
Subject: [PATCH v7 0/7] SUNRPC: make rpcbind clients allocated and destroyed dynamically

Posted by [Stanislav Kinsbursky](#) on Fri, 28 Oct 2011 10:52:19 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
and rebased on tag "v3.1".

v7:

- 1) Implemented "goto pattern" in __svc_create().
- 2) Pathes 5 and 6 from perious patch-set were squashed to make things looks clearer and to allow safe bisecting of the patch-set.

v6:

- 1) Fixes in rpcb_clients management.

v4:

- 1) creation and destruction on rpcbind clients now depends on service program versions "vs_hidden" flag.

This patch is required for further RPC layer virtualization, because rpcbind clients have to be per network namespace.
To achive this, we have to untie network namespace from rpcbind clients sockets.
The idea of this patch set is to make rpcbind clients non-static. I.e. rpcbind clients will be created during first RPC service creation, and destroyed when last RPC service is stopped.
With this patch set rpcbind clients can be virtualized easely.

The following series consists of:

Stanislav Kinsbursky (7):

- SUNRPC: introduce helpers for reference counted rpcbind clients
- SUNRPC: use rpcbind reference counting helpers
- SUNRPC: introduce svc helpers for prepairing rpcbind infrastructure
- SUNRPC: setup rpcbind clients if service requires it
- SUNRPC: cleanup service destruction
- SUNRPC: remove rpcbind clients creation during service registering
- SUNRPC: remove rpcbind clients destruction on module cleanup

```
fs/nfsd/nfssvc.c      | 2 +
include/linux/sunrpc/clnt.h | 2 +
include/linux/sunrpc/svc.h | 1
net/sunrpc/rpcb_clnt.c  | 88 ++++++-----
net/sunrpc/sunrpc_syms.c | 3 -
```

net/sunrpc/svc.c | 57 ++++++-----
6 files changed, 113 insertions(+), 40 deletions(-)

Subject: [PATCH v7 1/7] SUNRPC: introduce helpers for reference counted rpcbind clients

Posted by [Stanislav Kinsbursky](#) on Fri, 28 Oct 2011 10:52:37 GMT

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

v6:

- 1) added write memory barrier to rpcb_set_local to make sure, that rpcbind clients become valid before rpcb_users assignment
- 2) explicitly set rpcb_users to 1 instead of incrementing it (looks clearer from my pow).

v5: fixed races with rpcb_users in rpcb_get_local()

This helpers will be used for dynamical creation and destruction of rpcbind clients.

Variable rpcb_users is actually a counter of lauched RPC services. If rpcbind clients has been created already, then we just increase rpcb_users.

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

net/sunrpc/rpcb_clnt.c | 53 ++++++-----
1 files changed, 53 insertions(+), 0 deletions(-)

diff --git a/net/sunrpc/rpcb_clnt.c b/net/sunrpc/rpcb_clnt.c

index e45d2fb..9830d87 100644

--- a/net/sunrpc/rpcb_clnt.c

+++ b/net/sunrpc/rpcb_clnt.c

@@ -114,6 +114,9 @@ static struct rpc_program rpcb_program;

static struct rpc_clnt * rpcb_local_clnt;

static struct rpc_clnt * rpcb_local_clnt4;

+DEFINE_SPINLOCK(rpcb_clnt_lock);

+unsigned int rpcb_users;

+

struct rpcbind_args {
struct rpc_xprt * r_xprt;

@@ -161,6 +164,56 @@ static void rpcb_map_release(void *data)

kfree(map);

}

+static int rpcb_get_local(void)

