37th IETF

NFS Version 4 BOF

Brent Callaghan & Mike Eisler

Sun Microsystems, Inc

brent@eng.sun.com
mre@eng.sun.com

Agenda

- **Motivation for this BOF** (5 min)
- **Overview of NFS protocol** (45 min)
- **WebNFS** (15 min)
- **NFS Version 4** (60 min)
- **Formation of Working Group** (20 min)

Motivation

- We have a clear understanding of NFS direction and protocol requirements.
- Want a protocol that benefits from wide participation and strong expertise attracted to IETF process.
- Want a protocol with broad agreement and strong design.
- Internet needs a non-proprietary file access protocol.
- Change control agreement based on RFC 1790

NFS Design Principles

1. Platform Independence	B
- Different client APIs	
- Different Filesystems	
2. Simple design	A-
- Easy to implement.	
- Simple failure modes	
3. Fast & Reliable Crash Recovery	A
- Server statelessness	
4. Transparent Access	A-
- Standard APIs	
- No recompilation or relinking	
5. Standards Compliance	B
- POSIX 1003.1	
6. Reasonable Performance	A+

NFS Developments

- 1986: PC-NFS
- 1986: First Connectathon
- 1987: Diskless Workstations
- 1987: Automounter
- 1987: Auspex
- 1988: Duplicate request cache
- 1989: Legato’s Prestoserve
- 1992: Network Appliance
- 1993: CacheFS
- 1994: SPEC SFS (LADDIS) Benchmark
- 1994: NFS Version 3
- 1996: WebNFS

NFS Version 2: Protocol Problems

- File size limitation: 32 bits = 4 GB
- Slow synchronous writes
- Arbitrary transfer limit (8k)
- Permissions and cached attributes (Access)
- READDIR of large directories
- Lack of cache consistency
- Overloaded fields (-1)

NFS Version 3 Definition

- Connectathon meeting 1992
- Participants: Auspex, Cray, DEC, Data General, HP, IBM, Legato, Network Appliance, OSF, Sun
- Boston meeting, July 1992
- "Working Group" via Email
- Design Considerations:
  - Keep It Simple!
  - Get it done in a year
  - Avoid anything contraversial

NFS Version 3

Extend fields, Boost performance, Fix protocol bugs

- 64 bit sizes and offsets
- WRITE/COMMIT (safe aynchronous writes on server)
- READDIRPLUS returns filehandles and attributes
- Increased transfer sizes (V2 has 8k limit)
- Fewer GETATTR calls (attrs piggyback on replies)
- ACCESS over the wire (supports caching & ACLs)
- Exclusive CREATE
- Support for weak cache consistency
- Support for POSIX pathconf
- Growing number of NFS V3 implementations

NFS V2 & V3

Version 2	Version 3
	ACCESS
	COMMIT
CREATE	CREATE
	FSINFO
	FSSTAT
GETATTR	GETATTR
LINK	LINK
LOOKUP	LOOKUP
MKDIR	MKDIR
	MKNOD
	PATHCONF
READ	READ
READDIR	READDIR
	READDIRPLUS
READLINK	READLINK
REMOVE	REMOVE
RENAME	RENAME
RMDIR	RMDIR
SETATTR	SETATTR
STATFS	
SYMLINK	SYMLINK
WRITE	WRITE

NFS V3 Performance

- **NFS V2 write performance is poor**
  - **Write gathering**
  - **NVRAM (Prestoserv)**
- **NFS V3 write performance is outstanding.**
  Comparisons of NFS Versions 2 and 3 by Digital:
  - **NFS Version 2 = 2,263 KB/s**
  - **NFS Version 3 = 6,105 KB/s**
- **READDIRPLUS**
  Running the find command on a tree with 155 directories and 9,612 files:
  - **NFS Version 2 = 10,559 over the wire calls**
  - **NFS Version 3 = 1,133 over the wire calls**

NFS Security

- Based on security of underlying RPC layer
- SunOS 1987: Diffie-Hellman exchange of DES keys
- SunOS 1989: Kerberos V4
- Solaris 1997: RPCSEC_GSS
  - RPC meets GSS-API
  - ONCRPC WG
  - `draft-ietf-oncrpc-rpcsec_gss-01.txt`
- V3 negotiates through MOUNT protocol
- Is NFS *inherently* insecure ? NO!

