Subject: [PATCH 0/5] NFS: create blocklayout pipe per network namesapce context
Posted by Stanislav Kinsbursky on Tue, 29 Nov 2011 09:17:03 GMT
View Forum Message <> Reply to Message

This patch set was created in context of clone of git
branch: git://git.linux-nfs.org/projects/trondmy/nfs-2.6.git.
tag: v3.1

This patch set depends on previous patch sets titled:
1) "SUNRPC: initial part of making pipefs work in net ns"
2) "SUNPRC: cleanup PipeFS for network-namespace-aware users"
3) "SUNRPC: make RPC clients use network-namespace-aware PipeFS routines"
4) "NFS: create clients and IDMAP pipes per network namespace"

This patch set is the final part of making SUNRPC PipeFS and it's users work in
network namespace context.

The following series consists of:

Stanislav Kinsbursky (5):
    NFS: handle blocklayout pipe PipeFS dentry by network namespace aware routines
    NFS: blocklayout pipe creation per network namespace context introduced
    NFS: blocklayout PipeFS notifier introduced
    NFS: remove RPC PipeFS mount point reference from blocklayout routines
    SUNRPC: kernel PipeFS mount point creation routines removed

 fs/nfs/blocklayout/blocklayout.c | 154 ++++++++++++++++++++++++++++++-------
 fs/nfs/blocklayout/blocklayout.h | 3 -
 fs/nfs/blocklayout/blocklayoutdev.c | 5 +
 fs/nfs/blocklayout/blocklayoutdm.c | 7 +-
 fs/nfs/inode.c | 1
 fs/nfs/netns.h | 1
 include/linux/sunrpc/rpc_pipe_fs.h | 2
 net/sunrpc/rpc_pipe.c | 21 -----
 8 files changed, 137 insertions(+), 57 deletions(-)

Subject: Re: [PATCH 0/5] NFS: create blocklayout pipe per network namesapce context
Posted by Stanislav Kinsbursky on Tue, 29 Nov 2011 09:17:35 GMT
View Forum Message <> Reply to Message

Hello, Trond.
This is the final part of SUNRPC PipeFS virtualization.
I hope, that you'll find some time to review all series.

You can clone my working tree to have a look at what will be at the end:

git://github.com/skinsbursky/nfs-per-net-ns.git

BTW, I can provide a simple "sandbox" (a kind of container with it's own network namespace and veth device inside) which I use to test my changes.

--
Best regards,
Stanislav Kinsbursky

Subject: [PATCH 3/5] NFS: blocklayout PipeFS notifier introduced
Posted by Stanislav Kinsbursky on Tue, 29 Nov 2011 09:19:12 GMT
View Forum Message <> Reply to Message

This patch subscribes blocklayout pipes to RPC pipefs notifications. Notifier is registering on blocklayout module load. This notifier callback is responsible for creation/destruction of PipeFS blocklayout pipe dentry.
Note that no locking required in notifier callback because PipeFS superblock pointer is passed as an argument from it's creation or destruction routine and thus we can be sure about it's validity.

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

```
---
 fs/nfs/blocklayout/blocklayout.c |   48 ++++++++++++++++++++++++++++++++++++-
 1 files changed, 47 insertions(+), 1 deletions(-)

diff --git a/fs/nfs/blocklayout/blocklayout.c b/fs/nfs/blocklayout/blocklayout.c
index bf20187..acf7ac9 100644
--- a/fs/nfs/blocklayout/blocklayout.c
+++ b/fs/nfs/blocklayout/blocklayout.c
@@ -984,6 +984,46 @@ static void nfs4blocklayout_unregister_sb(struct super_block *sb,
   rpc_unlink(pipe->dentry);
 }

+static int rpc_pipefs_event(struct notifier_block *nb, unsigned long event,
+      void *ptr)
+{
+ struct super_block *sb = ptr;
+ struct net *net = sb->s_fs_info;
+ struct nfs_net *nn = net_generic(net, nfs_net_id);
+ struct dentry *dentry;
+ int ret = 0;
+
+ if (!try_module_get(THIS_MODULE))
```

```
+  return 0;
+
+ if (nn->bl_device_pipe == NULL)
+  return 0;
+
+ switch (event) {
+ case RPC_PIPEFS_MOUNT:
+  dentry = nfs4blocklayout_register_sb(sb, nn->bl_device_pipe);
+  if (IS_ERR(dentry)) {
+   ret = PTR_ERR(dentry);
+   break;
+  }
+  nn->bl_device_pipe->dentry = dentry;
+  break;
+ case RPC_PIPEFS_UMOUNT:
+  if (nn->bl_device_pipe->dentry)
+   nfs4blocklayout_unregister_sb(sb, nn->bl_device_pipe);
+  break;
+ default:
+  ret = -ENOTSUPP;
+  break;
+ }
+ module_put(THIS_MODULE);
+ return ret;
+}
+
+static struct notifier_block nfs4blocklayout_block = {
+ .notifier_call = rpc_pipefs_event,
+};
+
 static struct dentry *nfs4blocklayout_register_net(struct net *net,
          struct rpc_pipe *pipe)
 {
@@ -1059,12 +1099,17 @@ static int __init nfs4blocklayout_init(void)
   ret = PTR_ERR(mnt);
   goto out_remove;
  }
- ret = register_pernet_subsys(&nfs4blocklayout_net_ops);
+ ret = rpc_pipefs_notifier_register(&nfs4blocklayout_block);
  if (ret)
   goto out_remove;
+ ret = register_pernet_subsys(&nfs4blocklayout_net_ops);
+ if (ret)
+  goto out_notifier;
 out:
  return ret;

+out_notifier:
```

```
+ rpc_pipefs_notifier_unregister(&nfs4blocklayout_block);
 out_remove:
  pnfs_unregister_layoutdriver(&blocklayout_type);
  return ret;
@@ -1075,6 +1120,7 @@ static void __exit nfs4blocklayout_exit(void)
  dprintk("%s: NFSv4 Block Layout Driver Unregistering...\n",
         __func__);

+ rpc_pipefs_notifier_unregister(&nfs4blocklayout_block);
  unregister_pernet_subsys(&nfs4blocklayout_net_ops);
  pnfs_unregister_layoutdriver(&blocklayout_type);
 }
```

Subject: [PATCH 2/5] NFS: blocklayout pipe creation per network namespace context introduced

Posted by Stanislav Kinsbursky on Tue, 29 Nov 2011 09:19:15 GMT

View Forum Message <> Reply to Message

This patch implements blocklayout pipe creation and registration per each existent network namespace.
This was achived by registering NFS per-net operations, responsible for blocklayout pipe allocation/register and unregister/destruction instead of initialization and destruction of static "bl_device_pipe" pipe (this one was removed).
Note, than pointer to network blocklayout pipe is stored in per-net "nfs_net" structure, because allocating of one more per-net structure for blocklayout module looks redundant.
This patch also changes dev_remove() function prototype (and all it's callers, where it' requied) by adding network namespace pointer parameter, which is used to discover proper blocklayout pipe for rpc_queue_upcall() call.

