
Subject: [PATCH 0/4] nfsd: containerize export and expkey caches
Posted by [Stanislav Kinsbursky](#) on Wed, 11 Apr 2012 11:13:07 GMT
[View Forum Message](#) <> [Reply to Message](#)

This patch set depends on "NFSD: precursor and cleanup patch set".

The following series consists of:

Stanislav Kinsbursky (4):

nfsd: pass network context to export caches init/shutdown routines
nfsd: make export cache allocated per network namespace context
nfsd: make expkey cache allocated per network namespace context
nfsd: allocate export and expkey caches in per-net operations.

```
fs/nfsd/export.c      | 83 ++++++-----  
fs/nfsd/netns.h      |  3 +  
fs/nfsd/nfsctl.c     | 32 ++++++----  
fs/nfsd/nfssvc.c     |  2 -  
include/linux/nfsd/export.h | 10 +---  
include/linux/sunrpc/svcauth.h |  2 -  
net/sunrpc/svcauth_unix.c | 13 +----  
7 files changed, 90 insertions(+), 55 deletions(-)
```

Subject: [PATCH 1/4] nfsd: pass network context to export caches init/shutdown routines
Posted by [Stanislav Kinsbursky](#) on Wed, 11 Apr 2012 11:13:14 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/export.c      | 20 ++++++-----  
fs/nfsd/nfsctl.c     |  6 +----  
include/linux/nfsd/export.h |  4 +---  
3 files changed, 15 insertions(+), 15 deletions(-)
```

```
diff --git a/fs/nfsd/export.c b/fs/nfsd/export.c  
index 429b2f5..688264b 100644  
--- a/fs/nfsd/export.c  
+++ b/fs/nfsd/export.c  
@@ -1235,17 +1235,17 @@ const struct seq_operations nfs_exports_op = {  
 * Initialize the exports module.
```

```

*/
int
-nfsd_export_init(void)
+nfsd_export_init(struct net *net)
{
    int rv;
    - dprintk("nfsd: initializing export module.\n");
    + dprintk("nfsd: initializing export module (net: %p).\n", net);

    - rv = cache_register_net(&svc_export_cache, &init_net);
    + rv = cache_register_net(&svc_export_cache, net);
    if (rv)
        return rv;
    - rv = cache_register_net(&svc_expkey_cache, &init_net);
    + rv = cache_register_net(&svc_expkey_cache, net);
    if (rv)
        - cache_unregister_net(&svc_export_cache, &init_net);
        + cache_unregister_net(&svc_export_cache, net);
        return rv;

}
@@ -1264,14 +1264,14 @@ nfsd_export_flush(void)
 * Shutdown the exports module.
 */
void
-nfsd_export_shutdown(void)
+nfsd_export_shutdown(struct net *net)
{
    - dprintk("nfsd: shutting down export module.\n");
    + dprintk("nfsd: shutting down export module (net: %p).\n", net);

    - cache_unregister_net(&svc_expkey_cache, &init_net);
    - cache_unregister_net(&svc_export_cache, &init_net);
    + cache_unregister_net(&svc_expkey_cache, net);
    + cache_unregister_net(&svc_export_cache, net);
    svcauth_unix_purge();

    - dprintk("nfsd: export shutdown complete.\n");
    + dprintk("nfsd: export shutdown complete (net: %p).\n", net);
}

diff --git a/fs/nfsd/nfsctl.c b/fs/nfsd/nfsctl.c
index ae19293..bc76f8e 100644
--- a/fs/nfsd/nfsctl.c
+++ b/fs/nfsd/nfsctl.c
@@ -1163,7 +1163,7 @@ static int __init init_nfsd(void)
    retval = nfsd_reply_cache_init();
    if (retval)

```

