
Subject: [PATCH v2 0/4] SUNRPC: register services with per-net rpcbind
Posted by [Stanislav Kinsbursky](#) on Fri, 13 Jan 2012 10:02:31 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.
head: devel

v2:

- 1) Rebased on up-to-date state ("devel" head).
 - 2) 3 patches from previous series:
 - a) [1/7] "SUNRPC: create rpcbind client in passed network namespace context"
 - b) [2/7] "SUNRPC: register rpcbind programs in passed network namespace context"
 - c) [4/7] "SUNRPC: parametrize local rpcbind clients creation with net ns"
- moved to series:
[PATCH v5 0/6] SUNRPC: rcbind clients virtualization

This patch set makes service registered with per-net rpcbind and required for making Lockd and NFSd services able to handle requests from and to different network namespaces.

The following series consists of:

Stanislav Kinsbursky (4):

- SUNRPC: use proper network namespace in rpcbind RPCBPROC_GETADDR procedure
- SUNRPC: pass network namespace to service registering routines
- SUNRPC: register service on creation in current network namespace
- SUNRPC: unregister service on creation in current network namespace

```
fs/nfsd/nfssvc.c      |  4 +-  
include/linux/sunrpc/svc.h | 11 ++++----  
net/sunrpc/rpcb_clnt.c |  3 +-  
net/sunrpc/svc.c       | 61 ++++++-----  
net/sunrpc/svcsock.c   |  3 +-  
5 files changed, 45 insertions(+), 37 deletions(-)
```

--

Signature

Subject: [PATCH v2 1/4] SUNRPC: use proper network namespace in rpcbind
RPCBPROC_GETADDR procedure
Posted by [Stanislav Kinsbursky](#) on Fri, 13 Jan 2012 10:02:40 GMT
[View Forum Message](#) <> [Reply to Message](#)

Pass request socket network namespace to rpc_uaddr2sockaddr() instead of hardcoded "init_net", when decoding address in RPCBPROC_GETADDR procedure.

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

```
---
net/sunrpc/rpcb_clnt.c | 3 +-
1 files changed, 2 insertions(+), 1 deletions(-)

diff --git a/net/sunrpc/rpcb_clnt.c b/net/sunrpc/rpcb_clnt.c
index 7c6fac0..4ce3a8e 100644
--- a/net/sunrpc/rpcb_clnt.c
+++ b/net/sunrpc/rpcb_clnt.c
@@ -934,7 +934,8 @@ static int rpcb_dec_getaddr(struct rpc_rqst *req, struct xdr_stream *xdr,
    dprintk("RPC: %5u RPCB_%s reply: %s\n", task->tk_pid,
        task->tk_msg.rpc_proc->p_name, (char *)p);

- if (rpc_uaddr2sockaddr(&init_net, (char *)p, len, sap, sizeof(address)) == 0)
+ if (rpc_uaddr2sockaddr(req->rq_xprt->xprt_net, (char *)p, len,
+     sap, sizeof(address)) == 0)
    goto out_fail;
    rpcb->r_port = rpc_get_port(sap);
```

Subject: [PATCH v2 2/4] SUNRPC: pass network namespace to service registering routines

Posted by [Stanislav Kinsbursky](#) on Fri, 13 Jan 2012 10:02:48 GMT

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

Lockd and NFSd services will handle requests from and to many network namespaces. And thus have to be registered and unregistered per network namespace.

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

```
---
include/linux/sunrpc/svc.h | 2 +-
net/sunrpc/svc.c           | 42 ++++++-----
net/sunrpc/svcsock.c      | 3 +-
3 files changed, 26 insertions(+), 21 deletions(-)

diff --git a/include/linux/sunrpc/svc.h b/include/linux/sunrpc/svc.h
index 35b37b1..d3563c2 100644
--- a/include/linux/sunrpc/svc.h
+++ b/include/linux/sunrpc/svc.h
@@ -428,7 +428,7 @@ void    svc_destroy(struct svc_serv *);
int      svc_process(struct svc_rqst *);
int      bc_svc_process(struct svc_serv *, struct rpc_rqst *,
```

```

    struct svc_rqst *);
-int   svc_register(const struct svc_serv *, const int,
+int   svc_register(const struct svc_serv *, struct net *, const int,
    const unsigned short, const unsigned short);

void   svc_wake_up(struct svc_serv *);
diff --git a/net/sunrpc/svc.c b/net/sunrpc/svc.c
index 8f6c255..03d2dfc 100644
--- a/net/sunrpc/svc.c
+++ b/net/sunrpc/svc.c
@@ -30,7 +30,7 @@

#define RPCDBG_FACILITY RPCDBG_SVCDSP

-static void svc_unregister(const struct svc_serv *serv);
+static void svc_unregister(const struct svc_serv *serv, struct net *net);

#define svc_serv_is_pooled(serv) ((serv)->sv_function)

@@ -375,13 +375,13 @@ static int svc_rpcb_setup(struct svc_serv *serv)
    return err;

    /* Remove any stale portmap registrations */
-   svc_unregister(serv);
+   svc_unregister(serv, &init_net);
    return 0;
}

void svc_rpcb_cleanup(struct svc_serv *serv)
{
-   svc_unregister(serv);
+   svc_unregister(serv, &init_net);
    rpcb_put_local(&init_net);
}
EXPORT_SYMBOL_GPL(svc_rpcb_cleanup);
@@ -790,7 +790,8 @@ EXPORT_SYMBOL_GPL(svc_exit_thread);
    * Returns zero on success; a negative errno value is returned
    * if any error occurs.
    */
-static int __svc_rpcb_register4(const u32 program, const u32 version,
+static int __svc_rpcb_register4(struct net *net, const u32 program,
+   const u32 version,
    const unsigned short protocol,
    const unsigned short port)
{
@@ -813,7 +814,7 @@ static int __svc_rpcb_register4(const u32 program, const u32 version,
    return -ENOPROTOOPT;
}

