
Subject: [PATCH v3 13/16] nfs: introduce mount option 'rpcmount'

Posted by [Kirill A. Shutemov](#) on Fri, 14 Jan 2011 13:49:11 GMT

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

It specifies rpc_pipefs to use. /var/lib/nfs/rpc_pipefs, by default.

Signed-off-by: Kirill A. Shutemov <kas@openvz.org>

```
fs/nfs/callback.c      |  5 ++-
fs/nfs/callback.h      |  3 +-
fs/nfs/client.c        | 46 ++++++-----
fs/nfs/internal.h      | 10 ++++++-
fs/nfs/mount_clnt.c    |  3 +-
fs/nfs/namespace.c     |  3 +-
fs/nfs/nfs4namespace.c | 22 ++++++-----
fs/nfs/super.c         | 20 ++++++-----
include/linux/nfs_fs_sb.h |  1 +
9 files changed, 86 insertions(+), 27 deletions(-)
```

diff --git a/fs/nfs/callback.c b/fs/nfs/callback.c

index 7a535c8..c9814fb 100644

--- a/fs/nfs/callback.c

+++ b/fs/nfs/callback.c

@@ -276,7 +276,8 @@ int nfs4_set_callback_sessionid(struct nfs_client *clp)

/*

* Bring up the callback thread if it is not already up.

*/

-int nfs_callback_up(u32 minorversion, struct rpc_xprt *xprt)

+int nfs_callback_up(u32 minorversion, struct rpc_xprt *xprt,

+ struct vfsmount *rpcmount)

{

struct svc_serv *serv = NULL;

struct svc_rqst *rqstp;

@@ -291,7 +292,7 @@ int nfs_callback_up(u32 minorversion, struct rpc_xprt *xprt)

nfs_callback_bc_serv(minorversion, xprt, cb_info);

goto out;

}

- serv = svc_create(&nfs4_callback_program, init_rpc_pipefs,

+ serv = svc_create(&nfs4_callback_program, rpcmount,

NFS4_CALLBACK_BUFSIZE, NULL);

if (!serv) {

ret = -ENOMEM;

diff --git a/fs/nfs/callback.h b/fs/nfs/callback.h

index d3b44f9..9496e0f 100644

--- a/fs/nfs/callback.h

+++ b/fs/nfs/callback.h

@@ -175,7 +175,8 @@ extern __be32 nfs4_callback_getattr(struct cb_getattrargs *args,

extern __be32 nfs4_callback_recall(struct cb_recallargs *args, void *dummy,

```

    struct cb_process_state *cps);
#ifdef CONFIG_NFS_V4
-extern int nfs_callback_up(u32 minorversion, struct rpc_xprt *xprt);
+extern int nfs_callback_up(u32 minorversion, struct rpc_xprt *xprt,
+ struct vfsmount *rpcmount);
extern void nfs_callback_down(int minorversion);
extern int nfs4_validate_delegation_stateid(struct nfs_delegation *delegation,
    const nfs4_stateid *stateid);
diff --git a/fs/nfs/client.c b/fs/nfs/client.c
index ad3b5e8..8dd0e99 100644
--- a/fs/nfs/client.c
+++ b/fs/nfs/client.c
@@ -131,6 +131,7 @@ struct nfs_client_initdata {
    const struct nfs_rpc_ops *rpc_ops;
    int proto;
    u32 minorversion;
+ struct vfsmount *rpcmount;
};

/*
@@ -167,6 +168,7 @@ static struct nfs_client *nfs_alloc_client(const struct nfs_client_initdata
*cl_
    clp->cl_rpcclient = ERR_PTR(-EINVAL);

    clp->cl_proto = cl_init->proto;
+ clp->cl_rpcmount = mntget(cl_init->rpcmount);

#ifdef CONFIG_NFS_V4
    err = nfs_get_cb_ident_idr(clp, cl_init->minorversion);
@@ -286,6 +288,7 @@ static void nfs_free_client(struct nfs_client *clp)
    if (clp->cl_machine_cred != NULL)
        put_rpccred(clp->cl_machine_cred);

+ mntput(clp->cl_rpcmount);
    kfree(clp->cl_hostname);
    kfree(clp);

@@ -471,6 +474,9 @@ static struct nfs_client *nfs_match_client(const struct nfs_client_initdata
*dat
    /* Match the full socket address */
    if (!nfs_sockaddr_cmp(sap, clap))
        continue;
+ /* Match rpc_pipefs mount point */
+ if (clp->cl_rpcmount->mnt_sb != data->rpcmount->mnt_sb)
+     continue;

    atomic_inc(&clp->cl_count);
    return clp;