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

```
---
 fs/nfs/blocklayout/blocklayout.c    |   49 ++++++++++++++++++++++++++-----------
 fs/nfs/blocklayout/blocklayout.h    |    3 +-
 fs/nfs/blocklayout/blocklayoutdev.c |    5 +++-
 fs/nfs/blocklayout/blocklayoutdm.c  |    7 +++--
 fs/nfs/inode.c                      |    1 +
 fs/nfs/netns.h                      |    1 +
 6 files changed, 46 insertions(+), 20 deletions(-)

diff --git a/fs/nfs/blocklayout/blocklayout.c b/fs/nfs/blocklayout/blocklayout.c
index 50d5183..bf20187 100644
--- a/fs/nfs/blocklayout/blocklayout.c
+++ b/fs/nfs/blocklayout/blocklayout.c
@@ -46,7 +46,6 @@ MODULE_LICENSE("GPL");
```

```
 MODULE_AUTHOR("Andy Adamson <andros@citi.umich.edu>");
 MODULE_DESCRIPTION("The NFSv4.1 pNFS Block layout driver");

-struct rpc_pipe *bl_device_pipe;
 wait_queue_head_t bl_wq;

 static void print_page(struct page *page)
@@ -1011,6 +1010,37 @@ static void nfs4blocklayout_unregister_net(struct net *net,
  }
 }

+static int nfs4blocklayout_net_init(struct net *net)
+{
+ struct nfs_net *nn = net_generic(net, nfs_net_id);
+ struct dentry *dentry;
+
+ nn->bl_device_pipe = rpc_mkpipe_data(&bl_upcall_ops, 0);
+ if (IS_ERR(nn->bl_device_pipe))
+  return PTR_ERR(nn->bl_device_pipe);
+ dentry = nfs4blocklayout_register_net(net, nn->bl_device_pipe);
+ if (IS_ERR(dentry)) {
+  rpc_destroy_pipe_data(nn->bl_device_pipe);
+  return PTR_ERR(dentry);
+ }
+ nn->bl_device_pipe->dentry = dentry;
+ return 0;
+}
+
+static void nfs4blocklayout_net_exit(struct net *net)
+{
+ struct nfs_net *nn = net_generic(net, nfs_net_id);
+
+ nfs4blocklayout_unregister_net(net, nn->bl_device_pipe);
+ rpc_destroy_pipe_data(nn->bl_device_pipe);
+ nn->bl_device_pipe = NULL;
+}
+
+static struct pernet_operations nfs4blocklayout_net_ops = {
+ .init = nfs4blocklayout_net_init,
+ .exit = nfs4blocklayout_net_exit,
+};
+
 static int __init nfs4blocklayout_init(void)
 {
  struct vfsmount *mnt;
@@ -1029,22 +1059,12 @@ static int __init nfs4blocklayout_init(void)
   ret = PTR_ERR(mnt);
   goto out_remove;
```

```
  }
- bl_device_pipe = rpc_mkpipe_data(&bl_upcall_ops, 0);
- if (IS_ERR(bl_device_pipe)) {
-  ret = PTR_ERR(bl_device_pipe);
+ ret = register_pernet_subsys(&nfs4blocklayout_net_ops);
+ if (ret)
   goto out_remove;
- }
- bl_device_pipe->dentry = nfs4blocklayout_register_net(&init_net,
-             bl_device_pipe);
- if (IS_ERR(bl_device_pipe->dentry)) {
-  ret = PTR_ERR(bl_device_pipe->dentry);
-  goto out_destroy_pipe;
- }
 out:
  return ret;

-out_destroy_pipe:
- rpc_destroy_pipe_data(bl_device_pipe);
 out_remove:
  pnfs_unregister_layoutdriver(&blocklayout_type);
  return ret;
@@ -1055,9 +1075,8 @@ static void __exit nfs4blocklayout_exit(void)
  dprintk("%s: NFSv4 Block Layout Driver Unregistering...\n",
         __func__);

+ unregister_pernet_subsys(&nfs4blocklayout_net_ops);
  pnfs_unregister_layoutdriver(&blocklayout_type);
- nfs4blocklayout_unregister_net(&init_net, bl_device_pipe);
- rpc_destroy_pipe_data(bl_device_pipe);
 }

 MODULE_ALIAS("nfs-layouttype4-3");
diff --git a/fs/nfs/blocklayout/blocklayout.h b/fs/nfs/blocklayout/blocklayout.h
index 5f30941..059d95a 100644
--- a/fs/nfs/blocklayout/blocklayout.h
+++ b/fs/nfs/blocklayout/blocklayout.h
@@ -37,6 +37,7 @@
 #include <linux/sunrpc/rpc_pipe_fs.h>

 #include "../pnfs.h"
+#include "../netns.h"

 #define PAGE_CACHE_SECTORS (PAGE_CACHE_SIZE >> SECTOR_SHIFT)
 #define PAGE_CACHE_SECTOR_SHIFT (PAGE_CACHE_SHIFT - SECTOR_SHIFT)
@@ -50,6 +51,7 @@ struct pnfs_block_dev {
  struct list_head  bm_node;
  struct nfs4_deviceid  bm_mdevid;    /* associated devid */
```

```
  struct block_device  *bm_mdev;     /* meta device itself */
+ struct net   *net;
 };

 enum exstate4 {
@@ -159,7 +161,6 @@ struct bl_msg_hdr {
  u16 totallen; /* length of entire message, including hdr itself */
 };

-extern struct rpc_pipe *bl_device_pipe;
 extern wait_queue_head_t bl_wq;

 #define BL_DEVICE_UMOUNT               0x0 /* Umount--delete devices */
diff --git a/fs/nfs/blocklayout/blocklayoutdev.c b/fs/nfs/blocklayout/blocklayoutdev.c
index 79f4752..8893247 100644
--- a/fs/nfs/blocklayout/blocklayoutdev.c
+++ b/fs/nfs/blocklayout/blocklayoutdev.c
@@ -142,6 +142,8 @@ nfs4_blk_decode_device(struct nfs_server *server,
  DECLARE_WAITQUEUE(wq, current);
  struct bl_dev_msg *reply = &bl_mount_reply;
  int offset, len, i;
+ struct net *net = server->nfs_client->net;
+ struct nfs_net *nn = net_generic(net, nfs_net_id);

  dprintk("%s CREATING PIPEFS MESSAGE\n", __func__);
  dprintk("%s: deviceid: %s, mincount: %d\n", __func__, dev->dev_id.data,
@@ -168,7 +170,7 @@ nfs4_blk_decode_device(struct nfs_server *server,

  dprintk("%s CALLING USERSPACE DAEMON\n", __func__);
  add_wait_queue(&bl_wq, &wq);
- if (rpc_queue_upcall(bl_device_pipe, &msg) < 0) {
+ if (rpc_queue_upcall(nn->bl_device_pipe, &msg) < 0) {
   remove_wait_queue(&bl_wq, &wq);
   goto out;
  }
@@ -200,6 +202,7 @@ nfs4_blk_decode_device(struct nfs_server *server,

  rv->bm_mdev = bd;
  memcpy(&rv->bm_mdevid, &dev->dev_id, sizeof(struct nfs4_deviceid));
+ rv->net = net;
  dprintk("%s Created device %s with bd_block_size %u\n",
   __func__,
   bd->bd_disk->disk_name,
diff --git a/fs/nfs/blocklayout/blocklayoutdm.c b/fs/nfs/blocklayout/blocklayoutdm.c
index 631f254..970490f 100644
--- a/fs/nfs/blocklayout/blocklayoutdm.c
+++ b/fs/nfs/blocklayout/blocklayoutdm.c
@@ -38,7 +38,7 @@
```

```
 #define NFSDBG_FACILITY        NFSDBG_PNFS_LD

-static void dev_remove(dev_t dev)
+static void dev_remove(struct net *net, dev_t dev)
 {
  struct rpc_pipe_msg msg;
  struct bl_dev_msg bl_umount_request;
@@ -48,6 +48,7 @@ static void dev_remove(dev_t dev)
  };
  uint8_t *dataptr;
  DECLARE_WAITQUEUE(wq, current);
+ struct nfs_net *nn = net_generic(net, nfs_net_id);

  dprintk("Entering %s\n", __func__);

@@ -66,7 +67,7 @@ static void dev_remove(dev_t dev)
  msg.len = sizeof(bl_msg) + bl_msg.totallen;

  add_wait_queue(&bl_wq, &wq);
- if (rpc_queue_upcall(bl_device_pipe, &msg) < 0) {
+ if (rpc_queue_upcall(nn->bl_device_pipe, &msg) < 0) {
   remove_wait_queue(&bl_wq, &wq);
   goto out;
  }
@@ -93,7 +94,7 @@ static void nfs4_blk_metadev_release(struct pnfs_block_dev *bdev)
   printk(KERN_ERR "%s nfs4_blkdev_put returns %d\n",
     __func__, rv);

- dev_remove(bdev->bm_mdev->bd_dev);
+ dev_remove(bdev->net, bdev->bm_mdev->bd_dev);
 }

 void bl_free_block_dev(struct pnfs_block_dev *bdev)
diff --git a/fs/nfs/inode.c b/fs/nfs/inode.c
index 84d8506..f48790c 100644
--- a/fs/nfs/inode.c
+++ b/fs/nfs/inode.c
@@ -1552,6 +1552,7 @@ static void nfsiod_stop(void)
 }

 int nfs_net_id;
+EXPORT_SYMBOL_GPL(nfs_net_id);

 static int nfs_net_init(struct net *net)
 {
diff --git a/fs/nfs/netns.h b/fs/nfs/netns.h
index 8c1f130..39ae4ca 100644
```

```
--- a/fs/nfs/netns.h
+++ b/fs/nfs/netns.h
@@ -6,6 +6,7 @@

 struct nfs_net {
  struct cache_detail *nfs_dns_resolve;
+ struct rpc_pipe *bl_device_pipe;
 };

 extern int nfs_net_id;
```