```

goto out_free_stat;
- retval = nfsd_export_init();
+ retval = nfsd_export_init(&init_net);
if (retval)
    goto out_free_cache;
nfsd_lockd_init(); /* lockd->nfsd callbacks */
@@ -1184,7 +1184,7 @@ out_free_idmap:
nfsd_idmap_shutdown();
out_free_lockd:
nfsd_lockd_shutdown();
- nfsd_export_shutdown();
+ nfsd_export_shutdown(&init_net);
out_free_cache:
nfsd_reply_cache_shutdown();
out_free_stat:
@@ -1201,7 +1201,7 @@ out_unregister_notifier:

static void __exit exit_nfsd(void)
{
- nfsd_export_shutdown();
+ nfsd_export_shutdown(&init_net);
nfsd_reply_cache_shutdown();
remove_proc_entry("fs/nfs/exports", NULL);
remove_proc_entry("fs/nfs", NULL);
diff --git a/include/linux/nfsd/export.h b/include/linux/nfsd/export.h
index 485c2af..375096c 100644
--- a/include/linux/nfsd/export.h
+++ b/include/linux/nfsd/export.h
@@ -130,8 +130,8 @@ __be32 check_nfsd_access(struct svc_export *exp, struct svc_rqst
*qstp);
/*
 * Function declarations
 */
-int nfsd_export_init(void);
-void nfsd_export_shutdown(void);
+int nfsd_export_init(struct net *net);
+void nfsd_export_shutdown(struct net *net);
void nfsd_export_flush(void);
struct svc_export * rqst_exp_get_by_name(struct svc_rqst *,
    struct path *);
```

Subject: [PATCH 2/4] nfsd: make export cache allocated per network namespace context

Posted by [Stanislav Kinsbursky](#) on Wed, 11 Apr 2012 11:13:21 GMT

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

This patch also changes prototypes of nfsd_export_flush() and exp_rootfh():

network namespace parameter added.

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

```
---  
fs/nfsd/export.c      | 47 ++++++-----  
fs/nfsd/netns.h       | 2 ++  
fs/nfsd/nfsctl.c     | 2 +-  
fs/nfsd/nfssvc.c     | 2 +-  
include/linux/nfsd/export.h | 4 +---  
5 files changed, 38 insertions(+), 19 deletions(-)
```

```
diff --git a/fs/nfsd/export.c b/fs/nfsd/export.c  
index 688264b..84d020f 100644  
--- a/fs/nfsd/export.c  
+++ b/fs/nfsd/export.c  
@@ -15,11 +15,13 @@  
#include <linux/namei.h>  
#include <linux/module.h>  
#include <linux/exportfs.h>  
+#include <linux/sunrpc/svc_xprt.h>  
  
#include <net/ipv6.h>  
  
#include "nfsd.h"  
#include "nfsfh.h"  
+#include "netns.h"  
  
#define NFSDBG_FACILITY NFSDBG_EXPORT  
  
@@ -298,8 +300,6 @@ svc_expkey_update(struct cache_detail *cd, struct svc_expkey *new,  
#define EXPORT_HASHBITS 8  
#define EXPORT_HASHMAX (1<< EXPORT_HASHBITS)  
  
-static struct cache_head *export_table[EXPORT_HASHMAX];  
-  
static void nfsd4_fslocs_free(struct nfsd4_fs_locations *fsloc)  
{  
    int i;  
@@ -708,10 +708,9 @@ static struct cache_head *svc_export_alloc(void)  
    return NULL;  
}  
  
-struct cache_detail svc_export_cache = {  
+struct cache_detail svc_export_cache_template = {  
    .owner = THIS_MODULE,  
    .hash_size = EXPORT_HASHMAX,  
-   .hash_table = export_table,
```

```

.name = "nfsd.export",
.cache_put = svc_export_put,
.cache_upcall = svc_export_upcall,
@@ -835,7 +834,7 @@ static struct svc_export *exp_parent(struct cache_detail *cd, svc_client
*clp,
 * since its harder to fool a kernel module than a user space program.
 */
int
-exp_rootfh(svc_client *clp, char *name,
+exp_rootfh(struct net *net, svc_client *clp, char *name,
    struct knfsd_fh *f, int maxsize)
{
    struct svc_export *exp;
@@ -843,7 +842,8 @@ exp_rootfh(svc_client *clp, char *name,
    struct inode *inode;
    struct svc_fh fh;
    int err;
- struct cache_detail *cd = &svc_export_cache;
+ struct nfsd_net *nn = net_generic(net, nfsd_net_id);
+ struct cache_detail *cd = nn->svc_export_cache;