+{

```

+ int cnt;
+
+ spin_lock(&rpcb_clnt_lock);
+ if (rpcb_users)
+   rpcb_users++;
+ cnt = rpcb_users;
+ spin_unlock(&rpcb_clnt_lock);
+
+ return cnt;
+}
+
+void rpcb_put_local(void)
+{
+ struct rpc_clnt *clnt = rpcb_local_clnt;
+ struct rpc_clnt *clnt4 = rpcb_local_clnt4;
+ int shutdown;
+
+ spin_lock(&rpcb_clnt_lock);
+ if (--rpcb_users == 0) {
+   rpcb_local_clnt = NULL;
+   rpcb_local_clnt4 = NULL;
+ }
+ shutdown = !rpcb_users;
+ spin_unlock(&rpcb_clnt_lock);
+
+ if (shutdown) {
+   /*
+    * cleanup_rpcb_clnt - remove xprtsock's sysctls, unregister
+    */
+   if (clnt4)
+     rpc_shutdown_client(clnt4);
+   if (clnt)
+     rpc_shutdown_client(clnt);
+ }
+}
+
+static void rpcb_set_local(struct rpc_clnt *clnt, struct rpc_clnt *clnt4)
+{
+ /* Protected by rpcb_create_local_mutex */
+ rpcb_local_clnt = clnt;
+ rpcb_local_clnt4 = clnt4;
+ smp_wmb();
+ rpcb_users = 1;
+ dprintk("RPC:   created new rpcb local clients (rpcb_local_clnt: "
+   "%p, rpcb_local_clnt4: %p)\n", rpcb_local_clnt,
+   rpcb_local_clnt4);
+}
+

```

```
/*  
 * Returns zero on success, otherwise a negative errno value  
 * is returned.
```

Subject: [PATCH v7 2/7] SUNRPC: use rpcbind reference counting helpers
Posted by [Stanislav Kinsbursky](#) on Fri, 28 Oct 2011 10:52:44 GMT
[View Forum Message](#) <> [Reply to Message](#)

All is simple: we just increase users counter if rpcbind clients has been created already. Otherwise we create them and set users counter to 1.

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

```
---  
net/sunrpc/rpcb_clnt.c | 12 ++++-----  
1 files changed, 4 insertions(+), 8 deletions(-)  
  
diff --git a/net/sunrpc/rpcb_clnt.c b/net/sunrpc/rpcb_clnt.c  
index 9830d87..115df11 100644  
--- a/net/sunrpc/rpcb_clnt.c  
+++ b/net/sunrpc/rpcb_clnt.c  
@@ -258,9 +258,7 @@ static int rpcb_create_local_unix(void)  
     clnt4 = NULL;  
 }  
  
- /* Protected by rpcb_create_local_mutex */  
- rpcb_local_clnt = clnt;  
- rpcb_local_clnt4 = clnt4;  
+ rpcb_set_local(clnt, clnt4);  
  
out:  
     return result;  
@@ -312,9 +310,7 @@ static int rpcb_create_local_net(void)  
     clnt4 = NULL;  
 }  
  
- /* Protected by rpcb_create_local_mutex */  
- rpcb_local_clnt = clnt;  
- rpcb_local_clnt4 = clnt4;  
+ rpcb_set_local(clnt, clnt4);  
  
out:  
     return result;  
@@ -329,11 +325,11 @@ static int rpcb_create_local(void)  
     static DEFINE_MUTEX(rpcb_create_local_mutex);  
     int result = 0;
```

```

- if (rpcb_local_clnt)
+ if (rpcb_get_local())
    return result;

    mutex_lock(&rpcb_create_local_mutex);
- if (rpcb_local_clnt)
+ if (rpcb_get_local())
    goto out;

if (rpcb_create_local_unix() != 0)

```

Subject: [PATCH v7 4/7] SUNRPC: setup rpcbind clients if service requires it
 Posted by [Stanislav Kinsbursky](#) on Fri, 28 Oct 2011 10:53:07 GMT
[View Forum Message](#) <> [Reply to Message](#)

New function ("svc_uses_rpcbind") will be used to detect, that new service will send portmapper register calls. For such services we will create rpcbind clients and remove all stale portmap registrations.
 Also, svc_rpcb_cleanup() will be set as sv_shutdown callback for such services in case of this field wasn't initialized earlier. This will allow to destroy rpcbind clients when no other users of them left.