```

```

- error = rpcb_v4_register(&init_net, program, version,
+ error = rpcb_v4_register(net, program, version,
    (const struct sockaddr *)&sin, netid);

/*
@@ -821,7 +822,7 @@ static int __svc_rpcb_register4(const u32 program, const u32 version,
    * registration request with the legacy rpcbind v2 protocol.
    */
if (error == -EPROTONOSUPPORT)
- error = rpcb_register(&init_net, program, version, protocol, port);
+ error = rpcb_register(net, program, version, protocol, port);

return error;
}
@@ -837,7 +838,8 @@ static int __svc_rpcb_register4(const u32 program, const u32 version,
    * Returns zero on success; a negative errno value is returned
    * if any error occurs.
    */
-static int __svc_rpcb_register6(const u32 program, const u32 version,
+static int __svc_rpcb_register6(struct net *net, const u32 program,
+    const u32 version,
    const unsigned short protocol,
    const unsigned short port)
{
@@ -860,7 +862,7 @@ static int __svc_rpcb_register6(const u32 program, const u32 version,
    return -ENOPROTOOPT;
}

- error = rpcb_v4_register(&init_net, program, version,
+ error = rpcb_v4_register(net, program, version,
    (const struct sockaddr *)&sin6, netid);

/*
@@ -880,7 +882,7 @@ static int __svc_rpcb_register6(const u32 program, const u32 version,
    * Returns zero on success; a negative errno value is returned
    * if any error occurs.
    */
-static int __svc_register(const char *progname,
+static int __svc_register(struct net *net, const char *progname,
    const u32 program, const u32 version,
    const int family,
    const unsigned short protocol,
@@ -890,12 +892,12 @@ static int __svc_register(const char *progname,

switch (family) {
case PF_INET:
- error = __svc_rpcb_register4(program, version,

```

```

+ error = __svc_rpcb_register4(net, program, version,
    protocol, port);
    break;
#ifdef IS_ENABLED(CONFIG_IPV6)
    case PF_INET6:
- error = __svc_rpcb_register6(program, version,
+ error = __svc_rpcb_register6(net, program, version,
    protocol, port);
#endif
}
@@ -909,14 +911,16 @@ static int __svc_register(const char *progrname,
/**
 * svc_register - register an RPC service with the local portmapper
 * @serv: svc_serv struct for the service to register
+ * @net: net namespace for the service to register
 * @family: protocol family of service's listener socket
 * @proto: transport protocol number to advertise
 * @port: port to advertise
 *
 * Service is registered for any address in the passed-in protocol family
 */
-int svc_register(const struct svc_serv *serv, const int family,
- const unsigned short proto, const unsigned short port)
+int svc_register(const struct svc_serv *serv, struct net *net,
+ const int family, const unsigned short proto,
+ const unsigned short port)
{
    struct svc_program *progp;
    unsigned int i;
@@ -941,7 +945,7 @@ int svc_register(const struct svc_serv *serv, const int family,
    if (progp->pg_vers[i]->vs_hidden)
        continue;

- error = __svc_register(progp->pg_name, progr->pg_prog,
+ error = __svc_register(net, progr->pg_name, progr->pg_prog,
    i, family, proto, port);
    if (error < 0)
        break;
@@ -958,19 +962,19 @@ int svc_register(const struct svc_serv *serv, const int family,
 * any "inet6" entries anyway. So a PMAP_UNSET should be sufficient
 * in this case to clear all existing entries for [program, version].
 */
-static void __svc_unregister(const u32 program, const u32 version,
+static void __svc_unregister(struct net *net, const u32 program, const u32 version,
    const char *progrname)
{
    int error;

