
Subject: [PATCH] nfsv4 delegation: Convert to kthread API

Posted by [ebiederm](#) on Thu, 19 Apr 2007 06:55:55 GMT

[View Forum Message](#) <> [Reply to Message](#)

From: Eric W. Biederman <ebiederm@xmission.com> - unquoted

To start the nfsv4-delegreturn thread this patch uses
kthread_run instead of a combination of kernel_thread
and daemonize.

In addition allow_signal(SIGKILL) is removed from
the expire delegations thread.

Cc: Neil Brown <neilb@suse.de>

Cc: Trond Myklebust <trond.myklebust@fys.uio.no>

Signed-off-by: Eric W. Biederman <ebiederm@xmission.com>

fs/nfs/delegation.c | 11 ++++-----

1 files changed, 4 insertions(+), 7 deletions(-)

diff --git a/fs/nfs/delegation.c b/fs/nfs/delegation.c

index 841c99a..7b9b88c 100644

--- a/fs/nfs/delegation.c

+++ b/fs/nfs/delegation.c

@@ -232,7 +232,6 @@ int nfs_do_expire_all_delegations(void *ptr)
 struct nfs_delegation *delegation;
 struct inode *inode;

- allow_signal(SIGKILL);

restart:

spin_lock(&clp->cl_lock);

if (test_bit(NFS4CLNT_STATE_RECOVER, &clp->cl_state) != 0)

@@ -310,8 +309,6 @@ static int recall_thread(void *data)

struct nfs_inode *nfsi = NFS_I(inode);

struct nfs_delegation *delegation;

- daemonize("nfsv4-delegreturn");

-

nfs_msync_inode(inode);

down_read(&clp->cl_sem);

down_write(&nfsi->rwsem);

@@ -350,18 +347,18 @@ int nfs_async_inode_return_delegation(struct inode *inode, const
nfs4_stateid *s

.inode = inode,

.stateid = stateid,

};

- int status;

+ struct task_struct *task;

```
init_completion(&data.started);
__module_get(THIS_MODULE);
- status = kernel_thread(recall_thread, &data, CLONE_KERNEL);
- if (status < 0)
+ task = kthread_run(recall_thread, &data, "nfsv4-delegreturn");
+ if (IS_ERR(task))
    goto out_module_put;
    wait_for_completion(&data.started);
    return data.result;
out_module_put:
    module_put(THIS_MODULE);
- return status;
+ return PTR_ERR(task);
}

/*
--
1.5.0.g53756
```

Containers mailing list
Containers@lists.linux-foundation.org
<https://lists.linux-foundation.org/mailman/listinfo/containers>