NFS Locking: NLM Protocol

- NFS Version 2
  - No paradigm for locking across heterogeneous platforms
- Locking moved into optional Network Lock Manager protocol.
- Complex protocol: “messaging” RPCs, status monitor, stateful recovery, grace period for lock reclaim.
- Buggy implementation via user-level lock daemons.
- Solaris NLM is re-written and in-kernel (much better)

NFS over TCP

- First implemented in BSD4.4
- Better error recovery than UDP
- Congestion control
- Good support for larger transfers of NFS V3
- Connections reestablished quickly after server crash.
- Equivalent to UDP performance on LANs
- Far better performance on WANs or Internet
- More implementations moving to TCP

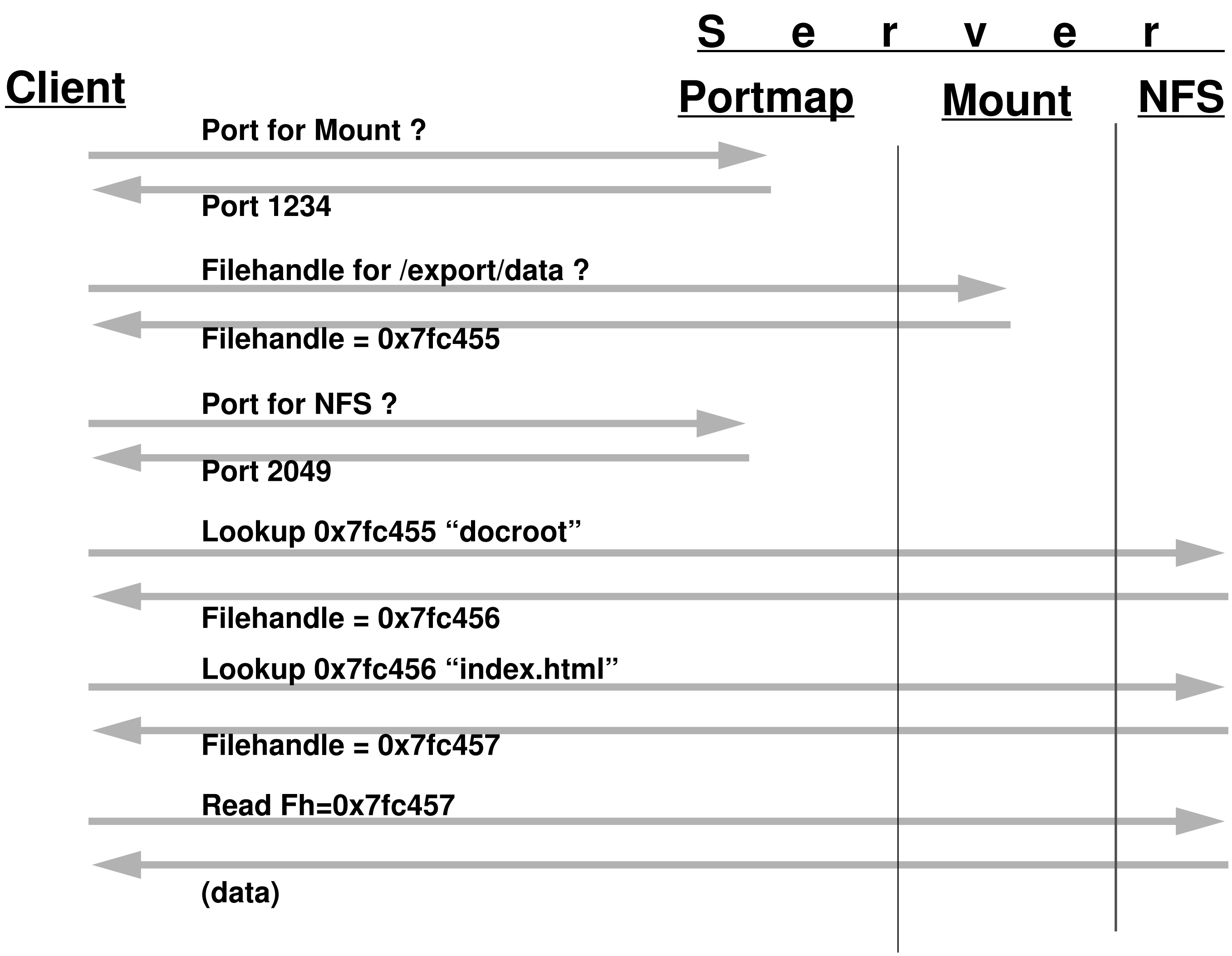
NFS on Internet

- NFS/TCP & NFS V3 use the Internet more efficiently.
- NFS servers are on the Internet now: e.g. wuarchive.wustl.edu
- NFS clients can browse remote archives with file browser.
- Access files without file transfer.

HOWEVER!

- Firewalls are a problem: portmap and MOUNT protocol
- WebNFS steps in.

