 files changed, 17 insertions(+), 4 deletions(-)

```
diff --git a/fs/nfsd/nfs4state.c b/fs/nfsd/nfs4state.c
index 59b9efc..33eccee 100644
--- a/fs/nfsd/nfs4state.c
+++ b/fs/nfsd/nfs4state.c
@@ -4579,7 +4579,7 @@ void nfsd_forget_openowners(u64 num)
    printk(KERN_INFO "NFSD: Forgot %d open owners", count);
}

-int nfsd_process_n_delegations(u64 num, void (*deleg_func)(struct nfs4_delegation *))
+int nfsd_process_n_delegations(u64 num, struct list_head *list)
{
    int i, count = 0;
    struct nfs4_file *fp, *fnext;
@@ -4588,7 +4588,7 @@ int nfsd_process_n_delegations(u64 num, void (*deleg_func)(struct
nfs4_delegatio
    for (i = 0; i < FILE_HASH_SIZE; i++) {
        list_for_each_entry_safe(fp, fnext, &file_hashtbl[i], fi_hash) {
            list_for_each_entry_safe(dp, dnnext, &fp->fi_delegations, dl_perfile) {
-            deleg_func(dp);
+            list_move(&dp->dl_recall_lru, list);
                if (++count == num)
                    return count;
            }
@@ -4601,9 +4601,16 @@ int nfsd_process_n_delegations(u64 num, void (*deleg_func)(struct
nfs4_delegatio
    void nfsd_forget_delegations(u64 num)
{
```

```

unsigned int count;
+ LIST_HEAD(victims);
+ struct nfs4_delegation *dp, *dnext;
+
+ spin_lock(&recall_lock);
+ count = nfsd_process_n_delegations(num, &victims);
+ spin_unlock(&recall_lock);

nfs4_lock_state();
- count = nfsd_process_n_delegations(num, unhash_delegation);
+ list_for_each_entry_safe(dp, dnext, &victims, dl_recall_lru)
+ unhash_delegation(dp);
nfs4_unlock_state();

printk(KERN_INFO "NFSD: Forgot %d delegations", count);
@@ -4612,10 +4619,16 @@ void nfsd_forget_delegations(u64 num)
void nfsd_recall_delegations(u64 num)
{
    unsigned int count;
+ LIST_HEAD(victims);
+ struct nfs4_delegation *dp, *dnext;

nfs4_lock_state();
spin_lock(&recall_lock);
- count = nfsd_process_n_delegations(num, nfsd_break_one_deleg);
+ count = nfsd_process_n_delegations(num, &victims);
+ list_for_each_entry_safe(dp, dnext, &victims, dl_recall_lru) {
+ list_del(&dp->dl_recall_lru);
+ nfsd_break_one_deleg(dp);
+ }
spin_unlock(&recall_lock);
nfs4_unlock_state();

```

---