    err = -EPERM;
    /* NB: we probably ought to check that it's NUL-terminated */
@@ -930,7 +930,8 @@ 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;
+ struct nfsd_net *nn = net_generic(rqstp->rq_xprt->xpt_net, nfsd_net_id);
+ struct cache_detail *cd = nn->svc_export_cache;

    if (rqstp->rq_client == NULL)
        goto gss;
@@ -960,7 +961,8 @@ 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;
+ struct nfsd_net *nn = net_generic(rqstp->rq_xprt->xpt_net, nfsd_net_id);
+ struct cache_detail *cd = nn->svc_export_cache;

    if (rqstp->rq_client == NULL)
        goto gss;
@@ -1238,26 +1240,39 @@ int
nfsd_export_init(struct net *net)
{
    int rv;
+ struct nfsd_net *nn = net_generic(net, nfsd_net_id);

```

```

+
dprintk("nfsd: initializing export module (net: %p).\n", net);

- rv = cache_register_net(&svc_export_cache, net);
+ nn->svc_export_cache = cache_create_net(&svc_export_cache_template, net);
+ if (IS_ERR(nn->svc_export_cache))
+ return PTR_ERR(nn->svc_export_cache);
+ rv = cache_register_net(nn->svc_export_cache, net);
if (rv)
- return rv;
+ goto destroy_export_cache;
+
rv = cache_register_net(&svc_expkey_cache, net);
if (rv)
- cache_unregister_net(&svc_export_cache, net);
- return rv;
+ goto unregister_export_cache;
+ return 0;

+unregister_export_cache:
+ cache_unregister_net(nn->svc_export_cache, net);
+destroy_export_cache:
+ cache_destroy_net(nn->svc_export_cache, net);
+ return rv;
}

/*
 * Flush exports table - called when last nfsd thread is killed
 */
void
-nfsd_export_flush(void)
+nfsd_export_flush(struct net *net)
{
+ struct nfsd_net *nn = net_generic(net, nfsd_net_id);
+
cache_purge(&svc_expkey_cache);
- cache_purge(&svc_export_cache);
+ cache_purge(nn->svc_export_cache);
}

/*
@@ -1266,11 +1281,13 @@ nfsd_export_flush(void)
void
nfsd_export_shutdown(struct net *net)
{
+ struct nfsd_net *nn = net_generic(net, nfsd_net_id);

dprintk("nfsd: shutting down export module (net: %p).\n", net);

```

```

cache_unregister_net(&svc_expkey_cache, net);
- cache_unregister_net(&svc_export_cache, net);
+ cache_unregister_net(nn->svc_export_cache, net);
+ cache_destroy_net(nn->svc_export_cache, net);
    svcauth_unix_purge();

dprintk("nfsd: export shutdown complete (net: %p).\n", net);
diff --git a/fs/nfsd/netns.h b/fs/nfsd/netns.h
index 12e0cff..c1c6242 100644
--- a/fs/nfsd/netns.h
+++ b/fs/nfsd/netns.h
@@ -28,6 +28,8 @@ struct cld_net;

struct nfsd_net {
    struct cld_net *cld_net;
+
+    struct cache_detail *svc_export_cache;
};

extern int nfsd_net_id;
diff --git a/fs/nfsd/nfsctl.c b/fs/nfsd/nfsctl.c
index bc76f8e..ddb9f87 100644
--- a/fs/nfsd/nfsctl.c
+++ b/fs/nfsd/nfsctl.c
@@ -354,7 +354,7 @@ static ssize_t write_filehandle(struct file *file, char *buf, size_t size)
    if (!dom)
        return -ENOMEM;

- len = exp_rootfh(dom, path, &fh, maxsize);
+ len = exp_rootfh(&init_net, dom, path, &fh, maxsize);
    auth_domain_put(dom);
    if (len)
        return len;
diff --git a/fs/nfsd/nfssvc.c b/fs/nfsd/nfssvc.c
index 78e5213..cb4d51d 100644
--- a/fs/nfsd/nfssvc.c
+++ b/fs/nfsd/nfssvc.c
@@ -261,7 +261,7 @@ static void nfsd_last_thread(struct svc_serv *serv, struct net *net)

    printk(KERN_WARNING "nfsd: last server has exited, flushing export "
          "cache\n");
- nfsd_export_flush();
+ nfsd_export_flush(net);
}

void nfsd_reset_versions(void)
diff --git a/include/linux/nfsd/export.h b/include/linux/nfsd/export.h

```

