Subject: [PATCH] NFS: init client before declaration
Posted by Stanislav Kinsbursky on Tue, 22 May 2012 12:40:49 GMT
View Forum Message <> Reply to Message

Client have to be initialized prior to adding it to per-net clients list,
because otherwise there are races, shown below:

CPU#0 CPU#1

_____ _____

nfs_get_client
nfs_alloc_client
list_add(..., nfs_client_list)
rpc_fill_super
rpc_pipefs_event
nfs_get_client_for_event
__rpc_pipefs_event
(clp->cl_rpcclient is uninitialized)
BUG()
init_client
clp->cl_rpcclient = ...

Signed-off-by: Stanislav Kinsbursky <skinsbursky@parallels.com>
---
fs/nfs/client.c | 22 ++++++++++++----------
1 files changed, 12 insertions(+), 10 deletions(-)

diff --git a/fs/nfs/client.c b/fs/nfs/client.c
index ae29d4f..9bf4702 100644
--- a/fs/nfs/client.c
+++ b/fs/nfs/client.c
@@ -525,7 +525,7 @@ nfs_get_client(const struct nfs_client_initdata *cl_init,
cl_init->hostname ?: "", cl_init->rpc_ops->version);

/* see if the client already exists */
- do {
+ while (1) {
spin_lock(&nn->nfs_client_lock);

clp = nfs_match_client(cl_init);
@@ -537,10 +537,18 @@ nfs_get_client(const struct nfs_client_initdata *cl_init,
spin_unlock(&nn->nfs_client_lock);

new = nfs_alloc_client(cl_init);
- } while (!IS_ERR(new));
+ if (IS_ERR(new)) {
+ dprintk("--> nfs_get_client() = %ld [failed]\n", PTR_ERR(new));

```
+   return new;
+  }

- dprintk("--> nfs_get_client() = %ld [failed]\n", PTR_ERR(new));
- return new;
+  error = cl_init->rpc_ops->init_client(new, timeparms, ip_addr,
+           authflavour, noresvport);
+  if (error < 0) {
+   nfs_put_client(new);
+   return ERR_PTR(error);
+  }
+ }

  /* install a new client and return with it unready */
 install_client:
@@ -548,12 +556,6 @@ install_client:
  list_add(&clp->cl_share_link, &nn->nfs_client_list);
  spin_unlock(&nn->nfs_client_lock);

- error = cl_init->rpc_ops->init_client(clp, timeparms, ip_addr,
-          authflavour, noresvport);
- if (error < 0) {
-  nfs_put_client(clp);
-  return ERR_PTR(error);
- }
  dprintk("--> nfs_get_client() = %p [new]\n", clp);
  return clp;
```

Subject: Re: [PATCH] NFS: init client before declaration
Posted by Chuck Lever on Tue, 22 May 2012 13:47:51 GMT
View Forum Message <> Reply to Message

On May 22, 2012, at 8:40 AM, Stanislav Kinsbursky wrote:

> Client have to be initialized prior to adding it to per-net clients list,
> because otherwise there are races, shown below:
>
> CPU#0   CPU#1
> _____   _____
>
> nfs_get_client
> nfs_alloc_client
> list_add(..., nfs_client_list)
>   rpc_fill_super
>   rpc_pipefs_event
>   nfs_get_client_for_event
>   __rpc_pipefs_event

>   (clp->cl_rpcclient is uninitialized)
>   BUG()
> init_client
> clp->cl_rpcclient = ...
>
>
> Signed-off-by: Stanislav Kinsbursky <skinsbursky@parallels.com>

This patch collides pretty hard with the server trunking detection work. If you agree this needs to be fixed, the best thing we can do, I guess, is take this patch and drop patch 11, 12, and 13 from my recent patch set, and I'll try to rework for 3.6.

> ---
> fs/nfs/client.c | 22 ++++++++++++----------
> 1 files changed, 12 insertions(+), 10 deletions(-)
>
> diff --git a/fs/nfs/client.c b/fs/nfs/client.c
> index ae29d4f..9bf4702 100644
> --- a/fs/nfs/client.c
> +++ b/fs/nfs/client.c
> @@ -525,7 +525,7 @@ nfs_get_client(const struct nfs_client_initdata *cl_init,
>   cl_init->hostname ?: "", cl_init->rpc_ops->version);
>
> /* see if the client already exists */
> - do {
> + while (1) {
>   spin_lock(&nn->nfs_client_lock);
>
>   clp = nfs_match_client(cl_init);
> @@ -537,10 +537,18 @@ nfs_get_client(const struct nfs_client_initdata *cl_init,
>   spin_unlock(&nn->nfs_client_lock);
>
>   new = nfs_alloc_client(cl_init);
> - } while (!IS_ERR(new));
> + if (IS_ERR(new)) {
> +   dprintk("--> nfs_get_client() = %ld [failed]\n", PTR_ERR(new));
> +   return new;
> + }
>
> - dprintk("--> nfs_get_client() = %ld [failed]\n", PTR_ERR(new));
> - return new;
> + error = cl_init->rpc_ops->init_client(new, timeparms, ip_addr,
> +   authflavour, noresvport);
> + if (error < 0) {
> +   nfs_put_client(new);
> +   return ERR_PTR(error);
> + }
> + }

>
> /* install a new client and return with it unready */
> install_client:
> @@ -548,12 +556,6 @@ install_client:
> list_add(&clp->cl_share_link, &nn->nfs_client_list);
> spin_unlock(&nn->nfs_client_lock);
>
> - error = cl_init->rpc_ops->init_client(clp, timeparms, ip_addr,
> -   authflavour, noresvport);
> - if (error < 0) {
> - nfs_put_client(clp);
> - return ERR_PTR(error);
> - }
> dprintk("--> nfs_get_client() = %p [new]\n", clp);
> return clp;
>
>
> --
> To unsubscribe from this list: send the line "unsubscribe linux-nfs" in
> the body of a message to majordomo@vger.kernel.org
> More majordomo info at http://vger.kernel.org/majordomo-info.html

