Subject: [PATCH 0/4] nfsd: containerize id-to-name and name-to-id caches
Posted by Stanislav Kinsbursky on Wed, 11 Apr 2012 13:32:37 GMT
View Forum Message <> Reply to Message

This patch set depends on "nfsd: remove hard-coded dereferences to name-to-id and id-to-name caches" patch.

The following series consists of:

Stanislav Kinsbursky (4):
      nfsd: pass network context to idmap init/exit functions
      nfsd: make id-to-name cache allocated per network namespace context
      nfsd: make name-to-id cache allocated per network namespace context
      nfsd: allocate id-to-name and name-to-id caches in per-net operations.

```
 fs/nfsd/idmap.h     |    8 +++----
 fs/nfsd/netns.h     |    3 +++
 fs/nfsd/nfs4idmap.c |   59 ++++++++++++++++++++++++++++++++++-----------------
 fs/nfsd/nfsctl.c    |   14 ++++++------
 4 files changed, 53 insertions(+), 31 deletions(-)
```

Subject: [PATCH 1/4] nfsd: pass network context to idmap init/exit functions
Posted by Stanislav Kinsbursky on Wed, 11 Apr 2012 13:32:44 GMT
View Forum Message <> Reply to Message

These functions will be called from per-net operations.

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

```
---
 fs/nfsd/idmap.h     |    8 ++++----
 fs/nfsd/nfs4idmap.c |   14 +++++++-------
 fs/nfsd/nfsctl.c    |    6 +++---
 3 files changed, 14 insertions(+), 14 deletions(-)

diff --git a/fs/nfsd/idmap.h b/fs/nfsd/idmap.h
index 2f3be13..9d513ef 100644
--- a/fs/nfsd/idmap.h
+++ b/fs/nfsd/idmap.h
@@ -42,14 +42,14 @@
 #define IDMAP_NAMESZ 128

 #ifdef CONFIG_NFSD_V4
-int nfsd_idmap_init(void);
```

```
-void nfsd_idmap_shutdown(void);
+int nfsd_idmap_init(struct net *);
+void nfsd_idmap_shutdown(struct net *);
 #else
-static inline int nfsd_idmap_init(void)
+static inline int nfsd_idmap_init(struct net *net)
 {
  return 0;
 }
-static inline void nfsd_idmap_shutdown(void)
+static inline void nfsd_idmap_shutdown(struct net *net)
 {
 }
 #endif
diff --git a/fs/nfsd/nfs4idmap.c b/fs/nfsd/nfs4idmap.c
index 2ff4470..d37405f 100644
--- a/fs/nfsd/nfs4idmap.c
+++ b/fs/nfsd/nfs4idmap.c
@@ -469,24 +469,24 @@ nametoid_update(struct cache_detail *cd, struct ent *new, struct ent
*old)
  */

 int
-nfsd_idmap_init(void)
+nfsd_idmap_init(struct net *net)
 {
  int rv;

- rv = cache_register_net(&idtoname_cache, &init_net);
+ rv = cache_register_net(&idtoname_cache, net);
  if (rv)
   return rv;
- rv = cache_register_net(&nametoid_cache, &init_net);
+ rv = cache_register_net(&nametoid_cache, net);
  if (rv)
-  cache_unregister_net(&idtoname_cache, &init_net);
+  cache_unregister_net(&idtoname_cache, net);
  return rv;
 }

 void
-nfsd_idmap_shutdown(void)
+nfsd_idmap_shutdown(struct net *net)
 {
- cache_unregister_net(&idtoname_cache, &init_net);
- cache_unregister_net(&nametoid_cache, &init_net);
+ cache_unregister_net(&idtoname_cache, net);
+ cache_unregister_net(&nametoid_cache, net);
```

```
 }

 static int
diff --git a/fs/nfsd/nfsctl.c b/fs/nfsd/nfsctl.c
index 08cd87a..d6e8b85 100644
--- a/fs/nfsd/nfsctl.c
+++ b/fs/nfsd/nfsctl.c
@@ -1186,7 +1186,7 @@ static int __init init_nfsd(void)
  if (retval)
   goto out_free_stat;
  nfsd_lockd_init(); /* lockd->nfsd callbacks */
- retval = nfsd_idmap_init();
+ retval = nfsd_idmap_init(&init_net);
  if (retval)
   goto out_free_lockd;
  retval = create_proc_exports_entry();
@@ -1200,7 +1200,7 @@ out_free_all:
  remove_proc_entry("fs/nfs/exports", NULL);
  remove_proc_entry("fs/nfs", NULL);
 out_free_idmap:
- nfsd_idmap_shutdown();
+ nfsd_idmap_shutdown(&init_net);
 out_free_lockd:
  nfsd_lockd_shutdown();
  nfsd_reply_cache_shutdown();
@@ -1223,7 +1223,7 @@ static void __exit exit_nfsd(void)
  remove_proc_entry("fs/nfs", NULL);
  nfsd_stat_shutdown();
  nfsd_lockd_shutdown();
- nfsd_idmap_shutdown();
+ nfsd_idmap_shutdown(&init_net);
  nfsd4_free_slabs();
  nfsd_fault_inject_cleanup();
  unregister_filesystem(&nfsd_fs_type);
```

