

---

Subject: Re: [PATCH 4/5] NFS: remove RPC PipeFS mount point reference from blocklayout routines

Posted by [Stanislav Kinsbursky](#) on Tue, 29 Nov 2011 13:13:36 GMT

[View Forum Message](#) <> [Reply to Message](#)

---

>> -----Original Message-----

>> From: Stanislav Kinsbursky [mailto:skinsbursky@parallels.com]

>> Sent: Tuesday, November 29, 2011 8:19 PM

>> To: Peng, Tao

>> Cc: Trond.Myklebust@netapp.com; linux-nfs@vger.kernel.org; Pavel Emelianov; neilb@suse.de;

>> netdev@vger.kernel.org; linux-kernel@vger.kernel.org; James Bottomley; bfields@fieldses.org;

>> davem@davemloft.net; devel@openvz.org

>> Subject: Re: [PATCH 4/5] NFS: remove RPC PipeFS mount point reference from blocklayout routines

>>

>>>> -----Original Message-----

>>>> From: linux-nfs-owner@vger.kernel.org [mailto:linux-nfs-owner@vger.kernel.org] On Behalf Of

>> Stanislav

>>>> Kinsbursky

>>>> Sent: Tuesday, November 29, 2011 6:11 PM

>>>> To: Trond.Myklebust@netapp.com

>>>> Cc: linux-nfs@vger.kernel.org; xemul@parallels.com; neilb@suse.de; netdev@vger.kernel.org; linux-

>>>> kernel@vger.kernel.org; jbottomley@parallels.com; bfields@fieldses.org; davem@davemloft.net;

>>>> devel@openvz.org

>>>> Subject: [PATCH 4/5] NFS: remove RPC PipeFS mount point reference from blocklayout routines

>>>>

>>>> This is a cleanup patch. We don't need this reference anymore, because

>>>> blocklayout pipes dentries now creates and destroys in per-net operations and

>>>> on PipeFS mount/umount notification.

>>>> Note that nfs4blocklayout\_register\_net() now returns 0 instead of -ENOENT in

>>>> case of PipeFS superblock absence. This is ok, because blocklayout pipe dentry

>>>> will be created on PipeFS mount event.

>>> When is the "pipefs mount event" going to happen? When inserting kernel modules or when user issues

>> mount command?

>>>

>>

>> When user issues mount command.

>> Kernel mounts of PipeFS has been removed with all these patch sets I've sent

>> already.

> Then it is going to break blocklayout user space program blkmapd, which is started before mounting any file system and it tries to open the pipe file when started.

Sorry, but I don't get it. Probably we have misunderstanding here.

You said, that "blkmapd ... tries to open the pipe file when started". This pipe file is located on PipeFS, isn't it?

If yes, then PipeFS have to be mounted already in user-space. And if it has been mounted - then pipe dentry is present.

IOW, pipe (without dentry) will be created on module load. Pipe dentry will be created right after that (like it was before) if PipeFS was mounted from user-space. If not - then pipe dentry will be created on PipeFS (!) mount (not NFS or pNFS mount) from user-space.

Or I'm missing something in your reply?

> Not sure if you implement the same logic on nfs pipe as well. But if you do, then nfs client user space program idmapd will fail to start for the same reason.

>

The same logic here.

> Why not just fail to load module if you fail to initialize pipefs? When is rpc\_get\_sb\_net() going to fail?

>

Sorry, but I don't understand, what is your idea. And why do we need to fail at all. BTW, rpc\_get\_sb\_net() just checks, was PipeFS mounted in passed net, or not. If not - not a problem. Dentries will be created on mount event. If yes, then it returns locked PipeFS sb and the next step is dentry creation.

>>

>>

>>> Thanks,

>>> Tao

>>>

>>>>

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

>>>>

>>>> ---

>>>> fs/nfs/blocklayout/blocklayout.c | 9 +-----

>>>> 1 files changed, 1 insertions(+), 8 deletions(-)

>>>>

>>>> diff --git a/fs/nfs/blocklayout/blocklayout.c b/fs/nfs/blocklayout/blocklayout.c

>>>> index acf7ac9..8211ffd 100644

>>>> --- a/fs/nfs/blocklayout/blocklayout.c

>>>> +++ b/fs/nfs/blocklayout/blocklayout.c

>>>> @@ -1032,7 +1032,7 @@ static struct dentry \*nfs4blocklayout\_register\_net(struct net \*net,

>>>>

```

>>>> pipefs_sb = rpc_get_sb_net(net);
>>>> if (!pipefs_sb)
>>>> - return ERR_PTR(-ENOENT);
>>>> + return 0;
>>>> dentry = nfs4blocklayout_register_sb(pipefs_sb, pipe);
>>>> rpc_put_sb_net(net);
>>>> return dentry;
>>>> @@ -1083,7 +1083,6 @@ static struct pernet_operations nfs4blocklayout_net_ops = {
>>>>
>>>> static int __init nfs4blocklayout_init(void)
>>>> {
>>>> - struct vfsmount *mnt;
>>>> int ret;
>>>>
>>>> dprintk("%s: NFSv4 Block Layout Driver Registering...\n", __func__);
>>>> @@ -1093,12 +1092,6 @@ static int __init nfs4blocklayout_init(void)
>>>> goto out;
>>>>
>>>> init_waitqueue_head(&bl_wq);
>>>> -
>>>> - mnt = rpc_get_mount();
>>>> - if (IS_ERR(mnt)) {
>>>> - ret = PTR_ERR(mnt);
>>>> - goto out_remove;
>>>> - }
>>>> ret = rpc_pipefs_notifier_register(&nfs4blocklayout_block);
>>>> if (ret)
>>>> goto out_remove;
>>>>
>>>> --
>>>> 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
>>>
>>
>>
>> --
>> Best regards,
>> Stanislav Kinsbursky
>

```

```

--
Best regards,
Stanislav Kinsbursky

```

---