If client is a clone, then it's parent can not be in the list.  
But parent's PipeFS dentries have to be created and destroyed as well.

Note: event skip helper for clients introduced

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

---

net/sunrpc/clnt.c | 30 ++++++++++++++++++++++++++----
1 files changed, 26 insertions(+), 4 deletions(-)

diff --git a/net/sunrpc/clnt.c b/net/sunrpc/clnt.c
index 8a19849..80b59f1 100644
--- a/net/sunrpc/clnt.c
+++ b/net/sunrpc/clnt.c
@@ -176,8 +176,16 @@ rpc_setup_pipedir(struct rpc_clnt *clnt, const char *dir_name)
    return 0;
 }

-static int __rpc_pipefs_event(struct rpc_clnt *clnt, unsigned long event,
-struct super_block *sb)
+static inline int rpc_clnt_skip_event(struct rpc_clnt *clnt, unsigned long event)
+{
+    if (((event == RPC_PIPEFS_MOUNT) && clnt->cl_dentry) ||
+        ((event == RPC_PIPEFS_UMOUNT) && !clnt->cl_dentry))
+        return 1;
+    return 0;
+
+static int __rpc_clnt_handle_event(struct rpc_clnt *clnt, unsigned long event,
    struct super_block *sb)
{
    struct dentry *dentry;
    int err = 0;
    @@ -206,6 +214,21 @@ static int __rpc_pipefs_event(struct rpc_clnt *clnt, unsigned long event,
        return err;
 }

+static int __rpc_pipefs_event(struct rpc_clnt *clnt, unsigned long event,
+struct super_block *sb)
+{
+    int error;
+    +
+    static int __rpc_pipefs_event(struct rpc_clnt *clnt, unsigned long event, 
+    struct super_block *sb)
{ 
    struct dentry *dentry;
    int err = 0;
    @@ -206,6 +214,21 @@ static int __rpc_pipefs_event(struct rpc_clnt *clnt, unsigned long event,
        return err;
 }

+static int __rpc_pipefs_event(struct rpc_clnt *clnt, unsigned long event,
+struct super_block *sb)
+{
+    int error;
+    +
+    int if (rpc_clnt_skip_event(clnt, event)) { 


Hi Stanislav,

Recursion in the kernel is generally frowned upon due to the stack size
limits. Could you please rewrite the above into a simple loop. Something along the lines of:

```c
for(;;) {
  ...

  if (clnt == clnt->cl_parent)
    break;
  clnt = clnt->cl_parent;
}
```

--
Trond Myklebust
Linux NFS client maintainer

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

Subject: Re: [PATCH 2/3] SUNRPC: traverse clients tree on PipeFS event
Posted by Stanislav Kinsbursky on Thu, 26 Apr 2012 18:26:51 GMT

> On Fri, 2012-04-20 at 18:19 +0400, Stanislav Kinsbursky wrote:
> 
> >> +static int __rpc_pipefs_event(struct rpc_clnt *clnt, unsigned long event,
> >> +struct super_block *sb)
> >> +{
> >> +error = __rpc_clnt_handle_event(clnt, event, sb);
> >> +}
> >> +if (!rpc_clnt_skip_event(clnt, event))
> >> +error = __rpc_clnt_handle_event(clnt, event, sb);
> >> +if (error)
> >> +return error;
> >> +}
> >> +if (clnt != clnt->cl_parent)
> >> +return __rpc_pipefs_event(clnt->cl_parent, event, sb);
> >> +return 0;
> >> +}
> Hi Stanislav,
> >
> Recursion in the kernel is generally frowned upon due to the stack size
> limits. Could you please rewrite the above into a simple loop. Something
> along the lines of:
> >
> for(;;) {
if (clnt == clnt->cl_parent)
    break;
clnt = clnt->cl_parent;
}

Hi, Trond.
Yes, sure, I can do this.

---

Subject: [PATCH v2 2/3] SUNRPC: traverse clients tree on PipeFS event
Posted by Stanislav Kinsbursky on Fri, 27 Apr 2012 09:00:17 GMT

v2: recursion was replaced by loop

If client is a clone, then it's parent can not be in the list.
But parent's Pipefs dentries have to be created and destroyed.

Note: event skip helper for clients introduced

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

---

diff --git a/net/sunrpc/clnt.c b/net/sunrpc/clnt.c
index 8a19849..d127bd7 100644
--- a/net/sunrpc/clnt.c
+++ b/net/sunrpc/clnt.c
@@ -176,8 +176,16 @@ rpc_setup_pipedir(struct rpc_clnt *clnt, const char *dir_name)
     return 0;
 }

-    static int __rpc_pipefs_event(struct rpc_clnt *clnt, unsigned long event,
+    static int rpc_clnt_skip_event(struct rpc_clnt *clnt, unsigned long event)
+    {
+        if (((event == RPC_PIPEFS_MOUNT) && clnt->cl_dentry) ||
+            ((event == RPC_PIPEFS_UMOUNT) && !clnt->cl_dentry))
+            return 1;
+        return 0;
+    }
+    
+    static int __rpc_clnt_handle_event(struct rpc_clnt *clnt, unsigned long event,

-    static inline int rpc_set
+    static inline int rpc_setup_pipedir(struct rpc_clnt *clnt, const char *dir_name)
     return 0;
 }

---

Page 4 of 5 ---- Generated from OpenVZ Forum
+ struct super_block *sb)
+ {
+ struct dentry *dentry;
+ int err = 0;
+ @-206,6 +214,20 @ static int __rpc_pipefs_event(struct rpc_clnt *clnt, unsigned long event, return err;
+ }
+
+ static int __rpc_pipefs_event(struct rpc_clnt *clnt, unsigned long event,
+ struct super_block *sb)
+ {
+ int error = 0;
+ 
+ for (; ; clnt = clnt->cl_parent) {
+ if (!rpc_clnt_skip_event(clnt, event))
+ error = __rpc_clnt_handle_event(clnt, event, sb);
+ if (error || clnt == clnt->cl_parent)
+ break;
+ } 
+ return error;
+ } 
+
+ static struct rpc_clnt *rpc_get_client_for_event(struct net *net, int event)
+ {
+ struct sunrpc_net *sn = net_generic(net, sunrpc_net_id);
+ @-215,8 +237,7 @ static struct rpc_clnt *rpc_get_client_for_event(struct net *net, int event)
+ list_for_each_entry(clnt, &sn->all_clients, cl_clients) {
+ if (clnt->cl_program->pipe_dir_name == NULL)
+ break;
+ -if (((event == RPC_PIPEFS_MOUNT) && clnt->cl_dentry) ||
+ - (event == RPC_PIPEFS_UMOUNT) && !clnt->cl_dentry))
+ if (rpc_clnt_skip_event(clnt, event))
+ continue;
+ if (atomic_inc_not_zero(&clnt->cl_count) == 0)
+ continue;
+ }