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);
```

```
  strlcpy(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;
 }
```

Subject: [PATCH 3/4] nfsd: make name-to-id cache allocated per network namespace context

Posted by Stanislav Kinsbursky on Wed, 11 Apr 2012 13:32:58 GMT

View Forum Message <> Reply to Message

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

```
---
 fs/nfsd/netns.h     |    2 +-
 fs/nfsd/nfs4idmap.c |   24 ++++++++++++++++++++---------
 2 files changed, 16 insertions(+), 10 deletions(-)

diff --git a/fs/nfsd/netns.h b/fs/nfsd/netns.h
index 948a718..3936563 100644
--- a/fs/nfsd/netns.h
+++ b/fs/nfsd/netns.h
@@ -33,7 +33,7 @@ struct nfsd_net {
  struct cache_detail *svc_export_cache;

  struct cache_detail *idtoname_cache;
-
+ struct cache_detail *nametoid_cache;
 };

 extern int nfsd_net_id;
diff --git a/fs/nfsd/nfs4idmap.c b/fs/nfsd/nfs4idmap.c
index b285a69..286a7f8 100644
--- a/fs/nfsd/nfs4idmap.c
+++ b/fs/nfsd/nfs4idmap.c
@@ -301,8 +301,6 @@ idtoname_update(struct cache_detail *cd, struct ent *new, struct ent *old)
  * Name -> ID cache
  */
```

```
-static struct cache_head *nametoid_table[ENT_HASHMAX];
-
 static inline int
 nametoid_hash(struct ent *ent)
 {
@@ -362,10 +360,9 @@ static struct ent *nametoid_update(struct cache_detail *, struct ent *,
        struct ent *);
 static int         nametoid_parse(struct cache_detail *, char *, int);

-static struct cache_detail nametoid_cache = {
+static struct cache_detail nametoid_cache_template = {
  .owner  = THIS_MODULE,
  .hash_size = ENT_HASHMAX,
- .hash_table = nametoid_table,
  .name  = "nfs4.nametoid",
  .cache_put = ent_put,
  .cache_upcall = nametoid_upcall,
@@ -479,11 +476,18 @@ nfsd_idmap_init(struct net *net)
  rv = cache_register_net(nn->idtoname_cache, net);
  if (rv)
   goto destroy_idtoname_cache;
- rv = cache_register_net(&nametoid_cache, net);
- if (rv)
+ nn->nametoid_cache = cache_create_net(&nametoid_cache_template, net);
+ if (IS_ERR(nn->nametoid_cache)) {
+  rv = PTR_ERR(nn->idtoname_cache);
   goto unregister_idtoname_cache;
+ }
+ rv = cache_register_net(nn->nametoid_cache, net);
+ if (rv)
+  goto destroy_nametoid_cache;
  return 0;

