
Subject: [PATCH 3/4] NFS: make cb_ident_idr per net ns
Posted by [Stanislav Kinsbursky](#) on Mon, 23 Jan 2012 17:26:22 GMT
[View Forum Message](#) <> [Reply to Message](#)

This patch makes ID's infrastructure network namespace aware. This was done mainly because of nfs_client_lock, which is desired to be per network namespace, but protects NFS clients ID's.

NOTE: NFS client's net pointer have to be set prior to ID initialization, proper assignment was moved.

Signed-off-by: Stanislav Kinsbursky <skinsbursky@parallels.com>

```
---
fs/nfs/callback_xdr.c | 2 +-
fs/nfs/client.c      | 28 ++++++-----
fs/nfs/inode.c       | 2 +-
fs/nfs/internal.h    | 4 +--
fs/nfs/netns.h       | 3 +++
5 files changed, 25 insertions(+), 14 deletions(-)
```

```
diff --git a/fs/nfs/callback_xdr.c b/fs/nfs/callback_xdr.c
index d50b274..f2be3e1e 100644
--- a/fs/nfs/callback_xdr.c
+++ b/fs/nfs/callback_xdr.c
@@ -876,7 +876,7 @@ static __be32 nfs4_callback_compound(struct svc_rqst *rqstp, void
*argp, void *r
    return rpc_garbage_args;

    if (hdr_arg.minorversion == 0) {
- cps.clp = nfs4_find_client_ident(hdr_arg.cb_ident);
+ cps.clp = nfs4_find_client_ident(rqstp->rq_xprt->xpt_net, hdr_arg.cb_ident);
    if (!cps.clp || !check_gss_callback_principal(cps.clp, rqstp))
        return rpc_drop_reply;
    }
}
```

```
diff --git a/fs/nfs/client.c b/fs/nfs/client.c
index d58e838..f51b279 100644
--- a/fs/nfs/client.c
+++ b/fs/nfs/client.c
@@ -58,7 +58,6 @@
#define SPINLOCK(nfs_client_lock);
static DECLARE_WAIT_QUEUE_HEAD(nfs_client_active_wq);
#ifdef CONFIG_NFS_V4
-static DEFINE_IDR(cb_ident_idr); /* Protected by nfs_client_lock */

/*
 * Get a unique NFSv4.0 callback identifier which will be used
@@ -67,14 +66,15 @@ static DEFINE_IDR(cb_ident_idr); /* Protected by nfs_client_lock */
```

```

static int nfs_get_cb_ident_idr(struct nfs_client *clp, int minorversion)
{
    int ret = 0;
+ struct nfs_net *nn = net_generic(clp->net, nfs_net_id);

    if (clp->rpc_ops->version != 4 || minorversion != 0)
        return ret;
    retry:
- if (!idr_pre_get(&cb_ident_idr, GFP_KERNEL))
+ if (!idr_pre_get(&nn->cb_ident_idr, GFP_KERNEL))
    return -ENOMEM;
    spin_lock(&nfs_client_lock);
- ret = idr_get_new(&cb_ident_idr, clp, &clp->cl_cb_ident);
+ ret = idr_get_new(&nn->cb_ident_idr, clp, &clp->cl_cb_ident);
    spin_unlock(&nfs_client_lock);
    if (ret == -EAGAIN)
        goto retry;
@@ -173,6 +173,7 @@ static struct nfs_client *nfs_alloc_client(const struct nfs_client_initdata
*cl_
    clp->cl_rpcclient = ERR_PTR(-EINVAL);

    clp->cl_proto = cl_init->proto;
+ clp->net = cl_init->net;

#ifdef CONFIG_NFS_V4
    err = nfs_get_cb_ident_idr(clp, cl_init->minorversion);
@@ -191,7 +192,6 @@ static struct nfs_client *nfs_alloc_client(const struct nfs_client_initdata
*cl_
    if (!IS_ERR(cred))
        clp->cl_machine_cred = cred;
    nfs_fscache_get_client_cookie(clp);
- clp->net = cl_init->net;

    return clp;

@@ -236,16 +236,20 @@ static void nfs4_shutdown_client(struct nfs_client *clp)
}

/* idr_remove_all is not needed as all id's are removed by nfs_put_client */
-void nfs_cleanup_cb_ident_idr(void)
+void nfs_cleanup_cb_ident_idr(struct net *net)
{
- idr_destroy(&cb_ident_idr);
+ struct nfs_net *nn = net_generic(net, nfs_net_id);
+
+ idr_destroy(&nn->cb_ident_idr);
}