Note: Currently, any creating service will be detected as portmap user. Probably, this is wrong. But now it depends on program versions "vs_hidden" flag.

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

```

---
net/sunrpc/svc.c | 20 ++++++
1 files changed, 14 insertions(+), 6 deletions(-)

```

```

diff --git a/net/sunrpc/svc.c b/net/sunrpc/svc.c
index d2d61bf..87a67b2 100644

```

```

--- a/net/sunrpc/svc.c
+++ b/net/sunrpc/svc.c
@@ -436,10 +436,8 @@ __svc_create(struct svc_program *prog, unsigned int bufsize, int
npools,
    serv->sv_pools =
        kcalloc(serv->sv_nr_pools, sizeof(struct svc_pool),
            GFP_KERNEL);
- if (!serv->sv_pools) {
-     kfree(serv);
-     return NULL;
- }
+ if (!serv->sv_pools)
+     goto err_serv;

```

```

    for (i = 0; i < serv->sv_nrpoools; i++) {
        struct svc_pool *pool = &serv->sv_poools[i];
@@ -454,10 +452,20 @@ __svc_create(struct svc_program *prog, unsigned int bufsize, int
        npools,
        spin_lock_init(&pool->sp_lock);
    }

- /* Remove any stale portmap registrations */
- svc_unregister(serv);
+ if (svc_uses_rpcbind(serv)) {
+     if (svc_rpcb_setup(serv) < 0)
+         goto err_rpcb;
+ if (!serv->sv_shutdown)
+     serv->sv_shutdown = svc_rpcb_cleanup;
+ }

    return serv;
+
+err_rpcb:
+ kfree(serv->sv_poools);
+err_serv:
+ kfree(serv);
+ return NULL;
+ }

struct svc_serv *

```

Subject: [PATCH v7 3/7] SUNRPC: introduce svc helpers for preparing rpcbind infrastructure

Posted by [Stanislav Kinsbursky](#) on Fri, 28 Oct 2011 10:53:12 GMT

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

This helpers will be used only for those services, that will send portmapper registration calls.

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

```

---
include/linux/sunrpc/clnt.h | 2 ++
net/sunrpc/rpcb_clnt.c      | 2 +-
net/sunrpc/svc.c            | 35 ++++++
3 files changed, 38 insertions(+), 1 deletions(-)

```

```

diff --git a/include/linux/sunrpc/clnt.h b/include/linux/sunrpc/clnt.h
index db7bcaf..1eb437d 100644
--- a/include/linux/sunrpc/clnt.h

```

```

+++ b/include/linux/sunrpc/clnt.h
@@ -135,6 +135,8 @@ void rpc_shutdown_client(struct rpc_clnt *);
void rpc_release_client(struct rpc_clnt *);
void rpc_task_release_client(struct rpc_task *);

+int rpcb_create_local(void);
+void rpcb_put_local(void);
int rpcb_register(u32, u32, int, unsigned short);
int rpcb_v4_register(const u32 program, const u32 version,
    const struct sockaddr *address,
diff --git a/net/sunrpc/rpcb_clnt.c b/net/sunrpc/rpcb_clnt.c
index 115df11..c144b95 100644
--- a/net/sunrpc/rpcb_clnt.c
+++ b/net/sunrpc/rpcb_clnt.c
@@ -320,7 +320,7 @@ out:
 * Returns zero on success, otherwise a negative errno value
 * is returned.
 */
-static int rpcb_create_local(void)
+int rpcb_create_local(void)
{
    static DEFINE_MUTEX(rpcb_create_local_mutex);
    int result = 0;
diff --git a/net/sunrpc/svc.c b/net/sunrpc/svc.c
index 6a69a11..d2d61bf 100644
--- a/net/sunrpc/svc.c
+++ b/net/sunrpc/svc.c
@@ -354,6 +354,41 @@ svc_pool_for_cpu(struct svc_serv *serv, int cpu)
    return &serv->sv_pools[pidx % serv->sv_nrpoools];
}

+static int svc_rpcb_setup(struct svc_serv *serv)
+{
+    int err;
+
+    err = rpcb_create_local();
+    if (err)
+        return err;
+
+    /* Remove any stale portmap registrations */
+    svc_unregister(serv);
+    return 0;
+}
+
+static void svc_rpcb_cleanup(struct svc_serv *serv)
+{
+    svc_unregister(serv);
+    rpcb_put_local();