```

```

@@ -629,7 +635,7 @@ static int nfs_create_rpc_client(struct nfs_client *clp,
    .program = &nfs_program,
    .version = clp->rpc_ops->version,
    .authflavor = flavor,
-   .rpcmount = init_rpc_pipefs,
+   .rpcmount = clp->cl_rpcmount,
};

    if (discrtry)
@@ -664,7 +670,7 @@ static void nfs_destroy_server(struct nfs_server *server)
/*
 * Version 2 or 3 lockd setup
 */
-static int nfs_start_lockd(struct nfs_server *server)
+static int nfs_start_lockd(struct nfs_server *server, struct vfsmount *rpcmount)
{
    struct nlm_host *host;
    struct nfs_client *clp = server->nfs_client;
@@ -675,7 +681,7 @@ static int nfs_start_lockd(struct nfs_server *server)
    .nfs_version = clp->rpc_ops->version,
    .noresvport = server->flags & NFS_MOUNT_NORESVPORT ?
        1 : 0,
-   .rpcmount = init_rpc_pipefs,
+   .rpcmount = rpcmount,
};

    if (nlm_init.nfs_version > 3)
@@ -823,8 +829,16 @@ static int nfs_init_server(struct nfs_server *server,
    cl_init.rpc_ops = &nfs_v3_clientops;
#endif

+   cl_init.rpcmount = get_rpc_pipefs(data->rpcmount);
+   if (IS_ERR(cl_init.rpcmount)) {
+       dprintk("<-- nfs_init_server() = error %ld\n",
+           PTR_ERR(cl_init.rpcmount));
+       return PTR_ERR(cl_init.rpcmount);
+   }
+
    /* Allocate or find a client reference we can use */
    clp = nfs_get_client(&cl_init);
+   mntput(cl_init.rpcmount);
    if (IS_ERR(clp)) {
        dprintk("<-- nfs_init_server() = error %ld\n", PTR_ERR(clp));
        return PTR_ERR(clp);
    }
@@ -856,7 +870,7 @@ static int nfs_init_server(struct nfs_server *server,
    server->acdirmax = data->acdirmax * HZ;

    /* Start lockd here, before we might error out */

```

```

- error = nfs_start_lockd(server);
+ error = nfs_start_lockd(server, clp->cl_rpcmount);
  if (error < 0)
    goto error;

@@ -1270,7 +1284,8 @@ static int nfs4_init_callback(struct nfs_client *clp)
{

  error = nfs_callback_up(clp->cl_mvops->minor_version,
-   clp->cl_rpcclient->cl_xprt);
+   clp->cl_rpcclient->cl_xprt,
+   clp->cl_rpcmount);
  if (error < 0) {
    dprintk("%s: failed to start callback. Error = %d\n",
      __func__, error);
@@ -1370,7 +1385,8 @@ static int nfs4_set_client(struct nfs_server *server,
  const char *ip_addr,
  rpc_authflavor_t authflavour,
  int proto, const struct rpc_timeout *timeparms,
- u32 minorversion)
+ u32 minorversion,
+ struct vfsmount *rpcmount)
{
  struct nfs_client_initdata cl_init = {
    .hostname = hostname,
@@ -1379,6 +1395,7 @@ static int nfs4_set_client(struct nfs_server *server,
    .rpc_ops = &nfs_v4_clientops,
    .proto = proto,
    .minorversion = minorversion,
+   .rpcmount = rpcmount,
  };
  struct nfs_client *clp;
  int error;
@@ -1485,6 +1502,7 @@ static int nfs4_init_server(struct nfs_server *server,
  const struct nfs_parsed_mount_data *data)
{
  struct rpc_timeout timeparms;
+ struct vfsmount *rpcmount;
  int error;

  dprintk("--> nfs4_init_server()\n");
@@ -1499,6 +1517,11 @@ static int nfs4_init_server(struct nfs_server *server,
  server->caps |= NFS_CAP_READDIRPLUS;
  server->options = data->options;

+ rpcmount = get_rpc_pipefs(data->rpcmount);
+ if (IS_ERR(rpcmount)) {
+   error = PTR_ERR(rpcmount);

