
Subject: [PATCH v2 1/5] SUNRPC: cache creation and destruction routines introduced

Posted by [Stanislav Kinsbursky](#) on Thu, 19 Jan 2012 17:42:21 GMT

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

This patch prepares infrastructure for network namespace aware cache detail allocation.

One note about adding network namespace link to cache structure. It's going to be used later in NFS DNS cache parsing routine (nfs_dns_parse for rpc_pton() call).

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

include/linux/sunrpc/cache.h | 4 ++++
net/sunrpc/cache.c | 26 ++++++
2 files changed, 30 insertions(+), 0 deletions(-)

diff --git a/include/linux/sunrpc/cache.h b/include/linux/sunrpc/cache.h

index 590a8ab..259381c 100644

--- a/include/linux/sunrpc/cache.h

+++ b/include/linux/sunrpc/cache.h

@@ -117,6 +117,7 @@ struct cache_detail {
 struct cache_detail_procfs procfs;
 struct cache_detail_pipefs pipefs;
} u;
+ struct net *net;
};

@@ -202,6 +203,9 @@ 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 struct cache_detail *cache_create_net(struct cache_detail *tmpl, struct net *net);

+extern void cache_destroy_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 *,

diff --git a/net/sunrpc/cache.c b/net/sunrpc/cache.c

index f128154..9ef5926 100644

--- a/net/sunrpc/cache.c

+++ b/net/sunrpc/cache.c

@@ -1662,6 +1662,32 @@ void cache_unregister(struct cache_detail *cd)
{
 EXPORT_SYMBOL_GPL(cache_unregister);

```

+struct cache_detail *cache_create_net(struct cache_detail *tmpl, struct net *net)
+{
+ struct cache_detail *cd;
+
+ cd = kmemdup(tmpl, sizeof(struct cache_detail), GFP_KERNEL);
+ if (cd == NULL)
+ return ERR_PTR(-ENOMEM);
+
+ cd->hash_table = kzalloc(cd->hash_size * sizeof(struct cache_head *),
+ GFP_KERNEL);
+ if (cd->hash_table == NULL) {
+ kfree(cd);
+ return ERR_PTR(-ENOMEM);
+ }
+ cd->net = net;
+ return cd;
+}
+EXPORT_SYMBOL_GPL(cache_create_net);
+
+void cache_destroy_net(struct cache_detail *cd, struct net *net)
+{
+ kfree(cd->hash_table);
+ kfree(cd);
+}
+EXPORT_SYMBOL_GPL(cache_destroy_net);
+
+static ssize_t cache_read_pipefs(struct file *filp, char __user *buf,
+ size_t count, loff_t *ppos)
+{

```
