Subject: [PATCH] nfsd: remove hard-coded dereferences to name-to-id and id-to-name caches
Posted by Stanislav Kinsbursky on Wed, 11 Apr 2012 11:18:58 GMT
View Forum Message <> Reply to Message

These dereferences to global static caches are redundant. They also prevents
converting these caches into per-net ones. So this patch is cleanup + precursor
of patch set, which will make them per-net.

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

```
---
 fs/nfsd/nfs4idmap.c |   50 ++++++++++++++++++++++++------------------------
 1 files changed, 24 insertions(+), 26 deletions(-)

diff --git a/fs/nfsd/nfs4idmap.c b/fs/nfsd/nfs4idmap.c
index 322d11c..2ff4470 100644
--- a/fs/nfsd/nfs4idmap.c
+++ b/fs/nfsd/nfs4idmap.c
@@ -183,8 +183,9 @@ warn_no_idmapd(struct cache_detail *detail, int has_died)


 static int         idtoname_parse(struct cache_detail *, char *, int);
-static struct ent *idtoname_lookup(struct ent *);
-static struct ent *idtoname_update(struct ent *, struct ent *);
+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 = {
  .owner  = THIS_MODULE,
@@ -244,7 +245,7 @@ idtoname_parse(struct cache_detail *cd, char *buf, int buflen)
   goto out;

  error = -ENOMEM;
- res = idtoname_lookup(&ent);
+ res = idtoname_lookup(cd, &ent);
  if (!res)
   goto out;

@@ -260,11 +261,11 @@ idtoname_parse(struct cache_detail *cd, char *buf, int buflen)
  else
   memcpy(ent.name, buf1, sizeof(ent.name));
  error = -ENOMEM;
- res = idtoname_update(&ent, res);
+ res = idtoname_update(cd, &ent, res);
  if (res == NULL)
   goto out;
```

```
- cache_put(&res->h, &idtoname_cache);
+ cache_put(&res->h, cd);

  error = 0;
 out:
@@ -275,10 +276,9 @@ out:


 static struct ent *
-idtoname_lookup(struct ent *item)
+idtoname_lookup(struct cache_detail *cd, struct ent *item)
 {
- struct cache_head *ch = sunrpc_cache_lookup(&idtoname_cache,
-          &item->h,
+ struct cache_head *ch = sunrpc_cache_lookup(cd, &item->h,
          idtoname_hash(item));
   if (ch)
    return container_of(ch, struct ent, h);
@@ -287,10 +287,9 @@ idtoname_lookup(struct ent *item)
 }

 static struct ent *
-idtoname_update(struct ent *new, struct ent *old)
+idtoname_update(struct cache_detail *cd, struct ent *new, struct ent *old)
 {
- struct cache_head *ch = sunrpc_cache_update(&idtoname_cache,
-          &new->h, &old->h,
+ struct cache_head *ch = sunrpc_cache_update(cd, &new->h, &old->h,
          idtoname_hash(new));
   if (ch)
    return container_of(ch, struct ent, h);
@@ -359,8 +358,9 @@ nametoid_show(struct seq_file *m, struct cache_detail *cd, struct
cache_head *h)
   return 0;
 }

-static struct ent *nametoid_lookup(struct ent *);
-static struct ent *nametoid_update(struct ent *, struct ent *);
+static struct ent *nametoid_lookup(struct cache_detail *, struct ent *);
+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 = {
@@ -426,14 +426,14 @@ nametoid_parse(struct cache_detail *cd, char *buf, int buflen)
    set_bit(CACHE_NEGATIVE, &ent.h.flags);
```

```
  error = -ENOMEM;
- res = nametoid_lookup(&ent);
+ res = nametoid_lookup(cd, &ent);
  if (res == NULL)
   goto out;
- res = nametoid_update(&ent, res);
+ res = nametoid_update(cd, &ent, res);
  if (res == NULL)
   goto out;

- cache_put(&res->h, &nametoid_cache);
+ cache_put(&res->h, cd);
  error = 0;
 out:
  kfree(buf1);
@@ -443,10 +443,9 @@ out:


 static struct ent *
-nametoid_lookup(struct ent *item)
+nametoid_lookup(struct cache_detail *cd, struct ent *item)
 {
- struct cache_head *ch = sunrpc_cache_lookup(&nametoid_cache,
-          &item->h,
+ struct cache_head *ch = sunrpc_cache_lookup(cd, &item->h,
          nametoid_hash(item));
  if (ch)
   return container_of(ch, struct ent, h);
@@ -455,10 +454,9 @@ nametoid_lookup(struct ent *item)
 }

 static struct ent *
-nametoid_update(struct ent *new, struct ent *old)
+nametoid_update(struct cache_detail *cd, struct ent *new, struct ent *old)
 {
- struct cache_head *ch = sunrpc_cache_update(&nametoid_cache,
-          &new->h, &old->h,
+ struct cache_head *ch = sunrpc_cache_update(cd, &new->h, &old->h,
          nametoid_hash(new));
  if (ch)
   return container_of(ch, struct ent, h);
@@ -493,12 +491,12 @@ nfsd_idmap_shutdown(void)

 static int
 idmap_lookup(struct svc_rqst *rqstp,
-  struct ent *(*lookup_fn)(struct ent *), struct ent *key,
-  struct cache_detail *detail, struct ent **item)
+  struct ent *(*lookup_fn)(struct cache_detail *, struct ent *),
```

```
+  struct ent *key, struct cache_detail *detail, struct ent **item)
 {
  int ret;

- *item = lookup_fn(key);
+ *item = lookup_fn(detail, key);
  if (!*item)
   return -ENOMEM;
  retry:
@@ -506,7 +504,7 @@ idmap_lookup(struct svc_rqst *rqstp,

  if (ret == -ETIMEDOUT) {
   struct ent *prev_item = *item;
-  *item = lookup_fn(key);
+  *item = lookup_fn(detail, key);
   if (*item != prev_item)
    goto retry;
   cache_put(&(*item)->h, detail);
```