--
Chuck Lever
chuck[dot]lever[at]oracle[dot]com

Subject: Re: [PATCH] NFS: init client before declaration
Posted by Myklebust, Trond on Tue, 22 May 2012 14:29:44 GMT
View Forum Message <> Reply to Message

On Tue, 2012-05-22 at 16:40 +0400, Stanislav Kinsbursky wrote:
> Client have to be initialized prior to adding it to per-net clients list,
> because otherwise there are races, shown below:
>
> CPU#0   CPU#1
> _____   _____
>
> nfs_get_client
> nfs_alloc_client
> list_add(..., nfs_client_list)
>   rpc_fill_super
>   rpc_pipefs_event
>   nfs_get_client_for_event
>   __rpc_pipefs_event
>   (clp->cl_rpcclient is uninitialized)
>   BUG()
> init_client

> clp->cl_rpcclient = ...
>

Why not simply change nfs_get_client_for_event() so that it doesn't
touch nfs_clients that have clp->cl_cons_state!=NFS_CS_READY?

That should ensure that it doesn't touch nfs_clients that failed to
initialise and/or are still in the process of being initialised.

Cheers
  Trond

--
Trond Myklebust
Linux NFS client maintainer

NetApp
Trond.Myklebust@netapp.com
www.netapp.com

Subject: Re: [PATCH] NFS: init client before declaration
Posted by Myklebust, Trond on Tue, 22 May 2012 15:00:17 GMT
View Forum Message <> Reply to Message

On Tue, 2012-05-22 at 10:29 -0400, Trond Myklebust wrote:
> On Tue, 2012-05-22 at 16:40 +0400, Stanislav Kinsbursky wrote:
> > Client have to be initialized prior to adding it to per-net clients list,
> > because otherwise there are races, shown below:
> >
> > CPU#0   CPU#1
> > _____   _____
> >
> > nfs_get_client
> > nfs_alloc_client
> > list_add(..., nfs_client_list)
> >   rpc_fill_super
> >   rpc_pipefs_event
> >   nfs_get_client_for_event
> >   __rpc_pipefs_event
> >   (clp->cl_rpcclient is uninitialized)
> >   BUG()
> > init_client
> > clp->cl_rpcclient = ...
> >
>
> Why not simply change nfs_get_client_for_event() so that it doesn't
> touch nfs_clients that have clp->cl_cons_state!=NFS_CS_READY?

>
> That should ensure that it doesn't touch nfs_clients that failed to
> initialise and/or are still in the process of being initialised.

...actually, come to think of it. Why not just add a helper function
"bool nfs_client_active(const struct nfs_client *clp)" to
fs/nfs/client.c that does a call to
 wait_event_killable(nfs_client_active_wq, clp->cl_cons_state < NFS_CS_INITING);
and checks the resulting value of clp->cl_cons_state?

--
Trond Myklebust
Linux NFS client maintainer

NetApp
Trond.Myklebust@netapp.com
www.netapp.com

Subject: Re: [PATCH] NFS: init client before declaration
Posted by Stanislav Kinsbursky on Tue, 22 May 2012 15:03:37 GMT
View Forum Message <> Reply to Message

On 22.05.2012 18:29, Myklebust, Trond wrote:
> On Tue, 2012-05-22 at 16:40 +0400, Stanislav Kinsbursky wrote:
>> Client have to be initialized prior to adding it to per-net clients list,
>> because otherwise there are races, shown below:
>>
>> CPU#0   CPU#1
>> _____   _____
>>
>> nfs_get_client
>> nfs_alloc_client
>> list_add(..., nfs_client_list)
>>   rpc_fill_super
>>   rpc_pipefs_event
>>   nfs_get_client_for_event
>>   __rpc_pipefs_event
>>   (clp->cl_rpcclient is uninitialized)
>>   BUG()
>> init_client
>> clp->cl_rpcclient = ...
>>
>
> Why not simply change nfs_get_client_for_event() so that it doesn't
> touch nfs_clients that have clp->cl_cons_state!=NFS_CS_READY?
>
> That should ensure that it doesn't touch nfs_clients that failed to

> initialise and/or are still in the process of being initialised.
>

It looks like in this case we will have another races:

CPU#0 CPU#1

_____ _____

nfs4_init_client
nfs_idmap_new
nfs_idmap_register
rpc_get_sb_net (fail - no pipefs)
rpc_fill_super
rpc_pipefs_event
nfs_get_client_for_event
(skip client - NFS_CS_READY is not set)
nfs_mark_client_ready(NFS_CS_READY)

And we are having client without idmap pipe...

> Cheers
> Trond
>

--
Best regards,
Stanislav Kinsbursky

Subject: Re: [PATCH] NFS: init client before declaration
Posted by Stanislav Kinsbursky on Tue, 22 May 2012 15:29:31 GMT
View Forum Message <> Reply to Message

On 22.05.2012 19:00, Myklebust, Trond wrote:
> On Tue, 2012-05-22 at 10:29 -0400, Trond Myklebust wrote:
>> On Tue, 2012-05-22 at 16:40 +0400, Stanislav Kinsbursky wrote:
>>> Client have to be initialized prior to adding it to per-net clients list,
>>> because otherwise there are races, shown below:
>>>
>>> CPU#0 CPU#1
>>> _____ _____
>>>
>>> nfs_get_client
>>> nfs_alloc_client
>>> list_add(..., nfs_client_list)

