
Subject: [PATCH 2/4] nfsd: make id-to-name cache allocated per network namespace context

Posted by [Stanislav Kinsbursky](#) on Wed, 11 Apr 2012 13:32:51 GMT

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

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

fs/nfsd/netns.h | 3 +++
fs/nfsd/nfs4idmap.c | 33 ++++++-----
2 files changed, 26 insertions(+), 10 deletions(-)

diff --git a/fs/nfsd/netns.h b/fs/nfsd/netns.h

index 9794c6c..948a718 100644

--- a/fs/nfsd/netns.h

+++ b/fs/nfsd/netns.h

@@ -31,6 +31,9 @@ struct nfsd_net {

struct cache_detail *svc_expkey_cache;

struct cache_detail *svc_export_cache;

+

+ struct cache_detail *idtoname_cache;

+

};

extern int nfsd_net_id;

diff --git a/fs/nfsd/nfs4idmap.c b/fs/nfsd/nfs4idmap.c

index d37405f..b285a69 100644

--- a/fs/nfsd/nfs4idmap.c

+++ b/fs/nfsd/nfs4idmap.c

@@ -36,9 +36,11 @@

#include <linux/seq_file.h>

#include <linux/sched.h>

#include <linux/slab.h>

+#include <linux/sunrpc/svc_xprt.h>

#include <net/net_namespace.h>

#include "idmap.h"

#include "nfsd.h"

+#include "netns.h"

/*

* Turn off idmapping when using AUTH_SYS.

@@ -107,8 +109,6 @@ ent_alloc(void)

* ID -> Name cache

*/

-static struct cache_head *idtoname_table[ENT_HASHMAX];

-

```

static uint32_t
idtoname_hash(struct ent *ent)
{
@@ -187,10 +187,9 @@ static struct ent *idtoname_lookup(struct cache_detail *, struct ent *);
static struct ent *idtoname_update(struct cache_detail *, struct ent *,
    struct ent *);

-static struct cache_detail idtoname_cache = {
+static struct cache_detail idtoname_cache_template = {
    .owner = THIS_MODULE,
    .hash_size = ENT_HASHMAX,
- .hash_table = idtoname_table,
    .name = "nfs4.idtoname",
    .cache_put = ent_put,
    .cache_upcall = idtoname_upcall,
@@ -472,21 +471,34 @@ int
nfsd_idmap_init(struct net *net)
{
    int rv;
+ struct nfsd_net *nn = net_generic(net, nfsd_net_id);

- rv = cache_register_net(&idtoname_cache, net);
+ nn->idtoname_cache = cache_create_net(&idtoname_cache_template, net);
+ if (IS_ERR(nn->idtoname_cache))
+ return PTR_ERR(nn->idtoname_cache);
+ rv = cache_register_net(nn->idtoname_cache, net);
    if (rv)
- return rv;
+ goto destroy_idtoname_cache;
    rv = cache_register_net(&nametoid_cache, net);
    if (rv)
- cache_unregister_net(&idtoname_cache, net);
+ goto unregister_idtoname_cache;
+ return 0;
+
+unregister_idtoname_cache:
+ cache_unregister_net(nn->idtoname_cache, net);
+destroy_idtoname_cache:
+ cache_destroy_net(nn->idtoname_cache, net);
    return rv;
}

void
nfsd_idmap_shutdown(struct net *net)
{
- cache_unregister_net(&idtoname_cache, 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_destroy_net(nn->idtoname_cache, net);
}

static int
@@ -553,9 +565,10 @@ idmap_id_to_name(struct svc_rqst *rqstp, int type, uid_t id, char *name)
  .type = type,
};
int ret;
+ struct nfsd_net *nn = net_generic(rqstp->rq_xprt->xpt_net, nfsd_net_id);

  strcpy(key.authname, rqst_authname(rqstp), sizeof(key.authname));
- ret = idmap_lookup(rqstp, idtoname_lookup, &key, &idtoname_cache, &item);
+ ret = idmap_lookup(rqstp, idtoname_lookup, &key, nn->idtoname_cache, &item);
  if (ret == -ENOENT)
    return sprintf(name, "%u", id);
  if (ret)
@@ -563,7 +576,7 @@ idmap_id_to_name(struct svc_rqst *rqstp, int type, uid_t id, char *name)
    ret = strlen(item->name);
    BUG_ON(ret > IDMAP_NAMESZ);
    memcpy(name, item->name, ret);
- cache_put(&item->h, &idtoname_cache);
+ cache_put(&item->h, nn->idtoname_cache);
  return ret;
}

```