```

```

+}
+
+static int svc_uses_rpcbnd(struct svc_serv *serv)
+{
+ struct svc_program *progp;
+ unsigned int i;
+
+ for (progp = serv->sv_program; progp; progp = progp->pg_next) {
+ for (i = 0; i < progp->pg_nvers; i++) {
+ if (progp->pg_vers[i] == NULL)
+ continue;
+ if (progp->pg_vers[i]->vs_hidden == 0)
+ return 1;
+ }
+ }
+
+ return 0;
+}

/*
 * Create an RPC service

```

Subject: [PATCH v7 5/7] SUNRPC: cleanup service destruction
 Posted by [Stanislav Kinsbursky](#) on Fri, 28 Oct 2011 10:53:31 GMT
[View Forum Message](#) <> [Reply to Message](#)

svc_unregister() call have to be removed from svc_destroy() since it will be called in sv_shutdown callback.

This also means, that we have to call svc_rpcb_cleanup() explicitly from nfsd_last_thread() since this function is registered as service shutdown callback and thus nobody else will done it for us.

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

```

---
fs/nfsd/nfssvc.c      | 2 ++
include/linux/sunrpc/svc.h | 1 +
net/sunrpc/svc.c      | 4 +++
3 files changed, 5 insertions(+), 2 deletions(-)

```

```

diff --git a/fs/nfsd/nfssvc.c b/fs/nfsd/nfssvc.c
index dc5a1bf..52cd976 100644
--- a/fs/nfsd/nfssvc.c
+++ b/fs/nfsd/nfssvc.c
@@ -256,6 +256,8 @@ static void nfsd_last_thread(struct svc_serv *serv)
     nfsd_serv = NULL;
     nfsd_shutdown();

```



```

+ svc_rpcb_cleanup(serv);
+
  printk(KERN_WARNING "nfsd: last server has exited, flushing export "
    "cache\n");
  nfsd_export_flush();
diff --git a/include/linux/sunrpc/svc.h b/include/linux/sunrpc/svc.h
index 223588a..5e71a30 100644
--- a/include/linux/sunrpc/svc.h
+++ b/include/linux/sunrpc/svc.h
@@ -401,6 +401,7 @@ struct svc_procedure {
/*
 * Function prototypes.
 */
+void svc_rpcb_cleanup(struct svc_serv *serv);
struct svc_serv *svc_create(struct svc_program *, unsigned int,
  void (*shutdown)(struct svc_serv *));
struct svc_rqst *svc_prepare_thread(struct svc_serv *serv,
diff --git a/net/sunrpc/svc.c b/net/sunrpc/svc.c
index 87a67b2..96df2ba 100644
--- a/net/sunrpc/svc.c
+++ b/net/sunrpc/svc.c
@@ -367,11 +367,12 @@ static int svc_rpcb_setup(struct svc_serv *serv)
  return 0;
}

-static void svc_rpcb_cleanup(struct svc_serv *serv)
+void svc_rpcb_cleanup(struct svc_serv *serv)
{
  svc_unregister(serv);
  rpcb_put_local();
}
+EXPORT_SYMBOL_GPL(svc_rpcb_cleanup);

static int svc_uses_rpcbind(struct svc_serv *serv)
{
@@ -531,7 +532,6 @@ svc_destroy(struct svc_serv *serv)
  if (svc_serv_is_pooled(serv))
    svc_pool_map_put();

- svc_unregister(serv);
  kfree(serv->sv_pools);
  kfree(serv);
}

```

Subject: [PATCH v7 6/7] SUNRPC: remove rpcbind clients creation during service

registering

Posted by [Stanislav Kinsbursky](#) on Fri, 28 Oct 2011 10:54:34 GMT

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

We don't need this code since rpcbind clients are creating during RPC service creation.