>>>   rpc_fill_super
>>>   rpc_pipefs_event
>>>   nfs_get_client_for_event
>>>   __rpc_pipefs_event
>>>   (clp->cl_rpcclient is uninitialized)
>>>   BUG()
>>> init_client
>>> clp->cl_rpcclient = ...
>>>
>>
>> Why not simply change nfs_get_client_for_event() so that it doesn't
>> touch nfs_clients that have clp->cl_cons_state!=NFS_CS_READY?
>>
>> That should ensure that it doesn't touch nfs_clients that failed to
>> initialise and/or are still in the process of being initialised.
>
> ...actually, come to think of it. Why not just add a helper function
> "bool nfs_client_active(const struct nfs_client *clp)" to
> fs/nfs/client.c that does a call to
> wait_event_killable(nfs_client_active_wq, clp->cl_cons_state< NFS_CS_INITING);
> and checks the resulting value of clp->cl_cons_state?
>

Sorry, but I don't understand the idea...
Where are you proposing to call this function?
In __rpc_pipefs_event() prior to dentries creatios?

--
Best regards,
Stanislav Kinsbursky

Subject: Re: [PATCH] NFS: init client before declaration
Posted by Myklebust, Trond on Tue, 22 May 2012 15:51:44 GMT
View Forum Message <> Reply to Message

On Tue, 2012-05-22 at 19:29 +0400, Stanislav Kinsbursky wrote:
> On 22.05.2012 19:00, Myklebust, Trond wrote:
> > On Tue, 2012-05-22 at 10:29 -0400, Trond Myklebust wrote:
> >> On Tue, 2012-05-22 at 16:40 +0400, Stanislav Kinsbursky wrote:
> >>> Client have to be initialized prior to adding it to per-net clients list,
> >>> because otherwise there are races, shown below:
> >>>
> >>> CPU#0   CPU#1
> >>> _____   _____
> >>>
> >>> nfs_get_client

> >>> nfs_alloc_client
> >>> list_add(..., nfs_client_list)
> >>> rpc_fill_super
> >>> rpc_pipefs_event
> >>> nfs_get_client_for_event
> >>> __rpc_pipefs_event
> >>> (clp->cl_rpcclient is uninitialized)
> >>> BUG()
> >>> init_client
> >>> clp->cl_rpcclient = ...
> >>>
> >>
> >> Why not simply change nfs_get_client_for_event() so that it doesn't
> >> touch nfs_clients that have clp->cl_cons_state!=NFS_CS_READY?
> >>
> >> That should ensure that it doesn't touch nfs_clients that failed to
> >> initialise and/or are still in the process of being initialised.
> >
> > ...actually, come to think of it. Why not just add a helper function
> > "bool nfs_client_active(const struct nfs_client *clp)" to
> > fs/nfs/client.c that does a call to
> > wait_event_killable(nfs_client_active_wq, clp->cl_cons_state< NFS_CS_INITING);
> > and checks the resulting value of clp->cl_cons_state?
> >
>
> Sorry, but I don't understand the idea...
> Where are you proposing to call this function?
> In __rpc_pipefs_event() prior to dentries creatios?

See below:

8< ---------- ----------
>From f5b90df6381a20395d9f88a199e9e52f44267457 Mon Sep 17 00:00:00 2001
From: Trond Myklebust <Trond.Myklebust@netapp.com>
Date: Tue, 22 May 2012 11:49:55 -0400
Subject: [PATCH] NFSv4: Fix a race in the net namespace mount notification

Since the struct nfs_client gets added to the global nfs_client_list
before it is initialised, it is possible that rpc_pipefs_event can
end up trying to create idmapper entries for such a thing.

The solution is to have the mount notification wait for the
nfs_client initialisation to complete.

Reported-by: Stanislav Kinsbursky <skinsbursky@parallels.com>
Signed-off-by: Trond Myklebust <Trond.Myklebust@netapp.com>
---
 fs/nfs/client.c | 14 ++++++++++++++

```
 fs/nfs/idmap.c    |    3 ++-
 fs/nfs/internal.h |    1 +
 3 files changed, 17 insertions(+), 1 deletions(-)

diff --git a/fs/nfs/client.c b/fs/nfs/client.c
index 60f7e4e..3fa44ef 100644
--- a/fs/nfs/client.c
+++ b/fs/nfs/client.c
@@ -592,6 +592,20 @@ void nfs_mark_client_ready(struct nfs_client *clp, int state)
  wake_up_all(&nfs_client_active_wq);
 }

+static bool nfs_client_ready(struct nfs_client *clp)
+{
+ return clp->cl_cons_state <= NFS_CS_READY;
+}
+
+int nfs_wait_client_ready(struct nfs_client *clp)
+{
+ if (wait_event_killable(nfs_client_active_wq, nfs_client_ready(clp)) < 0)
+  return -ERESTARTSYS;
+ if (clp->cl_cons_state < 0)
+  return clp->cl_cons_state;
+ return 0;
+}
+
 /*
  * With sessions, the client is not marked ready until after a
  * successful EXCHANGE_ID and CREATE_SESSION.
diff --git a/fs/nfs/idmap.c b/fs/nfs/idmap.c
index 3e8edbe..67962c8 100644
--- a/fs/nfs/idmap.c
+++ b/fs/nfs/idmap.c
@@ -558,7 +558,8 @@ static int rpc_pipefs_event(struct notifier_block *nb, unsigned long event,
   return 0;

  while ((clp = nfs_get_client_for_event(sb->s_fs_info, event))) {
-  error = __rpc_pipefs_event(clp, event, sb);
+  if (nfs_wait_client_ready(clp) == 0)
+   error = __rpc_pipefs_event(clp, event, sb);
   nfs_put_client(clp);
   if (error)
    break;
diff --git a/fs/nfs/internal.h b/fs/nfs/internal.h
index b777bda..3be00a0 100644
--- a/fs/nfs/internal.h
+++ b/fs/nfs/internal.h
@@ -168,6 +168,7 @@ extern struct nfs_server *nfs_clone_server(struct nfs_server *,
```

```
        struct nfs_fattr *,
        rpc_authflavor_t);
 extern void nfs_mark_client_ready(struct nfs_client *clp, int state);
+extern int nfs_wait_client_ready(struct nfs_client *clp);
 extern int nfs4_check_client_ready(struct nfs_client *clp);
 extern struct nfs_client *nfs4_set_ds_client(struct nfs_client* mds_clp,
         const struct sockaddr *ds_addr,
--
1.7.7.6
```