```

index 375096c..565c212 100644
--- a/include/linux/nfsd/export.h
+++ b/include/linux/nfsd/export.h
@@ -132,13 +132,13 @@ __be32 check_nfsd_access(struct svc_export *exp, struct svc_rqst
 *rqstp);
 */
int nfsd_export_init(struct net *);
void nfsd_export_shutdown(struct net *);
-void nfsd_export_flush(void);
+void nfsd_export_flush(struct net *);
struct svc_export * rqst_exp_get_by_name(struct svc_rqst *,
    struct path *);
struct svc_export * rqst_exp_parent(struct svc_rqst *,
    struct path *);
struct svc_export * rqst_find_fsidzero_export(struct svc_rqst *);
-int exp_rootfh(struct auth_domain *,
+int exp_rootfh(struct net *, struct auth_domain *,
    char *path, struct knfsd_fh *, int maxsize);
__be32 exp_pseudoroot(struct svc_rqst *, struct svc_fh *);
__be32 nfserrno(int errno);

```

Subject: [PATCH 3/4] nfsd: make expkey cache allocated per network namespace context

Posted by Stanislav Kinsbursky on Wed, 11 Apr 2012 11:13:28 GMT

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

This patch also changes svcauth_unix_purge() function: added network namespace as a parameter and thus loop over all networks was replaced by only one call for ip map cache purge.

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

```

---
fs/nfsd/export.c      | 28 ++++++-----+
fs/nfsd/netns.h       |  1 +
fs/nfsd/nfsctl.c     |  3 ++
include/linux/nfsd/export.h |  2 --
include/linux/sunrpc/svcauth.h |  2 ++
net/sunrpc/svcauth_unix.c | 13 +++++-----
6 files changed, 25 insertions(+), 24 deletions(-)
```

```

diff --git a/fs/nfsd/export.c b/fs/nfsd/export.c
index 84d020f..dcba52b8 100644
--- a/fs/nfsd/export.c
+++ b/fs/nfsd/export.c
@@ -40,7 +40,6 @@ typedef struct svc_export svc_export;
#define EXPKEY_HASHBITS 8
```

```

#define EXPKEY_HASHMAX (1 << EXPKEY_HASHBITS)
#define EXPKEY_HASHMASK (EXPKEY_HASHMAX -1)
-static struct cache_head *expkey_table[EXPKEY_HASHMAX];

static void expkey_put(struct kref *ref)
{
@@ -241,10 +240,9 @@ static struct cache_head *expkey_alloc(void)
    return NULL;
}

-static struct cache_detail svc_expkey_cache = {
+static struct cache_detail svc_expkey_cache_template = {
    .owner = THIS_MODULE,
    .hash_size = EXPKEY_HASHMAX,
-   .hash_table = expkey_table,
    .name = "nfsd.fh",
    .cache_put = expkey_put,
    .cache_upcall = expkey_upcall,
@@ -883,12 +881,13 @@ static struct svc_export *exp_find(struct cache_detail *cd,
    u32 *fsidv, struct cache_req *reqp)
{
    struct svc_export *exp;
-   struct svc_expkey *ek = exp_find_key(&svc_expkey_cache, clp, fsid_type, fsidv, reqp);
+   struct nfsd_net *nn = net_generic(cd->net, nfsd_net_id);
+   struct svc_expkey *ek = exp_find_key(nn->svc_expkey_cache, clp, fsid_type, fsidv, reqp);
    if (IS_ERR(ek))
        return ERR_CAST(ek);

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

    if (IS_ERR(exp))
        return ERR_CAST(exp);
@@ -1232,7 +1231,6 @@ const struct seq_operations nfs_exports_op = {
    .show = e_show,
};

/*
 * Initialize the exports module.
 */
@@ -1251,11 +1249,18 @@ nfsd_export_init(struct net *net)
if (rv)
    goto destroy_export_cache;