```

```

/* nfs_client_lock held */
static void nfs_cb_idr_remove_locked(struct nfs_client *clp)
{
+ struct nfs_net *nn = net_generic(clp->net, nfs_net_id);
+
  if (clp->cl_cb_ident)
- idr_remove(&cb_ident_idr, clp->cl_cb_ident);
+ idr_remove(&nn->cb_ident_idr, clp->cl_cb_ident);
}

static void pnfs_init_server(struct nfs_server *server)
@@ -263,7 +267,7 @@ static void nfs4_shutdown_client(struct nfs_client *clp)
{
}

-void nfs_cleanup_cb_ident_idr(void)
+void nfs_cleanup_cb_ident_idr(struct net *net)
{
}

@@ -1203,12 +1207,13 @@ error:
 * Find a client by callback identifier
 */
struct nfs_client *
-nfs4_find_client_ident(int cb_ident)
+nfs4_find_client_ident(struct net *net, int cb_ident)
{
  struct nfs_client *clp;
+ struct nfs_net *nn = net_generic(net, nfs_net_id);

  spin_lock(&nfs_client_lock);
- clp = idr_find(&cb_ident_idr, cb_ident);
+ clp = idr_find(&nn->cb_ident_idr, cb_ident);
  if (clp)
    atomic_inc(&clp->cl_count);
  spin_unlock(&nfs_client_lock);
@@ -1765,6 +1770,9 @@ void nfs_clients_init(struct net *net)

  INIT_LIST_HEAD(&nn->nfs_client_list);
  INIT_LIST_HEAD(&nn->nfs_volume_list);
+#ifdef CONFIG_NFS_V4
+ idr_init(&nn->cb_ident_idr);
+#endif
}

#ifdef CONFIG_PROC_FS
diff --git a/fs/nfs/inode.c b/fs/nfs/inode.c
index e8f81e5..d245064 100644

```

```

--- a/fs/nfs/inode.c
+++ b/fs/nfs/inode.c
@@ -1565,6 +1565,7 @@ static int nfs_net_init(struct net *net)
static void nfs_net_exit(struct net *net)
{
    nfs_dns_resolver_cache_destroy(net);
+ nfs_cleanup_cb_ident_idr(net);
}

static struct pernet_operations nfs_net_ops = {
@@ -1674,7 +1675,6 @@ static void __exit exit_nfs_fs(void)
#ifdef CONFIG_PROC_FS
    rpc_proc_unregister(&init_net, "nfs");
#endif
- nfs_cleanup_cb_ident_idr();
  unregister_nfs_fs();
  nfs_fs_proc_exit();
  nfsiod_stop();
diff --git a/fs/nfs/internal.h b/fs/nfs/internal.h
index a9ae806..958fff2 100644
--- a/fs/nfs/internal.h
+++ b/fs/nfs/internal.h
@@ -148,9 +148,9 @@ extern void nfs_umount(const struct nfs_mount_request *info);
extern const struct rpc_program nfs_program;
extern void nfs_clients_init(struct net *net);

-extern void nfs_cleanup_cb_ident_idr(void);
+extern void nfs_cleanup_cb_ident_idr(struct net *);
extern void nfs_put_client(struct nfs_client *);
-extern struct nfs_client *nfs4_find_client_ident(int);
+extern struct nfs_client *nfs4_find_client_ident(struct net *, int);
extern struct nfs_client *
nfs4_find_client_sessionid(const struct sockaddr *, struct nfs4_sessionid *);
extern struct nfs_server *nfs_create_server(
diff --git a/fs/nfs/netns.h b/fs/nfs/netns.h
index 0fbd4e0..547cc95 100644
--- a/fs/nfs/netns.h
+++ b/fs/nfs/netns.h
@@ -9,6 +9,9 @@ struct nfs_net {
    struct rpc_pipe *bl_device_pipe;
    struct list_head nfs_client_list;
    struct list_head nfs_volume_list;
+#ifdef CONFIG_NFS_V4
+ struct idr cb_ident_idr; /* Protected by nfs_client_lock */
+#endif
};

extern int nfs_net_id;

```