--
Trond Myklebust
Linux NFS client maintainer

NetApp
Trond.Myklebust@netapp.com
www.netapp.com

Subject: Re: [PATCH] NFS: init client before declaration
Posted by Stanislav Kinsbursky on Tue, 22 May 2012 16:18:20 GMT
View Forum Message <> Reply to Message

On 22.05.2012 19:51, Myklebust, Trond wrote:
> On Tue, 2012-05-22 at 19:29 +0400, Stanislav Kinsbursky wrote:
>> On 22.05.2012 19:00, Myklebust, Trond wrote:
>>> On Tue, 2012-05-22 at 10:29 -0400, Trond Myklebust wrote:
>>>> On Tue, 2012-05-22 at 16:40 +0400, Stanislav Kinsbursky wrote:
>>>>> Client have to be initialized prior to adding it to per-net clients list,
>>>>> because otherwise there are races, shown below:
>>>>>
>>>>> CPU#0   CPU#1
>>>>> _____   _____
>>>>>
>>>>> nfs_get_client
>>>>> nfs_alloc_client
>>>>> list_add(..., nfs_client_list)
>>>>>   rpc_fill_super
>>>>>   rpc_pipefs_event
>>>>>   nfs_get_client_for_event
>>>>>   __rpc_pipefs_event
>>>>>   (clp->cl_rpcclient is uninitialized)
>>>>>   BUG()
>>>>> init_client
>>>>> clp->cl_rpcclient = ...
>>>>>
>>>>
```
```

>>>> Why not simply change nfs_get_client_for_event() so that it doesn't
>>>> touch nfs_clients that have clp->cl_cons_state!=NFS_CS_READY?
>>>>
>>>> That should ensure that it doesn't touch nfs_clients that failed to
>>>> initialise and/or are still in the process of being initialised.
>>>
>>> ...actually, come to think of it. Why not just add a helper function
>>> "bool nfs_client_active(const struct nfs_client *clp)" to
>>> fs/nfs/client.c that does a call to
>>> wait_event_killable(nfs_client_active_wq, clp->cl_cons_state<   NFS_CS_INITING);
>>> and checks the resulting value of clp->cl_cons_state?
>>>
>>
>> Sorry, but I don't understand the idea...
>> Where are you proposing to call this function?
>> In __rpc_pipefs_event() prior to dentries creatios?
>
> See below:
>
> 8< ---------- ----------
> From f5b90df6381a20395d9f88a199e9e52f44267457 Mon Sep 17 00:00:00 2001
> From: Trond Myklebust<Trond.Myklebust@netapp.com>
> Date: Tue, 22 May 2012 11:49:55 -0400
> Subject: [PATCH] NFSv4: Fix a race in the net namespace mount notification
>
> Since the struct nfs_client gets added to the global nfs_client_list
> before it is initialised, it is possible that rpc_pipefs_event can
> end up trying to create idmapper entries for such a thing.
>
> The solution is to have the mount notification wait for the
> nfs_client initialisation to complete.
>
> Reported-by: Stanislav Kinsbursky<skinsbursky@parallels.com>
> Signed-off-by: Trond Myklebust<Trond.Myklebust@netapp.com>
> ---
> fs/nfs/client.c | 14 ++++++++++++++
> fs/nfs/idmap.c | 3 ++-
> fs/nfs/internal.h | 1 +
> 3 files changed, 17 insertions(+), 1 deletions(-)
>
> diff --git a/fs/nfs/client.c b/fs/nfs/client.c
> index 60f7e4e..3fa44ef 100644
> --- a/fs/nfs/client.c
> +++ b/fs/nfs/client.c
> @@ -592,6 +592,20 @@ void nfs_mark_client_ready(struct nfs_client *clp, int state)
>   wake_up_all(&nfs_client_active_wq);
> }
>

> +static bool nfs_client_ready(struct nfs_client *clp)
> +{
> + return clp->cl_cons_state<= NFS_CS_READY;
> +}
> +
> +int nfs_wait_client_ready(struct nfs_client *clp)
> +{
> + if (wait_event_killable(nfs_client_active_wq, nfs_client_ready(clp))< 0)
> + return -ERESTARTSYS;

Ok, I see...
BTW, caller of this function is pipefs mount operation call... And when this
mount call waits for NFS clients - it look a bit odd to me...

> + if (clp->cl_cons_state< 0)
> + return clp->cl_cons_state;
> + return 0;
> +}
> +
>   /*
>   * With sessions, the client is not marked ready until after a
>   * successful EXCHANGE_ID and CREATE_SESSION.
> diff --git a/fs/nfs/idmap.c b/fs/nfs/idmap.c
> index 3e8edbe..67962c8 100644
> --- a/fs/nfs/idmap.c
> +++ b/fs/nfs/idmap.c
> @@ -558,7 +558,8 @@ static int rpc_pipefs_event(struct notifier_block *nb, unsigned long
event,
>   return 0;
>
>   while ((clp = nfs_get_client_for_event(sb->s_fs_info, event))) {
> - error = __rpc_pipefs_event(clp, event, sb);
> + if (nfs_wait_client_ready(clp) == 0)
> +   error = __rpc_pipefs_event(clp, event, sb);

We have another problem here.
nfs4_init_client() will try to create pipe dentries prior to set of NFS_CS_READY
to the client. And dentries will be created since semaphore is dropped and
per-net superblock variable is initialized already.
But __rpc_pipefs_event() relays on the fact, that no dentries present.
Looks like the problem was introduced by me in aad9487c...
So maybe we should not call "continue" instead "__rpc_pipefs_event()", when
client becomes ready?
Looks like this will allow us to handle such races.

