
Subject: [PATCH 3/4] nfsd: make expkey cache allocated per network namespace context

Posted by [Stanislav Kinsbursky](#) on Wed, 11 Apr 2012 11:13:28 GMT

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

This patch also changes svcauth_unix_purge() function: added network namespace as a parameter and thus loop over all networks was replaced by only one call for ip map cache purge.

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

```
---
fs/nfsd/export.c      | 28 ++++++-----
fs/nfsd/netns.h       |  1 +
fs/nfsd/nfsctl.c      |  3 +-
include/linux/nfsd/export.h |  2 --
include/linux/sunrpc/svcauth.h |  2 +-
net/sunrpc/svcauth_unix.c | 13 ++++-----
6 files changed, 25 insertions(+), 24 deletions(-)
```

```
diff --git a/fs/nfsd/export.c b/fs/nfsd/export.c
```

```
index 84d020f..dcb52b8 100644
```

```
--- a/fs/nfsd/export.c
```

```
+++ b/fs/nfsd/export.c
```

```
@@ -40,7 +40,6 @@ typedef struct svc_export svc_export;
```

```
#define EXPKEY_HASHBITS 8
```

```
#define EXPKEY_HASHMAX (1 << EXPKEY_HASHBITS)
```

```
#define EXPKEY_HASHMASK (EXPKEY_HASHMAX -1)
```

```
-static struct cache_head *expkey_table[EXPKEY_HASHMAX];
```

```
static void expkey_put(struct kref *ref)
```

```
{
```

```
@@ -241,10 +240,9 @@ static struct cache_head *expkey_alloc(void)
```

```
return NULL;
```

```
}
```

```
-static struct cache_detail svc_expkey_cache = {
```

```
+static struct cache_detail svc_expkey_cache_template = {
```

```
.owner = THIS_MODULE,
```

```
.hash_size = EXPKEY_HASHMAX,
```

```
- .hash_table = expkey_table,
```

```
.name = "nfsd.fh",
```

```
.cache_put = expkey_put,
```

```
.cache_upcall = expkey_upcall,
```

```
@@ -883,12 +881,13 @@ static struct svc_export *exp_find(struct cache_detail *cd,
```

```
u32 *fsidv, struct cache_req *reqp)
```

```
{
```

```
struct svc_export *exp;
```

```

- struct svc_expkey *ek = exp_find_key(&svc_expkey_cache, clp, fsid_type, fsidv, reqp);
+ struct nfsd_net *nn = net_generic(cd->net, nfsd_net_id);
+ struct svc_expkey *ek = exp_find_key(nn->svc_expkey_cache, clp, fsid_type, fsidv, reqp);
  if (IS_ERR(ek))
    return ERR_CAST(ek);

  exp = exp_get_by_name(cd, clp, &ek->ek_path, reqp);
- cache_put(&ek->h, &svc_expkey_cache);
+ cache_put(&ek->h, nn->svc_expkey_cache);

  if (IS_ERR(exp))
    return ERR_CAST(exp);
@@ -1232,7 +1231,6 @@ const struct seq_operations nfs_exports_op = {
  .show = e_show,
};

-
/*
 * Initialize the exports module.
 */
@@ -1251,11 +1249,18 @@ nfsd_export_init(struct net *net)
  if (rv)
    goto destroy_export_cache;

- rv = cache_register_net(&svc_expkey_cache, net);
- if (rv)
+ nn->svc_expkey_cache = cache_create_net(&svc_expkey_cache_template, net);
+ if (IS_ERR(nn->svc_expkey_cache)) {
+   rv = PTR_ERR(nn->svc_expkey_cache);
+   goto unregister_export_cache;
+ }
+ rv = cache_register_net(nn->svc_expkey_cache, net);
+ if (rv)
+   goto destroy_expkey_cache;
  return 0;

+destroy_expkey_cache:
+ cache_destroy_net(nn->svc_expkey_cache, net);
unregister_export_cache:
  cache_unregister_net(nn->svc_export_cache, net);
destroy_export_cache:
@@ -1271,7 +1276,7 @@ nfsd_export_flush(struct net *net)
{
  struct nfsd_net *nn = net_generic(net, nfsd_net_id);

- cache_purge(&svc_expkey_cache);
+ cache_purge(nn->svc_expkey_cache);
  cache_purge(nn->svc_export_cache);