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

net/sunrpc/rpcb_clnt.c | 9 -----

1 files changed, 0 insertions(+), 9 deletions(-)

diff --git a/net/sunrpc/rpcb_clnt.c b/net/sunrpc/rpcb_clnt.c

index c144b95..115c691 100644

--- a/net/sunrpc/rpcb_clnt.c

+++ b/net/sunrpc/rpcb_clnt.c

@@ -431,11 +431,6 @@ int rpcb_register(u32 prog, u32 vers, int prot, unsigned short port)

struct rpc_message msg = {

.rpc_argp = &map,

};

- int error;

-

- error = rpcb_create_local();

- if (error)

- return error;

dprintk("RPC: %sregistering (%u, %u, %d, %u) with local "

"rpcbind\n", (port ? "" : "un"),

@@ -571,11 +566,7 @@ int rpcb_v4_register(const u32 program, const u32 version,

struct rpc_message msg = {

.rpc_argp = &map,

};

- int error;

- error = rpcb_create_local();

- if (error)

- return error;

if (rpcb_local_clnt4 == NULL)

return -EPROTONOSUPPORT;

Subject: [PATCH v7 7/7] SUNRPC: remove rpcbind clients destruction on module cleanup

Posted by [Stanislav Kinsbursky](#) on Fri, 28 Oct 2011 10:55:38 GMT

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

Rpcbnd clients destruction during SUNRPC module removing is obsolete since now

those clients are destroying during last RPC service shutdown.

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

```
net/sunrpc/rpcb_clnt.c | 12 -----
net/sunrpc/sunrpc_syms.c | 3 ---
2 files changed, 0 insertions(+), 15 deletions(-)
```

```
diff --git a/net/sunrpc/rpcb_clnt.c b/net/sunrpc/rpcb_clnt.c
index 115c691..983b74f 100644
```

```
--- a/net/sunrpc/rpcb_clnt.c
```

```
+++ b/net/sunrpc/rpcb_clnt.c
```

```
@ @ -1100,15 +1100,3 @ @ static struct rpc_program rpcb_program = {
    .version = rpcb_version,
    .stats = &rpcb_stats,
};
```

```
-
```

```
-/**
```

```
- * cleanup_rpcb_clnt - remove xprtsock's sysctls, unregister
```

```
- *
```

```
- */
```

```
-void cleanup_rpcb_clnt(void)
```

```
-{
```

```
- if (rpcb_local_clnt4)
```

```
- rpc_shutdown_client(rpcb_local_clnt4);
```

```
- if (rpcb_local_clnt)
```

```
- rpc_shutdown_client(rpcb_local_clnt);
```

```
-}
```

```
diff --git a/net/sunrpc/sunrpc_syms.c b/net/sunrpc/sunrpc_syms.c
```

```
index 9d08091..8ec9778 100644
```

```
--- a/net/sunrpc/sunrpc_syms.c
```

```
+++ b/net/sunrpc/sunrpc_syms.c
```

```
@ @ -61,8 +61,6 @ @ static struct pernet_operations sunrpc_net_ops = {
```

```
extern struct cache_detail unix_gid_cache;
```

```
-extern void cleanup_rpcb_clnt(void);
```

```
-
```

```
static int __init
```

```
init_sunrpc(void)
```

```
{
```

```
@ @ -102,7 +100,6 @ @ out:
```

```
static void __exit
```

```
cleanup_sunrpc(void)
```

```
{
```

```
- cleanup_rpcb_clnt();
```

```
rpcauth_remove_module();
```

```
cleanup_socket_xprt();
svc_cleanup_xprt_sock();
```

Subject: Re: [PATCH v7 3/7] SUNRPC: introduce svc helpers for preparing rpcbind infrastructure

Posted by [bfields](#) on Fri, 28 Oct 2011 14:22:15 GMT

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

On Fri, Oct 28, 2011 at 02:52:59PM +0300, Stanislav Kinsbursky wrote:

> This helpers will be used only for those services, that will send portmapper
> registration calls.