>   nfs_put_client(clp);
>   if (error)
>   break;
> diff --git a/fs/nfs/internal.h b/fs/nfs/internal.h
> index b777bda..3be00a0 100644
> --- a/fs/nfs/internal.h
> +++ b/fs/nfs/internal.h
> @@ -168,6 +168,7 @@ extern struct nfs_server *nfs_clone_server(struct nfs_server *,
>   struct nfs_fattr *,
>   rpc_authflavor_t);
>   extern void nfs_mark_client_ready(struct nfs_client *clp, int state);
> +extern int nfs_wait_client_ready(struct nfs_client *clp);
>   extern int nfs4_check_client_ready(struct nfs_client *clp);
>   extern struct nfs_client *nfs4_set_ds_client(struct nfs_client* mds_clp,
>   const struct sockaddr *ds_addr,

--
Best regards,
Stanislav Kinsbursky

Subject: Re: [PATCH] NFS: init client before declaration
Posted by Myklebust, Trond on Tue, 22 May 2012 16:43:23 GMT
View Forum Message <> Reply to Message

On Tue, 2012-05-22 at 20:18 +0400, Stanislav Kinsbursky wrote:
> On 22.05.2012 19:51, Myklebust, Trond wrote:
> > On Tue, 2012-05-22 at 19:29 +0400, Stanislav Kinsbursky wrote:
> >> On 22.05.2012 19:00, Myklebust, Trond wrote:
> >>> On Tue, 2012-05-22 at 10:29 -0400, Trond Myklebust wrote:
> >>>> On Tue, 2012-05-22 at 16:40 +0400, Stanislav Kinsbursky wrote:
> >>>>> Client have to be initialized prior to adding it to per-net clients list,
> >>>>> because otherwise there are races, shown below:
> >>>>>
> >>>>> CPU#0   CPU#1
> >>>>> _____   _____
> >>>>>
> >>>>> nfs_get_client
> >>>>> nfs_alloc_client
> >>>>> list_add(..., nfs_client_list)
> >>>>>   rpc_fill_super
> >>>>>   rpc_pipefs_event
> >>>>>   nfs_get_client_for_event
> >>>>>   __rpc_pipefs_event
> >>>>>   (clp->cl_rpcclient is uninitialized)
> >>>>>   BUG()
> >>>>> init_client

> >>>>> clp->cl_rpcclient = ...
> >>>>>
> >>>>
> >>>> Why not simply change nfs_get_client_for_event() so that it doesn't
> >>>> touch nfs_clients that have clp->cl_cons_state!=NFS_CS_READY?
> >>>>
> >>>> That should ensure that it doesn't touch nfs_clients that failed to
> >>>> initialise and/or are still in the process of being initialised.
> >>>
> >>> ...actually, come to think of it. Why not just add a helper function
> >>> "bool nfs_client_active(const struct nfs_client *clp)" to
> >>> fs/nfs/client.c that does a call to
> >>> wait_event_killable(nfs_client_active_wq, clp->cl_cons_state<   NFS_CS_INITING);
> >>> and checks the resulting value of clp->cl_cons_state?
> >>>
> >>
> >> Sorry, but I don't understand the idea...
> >> Where are you proposing to call this function?
> >> In __rpc_pipefs_event() prior to dentries creatios?
> >
> > See below:
> >
> > 8< ---------- ----------
> > From f5b90df6381a20395d9f88a199e9e52f44267457 Mon Sep 17 00:00:00 2001
> > From: Trond Myklebust<Trond.Myklebust@netapp.com>
> > Date: Tue, 22 May 2012 11:49:55 -0400
> > Subject: [PATCH] NFSv4: Fix a race in the net namespace mount notification
> >
> > Since the struct nfs_client gets added to the global nfs_client_list
> > before it is initialised, it is possible that rpc_pipefs_event can
> > end up trying to create idmapper entries for such a thing.
> >
> > The solution is to have the mount notification wait for the
> > nfs_client initialisation to complete.
> >
> > Reported-by: Stanislav Kinsbursky<skinsbursky@parallels.com>
> > Signed-off-by: Trond Myklebust<Trond.Myklebust@netapp.com>
> > ---
> >   fs/nfs/client.c | 14 ++++++++++++++
> >   fs/nfs/idmap.c | 3 ++-
> >   fs/nfs/internal.h | 1 +
> >   3 files changed, 17 insertions(+), 1 deletions(-)
> >
> > diff --git a/fs/nfs/client.c b/fs/nfs/client.c
> > index 60f7e4e..3fa44ef 100644
> > --- a/fs/nfs/client.c
> > +++ b/fs/nfs/client.c
> > @@ -592,6 +592,20 @@ void nfs_mark_client_ready(struct nfs_client *clp, int state)