-   rv = cache_register_net(&svc_expkey_cache, net);
-   if (rv)

```

```

+ nn->svc_expkey_cache = cache_create_net(&svc_expkey_cache_template, net);
+ if (IS_ERR(nn->svc_expkey_cache)) {
+   rv = PTR_ERR(nn->svc_expkey_cache);
   goto unregister_export_cache;
+ }
+ rv = cache_register_net(nn->svc_expkey_cache, net);
+ if (rv)
+   goto destroy_expkey_cache;
 return 0;

+destroy_expkey_cache:
+ cache_destroy_net(nn->svc_expkey_cache, net);
unregister_export_cache:
 cache_unregister_net(nn->svc_export_cache, net);
destroy_export_cache:
@@ -1271,7 +1276,7 @@ nfsd_export_flush(struct net *net)
{
 struct nfsd_net *nn = net_generic(net, nfsd_net_id);

- cache_purge(&svc_expkey_cache);
+ cache_purge(nn->svc_expkey_cache);
 cache_purge(nn->svc_export_cache);
}

@@ -1285,10 +1290,11 @@ nfsd_export_shutdown(struct net *net)

 dprintk("nfsd: shutting down export module (net: %p).\n", net);

- cache_unregister_net(&svc_expkey_cache, net);
+ cache_unregister_net(nn->svc_expkey_cache, net);
 cache_unregister_net(nn->svc_export_cache, net);
+ cache_destroy_net(nn->svc_expkey_cache, net);
 cache_destroy_net(nn->svc_export_cache, net);
- svcauth_unix_purge();
+ svcauth_unix_purge(net);

 dprintk("nfsd: export shutdown complete (net: %p).\n", net);
}
diff --git a/fs/nfsd/netns.h b/fs/nfsd/netns.h
index c1c6242..9794c6c 100644
--- a/fs/nfsd/netns.h
+++ b/fs/nfsd/netns.h
@@ -29,6 +29,7 @@ struct cld_net;
struct nfsd_net {
 struct cld_net *cld_net;

+ struct cache_detail *svc_expkey_cache;
 struct cache_detail *svc_export_cache;

```

};

```
diff --git a/fs/nfsd/nfsctl.c b/fs/nfsd/nfsctl.c
index ddb9f87..b144177 100644
--- a/fs/nfsd/nfsctl.c
+++ b/fs/nfsd/nfsctl.c
@@ -129,13 +129,14 @@ static int exports_open(struct inode *inode, struct file *file)
{
    int err;
    struct seq_file *seq;
+ struct nfsd_net *nn = net_generic(&init_net, nfsd_net_id);

    err = seq_open(file, &nfs_exports_op);
    if (err)
        return err;

    seq = file->private_data;
- seq->private = &svc_export_cache;
+ seq->private = nn->svc_export_cache;
    return 0;
}
```

```
diff --git a/include/linux/nfsd/export.h b/include/linux/nfsd/export.h
index 565c212..e33f747 100644
--- a/include/linux/nfsd/export.h
+++ b/include/linux/nfsd/export.h
@@ -143,8 +143,6 @@ int exp_rootfh(struct net *, struct auth_domain *,
__be32 exp_pseudoroot(struct svc_rqst *, struct svc_fh *);
__be32 nfserrno(int errno);

-extern struct cache_detail svc_export_cache;
-
static inline void exp_put(struct svc_export *exp)
{
    cache_put(&exp->h, exp->cd);
diff --git a/include/linux/sunrpc/svcauth.h b/include/linux/sunrpc/svcauth.h
index 548790e..153de45 100644
--- a/include/linux/sunrpc/svcauth.h
+++ b/include/linux/sunrpc/svcauth.h
@@ -131,7 +131,7 @@ extern struct auth_domain *auth_domain_lookup(char *name, struct
auth_domain *ne
    extern struct auth_domain *auth_domain_find(char *name);
    extern struct auth_domain *auth_unix_lookup(struct net *net, struct in6_addr *addr);
    extern int auth_unix_forget_old(struct auth_domain *dom);
-extern void svcauth_unix_purge(void);
+extern void svcauth_unix_purge(struct net *net);
    extern void svcauth_unix_info_release(struct svc_xprt *xpt);
    extern int svcauth_unix_set_client(struct svc_rqst *rqstp);
```