>

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

Acked-by: J. Bruce Fields <bfields@redhat.com>

>

> ---

> include/linux/sunrpc/clnt.h | 2 ++

> net/sunrpc/rpcb_clnt.c | 2 +-

> net/sunrpc/svc.c | 35 ++++++++++++++++++++++++++++++++++++++

> 3 files changed, 38 insertions(+), 1 deletions(-)

>

> diff --git a/include/linux/sunrpc/clnt.h b/include/linux/sunrpc/clnt.h

> index db7bcaf..1eb437d 100644

> --- a/include/linux/sunrpc/clnt.h

> +++ b/include/linux/sunrpc/clnt.h

> @@ -135,6 +135,8 @@ void rpc_shutdown_client(struct rpc_clnt *);

> void rpc_release_client(struct rpc_clnt *);

> void rpc_task_release_client(struct rpc_task *);

>

> +int rpcb_create_local(void);

> +void rpcb_put_local(void);

> int rpcb_register(u32, u32, int, unsigned short);

> int rpcb_v4_register(const u32 program, const u32 version,

> const struct sockaddr *address,

> diff --git a/net/sunrpc/rpcb_clnt.c b/net/sunrpc/rpcb_clnt.c

> index 115df11..c144b95 100644

> --- a/net/sunrpc/rpcb_clnt.c

> +++ b/net/sunrpc/rpcb_clnt.c

> @@ -320,7 +320,7 @@ out:

> * Returns zero on success, otherwise a negative errno value

> * is returned.

> */

> -static int rpcb_create_local(void)

> +int rpcb_create_local(void)

> {

```

> static DEFINE_MUTEX(rpcb_create_local_mutex);
> int result = 0;
> diff --git a/net/sunrpc/svc.c b/net/sunrpc/svc.c
> index 6a69a11..d2d61bf 100644
> --- a/net/sunrpc/svc.c
> +++ b/net/sunrpc/svc.c
> @@ -354,6 +354,41 @@ svc_pool_for_cpu(struct svc_serv *serv, int cpu)
>     return &serv->sv_pools[pidx % serv->sv_nrpools];
> }
>
> +static int svc_rpcb_setup(struct svc_serv *serv)
> +{
> + int err;
> +
> + err = rpcb_create_local();
> + if (err)
> + return err;
> +
> + /* Remove any stale portmap registrations */
> + svc_unregister(serv);
> + return 0;
> +}
> +
> +static void svc_rpcb_cleanup(struct svc_serv *serv)
> +{
> + svc_unregister(serv);
> + rpcb_put_local();
> +}
> +
> +static int svc_uses_rpcbbind(struct svc_serv *serv)
> +{
> + struct svc_program *progp;
> + unsigned int i;
> +
> + for (progp = serv->sv_program; progp; progp = progp->pg_next) {
> + for (i = 0; i < progp->pg_nvers; i++) {
> + if (progp->pg_vers[i] == NULL)
> + continue;
> + if (progp->pg_vers[i]->vs_hidden == 0)
> + return 1;
> + }
> + }
> +
> + return 0;
> +}
>
> /*
>  * Create an RPC service

```

>
> --
> 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 v7 4/7] SUNRPC: setup rpcbind clients if service requires it
Posted by [bfields](#) on Fri, 28 Oct 2011 14:22:32 GMT

