Subject: [PATCH 3/4] nfsd: make name-to-id cache allocated per network namespace context
Posted by Stanislav Kinsbursky on Wed, 11 Apr 2012 13:32:58 GMT
View Forum Message <> Reply to Message

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

```
---
 fs/nfsd/netns.h     |    2 +-
 fs/nfsd/nfs4idmap.c |   24 ++++++++++++++++++---------
 2 files changed, 16 insertions(+), 10 deletions(-)

diff --git a/fs/nfsd/netns.h b/fs/nfsd/netns.h
index 948a718..3936563 100644
--- a/fs/nfsd/netns.h
+++ b/fs/nfsd/netns.h
@@ -33,7 +33,7 @@ struct nfsd_net {
  struct cache_detail *svc_export_cache;

  struct cache_detail *idtoname_cache;
-
+ struct cache_detail *nametoid_cache;
 };

 extern int nfsd_net_id;
diff --git a/fs/nfsd/nfs4idmap.c b/fs/nfsd/nfs4idmap.c
index b285a69..286a7f8 100644
--- a/fs/nfsd/nfs4idmap.c
+++ b/fs/nfsd/nfs4idmap.c
@@ -301,8 +301,6 @@ idtoname_update(struct cache_detail *cd, struct ent *new, struct ent *old)
  * Name -> ID cache
  */

-static struct cache_head *nametoid_table[ENT_HASHMAX];
-
 static inline int
 nametoid_hash(struct ent *ent)
 {
@@ -362,10 +360,9 @@ static struct ent *nametoid_update(struct cache_detail *, struct ent *,
        struct ent *);
 static int        nametoid_parse(struct cache_detail *, char *, int);

-static struct cache_detail nametoid_cache = {
+static struct cache_detail nametoid_cache_template = {
  .owner  = THIS_MODULE,
  .hash_size = ENT_HASHMAX,
- .hash_table = nametoid_table,
  .name  = "nfs4.nametoid",
```

```
  .cache_put = ent_put,
  .cache_upcall = nametoid_upcall,
@@ -479,11 +476,18 @@ nfsd_idmap_init(struct net *net)
  rv = cache_register_net(nn->idtoname_cache, net);
  if (rv)
   goto destroy_idtoname_cache;
- rv = cache_register_net(&nametoid_cache, net);
- if (rv)
+ nn->nametoid_cache = cache_create_net(&nametoid_cache_template, net);
+ if (IS_ERR(nn->nametoid_cache)) {
+  rv = PTR_ERR(nn->idtoname_cache);
   goto unregister_idtoname_cache;
+ }
+ rv = cache_register_net(nn->nametoid_cache, net);
+ if (rv)
+  goto destroy_nametoid_cache;
  return 0;

+destroy_nametoid_cache:
+ cache_destroy_net(nn->nametoid_cache, net);
 unregister_idtoname_cache:
  cache_unregister_net(nn->idtoname_cache, net);
 destroy_idtoname_cache:
@@ -497,8 +501,9 @@ nfsd_idmap_shutdown(struct net *net)
  struct nfsd_net *nn = net_generic(net, nfsd_net_id);

  cache_unregister_net(nn->idtoname_cache, net);
- cache_unregister_net(&nametoid_cache, net);
+ cache_unregister_net(nn->nametoid_cache, net);
  cache_destroy_net(nn->idtoname_cache, net);
+ cache_destroy_net(nn->nametoid_cache, net);
 }

 static int
@@ -541,19 +546,20 @@ idmap_name_to_id(struct svc_rqst *rqstp, int type, const char *name,
u32 namelen
   .type = type,
  };
  int ret;
+ struct nfsd_net *nn = net_generic(rqstp->rq_xprt->xpt_net, nfsd_net_id);

  if (namelen + 1 > sizeof(key.name))
   return nfserr_badowner;
  memcpy(key.name, name, namelen);
  key.name[namelen] = '\0';
  strlcpy(key.authname, rqst_authname(rqstp), sizeof(key.authname));
- ret = idmap_lookup(rqstp, nametoid_lookup, &key, &nametoid_cache, &item);
+ ret = idmap_lookup(rqstp, nametoid_lookup, &key, nn->nametoid_cache, &item);
```

```
  if (ret == -ENOENT)
   return nfserr_badowner;
  if (ret)
   return nfserrno(ret);
  *id = item->id;
- cache_put(&item->h, &nametoid_cache);
+ cache_put(&item->h, nn->nametoid_cache);
  return 0;
 }
```