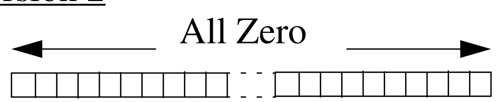
Access without WebNFS



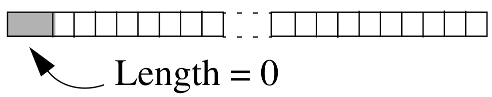
Public Filehandle

- **It's nothing really**

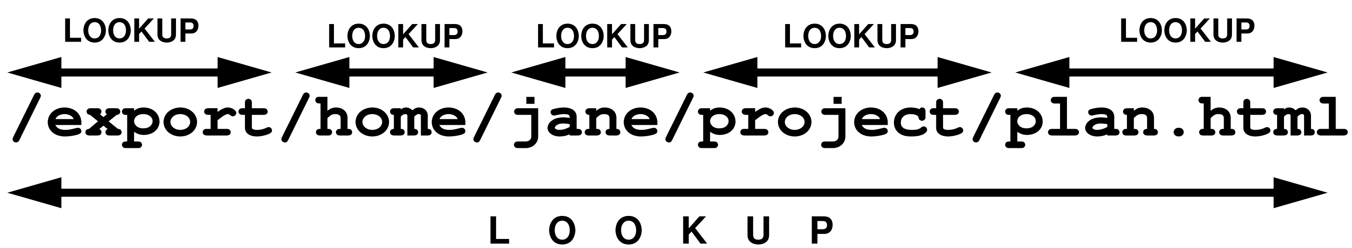
Version 2



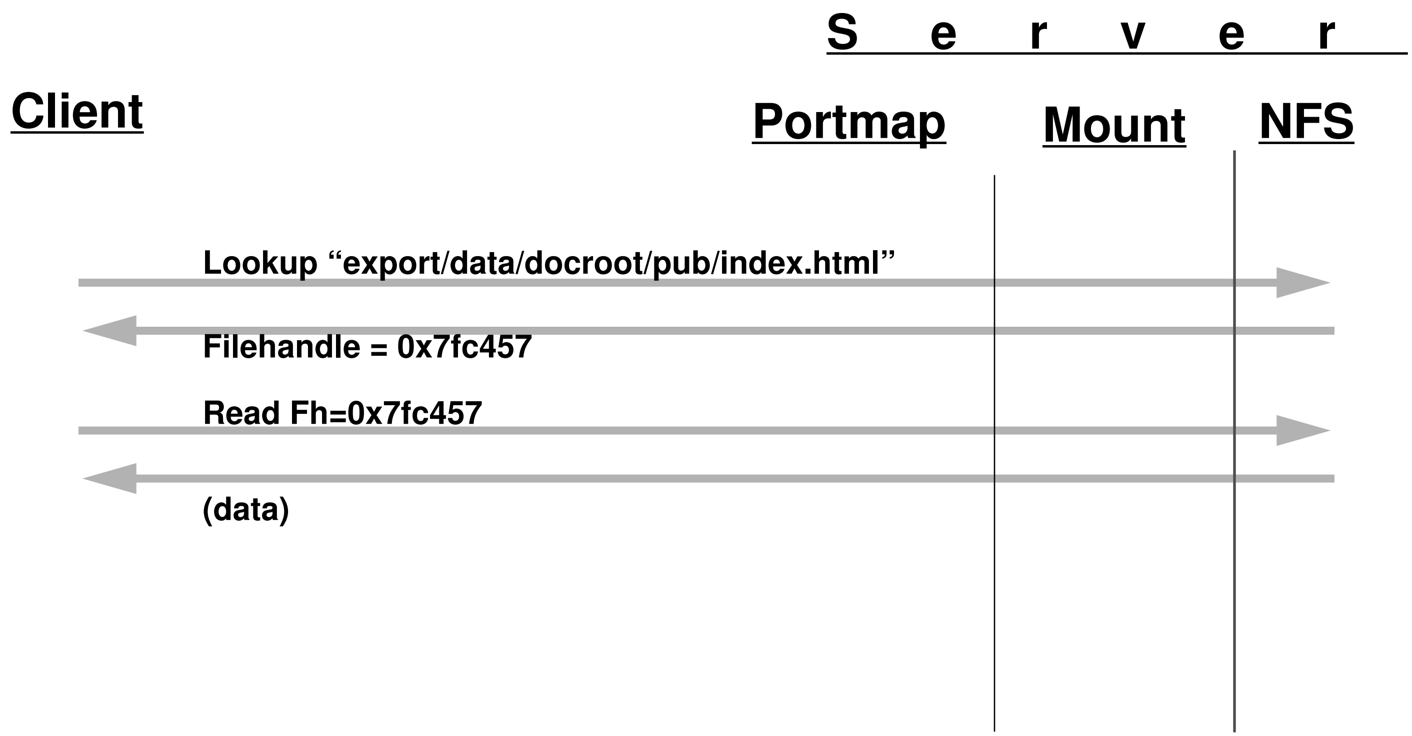
Version 3



- **Multi-component Lookup**



Access with WebNFS



WebNFS - Clients

- **WebNFS client RFC 2054**
- **Regular NFS clients - mount/automount from Internet servers, e.g.**
  `/net/wuarchive.wustl.edu/`
- **Lightweight Clients can use NFS directly:**
  - **Web browsers enable use of NFS URL's**
    `nfs://server/path`
  - **File transfer programs**

WebNFS - Server

- WebNFS server RFC 2055
- Small modification to existing v2 & v3 servers.
- New share options "`public`" and "`index`"

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

# share -o ro,public,index=index.html /export/docroot
```

- Not a complete HTTP replacement!
  - No CGI
  - No MIME headers

NFS URL

- Syntax: `nfs://server:port/pathname`
- Currently an Internet-Draft for URL scheme
  `draft-callaghan-url-nfs-00.txt`
- Locates a file on NFS servers anywhere
- Platform independent, names without backslashes!
- Name is location-dependent. URN's might be a solution.

WebNFS Issues

- Need security for the Internet
- No file locking
- Turnarounds: LOOKUP/READ
- No support for proxy caching

NFS Version 4

Mike Eisler

NFS V4 Working Group

- Initial Business
  - Charter. What does it expect to do ?
  - Milestones. When does it expect to do it ?
- Co-chairs
  - Brent Callaghan (Sun)
  - Spencer Shepler (IBM)
- Mailing List
  - `nfs4-wg@sunroof.eng.sun.com`
  - `nfs4-wg-request@sunroof.eng.sun.com`
- Strawman protocol spec
  - Available before first WG meeting