```

```

+ goto error;
+ }
/* Get a client record */
error = nfs4_set_client(server,
    data->nfs_server.hostname,
@@ -1508,7 +1531,9 @@ static int nfs4_init_server(struct nfs_server *server,
    data->auth_flavors[0],
    data->nfs_server.protocol,
    &timeparms,
- data->minorversion);
+ data->minorversion,
+ rpcmount);
+ mntput(rpcmount);
if (error < 0)
    goto error;

@@ -1598,7 +1623,10 @@ struct nfs_server *nfs4_create_referral_server(struct
nfs_clone_mount *data,
    data->authflavor,
    parent_server->client->cl_xprt->prot,
    parent_server->client->cl_timeout,
- parent_client->cl_mvops->minor_version);
+ parent_client->cl_mvops->minor_version,
+ parent_client->cl_rpcmount);
+
+
if (error < 0)
    goto error;

@@ -1672,7 +1700,7 @@ struct nfs_server *nfs_clone_server(struct nfs_server *source,
(unsigned long long) server->fsid.major,
(unsigned long long) server->fsid.minor);

- error = nfs_start_lockd(server);
+ error = nfs_start_lockd(server, server->nfs_client->cl_rpcmount);
if (error < 0)
    goto out_free_server;

diff --git a/fs/nfs/internal.h b/fs/nfs/internal.h
index bfa3a34..b9554d7 100644
--- a/fs/nfs/internal.h
+++ b/fs/nfs/internal.h
@@ -86,6 +86,7 @@ struct nfs_parsed_mount_data {
    unsigned int version;
    unsigned int minorversion;
    char *fscache_uniq;
+ char *rpcmount;

```

```

struct {
    struct sockaddr_storage address;
@@ -120,6 +121,7 @@ struct nfs_mount_request {
    int noresvport;
    unsigned int *auth_flav_len;
    rpc_authflavor_t *auth_flavs;
+ struct vfsmount *rpcmount;
};

extern int nfs_mount(struct nfs_mount_request *info);
@@ -164,10 +166,14 @@ static inline void nfs_fs_proc_exit(void)

/* nfs4namespace.c */
#ifdef CONFIG_NFS_V4
-extern struct vfsmount *nfs_do_refmount(const struct vfsmount *mnt_parent, struct dentry
*dentry);
+extern struct vfsmount *nfs_do_refmount(const struct vfsmount *mnt_parent,
+ struct dentry *dentry,
+ struct vfsmount *rpcmount);
#else
static inline
-struct vfsmount *nfs_do_refmount(const struct vfsmount *mnt_parent, struct dentry *dentry)
+struct vfsmount *nfs_do_refmount(const struct vfsmount *mnt_parent,
+ struct dentry *dentry,
+ struct vfsmount *rpcmount)
{
    return ERR_PTR(-ENOENT);
}
diff --git a/fs/nfs/mount_clnt.c b/fs/nfs/mount_clnt.c
index 6227875..205cc02 100644
--- a/fs/nfs/mount_clnt.c
+++ b/fs/nfs/mount_clnt.c
@@ -13,7 +13,6 @@
#include <linux/in.h>
#include <linux/sunrpc/clnt.h>
#include <linux/sunrpc/sched.h>
-#include <linux/sunrpc/rpc_pipe_fs.h>
#include <linux/nfs_fs.h>
#include "internal.h"

@@ -162,7 +161,7 @@ int nfs_mount(struct nfs_mount_request *info)
    .program = &mnt_program,
    .version = info->version,
    .authflavor = RPC_AUTH_UNIX,
- .rpcmount = init_rpc_pipefs,
+ .rpcmount = info->rpcmount,
};
struct rpc_clnt *mnt_clnt;

