The NFS client

The NFS version 2 protocol was first documented in RFC1094 (March 1989). Since then two more major releases of NFS have been published, with NFSv3 being documented in RFC1813 (June 1995), and NFSv4 in RFC3530 (April 2003).

The Linux NFS client currently supports all the above published versions, and work is in progress on adding support for minor version 1 of the NFSv4 protocol.

The purpose of this document is to provide information on some of the upcall interfaces that are used in order to provide the NFS client with some of the information that it requires in order to fully comply with the NFS spec.

The DNS resolver

NFSv4 allows for one server to refer the NFS client to data that has been migrated onto another server by means of the special "fs_locations" attribute. See
        http://tools.ietf.org/html/rfc3530#section-6
and
        http://tools.ietf.org/html/draft-ietf-nfsv4-referrals-00

The fs_locations information can take the form of either an ip address and a path, or a DNS hostname and a path. The latter requires the NFS client to do a DNS lookup in order to mount the new volume, and hence the need for an upcall to allow userland to provide this service.

Assuming that the user has the 'rpc_pipefs' filesystem mounted in the usual /var/lib/nfs/rpc_pipefs, the upcall consists of the following steps:

(1) The process checks the dns_resolve cache to see if it contains a valid entry. If so, it returns that entry and exits.

(2) If no valid entry exists, the helper script '/sbin/nfs_cache_getent' (may be changed using the 'nfs.cache_getent' kernel boot parameter) is run, with two arguments:
- the cache name, "dns_resolve"
- the hostname to resolve

(3) After looking up the corresponding ip address, the helper script writes the result into the rpc_pipefs pseudo-file '/var/lib/nfs/rpc_pipefs/cache/dns_resolve/channel' in the following (text) format:

```
"<ip address> <hostname> <ttl>\n"
```

Where <ip address> is in the usual IPv4 (123.456.78.90) or IPv6 (ffee:ddcc:bbaa:9988:7766:5544:3322:1100, ffee::1100, ...) format. <hostname> is identical to the second argument of the helper script, and <ttl> is the 'time to live' of this cache entry (in units of seconds).

Note: If <ip address> is invalid, say the string "0", then a negative entry is created, which will cause the kernel to treat the hostname as having no valid DNS translation.

A basic sample /sbin/nfs_cache_getent

```
#!/bin/bash
#
ttl=600
#
cut=/usr/bin/cut
getent=/usr/bin/getent
rpc_pipefs=/var/lib/nfs/rpc_pipefs
#
die()
{
        echo "Usage: $0 cache_name entry_name"
        exit 1
}

[ $# -lt 2 ] && die
cachename="$1"
cache_path=${rpc_pipefs}/cache/${cachename}/channel

case "${cachename}" in
        dns_resolve)
                name="$2"
                result="$(${getent} hosts ${name} | ${cut} -f1 -d\ )"
                [ -z "${result}" ] && result="0"
                ;;
        *)
                die
                ;;
esac
echo "${result} ${name} ${ttl}" >${cache_path}
```