```

```

}

@@ -1285,10 +1290,11 @@ nfsd_export_shutdown(struct net *net)

    dprintk("nfsd: shutting down export module (net: %p).\n", net);

- cache_unregister_net(&svc_expkey_cache, net);
+ cache_unregister_net(nn->svc_expkey_cache, net);
  cache_unregister_net(nn->svc_export_cache, net);
+ cache_destroy_net(nn->svc_expkey_cache, net);
  cache_destroy_net(nn->svc_export_cache, net);
- svcauth_unix_purge();
+ svcauth_unix_purge(net);

    dprintk("nfsd: export shutdown complete (net: %p).\n", net);
}
diff --git a/fs/nfsd/netns.h b/fs/nfsd/netns.h
index c1c6242..9794c6c 100644
--- a/fs/nfsd/netns.h
+++ b/fs/nfsd/netns.h
@@ -29,6 +29,7 @@ struct cld_net;
struct nfsd_net {
    struct cld_net *cld_net;

+ struct cache_detail *svc_expkey_cache;
  struct cache_detail *svc_export_cache;
};

diff --git a/fs/nfsd/nfsctl.c b/fs/nfsd/nfsctl.c
index ddb9f87..b144177 100644
--- a/fs/nfsd/nfsctl.c
+++ b/fs/nfsd/nfsctl.c
@@ -129,13 +129,14 @@ static int exports_open(struct inode *inode, struct file *file)
{
    int err;
    struct seq_file *seq;
+ struct nfsd_net *nn = net_generic(&init_net, nfsd_net_id);

    err = seq_open(file, &nfs_exports_op);
    if (err)
        return err;

    seq = file->private_data;
- seq->private = &svc_export_cache;
+ seq->private = nn->svc_export_cache;
    return 0;
}

```

```

diff --git a/include/linux/nfsd/export.h b/include/linux/nfsd/export.h
index 565c212..e33f747 100644
--- a/include/linux/nfsd/export.h
+++ b/include/linux/nfsd/export.h
@@ -143,8 +143,6 @@ int exp_rootfh(struct net *, struct auth_domain *,
__be32 exp_pseudoroot(struct svc_rqst *, struct svc_fh *);
__be32 nfserno(int errno);

-extern struct cache_detail svc_export_cache;
-
static inline void exp_put(struct svc_export *exp)
{
    cache_put(&exp->h, exp->cd);
diff --git a/include/linux/sunrpc/svcauth.h b/include/linux/sunrpc/svcauth.h
index 548790e..153de45 100644
--- a/include/linux/sunrpc/svcauth.h
+++ b/include/linux/sunrpc/svcauth.h
@@ -131,7 +131,7 @@ extern struct auth_domain *auth_domain_lookup(char *name, struct
auth_domain *ne
extern struct auth_domain *auth_domain_find(char *name);
extern struct auth_domain *auth_unix_lookup(struct net *net, struct in6_addr *addr);
extern int auth_unix_forget_old(struct auth_domain *dom);
-extern void svcauth_unix_purge(void);
+extern void svcauth_unix_purge(struct net *net);
extern void svcauth_unix_info_release(struct svc_xprt *xprt);
extern int svcauth_unix_set_client(struct svc_rqst *rqstp);

diff --git a/net/sunrpc/svcauth_unix.c b/net/sunrpc/svcauth_unix.c
index 521d8f7..9c3b9f0 100644
--- a/net/sunrpc/svcauth_unix.c
+++ b/net/sunrpc/svcauth_unix.c
@@ -346,17 +346,12 @@ static inline int ip_map_update(struct net *net, struct ip_map *ipm,
return __ip_map_update(sn->ip_map_cache, ipm, udom, expiry);
}

-
-void svcauth_unix_purge(void)
+void svcauth_unix_purge(struct net *net)
{
- struct net *net;
-
- for_each_net(net) {
- struct sunrpc_net *sn;
+ struct sunrpc_net *sn;

- sn = net_generic(net, sunrpc_net_id);
- cache_purge(sn->ip_map_cache);
- }

```

```
+ sn = net_generic(net, sunrpc_net_id);  
+ cache_purge(sn->ip_map_cache);  
}  
EXPORT_SYMBOL_GPL(svcauth_unix_purge);
```