Subject: [PATCH 4/5] NFS: remove RPC PipeFS mount point reference from blocklayout routines
Posted by Stanislav Kinsbursky on Tue, 29 Nov 2011 09:20:55 GMT
View Forum Message <> Reply to Message

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.

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;
```

Subject: [PATCH 5/5] SUNRPC: kernel PipeFS mount point creation routines removed

Posted by Stanislav Kinsbursky on Tue, 29 Nov 2011 09:20:58 GMT

View Forum Message <> Reply to Message

This patch removes static rpc_mnt variable and its creation and destruction
routines, because they are not used anymore.

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

```
 include/linux/sunrpc/rpc_pipe_fs.h |    2 --
 net/sunrpc/rpc_pipe.c              |   21 ----------------------
 2 files changed, 0 insertions(+), 23 deletions(-)

diff --git a/include/linux/sunrpc/rpc_pipe_fs.h b/include/linux/sunrpc/rpc_pipe_fs.h
index 7eb0160..a28d2de 100644
--- a/include/linux/sunrpc/rpc_pipe_fs.h
+++ b/include/linux/sunrpc/rpc_pipe_fs.h
@@ -90,8 +90,6 @@ void rpc_destroy_pipe_data(struct rpc_pipe *pipe);
 extern struct dentry *rpc_mkpipe_dentry(struct dentry *, const char *, void *,
      struct rpc_pipe *);
 extern int rpc_unlink(struct dentry *);
-extern struct vfsmount *rpc_get_mount(void);
-extern void rpc_put_mount(void);
 extern int register_rpc_pipefs(void);
 extern void unregister_rpc_pipefs(void);

diff --git a/net/sunrpc/rpc_pipe.c b/net/sunrpc/rpc_pipe.c
index e194e32..4b1d042 100644
```

```
--- a/net/sunrpc/rpc_pipe.c
+++ b/net/sunrpc/rpc_pipe.c
@@ -18,7 +18,6 @@
 #include <linux/kernel.h>

 #include <asm/ioctls.h>
-#include <linux/fs.h>
 #include <linux/poll.h>
 #include <linux/wait.h>
 #include <linux/seq_file.h>
@@ -37,9 +36,6 @@

 #define NET_NAME(net) ((net == &init_net) ? " (init_net)" : "")

-static struct vfsmount *rpc_mnt __read_mostly;
-static int rpc_mount_count;
-
 static struct file_system_type rpc_pipe_fs_type;


@@ -430,23 +426,6 @@ struct rpc_filelist {
  umode_t mode;
 };