+destroy_nametoid_cache:
+ cache_destroy_net(nn->nametoid_cache, net);
 unregister_idtoname_cache:
  cache_unregister_net(nn->idtoname_cache, net);
 destroy_idtoname_cache:
@@ -497,8 +501,9 @@ nfsd_idmap_shutdown(struct net *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_unregister_net(nn->nametoid_cache, net);
  cache_destroy_net(nn->idtoname_cache, net);
+ cache_destroy_net(nn->nametoid_cache, net);
 }
```

```
 static int
@@ -541,19 +546,20 @@ idmap_name_to_id(struct svc_rqst *rqstp, int type, const char *name,
u32 namelen
    .type = type,
  };
  int ret;
+ struct nfsd_net *nn = net_generic(rqstp->rq_xprt->xpt_net, nfsd_net_id);

  if (namelen + 1 > sizeof(key.name))
   return nfserr_badowner;
  memcpy(key.name, name, namelen);
  key.name[namelen] = '\0';
  strlcpy(key.authname, rqst_authname(rqstp), sizeof(key.authname));
- ret = idmap_lookup(rqstp, nametoid_lookup, &key, &nametoid_cache, &item);
+ ret = idmap_lookup(rqstp, nametoid_lookup, &key, nn->nametoid_cache, &item);
  if (ret == -ENOENT)
   return nfserr_badowner;
  if (ret)
   return nfserrno(ret);
  *id = item->id;
- cache_put(&item->h, &nametoid_cache);
+ cache_put(&item->h, nn->nametoid_cache);
  return 0;
 }
```

Subject: [PATCH 4/4] nfsd: allocate id-to-name and name-to-id caches in per-net operations.
Posted by Stanislav Kinsbursky on Wed, 11 Apr 2012 13:33:05 GMT
View Forum Message <> Reply to Message

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

```
---
 fs/nfsd/nfsctl.c |   14 +++++++-------
 1 files changed, 7 insertions(+), 7 deletions(-)

diff --git a/fs/nfsd/nfsctl.c b/fs/nfsd/nfsctl.c
index d6e8b85..7269988 100644
--- a/fs/nfsd/nfsctl.c
+++ b/fs/nfsd/nfsctl.c
@@ -1145,14 +1145,20 @@ static __net_init int nfsd_init_net(struct net *net)
  retval = nfsd_export_init(net);
  if (retval)
   goto out_export_error;
+ retval = nfsd_idmap_init(net);
+ if (retval)
```

```
+  goto out_idmap_error;
  return 0;

+out_idmap_error:
+ nfsd_export_shutdown(net);
 out_export_error:
  return retval;
 }

 static __net_exit void nfsd_exit_net(struct net *net)
 {
+ nfsd_idmap_shutdown(net);
  nfsd_export_shutdown(net);
 }

@@ -1186,12 +1192,9 @@ static int __init init_nfsd(void)
  if (retval)
   goto out_free_stat;
  nfsd_lockd_init(); /* lockd->nfsd callbacks */
- retval = nfsd_idmap_init(&init_net);
- if (retval)
-  goto out_free_lockd;
  retval = create_proc_exports_entry();
  if (retval)
-  goto out_free_idmap;
+  goto out_free_lockd;
  retval = register_filesystem(&nfsd_fs_type);
  if (retval)
   goto out_free_all;
@@ -1199,8 +1202,6 @@ static int __init init_nfsd(void)
 out_free_all:
  remove_proc_entry("fs/nfs/exports", NULL);
  remove_proc_entry("fs/nfs", NULL);
-out_free_idmap:
- nfsd_idmap_shutdown(&init_net);
 out_free_lockd:
  nfsd_lockd_shutdown();
  nfsd_reply_cache_shutdown();
@@ -1223,7 +1224,6 @@ static void __exit exit_nfsd(void)
  nfsd_stat_shutdown();
  nfsd_lockd_shutdown();
- nfsd_idmap_shutdown(&init_net);
  nfsd4_free_slabs();
  nfsd_fault_inject_cleanup();
  unregister_filesystem(&nfsd_fs_type);
```