Public NFS

Brent Callaghan

`brent@eng.sun.com`

NFS on the Internet: History

NFS does not have a good reputation

- **Don’t do NFS on the Internet!**
- **Large (8K) UDP transfers hog bandwidth (bad netiquette)**
- **No congestion control**
- **Inefficient error recovery - lose one fragment lose ‘em all.**
- **No security**
- **Use a nice TCP-based protocol like FTP.**

10 Years Later

- NFS over TCP is “nice” - just like FTP.
  - Congestion control
  - Error recovery
- V3 gives us:
  - Large transfers (no more 8k limit)
  - READDIRPLUS (fewer turnarounds)
  - Weak cache consistency (fewer GETATTRs)
- Security still an issue for *all* Internet protocols
  - Not an issue for archive download (anon FTP)

NFS vs FTP

• Hardly known on Internet	• Anonymous FTP popular for file download
• Efficient in-kernel implementations handle high loads	• User-level daemon - fork, data copy, context switch overhead
• Single, persistent TCP connection	• Control connection + one connection per file (incoming)
• Good for file transfer and file access	• Transfer of whole files only
• Competitive NFS server market	• No FTP server competition

Why not use NFS then ?

- **Firewalls typically block NFS because:**
  - **NFS uses UDP - replay attacks are easy.**
  - **NFS and MOUNT protocol require negotiated port via portmapper.**
  - **No security with AUTH_UNIX**
- **Need to make NFS easier to filter**
  - **Use TCP**
  - **Allow mounts without MOUNT protocol**
  - **Use de-facto port 2049 for NFS service.**

Public Filehandle

- **Allow one filesystem to be exported with a filehandle already known to clients.**

  ```
     share -o ro,public /export/ftp
  ```

- **Version 2: filehandle is all zeros.**
- **Version 3: filehandle is zero length.**
- **Client can now access this filesystem without using MOUNT protocol.**

Multi-component Lookup

- **Single component LOOKUP is expensive on WANs.**
- **Server 10,000 miles away, 100 ms/component.**
- **LOOKUP relative to public filehandle interpreted as a multi-component lookup.**
- **Version 2 pathnames limited to 255 characters.**
- **Pathname syntax determined by server OS as it is for MOUNT protocol.**

Multi-component Lookup (cont)

“pub/proj/src/cmd/index.html”

LOOKUP `F01` “`pub`” -->

<-- FH=`F02`

LOOKUP `F02` “`proj`” -->

<-- FH=`F03`

LOOKUP `F03` “`src`” -->

<-- FH=`F04`

LOOKUP `F04` “`cmd`” -->

<-- FH=`F05`

LOOKUP `F05` “`index.html`” -->

<-- FH=`F06`

LOOKUP `000` “`pub/proj/src/cmd/index.html`” -->

<-- FH=`F06`

Example

Read   `server:/export/foo/bar`

```
client -> server    PORTMAP C GETPORT prog=100005 (MOUNT) vers=2 proto=UDP
server -> client    PORTMAP R GETPORT port=32824

client -> server    MOUNT2 C Mount “/export”
server -> client    MOUNT2 R Mount OK FH=009A

client -> server    PORTMAP C GETPORT prog=100003 (NFS) vers=2 proto=UDP
server -> client    PORTMAP R GETPORT port=2049

client -> server    NFS C LOOKUP FH=009A “foo”
server -> client    NFS R LOOKUP FH=3F05

client -> server    NFS C LOOKUP FH=3F05 “bar”
server -> client    NFS R LOOKUP FH=03FC

client -> server    NFS C READ FH=03FC 0 for 4096
server -> client    NFS R READ OK
```

Example

Read `server:/export/foo/bar` **with public filehandle**

```
client -> server    NFS C LOOKUP FH=0000 “export/foo/bar”
server -> client    NFS R LOOKUP FH=03FC

client -> server    NFS C READ FH=03FC 0 for 4096
server -> client    NFS R READ OK
```

NFS URL

- **Web clients can handle FTP URL’s (RFC 1630):**
  `ftp://server:port/path`
- **Why not NFS URL’s ?**
  `nfs://server:port/path`
- **Assume `/path` is relative to public filehandle on server and `:port` defaults to 2049 if omitted.**
- **Browser supports a subset of NFS protocol: LOOKUP, READDIR, READ, READLINK.**
- **Have modified NCSA Mosaic to do this.**

NFS URL: Performance

- NFS was designed for good performance. Implementations are highly tuned.
    - HTTP: ~500 ops/sec NFS: ~5,000 ops/sec
    - HTTP: ~100 ms NFS: ~10 ms
- NFS servers tightly integrated with OS.
- Competitive NFS server market - performance is everything.
- Anyone working on FTPD performance ?
- Fledgling HTTP server market.

NFS URL: Intranets

- **NFS servers already deployed in "Intranets"**
- **Current access by static mount (PC-NFS) or automount (UNIX).**
- **Web access requires co-located HTTP server or "`file:`" URL.**
- **NFS URL's provide fast access to NFS data from *any* desktop.**

NFS URL: Internet

- Hardly any NFS servers on Internet.
  This is an opportunity!
- Alternative to anonymous FTP:
  - `ftp://ftp.sunsite.unc.edu/pub/src/`
  - `nfs://ftp.sunsite.unc.edu/pub/src/`
- Replace FTP
- Faster alternative to HTTP for non-CGI pages.
- Web pages are full of relative URL’s to small files.

Status

- **Modified NCSA Mosaic available.**
- **Browser mods will be given to WWW Consortium Library of Common Code.**
- **Will publish an Informational RFC that describes NFS URL and requirements to support Public NFS.**
- **Solaris servers will support Public NFS.**
- **Comparative benchmarking of HTTP, FTP, NFS.**
- **Persuade browser vendors support NFS URLs.**
- **Persuade firewall admins to allow outgoing TCP connections on port 2049.**