-struct vfsmount *rpc_get_mount(void)
-{
- int err;
-
- err = simple_pin_fs(&rpc_pipe_fs_type, &rpc_mnt, &rpc_mount_count);
- if (err != 0)
-  return ERR_PTR(err);
- return rpc_mnt;
-}
-EXPORT_SYMBOL_GPL(rpc_get_mount);
-
-void rpc_put_mount(void)
-{
- simple_release_fs(&rpc_mnt, &rpc_mount_count);
-}
-EXPORT_SYMBOL_GPL(rpc_put_mount);
-
 static int rpc_delete_dentry(const struct dentry *dentry)
 {
  return 1;
```

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

> -----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?

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

---

Subject: Re: [PATCH 4/5] NFS: remove RPC PipeFS mount point reference from blocklayout routines
Posted by Stanislav Kinsbursky on Tue, 29 Nov 2011 12:19:26 GMT
View Forum Message <> Reply to Message

>> -----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.

> 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

Subject: RE: [PATCH 4/5] NFS: remove RPC PipeFS mount point reference from blocklayout routines
Posted by <tao.peng on Tue, 29 Nov 2011 12:40:19 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 stared before
mounting any file system and it tries to open the pipe file when started.
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.

Why not just fail to load module if you fail to initialize pipefs? When is rpc_get_sb_net() going to
fail?

>
>
> > 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
```

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 stared 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

Subject: RE: [PATCH 4/5] NFS: remove RPC PipeFS mount point reference from blocklayout routines
Posted by Myklebust, Trond on Tue, 29 Nov 2011 13:35:51 GMT
View Forum Message <> Reply to Message

> -----Original Message-----
> From: tao.peng@emc.com [mailto:tao.peng@emc.com]

> Sent: Tuesday, November 29, 2011 7:40 AM
> To: skinsbursky@parallels.com
> Cc: Myklebust, Trond; 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: RE: [PATCH 4/5] NFS: remove RPC PipeFS mount point reference
> from blocklayout routines
>
> > -----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
> stared before mounting any file system and it tries to open the pipe file
> when started.

Why on earth is blkmapd doing this instead of listening for file creation notifications like the other rpc_pipefs daemons do?

Trond

Subject: Re: [PATCH 4/5] NFS: remove RPC PipeFS mount point reference from blocklayout routines
Posted by Peng Tao on Tue, 29 Nov 2011 15:05:33 GMT
View Forum Message <> Reply to Message

On Tue, Nov 29, 2011 at 9:13 PM, Stanislav Kinsbursky
<skinsbursky@parallels.com> wrote:

>
>>> -----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
>> stared 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.
Sorry, I misunderstood. I was thinking about mounting NFS or pNFS when
you say "when user issues mount command". Thanks for the explanation.