```

diff --git a/net/sunrpc/svcauth_unix.c b/net/sunrpc/svcauth_unix.c
index 521d8f7..9c3b9f0 100644
--- a/net/sunrpc/svcauth_unix.c
+++ b/net/sunrpc/svcauth_unix.c
@@@ -346,17 +346,12 @@ static inline int ip_map_update(struct net *net, struct ip_map *ipm,
    return __ip_map_update(sn->ip_map_cache, ipm, udom, expiry);
}

-
-void svcauth_unix_purge(void)
+void svcauth_unix_purge(struct net *net)
{
- struct net *net;
-
- for_each_net(net) {
- struct sunrpc_net *sn;
+ struct sunrpc_net *sn;

- sn = net_generic(net, sunrpc_net_id);
- cache_purge(sn->ip_map_cache);
- }
+ sn = net_generic(net, sunrpc_net_id);
+ cache_purge(sn->ip_map_cache);
}
EXPORT_SYMBOL_GPL(svcauth_unix_purge);

```

Subject: [PATCH 4/4] nfsd: allocate export and expkey caches in per-net operations.

Posted by [Stanislav Kinsbursky](#) on Wed, 11 Apr 2012 11:13:35 GMT

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

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

```

---
fs/nfsd/nfsctl.c | 27 ++++++-----+
1 files changed, 21 insertions(+), 6 deletions(-)
```

```

diff --git a/fs/nfsd/nfsctl.c b/fs/nfsd/nfsctl.c
index b144177..08cd87a 100644
--- a/fs/nfsd/nfsctl.c
+++ b/fs/nfsd/nfsctl.c
@@@ -1137,7 +1137,28 @@ static int create_proc_exports_entry(void)
#endif

int nfsd_net_id;
+
```

```

+static __net_init int nfsd_init_net(struct net *net)
+{
+ int retval;
+
+ retval = nfsd_export_init(net);
+ if (retval)
+ goto out_export_error;
+ return 0;
+
+out_export_error:
+ return retval;
+}
+
+static __net_exit void nfsd_exit_net(struct net *net)
+{
+ nfsd_export_shutdown(net);
+}
+
 static struct pernet_operations nfsd_net_ops = {
+ .init = nfsd_init_net,
+ .exit = nfsd_exit_net,
 .id = &nfsd_net_id,
 .size = sizeof(struct nfsd_net),
};

@@ -1164,9 +1185,6 @@ static int __init init_nfsd(void)
 retval = nfsd_reply_cache_init();
 if (retval)
 goto out_free_stat;
- retval = nfsd_export_init(&init_net);
- if (retval)
- goto out_free_cache;
 nfsd_lockd_init(); /* lockd->nfsd callbacks */
 retval = nfsd_idmap_init();
 if (retval)
@@ -1185,8 +1203,6 @@ out_free_idmap:
 nfsd_idmap_shutdown();
 out_free_lockd:
 nfsd_lockd_shutdown();
- nfsd_export_shutdown(&init_net);
-out_free_cache:
 nfsd_reply_cache_shutdown();
 out_free_stat:
 nfsd_stat_shutdown();
@@ -1202,7 +1218,6 @@ out_unregister_notifier:

 static void __exit exit_nfsd(void)
{
- nfsd_export_shutdown(&init_net);

```

```
nfsd_reply_cache_shutdown();
remove_proc_entry("fs/nfs/exports", NULL);
remove_proc_entry("fs/nfs", NULL);
```

Subject: Re: [PATCH 2/4] nfsd: make export cache allocated per network namespace context

Posted by [bfields](#) on Wed, 11 Apr 2012 22:14:29 GMT

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

On Wed, Apr 11, 2012 at 03:13:21PM +0400, Stanislav Kinsbursky wrote:

> This patch also changes prototypes of nfsd_export_flush() and exp_rootfh():
> network namespace parameter added.

This also isn't applying cleanly for me.

At this point I'm assuming any patches that aren't already applied to my git tree need to be resent.

--b.