```

```

- error = rpcb_v4_register(&init_net, program, version, NULL, "");
+ error = rpcb_v4_register(net, program, version, NULL, "");

/*
 * User space didn't support rpcbind v4, so retry this
 * request with the legacy rpcbind v2 protocol.
 */
if (error == -EPROTONOSUPPORT)
- error = rpcb_register(&init_net, program, version, 0, 0);
+ error = rpcb_register(net, program, version, 0, 0);

dprintf("svc: %s(%sv%u), error %d\n",
        __func__, progname, version, error);
@@ -984,7 +988,7 @@ static void __svc_unregister(const u32 program, const u32 version,
 * The result of unregistration is reported via dprintf for those who want
 * verification of the result, but is otherwise not important.
 */
-static void svc_unregister(const struct svc_serv *serv)
+static void svc_unregister(const struct svc_serv *serv, struct net *net)
{
    struct svc_program *progp;
    unsigned long flags;
@@ -1001,7 +1005,7 @@ static void svc_unregister(const struct svc_serv *serv)

    dprintf("svc: attempting to unregister %sv%u\n",
            progp->pg_name, i);
- __svc_unregister(progp->pg_prog, i, progp->pg_name);
+ __svc_unregister(net, progp->pg_prog, i, progp->pg_name);
}
}

diff --git a/net/sunrpc/svcsock.c b/net/sunrpc/svcsock.c
index 4645709..e8af0c9 100644
--- a/net/sunrpc/svcsock.c
+++ b/net/sunrpc/svcsock.c
@@ -1409,7 +1409,8 @@ static struct svc_sock *svc_setup_socket(struct svc_serv *serv,

/* Register socket with portmapper */
if (*errp >= 0 && pmap_register)
- *errp = svc_register(serv, inet->sk_family, inet->sk_protocol,
+ *errp = svc_register(serv, sock->sk->sk_net, inet->sk_family,
+ inet->sk_protocol,
    ntohs(inet_sk(inet)->inet_sport));

if (*errp < 0) {

```

Subject: [PATCH v2 3/4] SUNRPC: register service on creation in current network namespace

Posted by [Stanislav Kinsbursky](#) on Fri, 13 Jan 2012 10:02:56 GMT

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

Service, using rpcbind (Lockd, NFSd) are starting from userspace call and thus we can use current network namespace.

There could be a problem with NFSd service, because it's creation can be called through NFSd fs from different network namespace. But this is a part of "NFSd per net ns" task and will be fixed in future.

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

net/sunrpc/svc.c | 9 +++++----

1 files changed, 5 insertions(+), 4 deletions(-)

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

index 03d2dfc..b2cc1b3 100644

--- a/net/sunrpc/svc.c

+++ b/net/sunrpc/svc.c

@ @ -20,6 +20,7 @ @

#include <linux/module.h>

#include <linux/kthread.h>

#include <linux/slab.h>

+#include <linux/nsproxy.h>

#include <linux/sunrpc/types.h>

#include <linux/sunrpc/xdr.h>

@ @ -366,16 +367,16 @ @ 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)

+static int svc_rpcb_setup(struct svc_serv *serv, struct net *net)

{

int err;

- err = rpcb_create_local(&init_net);

+ err = rpcb_create_local(net);

if (err)

return err;

/* Remove any stale portmap registrations */

- svc_unregister(serv, &init_net);

+ svc_unregister(serv, net);

return 0;

}

```

@@ -468,7 +469,7 @@ __svc_create(struct svc_program *prog, unsigned int bufsize, int npools,
}

if (svc_uses_rpcbind(serv)) {
-   if (svc_rpcb_setup(serv) < 0) {
+   if (svc_rpcb_setup(serv, current->nsproxy->net_ns) < 0) {
       kfree(serv->sv_pools);
       kfree(serv);
       return NULL;

```

Subject: [PATCH v2 4/4] SUNRPC: unregister service on creation in current network namespace

Posted by [Stanislav Kinsbursky](#) on Fri, 13 Jan 2012 10:03:04 GMT

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

On service shutdown we can be sure, that no more users of it left except current. Thus it looks like using current network namespace context is safe in this case.