Regards,
Tao
>
> 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
> --
> 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
```

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

On Tue, Nov 29, 2011 at 9:35 PM, Myklebust, Trond
<Trond.Myklebust@netapp.com> wrote:
>> -----Original Message-----
>> From: tao.peng@emc.com [mailto:tao.peng@emc.com]
>> Sent: Tuesday, November 29, 2011 7:40 AM
>> To: skinsbursky@parallels.com
>> Cc: Myklebust, Trond; 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: RE: [PATCH 4/5] NFS: remove RPC PipeFS mount point reference
>> from blocklayout routines
>>
>> > -----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
>> stared before mounting any file system and it tries to open the pipe file
>> when started.
>
> Why on earth is blkmapd doing this instead of listening for file creation notifications like the other rpc_pipefs daemons do?
Not sure how the original implementer chose this but I think it is
likely because we do not expect the pipe file to be created or deleted
dynamically.

--
Thanks,
Tao

Subject: Re: [PATCH 4/5] NFS: remove RPC PipeFS mount point reference from blocklayout routines
Posted by Myklebust, Trond on Tue, 29 Nov 2011 15:18:26 GMT
View Forum Message <> Reply to Message

On Tue, 2011-11-29 at 23:10 +0800, Peng Tao wrote:
> On Tue, Nov 29, 2011 at 9:35 PM, Myklebust, Trond
> <Trond.Myklebust@netapp.com> wrote:
> >> -----Original Message-----
> >> From: tao.peng@emc.com [mailto:tao.peng@emc.com]
> >> Sent: Tuesday, November 29, 2011 7:40 AM
> >> To: skinsbursky@parallels.com
> >> Cc: Myklebust, Trond; 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: RE: [PATCH 4/5] NFS: remove RPC PipeFS mount point reference
> >> from blocklayout routines
> >>
> >> > -----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
> >> stared before mounting any file system and it tries to open the pipe file
> >> when started.
> >
> > Why on earth is blkmapd doing this instead of listening for file creation notifications like the
other rpc_pipefs daemons do?
> Not sure how the original implementer chose this but I think it is
> likely because we do not expect the pipe file to be created or deleted
> dynamically.

Unless blkmapd can pin the sunrpc module (which it shouldn't be able to)
then that assumption would be wrong. Please look into fixing blkmapd...

  Trond

--
Trond Myklebust
Linux NFS client maintainer

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

Subject: Re: [PATCH 4/5] NFS: remove RPC PipeFS mount point reference from blocklayout routines
Posted by Peng Tao on Tue, 29 Nov 2011 15:30:22 GMT
View Forum Message <> Reply to Message

On Tue, Nov 29, 2011 at 11:18 PM, Trond Myklebust
<Trond.Myklebust@netapp.com> wrote:
> On Tue, 2011-11-29 at 23:10 +0800, Peng Tao wrote:
>> On Tue, Nov 29, 2011 at 9:35 PM, Myklebust, Trond
>> <Trond.Myklebust@netapp.com> wrote:
>> >> -----Original Message-----
>> >> From: tao.peng@emc.com [mailto:tao.peng@emc.com]
>> >> Sent: Tuesday, November 29, 2011 7:40 AM
>> >> To: skinsbursky@parallels.com
>> >> Cc: Myklebust, Trond; 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: RE: [PATCH 4/5] NFS: remove RPC PipeFS mount point reference
>> >> from blocklayout routines
>> >>
>> >> > -----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
>> >> stared before mounting any file system and it tries to open the pipe file
>> >> when started.
>> >
>> > Why on earth is blkmapd doing this instead of listening for file creation notifications like the other rpc_pipefs daemons do?
>> Not sure how the original implementer chose this but I think it is
>> likely because we do not expect the pipe file to be created or deleted
>> dynamically.
>
> Unless blkmapd can pin the sunrpc module (which it shouldn't be able to)
> then that assumption would be wrong. Please look into fixing blkmapd...
Sorry, I don't quite get it. Do you mean sunrpc module may be removed
while nfs/blocklayout modules are still in use? Please explain it a
bit. Thanks.

Best,
Tao
>
> Trond
> --
> Trond Myklebust
> Linux NFS client maintainer
>
> NetApp
> Trond.Myklebust@netapp.com

> www.netapp.com
>

--
Thanks,
Tao

Subject: Re: [PATCH 4/5] NFS: remove RPC PipeFS mount point reference from blocklayout routines
Posted by Myklebust, Trond on Tue, 29 Nov 2011 16:40:30 GMT
View Forum Message <> Reply to Message

On Tue, 2011-11-29 at 23:30 +0800, Peng Tao wrote:
> On Tue, Nov 29, 2011 at 11:18 PM, Trond Myklebust
> <Trond.Myklebust@netapp.com> wrote:
> > On Tue, 2011-11-29 at 23:10 +0800, Peng Tao wrote:
> >> On Tue, Nov 29, 2011 at 9:35 PM, Myklebust, Trond
> >> <Trond.Myklebust@netapp.com> wrote:
> >> >> -----Original Message-----
> >> >> From: tao.peng@emc.com [mailto:tao.peng@emc.com]
> >> >> Sent: Tuesday, November 29, 2011 7:40 AM
> >> >> To: skinsbursky@parallels.com
> >> >> Cc: Myklebust, Trond; 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: RE: [PATCH 4/5] NFS: remove RPC PipeFS mount point reference
> >> >> from blocklayout routines
> >> >>
> >> >> > -----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
> >> >> stared before mounting any file system and it tries to open the pipe file
> >> >> when started.
> >> >
> >> Why on earth is blkmapd doing this instead of listening for file creation notifications like the other rpc_pipefs daemons do?
> >> Not sure how the original implementer chose this but I think it is
> >> likely because we do not expect the pipe file to be created or deleted
> >> dynamically.
> >
> > Unless blkmapd can pin the sunrpc module (which it shouldn't be able to)
> > then that assumption would be wrong. Please look into fixing blkmapd...
> Sorry, I don't quite get it. Do you mean sunrpc module may be removed
> while nfs/blocklayout modules are still in use? Please explain it a
> bit. Thanks.

I mean that I'm perfectly entitled to do

'modprobe -r blocklayoutdriver'

and when I do that, then I expect blkmapd to close the rpc pipe and wait
for a new one to be created just like rpc.idmapd and rpc.gssd do when I
remove the nfs and sunrpc modules.

--
Trond Myklebust
Linux NFS client maintainer

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

Subject: Re: [PATCH 4/5] NFS: remove RPC PipeFS mount point reference from blocklayout routines
Posted by bfields on Tue, 29 Nov 2011 16:42:52 GMT
View Forum Message <> Reply to Message

On Tue, Nov 29, 2011 at 11:40:30AM -0500, Trond Myklebust wrote:
> I mean that I'm perfectly entitled to do
>
> 'modprobe -r blocklayoutdriver'
>
> and when I do that, then I expect blkmapd to close the rpc pipe and wait
> for a new one to be created just like rpc.idmapd and rpc.gssd do when I
> remove the nfs and sunrpc modules.

The rpc pipefs mount doesn't hold a reference on the sunrpc module?

--b.

Subject: Re: [PATCH 4/5] NFS: remove RPC PipeFS mount point reference from blocklayout routines
Posted by Myklebust, Trond on Tue, 29 Nov 2011 17:19:25 GMT
View Forum Message <> Reply to Message