```
>
> Signed-off-by: Stanislav Kinsbursky <skinsbursky@parallels.com>
>
> ---
> fs/nfsd/export.c      | 47 ++++++-----+
> fs/nfsd/netns.h       |  2 ++
> fs/nfsd/nfsctl.c     |  2 ++
> fs/nfsd/nfssvc.c     |  2 ++
> include/linux/nfsd/export.h |  4 +-
> 5 files changed, 38 insertions(+), 19 deletions(-)
>
> diff --git a/fs/nfsd/export.c b/fs/nfsd/export.c
> index 688264b..84d020f 100644
> --- a/fs/nfsd/export.c
> +++ b/fs/nfsd/export.c
> @@ -15,11 +15,13 @@
> #include <linux/namei.h>
> #include <linux/module.h>
> #include <linux/exportfs.h>
> +#include <linux/sunrpc/svc_xprt.h>
>
> #include <net/ipv6.h>
>
> #include "nfsd.h"
> #include "nfsfh.h"
> +#include "netns.h"
>
```

```

> #define NFSDBG_FACILITY NFSDBG_EXPORT
>
> @@ -298,8 +300,6 @@ svc_expkey_update(struct cache_detail *cd, struct svc_expkey *new,
> #define EXPORT_HASHBITS 8
> #define EXPORT_HASHMAX (1<< EXPORT_HASHBITS)
>
> -static struct cache_head *export_table[EXPORT_HASHMAX];
> -
> static void nfsd4_fslocs_free(struct nfsd4_fs_locations *fsloc)
> {
>     int i;
> @@ -708,10 +708,9 @@ static struct cache_head *svc_export_alloc(void)
>     return NULL;
> }
>
> -struct cache_detail svc_export_cache = {
> +struct cache_detail svc_export_cache_template = {
>     .owner = THIS_MODULE,
>     .hash_size = EXPORT_HASHMAX,
>     .hash_table = export_table,
>     .name = "nfsd.export",
>     .cache_put = svc_export_put,
>     .cache_upcall = svc_export_upcall,
> @@ -835,7 +834,7 @@ static struct svc_export *exp_parent(struct cache_detail *cd, svc_client
*clp,
>     * since its harder to fool a kernel module than a user space program.
> */
> int
> -exp_rootfh(svc_client *clp, char *name,
> +exp_rootfh(struct net *net, svc_client *clp, char *name,
>     struct knfsd_fh *f, int maxsize)
> {
>     struct svc_export *exp;
> @@ -843,7 +842,8 @@ exp_rootfh(svc_client *clp, char *name,
>     struct inode *inode;
>     struct svc_fh fh;
>     int err;
>     - struct cache_detail *cd = &svc_export_cache;
>     + struct nfsd_net *nn = net_generic(net, nfsd_net_id);
>     + struct cache_detail *cd = nn->svc_export_cache;
>
>     err = -EPERM;
>     /* NB: we probably ought to check that it's NUL-terminated */
> @@ -930,7 +930,8 @@ 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;

```

```

> + struct nfsd_net *nn = net_generic(rqstp->rq_xprt->xpt_net, nfsd_net_id);
> + struct cache_detail *cd = nn->svc_export_cache;
>
> if (rqstp->rq_client == NULL)
>   goto gss;
> @@ -960,7 +961,8 @@ 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;
> + struct nfsd_net *nn = net_generic(rqstp->rq_xprt->xpt_net, nfsd_net_id);
> + struct cache_detail *cd = nn->svc_export_cache;
>
> if (rqstp->rq_client == NULL)
>   goto gss;
> @@ -1238,26 +1240,39 @@ int
> nfsd_export_init(struct net *net)
> {
>   int rv;
> + struct nfsd_net *nn = net_generic(net, nfsd_net_id);
> +
>   dprintk("nfsd: initializing export module (net: %p).\n", net);
>
> - rv = cache_register_net(&svc_export_cache, net);
> + nn->svc_export_cache = cache_create_net(&svc_export_cache_template, net);
> + if (IS_ERR(nn->svc_export_cache))
> +   return PTR_ERR(nn->svc_export_cache);
> + rv = cache_register_net(nn->svc_export_cache, net);
>   if (rv)
> -   return rv;
> +   goto destroy_export_cache;
> +
>   rv = cache_register_net(&svc_expkey_cache, net);
>   if (rv)
> -   cache_unregister_net(&svc_export_cache, net);
> -   return rv;
> +   goto unregister_export_cache;
> + return 0;
>
> +unregister_export_cache:
> + cache_unregister_net(nn->svc_export_cache, net);
> +destroy_export_cache:
> + cache_destroy_net(nn->svc_export_cache, net);
> + return rv;
> }
>
> /*
> * Flush exports table - called when last nfsd thread is killed

```

