
Subject: [PATCH 5/6] NFS: DNS resolver PipeFS notifier introduced
Posted by [Stanislav Kinsbursky](#) on Fri, 25 Nov 2011 13:19:22 GMT
[View Forum Message](#) <> [Reply to Message](#)

This patch subscribes DNS resolver caches to RPC pipefs notifications. Notifier is registering on NFS module load. This notifier callback is responsible for creation/destruction of PipeFS DNS resolver cache directory.
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/cache_lib.c | 4 ++--
fs/nfs/cache_lib.h | 4 ++++
fs/nfs/dns_resolve.c | 38 ++++++++++++++++++++++++++++++++++++++
3 files changed, 43 insertions(+), 3 deletions(-)

diff --git a/fs/nfs/cache_lib.c b/fs/nfs/cache_lib.c
index 5dd017b..5905a31 100644
--- a/fs/nfs/cache_lib.c
+++ b/fs/nfs/cache_lib.c
@@ -112,7 +112,7 @@ int nfs_cache_wait_for_upcall(struct nfs_cache_defer_req *dreq)
    return 0;
}

-static int nfs_cache_register_sb(struct super_block *sb, struct cache_detail *cd)
+int nfs_cache_register_sb(struct super_block *sb, struct cache_detail *cd)
{
    int ret;
    struct dentry *dir;
@@ -147,7 +147,7 @@ err:
    return ret;
}

-static void nfs_cache_unregister_sb(struct super_block *sb, struct cache_detail *cd)
+void nfs_cache_unregister_sb(struct super_block *sb, struct cache_detail *cd)
{
    if (cd->u.pipefs.dir)
        sunrpc_cache_unregister_pipefs(cd);
diff --git a/fs/nfs/cache_lib.h b/fs/nfs/cache_lib.h
index e0a6cc4..317db95 100644
--- a/fs/nfs/cache_lib.h
+++ b/fs/nfs/cache_lib.h
@@ -27,3 +27,7 @@ extern void nfs_cache_init(struct cache_detail *cd);
extern void nfs_cache_destroy(struct cache_detail *cd);
extern int nfs_cache_register_net(struct net *net, struct cache_detail *cd);
```

```

extern void nfs_cache_unregister_net(struct net *net, struct cache_detail *cd);
+extern int nfs_cache_register_sb(struct super_block *sb,
+ struct cache_detail *cd);
+extern void nfs_cache_unregister_sb(struct super_block *sb,
+ struct cache_detail *cd);
diff --git a/fs/nfs/dns_resolve.c b/fs/nfs/dns_resolve.c
index 9aea78a..200eb67 100644
--- a/fs/nfs/dns_resolve.c
+++ b/fs/nfs/dns_resolve.c
@@ -40,6 +40,7 @@ ssize_t nfs_dns_resolve_name(struct net *net, char *name, size_t namelen,
#include <linux/sunrpc/clnt.h>
#include <linux/sunrpc/cache.h>
#include <linux/sunrpc/svcauth.h>
+#include <linux/sunrpc/rpc_pipe_fs.h>

#include "dns_resolve.h"
#include "cache_lib.h"
@@ -400,12 +401,47 @@ void nfs_dns_resolver_cache_destroy(struct net *net)
    kfree(cd);
}

+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 cache_detail *cd = nn->nfs_dns_resolve;
+ int ret = 0;
+
+ if (cd == NULL)
+ return 0;
+
+ if (!try_module_get(THIS_MODULE))
+ return 0;
+
+ switch (event) {
+ case RPC_PIPEFS_MOUNT:
+ ret = nfs_cache_register_sb(sb, cd);
+ break;
+ case RPC_PIPEFS_UMOUNT:
+ nfs_cache_unregister_sb(sb, cd);
+ break;
+ default:
+ ret = -ENOTSUPP;
+ break;
+ }
+ module_put(THIS_MODULE);

```

```
+ return ret;
+}
+
+static struct notifier_block nfs_dns_resolver_block = {
+ .notifier_call = rpc_pipefs_event,
+};
+
+int nfs_dns_resolver_init(void)
+{
- return 0;
+ return rpc_pipefs_notifier_register(&nfs_dns_resolver_block);
+}

void nfs_dns_resolver_destroy(void)
+{
+ rpc_pipefs_notifier_unregister(&nfs_dns_resolver_block);
+}
#endif
```