On Tue, 2011-11-29 at 11:42 -0500, J. Bruce Fields wrote:
> On Tue, Nov 29, 2011 at 11:40:30AM -0500, Trond Myklebust wrote:
> > I mean that I'm perfectly entitled to do
> >
> > 'modprobe -r blocklayoutdriver'
> >
> > and when I do that, then I expect blkmapd to close the rpc pipe and wait
> > for a new one to be created just like rpc.idmapd and rpc.gssd do when I
> > remove the nfs and sunrpc modules.
>
> The rpc pipefs mount doesn't hold a reference on the sunrpc module?

I stand corrected: the mount does hold a reference to the sunrpc
module.

However nothing holds a reference to the blocklayoutdriver module, so
the main point that the "blocklayout" pipe can disappear from underneath
the blkmapd stands.

--
Trond Myklebust
Linux NFS client maintainer

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

Subject: Re: [PATCH 4/5] NFS: remove RPC PipeFS mount point reference from blocklayout routines
Posted by bfields on Tue, 29 Nov 2011 17:27:02 GMT
View Forum Message <> Reply to Message

On Tue, Nov 29, 2011 at 12:19:25PM -0500, Trond Myklebust wrote:
> On Tue, 2011-11-29 at 11:42 -0500, J. Bruce Fields wrote:
> > On Tue, Nov 29, 2011 at 11:40:30AM -0500, Trond Myklebust wrote:
> > > I mean that I'm perfectly entitled to do
> > >
> > > 'modprobe -r blocklayoutdriver'
> > >
> > > and when I do that, then I expect blkmapd to close the rpc pipe and wait
> > > for a new one to be created just like rpc.idmapd and rpc.gssd do when I
> > > remove the nfs and sunrpc modules.
> >
> > The rpc pipefs mount doesn't hold a reference on the sunrpc module?
>
> I stand corrected: the mount does hold a reference to the sunrpc
> module.
> However nothing holds a reference to the blocklayoutdriver module, so
> the main point that the "blocklayout" pipe can disappear from underneath
> the blkmapd stands.

OK, that makes sense.

--b.

Subject: Re: [PATCH 4/5] NFS: remove RPC PipeFS mount point reference from blocklayout routines
Posted by Peng Tao on Tue, 29 Nov 2011 17:30:27 GMT
View Forum Message <> Reply to Message

On Wed, Nov 30, 2011 at 1:19 AM, Trond Myklebust
<Trond.Myklebust@netapp.com> wrote:
> On Tue, 2011-11-29 at 11:42 -0500, J. Bruce Fields wrote:
>> On Tue, Nov 29, 2011 at 11:40:30AM -0500, Trond Myklebust wrote:
>> > I mean that I'm perfectly entitled to do
>> >
>> > 'modprobe -r blocklayoutdriver'
>> >
>> > and when I do that, then I expect blkmapd to close the rpc pipe and wait
>> > for a new one to be created just like rpc.idmapd and rpc.gssd do when I
>> > remove the nfs and sunrpc modules.
>>
>> The rpc pipefs mount doesn't hold a reference on the sunrpc module?
>
> I stand corrected: the mount does hold a reference to the sunrpc
> module.
> However nothing holds a reference to the blocklayoutdriver module, so
> the main point that the "blocklayout" pipe can disappear from underneath
> the blkmapd stands.
Thanks for the explanation and I agree it can cause problem if user
reload blocklayout module. I will look into a fix to blkmapd.

Best,
Tao

Subject: Re: [PATCH 0/5] NFS: create blocklayout pipe per network namesapce context
Posted by Myklebust, Trond on Fri, 30 Dec 2011 22:55:05 GMT
View Forum Message <> Reply to Message

On Tue, 2011-11-29 at 13:10 +0300, Stanislav Kinsbursky wrote:
> This patch set was created in context of clone of git
> branch: git://git.linux-nfs.org/projects/trondmy/nfs-2.6.git.
> tag: v3.1
>
> This patch set depends on previous patch sets titled:
> 1) "SUNRPC: initial part of making pipefs work in net ns"
> 2) "SUNPRC: cleanup PipeFS for network-namespace-aware users"
> 3) "SUNRPC: make RPC clients use network-namespace-aware PipeFS routines"
> 4) "NFS: create clients and IDMAP pipes per network namespace"
>
> This patch set is the final part of making SUNRPC PipeFS and it's users work in
> network namespace context.
>
> The following series consists of:
>
> ---

>
> Stanislav Kinsbursky (5):
>   NFS: handle blocklayout pipe PipeFS dentry by network namespace aware routines
>   NFS: blocklayout pipe creation per network namespace context introduced
>   NFS: blocklayout PipeFS notifier introduced
>   NFS: remove RPC PipeFS mount point reference from blocklayout routines
>   SUNRPC: kernel PipeFS mount point creation routines removed
>
>
> fs/nfs/blocklayout/blocklayout.c | 154 ++++++++++++++++++++++++++-------
> fs/nfs/blocklayout/blocklayout.h | 3 -
> fs/nfs/blocklayout/blocklayoutdev.c | 5 +
> fs/nfs/blocklayout/blocklayoutdm.c | 7 +-
> fs/nfs/inode.c | 1
> fs/nfs/netns.h | 1
> include/linux/sunrpc/rpc_pipe_fs.h | 2
> net/sunrpc/rpc_pipe.c | 21 -----
> 8 files changed, 137 insertions(+), 57 deletions(-)

These patches need rebasing in order to apply on top of 3.2-rc7...

Cheers
  Trond

--
Trond Myklebust
Linux NFS client maintainer

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

Subject: Re: [PATCH 0/5] NFS: create blocklayout pipe per network namesapce context
Posted by Myklebust, Trond on Thu, 05 Jan 2012 20:58:31 GMT
View Forum Message <> Reply to Message

On Fri, 2011-12-30 at 17:55 -0500, Trond Myklebust wrote:
> On Tue, 2011-11-29 at 13:10 +0300, Stanislav Kinsbursky wrote:
> > This patch set was created in context of clone of git
> > branch: git://git.linux-nfs.org/projects/trondmy/nfs-2.6.git.
> > tag: v3.1
> >
> > This patch set depends on previous patch sets titled:
> > 1) "SUNRPC: initial part of making pipefs work in net ns"
> > 2) "SUNPRC: cleanup PipeFS for network-namespace-aware users"

> > 3) "SUNRPC: make RPC clients use network-namespace-aware PipeFS routines"
> > 4) "NFS: create clients and IDMAP pipes per network namespace"
> >
> > This patch set is the final part of making SUNRPC PipeFS and it's users work in
> > network namespace context.
> >
> > The following series consists of:
> >
> > ---
> >
> > Stanislav Kinsbursky (5):
> >   NFS: handle blocklayout pipe PipeFS dentry by network namespace aware routines
> >   NFS: blocklayout pipe creation per network namespace context introduced
> >   NFS: blocklayout PipeFS notifier introduced
> >   NFS: remove RPC PipeFS mount point reference from blocklayout routines
> >   SUNRPC: kernel PipeFS mount point creation routines removed
> >
> >
> > fs/nfs/blocklayout/blocklayout.c | 154 ++++++++++++++++++++++++++++++-------
> > fs/nfs/blocklayout/blocklayout.h | 3 -
> > fs/nfs/blocklayout/blocklayoutdev.c | 5 +
> > fs/nfs/blocklayout/blocklayoutdm.c | 7 +-
> > fs/nfs/inode.c | 1
> > fs/nfs/netns.h | 1
> > include/linux/sunrpc/rpc_pipe_fs.h | 2
> > net/sunrpc/rpc_pipe.c | 21 -----
> > 8 files changed, 137 insertions(+), 57 deletions(-)
>
>
> These patches need rebasing in order to apply on top of 3.2-rc7...

OK. Further testing seems to indicate that we're going to have to
postpone merging these patches until the 3.4 merge window.

The problems are twofold:

As the patches stand now in the linux-next tree, they can (and
occasionally do) Oops on unmount. The reason was that I rejected the
PipeFS notifier patch for the idmapper (due to the reported problem of