```

> /*
> void
> -nfsd_export_flush(void)
> +nfsd_export_flush(struct net *net)
> {
> + struct nfsd_net *nn = net_generic(net, nfsd_net_id);
> +
> cache_purge(&svc_expkey_cache);
> - cache_purge(&svc_export_cache);
> + cache_purge(nn->svc_export_cache);
> }
>
> /*
> @@ -1266,11 +1281,13 @@ nfsd_export_flush(void)
> void
> nfsd_export_shutdown(struct net *net)
> {
> + struct nfsd_net *nn = net_generic(net, nfsd_net_id);
>
> dprintk("nfsd: shutting down export module (net: %p).\n", net);
>
> cache_unregister_net(&svc_expkey_cache, net);
> - cache_unregister_net(&svc_export_cache, net);
> + cache_unregister_net(nn->svc_export_cache, net);
> + cache_destroy_net(nn->svc_export_cache, net);
> svcauth_unix_purge();
>
> dprintk("nfsd: export shutdown complete (net: %p).\n", net);
> diff --git a/fs/nfsd/netns.h b/fs/nfsd/netns.h
> index 12e0cff..c1c6242 100644
> --- a/fs/nfsd/netns.h
> +++ b/fs/nfsd/netns.h
> @@ -28,6 +28,8 @@ struct cld_net;
>
> struct nfsd_net {
> struct cld_net *cld_net;
> +
> + struct cache_detail *svc_export_cache;
> };
>
> extern int nfsd_net_id;
> diff --git a/fs/nfsd/nfsctl.c b/fs/nfsd/nfsctl.c
> index bc76f8e..ddb9f87 100644
> --- a/fs/nfsd/nfsctl.c
> +++ b/fs/nfsd/nfsctl.c
> @@ -354,7 +354,7 @@ static ssize_t write_filehandle(struct file *file, char *buf, size_t size)
> if (!dom)
> return -ENOMEM;

```

```

>
> - len = exp_rootfh(dom, path, &fh, maxsize);
> + len = exp_rootfh(&init_net, dom, path, &fh, maxsize);
> auth_domain_put(dom);
> if (len)
> return len;
> diff --git a/fs/nfsd/nfssvc.c b/fs/nfsd/nfssvc.c
> index 78e5213..cb4d51d 100644
> --- a/fs/nfsd/nfssvc.c
> +++ b/fs/nfsd/nfssvc.c
> @@ -261,7 +261,7 @@ static void nfsd_last_thread(struct svc_serv *serv, struct net *net)
>
> printk(KERN_WARNING "nfsd: last server has exited, flushing export "
>       "cache\n");
> - nfsd_export_flush();
> + nfsd_export_flush(net);
> }
>
> void nfsd_reset_versions(void)
> diff --git a/include/linux/nfsd/export.h b/include/linux/nfsd/export.h
> index 375096c..565c212 100644
> --- a/include/linux/nfsd/export.h
> +++ b/include/linux/nfsd/export.h
> @@ -132,13 +132,13 @@ __be32 check_nfsd_access(struct svc_export *exp, struct svc_rqst
*rqstp);
> */
> int nfsd_export_init(struct net *);
> void nfsd_export_shutdown(struct net *);
> -void nfsd_export_flush(void);
> +void nfsd_export_flush(struct net *);
> struct svc_export * rqst_exp_get_by_name(struct svc_rqst *,
>           struct path *);
> struct svc_export * rqst_exp_parent(struct svc_rqst *,
>           struct path *);
> struct svc_export * rqst_find_fsidzero_export(struct svc_rqst *);
> -int exp_rootfh(struct auth_domain *,
> +int exp_rootfh(struct net *, struct auth_domain *,
>           char *path, struct knfsd_fh *, int maxsize);
> __be32 exp_pseudoroot(struct svc_rqst *, struct svc_fh *);
> __be32 nfserrno(int errno);
>
```