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

On Fri, Oct 28, 2011 at 02:53:18PM +0300, Stanislav Kinsbursky wrote:
> New function ("svc_uses_rpcbind") will be used to detect, that new service will
> send portmapper register calls. For such services we will create rpcbind
> clients and remove all stale portmap registrations.
> Also, svc_rpcb_cleanup() will be set as sv_shutdown callback for such services
> in case of this field wasn't initialized earlier. This will allow to destroy
> rpcbind clients when no other users of them left.
>
> Note: Currently, any creating service will be detected as portmap user.
> Probably, this is wrong. But now it depends on program versions "vs_hidden"
> flag.
>
> Signed-off-by: Stanislav Kinsbursky <skinsbursky@parallels.com>

Acked-by: J. Bruce Fields <bfields@redhat.com>

>
> ---
> net/sunrpc/svc.c | 20 ++++++-----
> 1 files changed, 14 insertions(+), 6 deletions(-)
>
> diff --git a/net/sunrpc/svc.c b/net/sunrpc/svc.c
> index d2d61bf..87a67b2 100644
> --- a/net/sunrpc/svc.c
> +++ b/net/sunrpc/svc.c
> @@ -436,10 +436,8 @@ __svc_create(struct svc_program *prog, unsigned int bufsize, int
npools,
> serv->sv_pools =
> kcalloc(serv->sv_nr_pools, sizeof(struct svc_pool),
> GFP_KERNEL);
> - if (!serv->sv_pools) {
> - kfree(serv);
> - return NULL;
> - }
> + if (!serv->sv_pools)
> + goto err_serv;
>

```

> for (i = 0; i < serv->sv_nrpoools; i++) {
> struct svc_pool *pool = &serv->sv_pools[i];
> @@ -454,10 +452,20 @@ __svc_create(struct svc_program *prog, unsigned int bufsize, int
npools,
> spin_lock_init(&pool->sp_lock);
> }
>
> - /* Remove any stale portmap registrations */
> - svc_unregister(serv);
> + if (svc_uses_rpcbind(serv)) {
> +     if (svc_rpcb_setup(serv) < 0)
> + goto err_rpcb;
> + if (!serv->sv_shutdown)
> + serv->sv_shutdown = svc_rpcb_cleanup;
> + }
>
> return serv;
> +
> +err_rpcb:
> + kfree(serv->sv_pools);
> +err_serv:
> + kfree(serv);
> + return NULL;
> }
>
> struct svc_serv *
>
> --
> 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 v7 5/7] SUNRPC: cleanup service destruction

Posted by [bfields](#) on Fri, 28 Oct 2011 14:22:50 GMT

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

On Fri, Oct 28, 2011 at 02:53:43PM +0300, Stanislav Kinsbursky wrote:

```

> svc_unregister() call have to be removed from svc_destroy() since it will be
> called in sv_shutdown callback.
> This also means, that we have to call svc_rpcb_cleanup() explicitly from
> nfsd_last_thread() since this function is registered as service shutdown
> callback and thus nobody else will done it for us.
>
> Signed-off-by: Stanislav Kinsbursky <skinsbursky@parallels.com>

```

Acked-by: J. Bruce Fields <bfields@redhat.com>