Subject: Re: [PATCH] nfsd: remove hard-coded dereferences to name-to-id and id-to-name caches
Posted by bfields on Wed, 11 Apr 2012 22:07:07 GMT
View Forum Message <> Reply to Message

On Wed, Apr 11, 2012 at 03:18:58PM +0400, Stanislav Kinsbursky wrote:
> These dereferences to global static caches are redundant. They also prevents
> converting these caches into per-net ones. So this patch is cleanup + precursor
> of patch set, which will make them per-net.

This isn't applying to my latest tree; could you look into that and
resend?

I've applied and pushed out your cleanup patches, but not yet the two
cache-containerization patches.

--b.

>
> Signed-off-by: Stanislav Kinsbursky <skinsbursky@parallels.com>
>
> ---
> fs/nfsd/nfs4idmap.c | 50 ++++++++++++++++++++++++++++----------------------
> 1 files changed, 24 insertions(+), 26 deletions(-)
>
> diff --git a/fs/nfsd/nfs4idmap.c b/fs/nfsd/nfs4idmap.c
> index 322d11c..2ff4470 100644
> --- a/fs/nfsd/nfs4idmap.c

```
> +++ b/fs/nfsd/nfs4idmap.c
> @@ -183,8 +183,9 @@ warn_no_idmapd(struct cache_detail *detail, int has_died)
>
>
>  static int         idtoname_parse(struct cache_detail *, char *, int);
> -static struct ent *idtoname_lookup(struct ent *);
> -static struct ent *idtoname_update(struct ent *, struct ent *);
> +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 = {
>   .owner  = THIS_MODULE,
> @@ -244,7 +245,7 @@ idtoname_parse(struct cache_detail *cd, char *buf, int buflen)
>    goto out;
>
>   error = -ENOMEM;
> - res = idtoname_lookup(&ent);
> + res = idtoname_lookup(cd, &ent);
>   if (!res)
>    goto out;
>
> @@ -260,11 +261,11 @@ idtoname_parse(struct cache_detail *cd, char *buf, int buflen)
>   else
>    memcpy(ent.name, buf1, sizeof(ent.name));
>   error = -ENOMEM;
> - res = idtoname_update(&ent, res);
> + res = idtoname_update(cd, &ent, res);
>   if (res == NULL)
>    goto out;
>
> - cache_put(&res->h, &idtoname_cache);
> + cache_put(&res->h, cd);
>
>   error = 0;
>  out:
> @@ -275,10 +276,9 @@ out:
>
>
>  static struct ent *
> -idtoname_lookup(struct ent *item)
> +idtoname_lookup(struct cache_detail *cd, struct ent *item)
>  {
> - struct cache_head *ch = sunrpc_cache_lookup(&idtoname_cache,
> -         &item->h,
> + struct cache_head *ch = sunrpc_cache_lookup(cd, &item->h,
>           idtoname_hash(item));
>   if (ch)
```

