
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 nfserro(int errno);
>

```