> >   wake_up_all(&nfs_client_active_wq);
> >   }
> >
> > +static bool nfs_client_ready(struct nfs_client *clp)
> > +{
> > + return clp->cl_cons_state<= NFS_CS_READY;
> > +}
> > +
> > +int nfs_wait_client_ready(struct nfs_client *clp)
> > +{
> > + if (wait_event_killable(nfs_client_active_wq, nfs_client_ready(clp))< 0)
> > + return -ERESTARTSYS;
>
> Ok, I see...
> BTW, caller of this function is pipefs mount operation call... And when this
> mount call waits for NFS clients - it look a bit odd to me...
>
>
> > + if (clp->cl_cons_state< 0)
> > + return clp->cl_cons_state;
> > + return 0;
> > +}
> > +
> >   /*
> >   * With sessions, the client is not marked ready until after a
> >   * successful EXCHANGE_ID and CREATE_SESSION.
> > diff --git a/fs/nfs/idmap.c b/fs/nfs/idmap.c
> > index 3e8edbe..67962c8 100644
> > --- a/fs/nfs/idmap.c
> > +++ b/fs/nfs/idmap.c
> > @@ -558,7 +558,8 @@ static int rpc_pipefs_event(struct notifier_block *nb, unsigned long
event,
> >   return 0;
> >
> >   while ((clp = nfs_get_client_for_event(sb->s_fs_info, event))) {
> > - error = __rpc_pipefs_event(clp, event, sb);
> > + if (nfs_wait_client_ready(clp) == 0)
> > +   error = __rpc_pipefs_event(clp, event, sb);
>
>
> We have another problem here.
> nfs4_init_client() will try to create pipe dentries prior to set of NFS_CS_READY
> to the client. And dentries will be created since semaphore is dropped and
> per-net superblock variable is initialized already.
> But __rpc_pipefs_event() relays on the fact, that no dentries present.
> Looks like the problem was introduced by me in aad9487c...
> So maybe we should not call "continue" instead "__rpc_pipefs_event()", when
> client becomes ready?

> Looks like this will allow us to handle such races.

Let me rework this patch a bit...

--
Trond Myklebust
Linux NFS client maintainer

NetApp
Trond.Myklebust@netapp.com
www.netapp.com

Subject: Re: [PATCH] NFS: init client before declaration
Posted by Myklebust, Trond on Tue, 22 May 2012 20:32:51 GMT
View Forum Message <> Reply to Message

On Tue, 2012-05-22 at 12:43 -0400, Trond Myklebust wrote:
> On Tue, 2012-05-22 at 20:18 +0400, Stanislav Kinsbursky wrote:
> > We have another problem here.
> > nfs4_init_client() will try to create pipe dentries prior to set of NFS_CS_READY
> > to the client. And dentries will be created since semaphore is dropped and
> > per-net superblock variable is initialized already.
> > But __rpc_pipefs_event() relays on the fact, that no dentries present.
> > Looks like the problem was introduced by me in aad9487c...
> > So maybe we should not call "continue" instead "__rpc_pipefs_event()", when
> > client becomes ready?
> > Looks like this will allow us to handle such races.
>
> Let me rework this patch a bit...

The following is ugly, but it should be demonstrably correct, and will
ensure that __rpc_pipefs_event() will only be called for fully
initialised nfs_clients...

8< ---------- ---------
>From 90c3b9fe9faeae32c8f629e8b6cbf5f50bb9b295 Mon Sep 17 00:00:00 2001
From: Trond Myklebust <Trond.Myklebust@netapp.com>
Date: Tue, 22 May 2012 16:22:50 -0400
Subject: [PATCH 1/2] NFSv4: Fix a race in the net namespace mount
 notification

Since the struct nfs_client gets added to the global nfs_client_list
before it is initialised, it is possible that rpc_pipefs_event can
end up trying to create idmapper entries on such a thing.

The solution is to have the mount notification wait for the
initialisation of each nfs_client to complete, and then to

skip any entries for which the it failed.

Reported-by: Stanislav Kinsbursky <skinsbursky@parallels.com>
Signed-off-by: Trond Myklebust <Trond.Myklebust@netapp.com>
---
 fs/nfs/client.c | 10 ++++++++++
 fs/nfs/idmap.c | 15 +++++++++++++++
 fs/nfs/internal.h | 1 +
 3 files changed, 26 insertions(+), 0 deletions(-)

diff --git a/fs/nfs/client.c b/fs/nfs/client.c
index 60f7e4e..d3c8553 100644
--- a/fs/nfs/client.c
+++ b/fs/nfs/client.c
@@ -583,6 +583,16 @@ found_client:
  return clp;
 }

+static bool nfs_client_ready(const struct nfs_client *clp)
+{
+ return clp->cl_cons_state <= NFS_CS_READY;
+}
+
+int nfs_wait_client_ready(const struct nfs_client *clp)
+{
+ return wait_event_killable(nfs_client_active_wq, nfs_client_ready(clp));
+}
+
 /*
  * Mark a server as ready or failed
  */
diff --git a/fs/nfs/idmap.c b/fs/nfs/idmap.c
index 3e8edbe..c0753c5 100644
--- a/fs/nfs/idmap.c
+++ b/fs/nfs/idmap.c
@@ -530,9 +530,24 @@ static struct nfs_client *nfs_get_client_for_event(struct net *net, int
event)
  struct nfs_net *nn = net_generic(net, nfs_net_id);
  struct dentry *cl_dentry;
  struct nfs_client *clp;
+ int err;

+restart:
  spin_lock(&nn->nfs_client_lock);
  list_for_each_entry(clp, &nn->nfs_client_list, cl_share_link) {
+ /* Wait for initialisation to finish */
+ if (clp->cl_cons_state > NFS_CS_READY) {
+   atomic_inc(&clp->cl_count);

```
+   spin_unlock(&nn->nfs_client_lock);
+   err = nfs_wait_client_ready(clp);
+   nfs_put_client(clp);
+   if (err)
+    return NULL;
+   goto restart;
+  }
+  /* Skip nfs_clients that failed to initialise */
+  if (clp->cl_cons_state < 0)
+   continue;
   if (clp->rpc_ops != &nfs_v4_clientops)
    continue;
   cl_dentry = clp->cl_idmap->idmap_pipe->dentry;
diff --git a/fs/nfs/internal.h b/fs/nfs/internal.h
index b777bda..3ee4040 100644
--- a/fs/nfs/internal.h
+++ b/fs/nfs/internal.h
@@ -168,6 +168,7 @@ extern struct nfs_server *nfs_clone_server(struct nfs_server *,
        struct nfs_fattr *,
        rpc_authflavor_t);
 extern void nfs_mark_client_ready(struct nfs_client *clp, int state);
+extern int nfs_wait_client_ready(const struct nfs_client *clp);
 extern int nfs4_check_client_ready(struct nfs_client *clp);
 extern struct nfs_client *nfs4_set_ds_client(struct nfs_client* mds_clp,
          const struct sockaddr *ds_addr,
--
1.7.7.6
```