nfs_idmap_init/nfs_idmap_quit being undefined when CONFIG_NFS_V4 is
undefined), and the fact that it is missing causes the unmount at the
end of our tests to hit the BUG() in fs/dcache.c:905. This suggests that
we will have the same problem with the pNFS block layout driver, since I
still haven't received a rebased update of the 5 'create blocklayout
pipe per network namesapce context' patches.

The second problem that was highlighted was the fact that as they stand
today, these patchsets do not allow for bisection. When we hit the Oops,

I had Bryan try to bisect where the problem arose. He ended up pointing at the patch "SUNRPC: handle RPC client pipefs dentries by network namespace aware routine", which is indeed the cause, but which is one of the _dependencies_ for all the PipeFS notifier patches that fix the problem.

--
Trond Myklebust
Linux NFS client maintainer

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

Subject: Re: [PATCH 0/5] NFS: create blocklayout pipe per network namesapce context
Posted by Stanislav Kinsbursky on Tue, 10 Jan 2012 10:50:10 GMT
View Forum Message <> Reply to Message

> On Tue, 2011-11-29 at 13:10 +0300, Stanislav Kinsbursky wrote:
>> This patch set was created in context of clone of git
>> branch: git://git.linux-nfs.org/projects/trondmy/nfs-2.6.git.
>> tag: v3.1
>>
>> This patch set depends on previous patch sets titled:
>> 1) "SUNRPC: initial part of making pipefs work in net ns"
>> 2) "SUNPRC: cleanup PipeFS for network-namespace-aware users"
>> 3) "SUNRPC: make RPC clients use network-namespace-aware PipeFS routines"
>> 4) "NFS: create clients and IDMAP pipes per network namespace"
>>
>> This patch set is the final part of making SUNRPC PipeFS and it's users work in
>> network namespace context.
>>
>> The following series consists of:
>>
>> ---
>>
>> Stanislav Kinsbursky (5):
>>   NFS: handle blocklayout pipe PipeFS dentry by network namespace aware routines
>>   NFS: blocklayout pipe creation per network namespace context introduced
>>   NFS: blocklayout PipeFS notifier introduced
>>   NFS: remove RPC PipeFS mount point reference from blocklayout routines
>>   SUNRPC: kernel PipeFS mount point creation routines removed
>>
>>
>> fs/nfs/blocklayout/blocklayout.c | 154 ++++++++++++++++++++++++++++-------

>> fs/nfs/blocklayout/blocklayout.h | 3 -
>> fs/nfs/blocklayout/blocklayoutdev.c | 5 +
>> fs/nfs/blocklayout/blocklayoutdm.c | 7 +-
>> fs/nfs/inode.c | 1
>> fs/nfs/netns.h | 1
>> include/linux/sunrpc/rpc_pipe_fs.h | 2
>> net/sunrpc/rpc_pipe.c | 21 -----
>> 8 files changed, 137 insertions(+), 57 deletions(-)
>
>
> These patches need rebasing in order to apply on top of 3.2-rc7...
>

Will resend rebased version soon.

--
Best regards,
Stanislav Kinsbursky

Subject: Re: [PATCH 0/5] NFS: create blocklayout pipe per network namesapce context
Posted by Stanislav Kinsbursky on Tue, 10 Jan 2012 12:58:27 GMT
View Forum Message <> Reply to Message

> On Fri, 2011-12-30 at 17:55 -0500, Trond Myklebust wrote:
>> On Tue, 2011-11-29 at 13:10 +0300, Stanislav Kinsbursky wrote:
>>> This patch set was created in context of clone of git
>>> branch: git://git.linux-nfs.org/projects/trondmy/nfs-2.6.git.
>>> tag: v3.1
>>>
>>> This patch set depends on previous patch sets titled:
>>> 1) "SUNRPC: initial part of making pipefs work in net ns"
>>> 2) "SUNPRC: cleanup PipeFS for network-namespace-aware users"
>>> 3) "SUNRPC: make RPC clients use network-namespace-aware PipeFS routines"
>>> 4) "NFS: create clients and IDMAP pipes per network namespace"
>>>
>>> This patch set is the final part of making SUNRPC PipeFS and it's users work in
>>> network namespace context.
>>>
>>> The following series consists of:
>>>
>>> ---
>>>
>>> Stanislav Kinsbursky (5):
>>> NFS: handle blocklayout pipe PipeFS dentry by network namespace aware routines
>>> NFS: blocklayout pipe creation per network namespace context introduced

>>>   NFS: blocklayout PipeFS notifier introduced
>>>   NFS: remove RPC PipeFS mount point reference from blocklayout routines
>>>   SUNRPC: kernel PipeFS mount point creation routines removed
>>>
>>>
>>>   fs/nfs/blocklayout/blocklayout.c | 154 ++++++++++++++++++++++++++++++-------
>>>   fs/nfs/blocklayout/blocklayout.h | 3 -
>>>   fs/nfs/blocklayout/blocklayoutdev.c | 5 +
>>>   fs/nfs/blocklayout/blocklayoutdm.c | 7 +-
>>>   fs/nfs/inode.c | 1
>>>   fs/nfs/netns.h | 1
>>>   include/linux/sunrpc/rpc_pipe_fs.h | 2
>>>   net/sunrpc/rpc_pipe.c | 21 -----
>>>   8 files changed, 137 insertions(+), 57 deletions(-)
>>
>>
>> These patches need rebasing in order to apply on top of 3.2-rc7...
>
> OK. Further testing seems to indicate that we're going to have to
> postpone merging these patches until the 3.4 merge window.
>
> The problems are twofold:
>
> As the patches stand now in the linux-next tree, they can (and
> occasionally do) Oops on unmount. The reason was that I rejected the
> PipeFS notifier patch for the idmapper (due to the reported problem of
> nfs_idmap_init/nfs_idmap_quit being undefined when CONFIG_NFS_V4 is
> undefined), and the fact that it is missing causes the unmount at the
> end of our tests to hit the BUG() in fs/dcache.c:905. This suggests that
> we will have the same problem with the pNFS block layout driver, since I
> still haven't received a rebased update of the 5 'create blocklayout
> pipe per network namesapce context' patches.
>

Hello, Trond.
I've resend the patch set (rebased with fix for nfs_idmap_init/nfs_idmap_quit).

> The second problem that was highlighted was the fact that as they stand
> today, these patchsets do not allow for bisection. When we hit the Oops,
> I had Bryan try to bisect where the problem arose. He ended up pointing
> at the patch "SUNRPC: handle RPC client pipefs dentries by network
> namespace aware routine", which is indeed the cause, but which is one of
> the _dependencies_ for all the PipeFS notifier patches that fix the
> problem.
>

I'm confused here. Does this means, that I have to fix patch "SUNRPC: handle RPC client pipefs dentries by network namespace aware routine" to make it able to

bisect?

--
Best regards,
Stanislav Kinsbursky

Subject: Re: [PATCH 0/5] NFS: create blocklayout pipe per network namesapce context
Posted by Myklebust, Trond on Wed, 11 Jan 2012 16:23:23 GMT
View Forum Message <> Reply to Message

On Tue, 2012-01-10 at 16:58 +0400, Stanislav Kinsbursky wrote:

> > The second problem that was highlighted was the fact that as they stand
> > today, these patchsets do not allow for bisection. When we hit the Oops,
> > I had Bryan try to bisect where the problem arose. He ended up pointing
> > at the patch "SUNRPC: handle RPC client pipefs dentries by network
> > namespace aware routine", which is indeed the cause, but which is one of
> > the _dependencies_ for all the PipeFS notifier patches that fix the
> > problem.
> >
>
> I'm confused here. Does this means, that I have to fix patch "SUNRPC: handle RPC
