Subject: Re: [PATCH v2 2/4] NFS: release per-net clients lock before calling PipeFS dentries creation

Posted by Stanislav Kinsbursky on Mon, 27 Feb 2012 16:20:46 GMT

View Forum Message <> Reply to Message

```
>
>>    spin_lock(&nn->nfs_client_lock);
>> - list_for_each_entry(clp,&nn->nfs_client_list, cl_share_link) {
>> + list_for_each_entry_safe(clp, tmp,&nn->nfs_client_list,
> cl_share_link) {
>>     if (clp->rpc_ops !=&nfs_v4_clientops)
>>      continue;
>> +  atomic_inc(&clp->cl_count);
>> +  spin_unlock(&nn->nfs_client_lock);
>>     error = __rpc_pipefs_event(clp, event, sb);
>> +  nfs_put_client(clp);
>>     if (error)
>>      break;
>> +  spin_lock(&nn->nfs_client_lock);
>>    }
>>    spin_unlock(&nn->nfs_client_lock);
>>    return error;
>
```

> The locking doesn't look right if the loop breaks on error.
> (Same applied to patch v2 1/4)
>

Thanks for the catch. I'll fix this.

> Although list_fo_each_entry_safe() allows the current entry
> to be freed, I don't believe it allows the 'next' to be freed.
> I doubt there is protection against that happening.
>

We need to use safe macro, because client can be destroyed on nfs_put_client() call.
About "protection against ... the 'next' to be freed" - I dont' think, that we
need any protection against it. This will be done under nfs_client_lock, and
current entry list pointers will be updated properly.

> Do you need to use an atomic_inc() for cl_count.
> I'd guess the nfs_client_lock is usually held?
>

Sorry, I don't understand this question.

--
Best regards,

Stanislav Kinsbursky