--
Trond Myklebust
Linux NFS client maintainer

NetApp
Trond.Myklebust@netapp.com
www.netapp.com

Subject: Re: [PATCH] NFS: init client before declaration
Posted by Stanislav Kinsbursky on Wed, 23 May 2012 11:30:23 GMT
View Forum Message <> Reply to Message

On 23.05.2012 00:32, Myklebust, Trond wrote:
> On Tue, 2012-05-22 at 12:43 -0400, Trond Myklebust wrote:
>> On Tue, 2012-05-22 at 20:18 +0400, Stanislav Kinsbursky wrote:
>>> We have another problem here.
>>> nfs4_init_client() will try to create pipe dentries prior to set of NFS_CS_READY
>>> to the client. And dentries will be created since semaphore is dropped and

>>> per-net superblock variable is initialized already.
>>> But __rpc_pipefs_event() relays on the fact, that no dentries present.
>>> Looks like the problem was introduced by me in aad9487c...
>>> So maybe we should not call "continue" instead "__rpc_pipefs_event()", when
>>> client becomes ready?
>>> Looks like this will allow us to handle such races.
>>
>> Let me rework this patch a bit...
>
> The following is ugly, but it should be demonstrably correct, and will
> ensure that __rpc_pipefs_event() will only be called for fully
> initialised nfs_clients...
>
> 8< -- ---------
> From 90c3b9fe9faeae32c8f629e8b6cbf5f50bb9b295 Mon Sep 17 00:00:00 2001
> From: Trond Myklebust<Trond.Myklebust@netapp.com>
> Date: Tue, 22 May 2012 16:22:50 -0400
> Subject: [PATCH 1/2] NFSv4: Fix a race in the net namespace mount
>   notification
>
> Since the struct nfs_client gets added to the global nfs_client_list
> before it is initialised, it is possible that rpc_pipefs_event can
> end up trying to create idmapper entries on such a thing.
>
> The solution is to have the mount notification wait for the
> initialisation of each nfs_client to complete, and then to
> skip any entries for which the it failed.
>
> Reported-by: Stanislav Kinsbursky<skinsbursky@parallels.com>
> Signed-off-by: Trond Myklebust<Trond.Myklebust@netapp.com>
> ---
>   fs/nfs/client.c | 10 ++++++++++
>   fs/nfs/idmap.c | 15 +++++++++++++++
>   fs/nfs/internal.h | 1 +
>   3 files changed, 26 insertions(+), 0 deletions(-)
>
> diff --git a/fs/nfs/client.c b/fs/nfs/client.c
> index 60f7e4e..d3c8553 100644
> --- a/fs/nfs/client.c
> +++ b/fs/nfs/client.c
> @@ -583,6 +583,16 @@ found_client:
>   return clp;
>   }
>
> +static bool nfs_client_ready(const struct nfs_client *clp)
> +{
> + return clp->cl_cons_state<= NFS_CS_READY;
> +}

```
> +
> +int nfs_wait_client_ready(const struct nfs_client *clp)
> +{
> + return wait_event_killable(nfs_client_active_wq, nfs_client_ready(clp));
> +}
> +
>   /*
>    * Mark a server as ready or failed
>    */
> diff --git a/fs/nfs/idmap.c b/fs/nfs/idmap.c
> index 3e8edbe..c0753c5 100644
> --- a/fs/nfs/idmap.c
> +++ b/fs/nfs/idmap.c
> @@ -530,9 +530,24 @@ static struct nfs_client *nfs_get_client_for_event(struct net *net, int
event)
>    struct nfs_net *nn = net_generic(net, nfs_net_id);
>    struct dentry *cl_dentry;
>    struct nfs_client *clp;
> + int err;
>
> +restart:
>    spin_lock(&nn->nfs_client_lock);
>    list_for_each_entry(clp,&nn->nfs_client_list, cl_share_link) {
> +  /* Wait for initialisation to finish */
> +  if (clp->cl_cons_state>  NFS_CS_READY) {
> +   atomic_inc(&clp->cl_count);
> +   spin_unlock(&nn->nfs_client_lock);
> +   err = nfs_wait_client_ready(clp);
```

What about NFSv4.1 ?
It's clients NFS_CS_READY status depends on session establishing RPC calls...
Which in turn can hung up pipefs mount call...