```

>
> ---
> fs/nfsd/nfssvc.c      | 2 ++
> include/linux/sunrpc/svc.h | 1 +
> net/sunrpc/svc.c      | 4 +---
> 3 files changed, 5 insertions(+), 2 deletions(-)
>
> diff --git a/fs/nfsd/nfssvc.c b/fs/nfsd/nfssvc.c
> index dc5a1bf..52cd976 100644
> --- a/fs/nfsd/nfssvc.c
> +++ b/fs/nfsd/nfssvc.c
> @@ -256,6 +256,8 @@ static void nfsd_last_thread(struct svc_serv *serv)
>  nfsd_serv = NULL;
>  nfsd_shutdown();
>
> + svc_rpcb_cleanup(serv);
> +
>  printk(KERN_WARNING "nfsd: last server has exited, flushing export "
>    "cache\n");
>  nfsd_export_flush();
> diff --git a/include/linux/sunrpc/svc.h b/include/linux/sunrpc/svc.h
> index 223588a..5e71a30 100644
> --- a/include/linux/sunrpc/svc.h
> +++ b/include/linux/sunrpc/svc.h
> @@ -401,6 +401,7 @@ struct svc_procedure {
>  /*
>   * Function prototypes.
>   */
> +void svc_rpcb_cleanup(struct svc_serv *serv);
> struct svc_serv *svc_create(struct svc_program *, unsigned int,
>   void (*shutdown)(struct svc_serv *));
> struct svc_rqst *svc_prepare_thread(struct svc_serv *serv,
> diff --git a/net/sunrpc/svc.c b/net/sunrpc/svc.c
> index 87a67b2..96df2ba 100644
> --- a/net/sunrpc/svc.c
> +++ b/net/sunrpc/svc.c
> @@ -367,11 +367,12 @@ static int svc_rpcb_setup(struct svc_serv *serv)
>  return 0;
>  }
>
> -static void svc_rpcb_cleanup(struct svc_serv *serv)
> +void svc_rpcb_cleanup(struct svc_serv *serv)
>  {
>   svc_unregister(serv);
>   rpcb_put_local();
>  }
> +EXPORT_SYMBOL_GPL(svc_rpcb_cleanup);
>

```



```
> static int svc_uses_rpcbind(struct svc_serv *serv)
> {
> @@ -531,7 +532,6 @@ svc_destroy(struct svc_serv *serv)
> if (svc_serv_is_pooled(serv))
>   svc_pool_map_put();
>
> - svc_unregister(serv);
> kfree(serv->sv_pools);
> kfree(serv);
> }
>
> --
> 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 v7 0/7] SUNRPC: make rpcbind clients allocated and destroyed dynamically

Posted by [bfields](#) on Fri, 28 Oct 2011 14:23:28 GMT

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

On Fri, Oct 28, 2011 at 02:52:09PM +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
> and rebased on tag "v3.1".
>
> v7:
> 1) Implemented "goto pattern" in __svc_create().
> 2) Pathes 5 and 6 from perious patch-set were squashed to make things looks
> clearer and to allow safe bisecting of the patch-set.
```

This series looks fine to me; thanks!

--b.

```
>
> v6:
> 1) Fixes in rpcb_clients management.
>
> v4:
> 1) creation and destruction on rpcbind clients now depends on service program
> versions "vs_hidden" flag.
>
> This patch is required for further RPC layer virtualization, because rpcbind
> clients have to be per network namespace.
> To achive this, we have to untie network namespace from rpcbind clients sockets.
> The idea of this patch set is to make rpcbind clients non-static. I.e. rpcbind
```

> clients will be created during first RPC service creation, and destroyed when
> last RPC service is stopped.
> With this patch set rpcbind clients can be virtualized easily.
>
> The following series consists of:
>
> ---
>
> Stanislav Kinsbursky (7):
> SUNRPC: introduce helpers for reference counted rpcbind clients
> SUNRPC: use rpcbind reference counting helpers
> SUNRPC: introduce svc helpers for prepairing rpcbind infrastructure
> SUNRPC: setup rpcbind clients if service requires it
> SUNRPC: cleanup service destruction
> SUNRPC: remove rpcbind clients creation during service registering
> SUNRPC: remove rpcbind clients destruction on module cleanup
>
>
> fs/nfsd/nfssvc.c | 2 +
> include/linux/sunrpc/clnt.h | 2 +
> include/linux/sunrpc/svc.h | 1
> net/sunrpc/rpcb_clnt.c | 88 ++++++-----
> net/sunrpc/sunrpc_syms.c | 3 -
> net/sunrpc/svc.c | 57 ++++++-----
> 6 files changed, 113 insertions(+), 40 deletions(-)
>
> --
> 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>
