
Subject: [PATCH 7/8] nfsd: pass pointer to export cache down to stack wherever possible.

Posted by [Stanislav Kinsbursky](#) on Wed, 28 Mar 2012 15:09:42 GMT

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

This cache will be per-net soon. And it's easier to get the pointer to desired per-net instance only once and then pass it down instead of discovering it in every place were required.

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

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

```
---
fs/nfsd/export.c | 37 ++++++-----
1 files changed, 22 insertions(+), 15 deletions(-)

diff --git a/fs/nfsd/export.c b/fs/nfsd/export.c
index 9fe7156..c237b0a 100644
--- a/fs/nfsd/export.c
+++ b/fs/nfsd/export.c
@@ -785,8 +785,8 @@ exp_find_key(svc_client *clp, int fsid_type, u32 *fsidv, struct cache_req
 *reqp)
 }

-static svc_export *exp_get_by_name(svc_client *clp, const struct path *path,
-    struct cache_req *reqp)
+static svc_export *exp_get_by_name(struct cache_detail *cd, svc_client *clp,
+    const struct path *path, struct cache_req *reqp)
 {
     struct svc_export *exp, key;
     int err;
@@ -796,12 +796,12 @@ static svc_export *exp_get_by_name(svc_client *clp, const struct path
 *path,
     key.ex_client = clp;
     key.ex_path = *path;
-    key.cd = &svc_export_cache;
+    key.cd = cd;

     exp = svc_export_lookup(&key);
     if (exp == NULL)
         return ERR_PTR(-ENOMEM);
-    err = cache_check(&svc_export_cache, &exp->h, reqp);
+    err = cache_check(cd, &exp->h, reqp);
     if (err)
         return ERR_PTR(err);
```

```

    return exp;
@@ -810,16 +810,17 @@ static svc_export *exp_get_by_name(svc_client *clp, const struct path
*path,
/*
 * Find the export entry for a given dentry.
 */
-static struct svc_export *exp_parent(svc_client *clp, struct path *path)
+static struct svc_export *exp_parent(struct cache_detail *cd, svc_client *clp,
+    struct path *path)
{
    struct dentry *saved = dget(path->dentry);
- svc_export *exp = exp_get_by_name(clp, path, NULL);
+ svc_export *exp = exp_get_by_name(cd, clp, path, NULL);

    while (PTR_ERR(exp) == -ENOENT && !IS_ROOT(path->dentry)) {
        struct dentry *parent = dget_parent(path->dentry);
        dput(path->dentry);
        path->dentry = parent;
- exp = exp_get_by_name(clp, path, NULL);
+ exp = exp_get_by_name(cd, clp, path, NULL);
    }
    dput(path->dentry);
    path->dentry = saved;
@@ -834,13 +835,15 @@ static struct svc_export *exp_parent(svc_client *clp, struct path *path)
 * since its harder to fool a kernel module than a user space program.
 */
int
-exp_rootfh(svc_client *clp, char *name, struct knfsd_fh *f, int maxsize)
+exp_rootfh(svc_client *clp, char *name,
+    struct knfsd_fh *f, int maxsize)
{
    struct svc_export *exp;
    struct path path;
    struct inode *inode;
    struct svc_fh fh;
    int err;
+ struct cache_detail *cd = &svc_export_cache;

    err = -EPERM;
    /* NB: we probably ought to check that it's NUL-terminated */
@@ -853,7 +856,7 @@ exp_rootfh(svc_client *clp, char *name, struct knfsd_fh *f, int maxsize)
    dprintk("nfds: exp_rootfh(%s [%p] %s:%s/%ld)\n",
        name, path.dentry, clp->name,
        inode->i_sb->s_id, inode->i_ino);
- exp = exp_parent(clp, &path);
+ exp = exp_parent(cd, clp, &path);
    if (IS_ERR(exp)) {
        err = PTR_ERR(exp);

```

```

    goto out;
@@ -875,7 +878,8 @@ out:
    return err;
}

-static struct svc_export *exp_find(struct auth_domain *clp, int fsid_type,
+static struct svc_export *exp_find(struct cache_detail *cd,
+    struct auth_domain *clp, int fsid_type,
+    u32 *fsidv, struct cache_req *reqp)
{
    struct svc_export *exp;
@@ -883,7 +887,7 @@ static struct svc_export *exp_find(struct auth_domain *clp, int fsid_type,
    if (IS_ERR(ek))
        return ERR_CAST(ek);

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

    if (IS_ERR(exp))
@@ -926,12 +930,13 @@ struct svc_export *
    rqst_exp_get_by_name(struct svc_rqst *rqstp, struct path *path)
{
    struct svc_export *gssexp, *exp = ERR_PTR(-ENOENT);
+ struct cache_detail *cd = &svc_export_cache;

    if (rqstp->rq_client == NULL)
        goto gss;

    /* First try the auth_unix client: */
    - exp = exp_get_by_name(rqstp->rq_client, path, &rqstp->rq_chandle);
+ exp = exp_get_by_name(cd, rqstp->rq_client, path, &rqstp->rq_chandle);
    if (PTR_ERR(exp) == -ENOENT)
        goto gss;
    if (IS_ERR(exp))
@@ -943,7 +948,7 @@ gss:
    /* Otherwise, try falling back on gss client */
    if (rqstp->rq_gssclient == NULL)
        return exp;
    - gssexp = exp_get_by_name(rqstp->rq_gssclient, path, &rqstp->rq_chandle);
+ gssexp = exp_get_by_name(cd, rqstp->rq_gssclient, path, &rqstp->rq_chandle);
    if (PTR_ERR(gssexp) == -ENOENT)
        return exp;
    if (!IS_ERR(exp))
@@ -955,12 +960,13 @@ struct svc_export *
    rqst_exp_find(struct svc_rqst *rqstp, int fsid_type, u32 *fsidv)
{
    struct svc_export *gssexp, *exp = ERR_PTR(-ENOENT);

```

```

+ struct cache_detail *cd = &svc_export_cache;

if (rqstp->rq_client == NULL)
    goto gss;

/* First try the auth_unix client: */
- exp = exp_find(rqstp->rq_client, fsid_type, fsidv, &rqstp->rq_chandle);
+ exp = exp_find(cd, rqstp->rq_client, fsid_type,
+     fsidv, &rqstp->rq_chandle);
if (PTR_ERR(exp) == -ENOENT)
    goto gss;
if (IS_ERR(exp))
@@ -972,7 +979,7 @@ gss:
/* Otherwise, try falling back on gss client */
if (rqstp->rq_gssclient == NULL)
    return exp;
- gssexp = exp_find(rqstp->rq_gssclient, fsid_type, fsidv,
+ gssexp = exp_find(cd, rqstp->rq_gssclient, fsid_type, fsidv,
    &rqstp->rq_chandle);
if (PTR_ERR(gssexp) == -ENOENT)
    return exp;

```