Moreover, looks like pipefs dentries creation have to be synchronized by
nfs_client_lock somehow... Otherwise because of races we can get a client
without pipe dentry....

```
> +   nfs_put_client(clp);
> +   if (err)
> +    return NULL;
> +   goto restart;
> +  }
> +  /* Skip nfs_clients that failed to initialise */
> +  if (clp->cl_cons_state<  0)
> +   continue;
>     if (clp->rpc_ops !=&nfs_v4_clientops)
>      continue;
```

>   cl_dentry = clp->cl_idmap->idmap_pipe->dentry;
> diff --git a/fs/nfs/internal.h b/fs/nfs/internal.h
> index b777bda..3ee4040 100644
> --- a/fs/nfs/internal.h
> +++ b/fs/nfs/internal.h
> @@ -168,6 +168,7 @@ extern struct nfs_server *nfs_clone_server(struct nfs_server *,
>   struct nfs_fattr *,
>   rpc_authflavor_t);
>   extern void nfs_mark_client_ready(struct nfs_client *clp, int state);
> +extern int nfs_wait_client_ready(const struct nfs_client *clp);
>   extern int nfs4_check_client_ready(struct nfs_client *clp);
>   extern struct nfs_client *nfs4_set_ds_client(struct nfs_client* mds_clp,
>   const struct sockaddr *ds_addr,

--
Best regards,
Stanislav Kinsbursky

Subject: Re: [PATCH] NFS: init client before declaration
Posted by Kinsbursky Stanislav on Wed, 23 May 2012 12:09:01 GMT
View Forum Message <> Reply to Message

On 23.05.2012 15:30, Stanislav Kinsbursky wrote:
> Moreover, looks like pipefs dentries creation have to be synchronized by
> nfs_client_lock somehow... Otherwise because of races we can get a client
> without pipe dentry....

Taking this back.

--
Best regards,
Stanislav Kinsbursky

Subject: Re: [PATCH] NFS: init client before declaration
Posted by Myklebust, Trond on Wed, 23 May 2012 13:57:44 GMT
View Forum Message <> Reply to Message

On Wed, 2012-05-23 at 15:30 +0400, Stanislav Kinsbursky wrote:
> On 23.05.2012 00:32, Myklebust, Trond wrote:
> > On Tue, 2012-05-22 at 12:43 -0400, Trond Myklebust wrote:
> >> On Tue, 2012-05-22 at 20:18 +0400, Stanislav Kinsbursky wrote:
> >>> We have another problem here.
> >>> nfs4_init_client() will try to create pipe dentries prior to set of NFS_CS_READY
> >>> to the client. And dentries will be created since semaphore is dropped and

> >>> per-net superblock variable is initialized already.
> >>> But __rpc_pipefs_event() relays on the fact, that no dentries present.
> >>> Looks like the problem was introduced by me in aad9487c...
> >>> So maybe we should not call "continue" instead "__rpc_pipefs_event()", when
> >>> client becomes ready?
> >>> Looks like this will allow us to handle such races.
> >>
> >> Let me rework this patch a bit...
> >
> > The following is ugly, but it should be demonstrably correct, and will
> > ensure that __rpc_pipefs_event() will only be called for fully
> > initialised nfs_clients...
> >
> > 8< ---------- ---------
> > From 90c3b9fe9faeae32c8f629e8b6cbf5f50bb9b295 Mon Sep 17 00:00:00 2001
> > From: Trond Myklebust<Trond.Myklebust@netapp.com>
> > Date: Tue, 22 May 2012 16:22:50 -0400
> > Subject: [PATCH 1/2] NFSv4: Fix a race in the net namespace mount
> >   notification
> >
> > Since the struct nfs_client gets added to the global nfs_client_list
> > before it is initialised, it is possible that rpc_pipefs_event can
> > end up trying to create idmapper entries on such a thing.
> >
> > The solution is to have the mount notification wait for the
> > initialisation of each nfs_client to complete, and then to
> > skip any entries for which the it failed.
> >
> > Reported-by: Stanislav Kinsbursky<skinsbursky@parallels.com>
> > Signed-off-by: Trond Myklebust<Trond.Myklebust@netapp.com>
> > ---
> >   fs/nfs/client.c | 10 ++++++++++
> >   fs/nfs/idmap.c | 15 +++++++++++++++
> >   fs/nfs/internal.h | 1 +
> >   3 files changed, 26 insertions(+), 0 deletions(-)
> >
> > diff --git a/fs/nfs/client.c b/fs/nfs/client.c
> > index 60f7e4e..d3c8553 100644
> > --- a/fs/nfs/client.c
> > +++ b/fs/nfs/client.c
> > @@ -583,6 +583,16 @@ found_client:
> >   return clp;
> >   }
> >
> > +static bool nfs_client_ready(const struct nfs_client *clp)
> > +{
> > + return clp->cl_cons_state<= NFS_CS_READY;
> > +}

> > +
> > +int nfs_wait_client_ready(const struct nfs_client *clp)
> > +{
> > + return wait_event_killable(nfs_client_active_wq, nfs_client_ready(clp));
> > +}
> > +
> > /*
> > * Mark a server as ready or failed
> > */
> > diff --git a/fs/nfs/idmap.c b/fs/nfs/idmap.c
> > index 3e8edbe..c0753c5 100644
> > --- a/fs/nfs/idmap.c
> > +++ b/fs/nfs/idmap.c
> > @@ -530,9 +530,24 @@ static struct nfs_client *nfs_get_client_for_event(struct net *net, int
event)
> > struct nfs_net *nn = net_generic(net, nfs_net_id);
> > struct dentry *cl_dentry;
> > struct nfs_client *clp;
> > + int err;
> >
> > +restart:
> > spin_lock(&nn->nfs_client_lock);
> > list_for_each_entry(clp,&nn->nfs_client_list, cl_share_link) {
> > + /* Wait for initialisation to finish */
> > + if (clp->cl_cons_state> NFS_CS_READY) {
> > + atomic_inc(&clp->cl_count);
> > + spin_unlock(&nn->nfs_client_lock);
> > + err = nfs_wait_client_ready(clp);
>
>
> What about NFSv4.1 ?
> It's clients NFS_CS_READY status depends on session establishing RPC calls...
> Which in turn can hung up pipefs mount call...

The alternative then is to wait for cl_cons_state != NFS_CS_INITING.
That shouldn't require any upcalls, and so shouldn't be able to
deadlock.

--
Trond Myklebust
Linux NFS client maintainer

NetApp
Trond.Myklebust@netapp.com
www.netapp.com