```

```

int status;
diff --git a/fs/nfs/namespace.c b/fs/nfs/namespace.c
index 74aaf39..dad0129 100644
--- a/fs/nfs/namespace.c
+++ b/fs/nfs/namespace.c
@@ -144,7 +144,8 @@ static void * nfs_follow_mountpoint(struct dentry *dentry, struct
nameidata *nd)
    goto out_err;

    if (fattr->valid & NFS_ATTR_FATTR_V4_REFERRAL)
-   mnt = nfs_do_refmount(nd->path.mnt, nd->path.dentry);
+   mnt = nfs_do_refmount(nd->path.mnt, nd->path.dentry,
+   server->nfs_client->cl_rpcmount);
    else
        mnt = nfs_do_submount(nd->path.mnt, nd->path.dentry, fh,
            fattr);
diff --git a/fs/nfs/nfs4namespace.c b/fs/nfs/nfs4namespace.c
index 7a61fdb..92d5d63 100644
--- a/fs/nfs/nfs4namespace.c
+++ b/fs/nfs/nfs4namespace.c
@@ -14,7 +14,6 @@
#include <linux/slab.h>
#include <linux/string.h>
#include <linux/sunrpc/clnt.h>
-#include <linux/sunrpc/rpc_pipe_fs.h>
#include <linux/vfs.h>
#include <linux/inet.h>
#include "internal.h"
@@ -99,14 +98,13 @@ static int nfs4_validate_fspath(const struct vfsmount *mnt_parent,
}

static size_t nfs_parse_server_name(char *string, size_t len,
- struct sockaddr *sa, size_t salen)
+ struct sockaddr *sa, size_t salen, struct vfsmount *rpcmount)
{
    ssize_t ret;

    ret = rpc_pton(string, len, sa, salen);
    if (ret == 0) {
-   ret = nfs_dns_resolve_name(string, len, sa, salen,
-   init_rpc_pipefs);
+   ret = nfs_dns_resolve_name(string, len, sa, salen, rpcmount);
        if (ret < 0)
            ret = 0;
    }
@@ -115,7 +113,8 @@ static size_t nfs_parse_server_name(char *string, size_t len,

static struct vfsmount *try_location(struct nfs_clone_mount *mountdata,

```

```

    char *page, char *page2,
-    const struct nfs4_fs_location *location)
+    const struct nfs4_fs_location *location,
+    struct vfsmount *rpcmount)
{
    const size_t addr_bufsize = sizeof(struct sockaddr_storage);
    struct vfsmount *mnt = ERR_PTR(-ENOENT);
@@ -143,7 +142,7 @@ static struct vfsmount *try_location(struct nfs_clone_mount *mountdata,
    continue;

    mountdata->addrlen = nfs_parse_server_name(buf->data, buf->len,
-    mountdata->addr, addr_bufsize);
+    mountdata->addr, addr_bufsize, rpcmount);
    if (mountdata->addrlen == 0)
        continue;

@@ -174,7 +173,8 @@ static struct vfsmount *try_location(struct nfs_clone_mount *mountdata,
    */
    static struct vfsmount *nfs_follow_referral(const struct vfsmount *mnt_parent,
        const struct dentry *dentry,
-    const struct nfs4_fs_locations *locations)
+    const struct nfs4_fs_locations *locations,
+    struct vfsmount *rpcmount)
    {
        struct vfsmount *mnt = ERR_PTR(-ENOENT);
        struct nfs_clone_mount mountdata = {
@@ -213,7 +213,7 @@ static struct vfsmount *nfs_follow_referral(const struct vfsmount
    *mnt_parent,
        location->rootpath.ncomponents == 0)
        continue;

-    mnt = try_location(&mountdata, page, page2, location);
+    mnt = try_location(&mountdata, page, page2, location, rpcmount);
    if (!IS_ERR(mnt))
        break;
}
@@ -231,7 +231,9 @@ out:
    * @dentry - dentry of referral
    *
    */
-struct vfsmount *nfs_do_refmount(const struct vfsmount *mnt_parent, struct dentry *dentry)
+struct vfsmount *nfs_do_refmount(const struct vfsmount *mnt_parent,
+    struct dentry *dentry,
+    struct vfsmount *rpcmount)
    {
        struct vfsmount *mnt = ERR_PTR(-ENOMEM);
        struct dentry *parent;
@@ -264,7 +266,7 @@ struct vfsmount *nfs_do_refmount(const struct vfsmount *mnt_parent,