>   return container_of(ch, struct ent, h);
> @@ -287,10 +287,9 @@ idtoname_lookup(struct ent *item)
> }
>
> static struct ent *
> -idtoname_update(struct ent *new, struct ent *old)
> +idtoname_update(struct cache_detail *cd, struct ent *new, struct ent *old)
> {
> - struct cache_head *ch = sunrpc_cache_update(&idtoname_cache,
> -   &new->h, &old->h,
> + struct cache_head *ch = sunrpc_cache_update(cd, &new->h, &old->h,
>   idtoname_hash(new));
>   if (ch)
>   return container_of(ch, struct ent, h);
> @@ -359,8 +358,9 @@ nametoid_show(struct seq_file *m, struct cache_detail *cd, struct
cache_head *h)
>   return 0;
> }
>
> -static struct ent *nametoid_lookup(struct ent *);
> -static struct ent *nametoid_update(struct ent *, struct ent *);
> +static struct ent *nametoid_lookup(struct cache_detail *, struct ent *);
> +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 = {
> @@ -426,14 +426,14 @@ nametoid_parse(struct cache_detail *cd, char *buf, int buflen)
>   set_bit(CACHE_NEGATIVE, &ent.h.flags);
>
>   error = -ENOMEM;
> - res = nametoid_lookup(&ent);
> + res = nametoid_lookup(cd, &ent);
>   if (res == NULL)
>   goto out;
> - res = nametoid_update(&ent, res);
> + res = nametoid_update(cd, &ent, res);
>   if (res == NULL)
>   goto out;
>
> - cache_put(&res->h, &nametoid_cache);
> + cache_put(&res->h, cd);
>   error = 0;
> out:
>   kfree(buf1);
> @@ -443,10 +443,9 @@ out:
>
>

```
>  static struct ent *
> -nametoid_lookup(struct ent *item)
> +nametoid_lookup(struct cache_detail *cd, struct ent *item)
>  {
> - struct cache_head *ch = sunrpc_cache_lookup(&nametoid_cache,
> -          &item->h,
> + struct cache_head *ch = sunrpc_cache_lookup(cd, &item->h,
>           nametoid_hash(item));
>   if (ch)
>    return container_of(ch, struct ent, h);
> @@ -455,10 +454,9 @@ nametoid_lookup(struct ent *item)
>  }
>
>  static struct ent *
> -nametoid_update(struct ent *new, struct ent *old)
> +nametoid_update(struct cache_detail *cd, struct ent *new, struct ent *old)
>  {
> - struct cache_head *ch = sunrpc_cache_update(&nametoid_cache,
> -          &new->h, &old->h,
> + struct cache_head *ch = sunrpc_cache_update(cd, &new->h, &old->h,
>           nametoid_hash(new));
>   if (ch)
>    return container_of(ch, struct ent, h);
> @@ -493,12 +491,12 @@ nfsd_idmap_shutdown(void)
>
>  static int
>  idmap_lookup(struct svc_rqst *rqstp,
> -  struct ent *(*lookup_fn)(struct ent *), struct ent *key,
> -  struct cache_detail *detail, struct ent **item)
> +  struct ent *(*lookup_fn)(struct cache_detail *, struct ent *),
> +  struct ent *key, struct cache_detail *detail, struct ent **item)
>  {
>   int ret;
>
> - *item = lookup_fn(key);
> + *item = lookup_fn(detail, key);
>   if (!*item)
>    return -ENOMEM;
>   retry:
> @@ -506,7 +504,7 @@ idmap_lookup(struct svc_rqst *rqstp,
>
>   if (ret == -ETIMEDOUT) {
>    struct ent *prev_item = *item;
> -  *item = lookup_fn(key);
> +  *item = lookup_fn(detail, key);
>    if (*item != prev_item)
>     goto retry;
>    cache_put(&(*item)->h, detail);
```

>