Subject: [PATCH v2] SUNRPC: skip dead but not buried clients on PipeFS events
Posted by Stanislav Kinsbursky on Fri, 20 Apr 2012 14:11:02 GMT

View Forum Message <> Reply to Message

v2: atomic_inc_return() was replaced by atomic_inc_not_zero().

These clients can't be safely dereferenced if their counter in 0.

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

```
---
 net/sunrpc/clnt.c |    3 ++-
 1 files changed, 2 insertions(+), 1 deletions(-)

diff --git a/net/sunrpc/clnt.c b/net/sunrpc/clnt.c
index 6797246..d10ebc4 100644
--- a/net/sunrpc/clnt.c
+++ b/net/sunrpc/clnt.c
@@ -218,7 +218,8 @@ static struct rpc_clnt *rpc_get_client_for_event(struct net *net, int event)
   if (((event == RPC_PIPEFS_MOUNT) && clnt->cl_dentry) ||
       ((event == RPC_PIPEFS_UMOUNT) && !clnt->cl_dentry))
    continue;
-  atomic_inc(&clnt->cl_count);
+  if (atomic_inc_not_zero(&clnt->cl_count) == 0)
+   continue;
   spin_unlock(&sn->rpc_client_lock);
   return clnt;
  }
```

Subject: Re: [PATCH v2] SUNRPC: skip dead but not buried clients on PipeFS events
Posted by bfields on Wed, 25 Apr 2012 17:30:05 GMT

View Forum Message <> Reply to Message

On Fri, Apr 20, 2012 at 06:11:02PM +0400, Stanislav Kinsbursky wrote:

> v2: atomic_inc_return() was replaced by atomic_inc_not_zero().
>
> These clients can't be safely dereferenced if their counter in 0.

I'm pretty confused by how these notifiers work....

rpc_release_client decrements cl_count to zero temporarily, to have it
immediately re-incremented by rpc_free_auth.

So if we're called concurrently with rpc_release_client then it's sort
of random whether someone gets this callback.

Is that a problem?

Also, is this an existing bug? (In which case Trond should take it
now.)

--b.

```
>
> Signed-off-by: Stanislav Kinsbursky <skinsbursky@parallels.com>
>
> ---
>  net/sunrpc/clnt.c |    3 ++-
>  1 files changed, 2 insertions(+), 1 deletions(-)
>
> diff --git a/net/sunrpc/clnt.c b/net/sunrpc/clnt.c
> index 6797246..d10ebc4 100644
> --- a/net/sunrpc/clnt.c
> +++ b/net/sunrpc/clnt.c
> @@ -218,7 +218,8 @@ static struct rpc_clnt *rpc_get_client_for_event(struct net *net, int
event)
>    if (((event == RPC_PIPEFS_MOUNT) && clnt->cl_dentry) ||
>        ((event == RPC_PIPEFS_UMOUNT) && !clnt->cl_dentry))
>     continue;
> -  atomic_inc(&clnt->cl_count);
> +  if (atomic_inc_not_zero(&clnt->cl_count) == 0)
> +   continue;
>    spin_unlock(&sn->rpc_client_lock);
>    return clnt;
>   }
>
```

Subject: Re: [PATCH v2] SUNRPC: skip dead but not buried clients on PipeFS events
Posted by Myklebust, Trond on Wed, 25 Apr 2012 18:54:55 GMT

View Forum Message <> Reply to Message

On Wed, 2012-04-25 at 13:30 -0400, J. Bruce Fields wrote:
> On Fri, Apr 20, 2012 at 06:11:02PM +0400, Stanislav Kinsbursky wrote:
> > v2: atomic_inc_return() was replaced by atomic_inc_not_zero().
> >
> > These clients can't be safely dereferenced if their counter in 0.
>
> I'm pretty confused by how these notifiers work....
>
> rpc_release_client decrements cl_count to zero temporarily, to have it
> immediately re-incremented by rpc_free_auth.
>

> So if we're called concurrently with rpc_release_client then it's sort
> of random whether someone gets this callback.
>
> Is that a problem?

Not really. If we re-increment the client->cl_count in rpc_free_auth()
then it would be so that we can send off a bunch of NULL rpc calls to
destroy existing RPCSEC_GSS contexts. We shouldn't need to do any more
upcalls in pipefs.