```

```

struct dentr
    fs_locations->fs_path.ncomponents <= 0)
    goto out_free;

- mnt = nfs_follow_referral(mnt_parent, dentry, fs_locations);
+ mnt = nfs_follow_referral(mnt_parent, dentry, fs_locations, rpcmount);
out_free:
    __free_page(page);
    kfree(fs_locations);
diff --git a/fs/nfs/super.c b/fs/nfs/super.c
index b68c860..be4852b 100644
--- a/fs/nfs/super.c
+++ b/fs/nfs/super.c
@@ -35,6 +35,7 @@
#include <linux/sunrpc/metrics.h>
#include <linux/sunrpc/xprtsock.h>
#include <linux/sunrpc/xprtrdma.h>
+#include <linux/sunrpc/rpc_pipe_fs.h>
#include <linux/nfs_fs.h>
#include <linux/nfs_mount.h>
#include <linux/nfs4_mount.h>
@@ -106,6 +107,7 @@ enum {
    Opt_lookupcache,
    Opt_fscache_uniq,
    Opt_local_lock,
+ Opt_rpcmount,

    /* Special mount options */
    Opt_userspace, Opt_deprecated, Opt_sloppy,
@@ -178,6 +180,7 @@ static const match_table_t nfs_mount_option_tokens = {
    { Opt_lookupcache, "lookupcache=%s" },
    { Opt_fscache_uniq, "fsc=%s" },
    { Opt_local_lock, "local_lock=%s" },
+ { Opt_rpcmount, "rpcmount=%s" },

    { Opt_err, NULL }
};
@@ -1486,6 +1489,13 @@ static int nfs_parse_mount_options(char *raw,
    return 0;
};
break;
+ case Opt_rpcmount:
+ string = match_strdup(args);
+ if (string == NULL)
+ goto out_nomem;
+ kfree(mnt->rpcmount);
+ mnt->rpcmount = string;
+ break;

```

```

/*
 * Special options
@@ -1646,11 +1656,19 @@ static int nfs_try_mount(struct nfs_parsed_mount_data *args,
    request salen = args->mount_server.addrlen;
    nfs_set_port(request.sap, &args->mount_server.port, 0);

+ request.rpcmount = get_rpc_pipefs(args->rpcmount);
+ if (IS_ERR(request.rpcmount)) {
+   dfprintk(MOUNT, "NFS: unable get rpc_pipefs mount point, "
+   "error %ld\n", PTR_ERR(request.rpcmount));
+   return PTR_ERR(request.rpcmount);
+ }
+
/*
 * Now ask the mount server to map our export path
 * to a file handle.
 */
status = nfs_mount(&request);
+ mntput(request.rpcmount);
if (status != 0) {
    dfprintk(MOUNT, "NFS: unable to mount server %s, error %d\n",
        request.hostname, status);
@@ -2355,6 +2373,7 @@ out:
    kfree(data->nfs_server.hostname);
    kfree(data->mount_server.hostname);
    kfree(data->fscache_uniq);
+ kfree(data->rpcmount);
    security_free_mnt_opts(&data->lsm_opts);
out_free_fh:
    nfs_free_fhandle(mntfh);
@@ -2962,6 +2981,7 @@ out:
    kfree(data->nfs_server.export_path);
    kfree(data->nfs_server.hostname);
    kfree(data->fscache_uniq);
+ kfree(data->rpcmount);
out_free_data:
    kfree(data);
    dprintk("<-- nfs4_get_sb() = %d%s\n", error,
diff --git a/include/linux/nfs_fs_sb.h b/include/linux/nfs_fs_sb.h
index b197563..ad8d913 100644
--- a/include/linux/nfs_fs_sb.h
+++ b/include/linux/nfs_fs_sb.h
@@ -36,6 +36,7 @@ struct nfs_client {
    struct list_head cl_share_link; /* link in global client list */
    struct list_head cl_superblocks; /* List of nfs_server structs */

+ struct vfsmount *cl_rpcmount; /* rpc_pipefs mount point */

```

```
struct rpc_clnt * cl_rpcclient;  
const struct nfs_rpc_ops *rpc_ops; /* NFS protocol vector */  
int cl_proto; /* Network transport protocol */
```

--

1.7.3.4

Containers mailing list

Containers@lists.linux-foundation.org

<https://lists.linux-foundation.org/mailman/listinfo/containers>