> client pipefs dentries by network namespace aware routine" to make it able to
> bisect?

What I mean is that currently, I have various ways to Oops the kernel when I apply "SUNRPC: handle RPC client pipefs dentries by network namespace aware routine" before all these other followup patches are applied.

One way to could fix this, might be to add dummy versions of rpc_pipefs_notifier_register()/unregister() so that "NFS: idmap PipeFS notifier introduced" and the other such patches can be applied without compilation errors or Oopses before the "handle RPC client pipefs dentries..." patch is applied. The latter could then enable the real rpc_pipefs_notifier_register()/....

The point is to not have these patches add _known_ bugs to the kernel at any point, so that someone who is trying to track down an unknown bug via "git bisect" doesn't have to also cope with these avoidable issues...

--
Trond Myklebust
Linux NFS client maintainer

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

Subject: Re: [PATCH 0/5] NFS: create blocklayout pipe per network namesapce context
Posted by Stanislav Kinsbursky on Wed, 11 Jan 2012 17:23:14 GMT
View Forum Message <> Reply to Message

> On Tue, 2012-01-10 at 16:58 +0400, Stanislav Kinsbursky wrote:

>>> The second problem that was highlighted was the fact that as they stand
>>> today, these patchsets do not allow for bisection. When we hit the Oops,
>>> I had Bryan try to bisect where the problem arose. He ended up pointing
>>> at the patch "SUNRPC: handle RPC client pipefs dentries by network
>>> namespace aware routine", which is indeed the cause, but which is one of
>>> the _dependencies_ for all the PipeFS notifier patches that fix the
>>> problem.
>>>
>>
>> I'm confused here. Does this means, that I have to fix patch "SUNRPC: handle RPC
>> client pipefs dentries by network namespace aware routine" to make it able to
>> bisect?
>
> What I mean is that currently, I have various ways to Oops the kernel
> when I apply "SUNRPC: handle RPC client pipefs dentries by network
> namespace aware routine" before all these other followup patches are
> applied.
>
> One way to could fix this, might be to add dummy versions of
> rpc_pipefs_notifier_register()/unregister() so that "NFS: idmap PipeFS
> notifier introduced" and the other such patches can be applied without
> compilation errors or Oopses before the "handle RPC client pipefs
> dentries..." patch is applied. The latter could then enable the real
> rpc_pipefs_notifier_register()/....
>
> The point is to not have these patches add _known_ bugs to the kernel at
> any point, so that someone who is trying to track down an unknown bug
> via "git bisect" doesn't have to also cope with these avoidable
> issues...
>

Ok, thanks for explanation.
I've sent rebased "v2" of the patch set, contains updated patch "SUNRPC: handle RPC client pipefs dentries by network namespace aware routine", which, I believe, fixes oops, spotted by Bryan (it was caused by excessive call of

rpc_put_mount() on PipeFS dentries unlink).
So, if I'm not mistaken here, there's no need in implementing of dummy versions
of rpc_pipefs_notifier_(un)register() or any other dummy stuff.

BTW, it looks like that in last 2 days I've sent all updates to the issues you
pointed out. If not, please, ping me once more.

--
Best regards,
Stanislav Kinsbursky

Subject: Re: [PATCH 0/5] NFS: create blocklayout pipe per network namesapce context
Posted by Myklebust, Trond on Wed, 11 Jan 2012 17:46:03 GMT
View Forum Message <> Reply to Message

On Wed, 2012-01-11 at 21:23 +0400, Stanislav Kinsbursky wrote:
> I've sent rebased "v2" of the patch set, contains updated patch "SUNRPC: handle
> RPC client pipefs dentries by network namespace aware routine", which, I
> believe, fixes oops, spotted by Bryan (it was caused by excessive call of
> rpc_put_mount() on PipeFS dentries unlink).
> So, if I'm not mistaken here, there's no need in implementing of dummy versions
> of rpc_pipefs_notifier_(un)register() or any other dummy stuff.

OK. Fair enough. We'll give it some testing to make sure that we don't
hit it again...

> BTW, it looks like that in last 2 days I've sent all updates to the issues you
> pointed out. If not, please, ping me once more.

I'm in the process of reviewing it right now.

Cheers
  Trond

--
Trond Myklebust
Linux NFS client maintainer

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

Subject: Re: [PATCH 0/5] NFS: create blocklayout pipe per network namesapce context

Posted by Stanislav Kinsbursky on Wed, 11 Jan 2012 18:03:17 GMT
View Forum Message <> Reply to Message

> On Wed, 2012-01-11 at 21:23 +0400, Stanislav Kinsbursky wrote:
>> I've sent rebased "v2" of the patch set, contains updated patch "SUNRPC: handle
>> RPC client pipefs dentries by network namespace aware routine", which, I
>> believe, fixes oops, spotted by Bryan (it was caused by excessive call of
>> rpc_put_mount() on PipeFS dentries unlink).
>> So, if I'm not mistaken here, there's no need in implementing of dummy versions
>> of rpc_pipefs_notifier_(un)register() or any other dummy stuff.
>
> OK. Fair enough. We'll give it some testing to make sure that we don't
> hit it again...
>
>> BTW, it looks like that in last 2 days I've sent all updates to the issues you
>> pointed out. If not, please, ping me once more.
>
> I'm in the process of reviewing it right now.
>

Cool, thanks, Trond.

> Cheers
> Trond
>

--
Best regards,
Stanislav Kinsbursky

Subject: Re: [PATCH 4/5] NFS: remove RPC PipeFS mount point reference from blocklayout routines
Posted by Boaz Harrosh on Mon, 28 May 2012 11:43:53 GMT
View Forum Message <> Reply to Message

On 11/29/2011 07:30 PM, Peng Tao wrote:

> On Wed, Nov 30, 2011 at 1:19 AM, Trond Myklebust
> <Trond.Myklebust@netapp.com> wrote:
>> On Tue, 2011-11-29 at 11:42 -0500, J. Bruce Fields wrote:
>>> On Tue, Nov 29, 2011 at 11:40:30AM -0500, Trond Myklebust wrote:
>>>> I mean that I'm perfectly entitled to do
>>>>
>>>> 'modprobe -r blocklayoutdriver'
>>>>

>>>> and when I do that, then I expect blkmapd to close the rpc pipe and wait
>>>> for a new one to be created just like rpc.idmapd and rpc.gssd do when I
>>>> remove the nfs and sunrpc modules.
>>>
>>> The rpc pipefs mount doesn't hold a reference on the sunrpc module?
>>
>> I stand corrected: the mount does hold a reference to the sunrpc
>> module.
>> However nothing holds a reference to the blocklayoutdriver module, so
>> the main point that the "blocklayout" pipe can disappear from underneath
>> the blkmapd stands.
> Thanks for the explanation and I agree it can cause problem if user
> reload blocklayout module. I will look into a fix to blkmapd.
>

You might want to consider converting to call_usermodehelper()

I know that it greatly simplified our code both in Kernel and
in user-mode. And it made nfs-utils maintainer much happier
as well.

The speed is not Cardinal here I think. Like in objects it's
done once per new device_id

> Best,
> Tao

Just my $0.017
Boaz
