
Subject: [PATCH 1/6] SUNRPC: split cache creation and PipeFS registration
Posted by [Stanislav Kinsbursky](#) on Fri, 25 Nov 2011 13:16:15 GMT
[View Forum Message](#) <> [Reply to Message](#)

This precursor patch splits SUNRPC cache creation and PipeFS registration.
It's required for latter split of NFS DNS resolver cache creation per network namespace context and PipeFS registration/unregistration on MOUNT/UMOUNT events.

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

```
fs/nfs/cache_lib.c      | 3 +++
include/linux/sunrpc/cache.h | 2 ++
net/sunrpc/cache.c      | 12 ++++++-----
3 files changed, 10 insertions(+), 7 deletions(-)
```

```
diff --git a/fs/nfs/cache_lib.c b/fs/nfs/cache_lib.c
index c98b439..d62a895 100644
--- a/fs/nfs/cache_lib.c
+++ b/fs/nfs/cache_lib.c
@@ -120,6 +120,7 @@ int nfs_cache_register(struct cache_detail *cd)
     mnt = rpc_get_mount();
     if (IS_ERR(mnt))
         return PTR_ERR(mnt);
+    sunrpc_init_cache_detail(cd);
     ret = vfs_path_lookup(mnt->mnt_root, mnt, "/cache", 0, &path);
     if (ret)
         goto err;
@@ -128,6 +129,7 @@ int nfs_cache_register(struct cache_detail *cd)
     if (!ret)
         return ret;
err:
+    sunrpc_destroy_cache_detail(cd);
     rpc_put_mount();
     return ret;
}
@@ -135,6 +137,7 @@ void nfs_cache_unregister(struct cache_detail *cd)
{
    sunrpc_cache_unregister_pipefs(cd);
+    sunrpc_destroy_cache_detail(cd);
    rpc_put_mount();
}
```

```
diff --git a/include/linux/sunrpc/cache.h b/include/linux/sunrpc/cache.h
index 5efd8ce..57d9fa7 100644
--- a/include/linux/sunrpc/cache.h
```

```

+++ b/include/linux/sunrpc/cache.h
@@ -202,6 +202,8 @@ extern int cache_register_net(struct cache_detail *cd, struct net *net);
extern void cache_unregister(struct cache_detail *cd);
extern void cache_unregister_net(struct cache_detail *cd, struct net *net);

+extern void sunrpc_init_cache_detail(struct cache_detail *cd);
+extern void sunrpc_destroy_cache_detail(struct cache_detail *cd);
extern int sunrpc_cache_register_pipefs(struct dentry *parent, const char *,
    mode_t, struct cache_detail *);
extern void sunrpc_cache_unregister_pipefs(struct cache_detail *);
diff --git a/net/sunrpc/cache.c b/net/sunrpc/cache.c
index 72ad836..320e549 100644
--- a/net/sunrpc/cache.c
+++ b/net/sunrpc/cache.c
@@ -344,7 +344,7 @@ static int current_index;
static void do_cache_clean(struct work_struct *work);
static struct delayed_work cache_cleaner;

-static void sunrpc_init_cache_detail(struct cache_detail *cd)
+void sunrpc_init_cache_detail(struct cache_detail *cd)
{
    rwlock_init(&cd->hash_lock);
    INIT_LIST_HEAD(&cd->queue);
@@ -360,8 +360,9 @@ static void sunrpc_init_cache_detail(struct cache_detail *cd)
    /* start the cleaning process */
    schedule_delayed_work(&cache_cleaner, 0);
}
+EXPORT_SYMBOL_GPL(sunrpc_init_cache_detail);

-static void sunrpc_destroy_cache_detail(struct cache_detail *cd)
+void sunrpc_destroy_cache_detail(struct cache_detail *cd)
{
    cache_purge(cd);
    spin_lock(&cache_list_lock);
@@ -384,6 +385,7 @@ static void sunrpc_destroy_cache_detail(struct cache_detail *cd)
out:
    printk(KERN_ERR "nfsd: failed to unregister %s cache\n", cd->name);
}
+EXPORT_SYMBOL_GPL(sunrpc_destroy_cache_detail);

/* clean cache tries to find something to clean
 * and cleans it.
@@ -1785,17 +1787,14 @@ int sunrpc_cache_register_pipefs(struct dentry *parent,
    struct dentry *dir;
    int ret = 0;

- sunrpc_init_cache_detail(cd);
    q.name = name;

```

```

q.len = strlen(name);
q.hash = full_name_hash(q.name, q.len);
dir = rpc_create_cache_dir(parent, &q, umode, cd);
if (!IS_ERR(dir))
    cd->u.pipefs.dir = dir;
- else {
- sunrpc_destroy_cache_detail(cd);
+ else
    ret = PTR_ERR(dir);
- }
    return ret;
}
EXPORT_SYMBOL_GPL(sunrpc_cache_register_pipefs);
@@ -1804,7 +1803,6 @@ void sunrpc_cache_unregister_pipefs(struct cache_detail *cd)
{
    rpc_remove_cache_dir(cd->u.pipefs.dir);
    cd->u.pipefs.dir = NULL;
- sunrpc_destroy_cache_detail(cd);
}
EXPORT_SYMBOL_GPL(sunrpc_cache_unregister_pipefs);

```