If we care, we could simply move the call to rpc_unregister_client()
into rpc_free_auth() so that the pipefs notifier doesn't see us, or we
could set a flag to have it ignore us.

> Also, is this an existing bug? (In which case Trond should take it
> now.)

--
Trond Myklebust
Linux NFS client maintainer

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

Subject: Re: [PATCH v2] SUNRPC: skip dead but not buried clients on PipeFS events
Posted by Stanislav Kinsbursky on Wed, 25 Apr 2012 21:14:57 GMT
View Forum Message <> Reply to Message

> On Fri, Apr 20, 2012 at 06:11:02PM +0400, Stanislav Kinsbursky wrote:
>> v2: atomic_inc_return() was replaced by atomic_inc_not_zero().
>>
>> These clients can't be safely dereferenced if their counter in 0.
> I'm pretty confused by how these notifiers work....

There were made as simple as possible - i.e. notifier holds a client
while creating of destroying PipeFS dentries. But event in this case
there were races.

> rpc_release_client decrements cl_count to zero temporarily, to have it
> immediately re-incremented by rpc_free_auth.

BTW, I'm really confused with these re-incrementing reference counter
technic. It makes life-time of RPC client unpredictable.

Is this a real-world valid situation, when usage of it reached zero, but
while we destroying auth, there can some other user of client appear and
client become alive again?
It it was done just to make sure that client is still active while we
destroying auth, then maybe we can just remove the client from the
clients list before rpc_free_auth? It will simplify the notifier
callback logic greatly...

> So if we're called concurrently with rpc_release_client then it's sort
> of random whether someone gets this callback.
>
> Is that a problem?
>
> Also, is this an existing bug? (In which case Trond should take it
> now.)

This is probably not a bug (I can't llok at the code right now; because
these dentries will be destroyed), but a flaw.
Today (without this patch) notifier can try to create dentries for
clients, which are dead already (i.e. auth was destroyed and client is
going to be destroyed very soon, but notifier gained lock first.

```
>
> --b.
>
>> Signed-off-by: Stanislav Kinsbursky<skinsbursky@parallels.com>
>>
>> ---
>>   net/sunrpc/clnt.c |    3 ++-
>>   1 files changed, 2 insertions(+), 1 deletions(-)
>>
>> diff --git a/net/sunrpc/clnt.c b/net/sunrpc/clnt.c
>> index 6797246..d10ebc4 100644
>> --- a/net/sunrpc/clnt.c
>> +++ b/net/sunrpc/clnt.c
>> @@ -218,7 +218,8 @@ static struct rpc_clnt *rpc_get_client_for_event(struct net *net, int event)
>>    if (((event == RPC_PIPEFS_MOUNT)&&  clnt->cl_dentry) ||
>>        ((event == RPC_PIPEFS_UMOUNT)&&  !clnt->cl_dentry))
>>     continue;
>> -  atomic_inc(&clnt->cl_count);
>> +  if (atomic_inc_not_zero(&clnt->cl_count) == 0)
>> +   continue;
>>    spin_unlock(&sn->rpc_client_lock);
>>    return clnt;
>>   }
```

>>

Subject: Re: [PATCH v2] SUNRPC: skip dead but not buried clients on PipeFS events
Posted by Stanislav Kinsbursky on Thu, 26 Apr 2012 06:31:45 GMT
View Forum Message <> Reply to Message

> On Fri, Apr 20, 2012 at 06:11:02PM +0400, Stanislav Kinsbursky wrote:
>> v2: atomic_inc_return() was replaced by atomic_inc_not_zero().
>>
>> These clients can't be safely dereferenced if their counter in 0.
> I'm pretty confused by how these notifiers work....
>
> rpc_release_client decrements cl_count to zero temporarily, to have it
> immediately re-incremented by rpc_free_auth.
>
> So if we're called concurrently with rpc_release_client then it's sort
> of random whether someone gets this callback.
>
> Is that a problem?
>
> Also, is this an existing bug? (In which case Trond should take it
> now.)

Sorry, I was mistaken in previous letter.
Yes, this is an existent bug.
I.e. without this patch notifier can dereference a client, which is
actually dead already, but haven't deleted itself from the client's list.
And then notifier will try to work with this client and even release it
at the end.