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

```

---
fs/nfsd/nfssvc.c      |  4 ++--
include/linux/sunrpc/svc.h |  9 +++++----
net/sunrpc/svc.c      | 14 ++++++++-----
3 files changed, 14 insertions(+), 13 deletions(-)

```

```
diff --git a/fs/nfsd/nfssvc.c b/fs/nfsd/nfssvc.c
```

```
index eda7d7e..fce472f 100644
```

```
--- a/fs/nfsd/nfssvc.c
```

```
+++ b/fs/nfsd/nfssvc.c
```

```

@@ -251,13 +251,13 @@ static void nfsd_shutdown(void)
     nfsd_up = false;
}

```

```
-static void nfsd_last_thread(struct svc_serv *serv)
```

```
+static void nfsd_last_thread(struct svc_serv *serv, struct net *net)
```

```

{
    /* When last nfsd thread exits we need to do some clean-up */
    nfsd_serv = NULL;
    nfsd_shutdown();

```

```
- svc_rpcb_cleanup(serv);
```

```
+ svc_rpcb_cleanup(serv, net);
```

```

    printk(KERN_WARNING "nfsd: last server has exited, flushing export "
           "cache\n");

```



```

diff --git a/include/linux/sunrpc/svc.h b/include/linux/sunrpc/svc.h
index d3563c2..7b65495 100644
--- a/include/linux/sunrpc/svc.h
+++ b/include/linux/sunrpc/svc.h
@@ -84,7 +84,8 @@ struct svc_serv {
    unsigned int sv_nrpoools; /* number of thread pools */
    struct svc_pool * sv_pools; /* array of thread pools */

- void (*sv_shutdown)(struct svc_serv *serv);
+ void (*sv_shutdown)(struct svc_serv *serv,
+    struct net *net);
    /* Callback to use when last thread
     * exits.
     */
@@ -413,14 +414,14 @@ struct svc_procedure {
/*
 * Function prototypes.
 */
-void svc_rpcb_cleanup(struct svc_serv *serv);
+void svc_rpcb_cleanup(struct svc_serv *serv, struct net *net);
struct svc_serv *svc_create(struct svc_program *, unsigned int,
-    void (*shutdown)(struct svc_serv *));
+    void (*shutdown)(struct svc_serv *, struct net *net));
struct svc_rqst *svc_prepare_thread(struct svc_serv *serv,
    struct svc_pool *pool, int node);
void svc_exit_thread(struct svc_rqst *);
struct svc_serv * svc_create_pooled(struct svc_program *, unsigned int,
-    void (*shutdown)(struct svc_serv *),
+    void (*shutdown)(struct svc_serv *, struct net *net),
    svc_thread_fn, struct module *);
int svc_set_num_threads(struct svc_serv *, struct svc_pool *, int);
int svc_pool_stats_open(struct svc_serv *serv, struct file *file);
diff --git a/net/sunrpc/svc.c b/net/sunrpc/svc.c
index b2cc1b3..1bcfc3c 100644
--- a/net/sunrpc/svc.c
+++ b/net/sunrpc/svc.c
@@ -380,10 +380,10 @@ static int svc_rpcb_setup(struct svc_serv *serv, struct net *net)
    return 0;
}

-void svc_rpcb_cleanup(struct svc_serv *serv)
+void svc_rpcb_cleanup(struct svc_serv *serv, struct net *net)
{
-    svc_unregister(serv, &init_net);
-    rpcb_put_local(&init_net);
+    svc_unregister(serv, net);
+    rpcb_put_local(net);
}

```

```
EXPORT_SYMBOL_GPL(svc_rpcb_cleanup);
```

```
@@ -409,7 +409,7 @@ static int svc_uses_rpcbind(struct svc_serv *serv)
*/
```

```
static struct svc_serv *
```

```
__svc_create(struct svc_program *prog, unsigned int bufsize, int npools,
```

```
- void (*shutdown)(struct svc_serv *serv))
```

```
+ void (*shutdown)(struct svc_serv *serv, struct net *net))
```

```
{
```

```
    struct svc_serv *serv;
```

```
    unsigned int vers;
```

```
@@ -483,7 +483,7 @@ __svc_create(struct svc_program *prog, unsigned int bufsize, int npools,
```

```
struct svc_serv *
```

```
svc_create(struct svc_program *prog, unsigned int bufsize,
```

```
- void (*shutdown)(struct svc_serv *serv))
```

```
+ void (*shutdown)(struct svc_serv *serv, struct net *net))
```

```
{
```

```
    return __svc_create(prog, bufsize, /*npools*/1, shutdown);
```

```
}
```

```
@@ -491,7 +491,7 @@ EXPORT_SYMBOL_GPL(svc_create);
```

```
struct svc_serv *
```

```
svc_create_pooled(struct svc_program *prog, unsigned int bufsize,
```

```
- void (*shutdown)(struct svc_serv *serv),
```

```
+ void (*shutdown)(struct svc_serv *serv, struct net *net),
```

```
    svc_thread_fn func, struct module *mod)
```

```
{
```

```
    struct svc_serv *serv;
```

```
@@ -532,7 +532,7 @@ svc_destroy(struct svc_serv *serv)
```

```
    svc_close_all(&serv->sv_tempsocks);
```

```
    if (serv->sv_shutdown)
```

```
- serv->sv_shutdown(serv);
```

```
+ serv->sv_shutdown(serv, current->nsproxy->net_ns);
```

```
    svc_close_all(&serv->sv_permsocks);
```