NFS version 4

Peter Åstrand

UppLYSning 2002–03–19

ISO 8601 är bra för dig

Agenda

- Översikt över v4. Historik
- Operationer & procedurer
- Tillstånd
- **Nyheter i v4**
- Nackdelar
- Jämförelse med CIFS
- Implementationer
- Frågor

Network File System (NFS)

- Ett av de första distribuerade filsystemen
- Utvecklades ursprungligen av Sun Microsystems
- Öppen standard
- Version 2 kom 1985. Version 3 blev klar 1995. WebNFS kom 1996.

NFS version 4

- Började utvecklas 1996
- Löser flertalet problem med tidigare versioner. Förbättringarna innefattar:
  - Anpassning för Internet
  - Bättre säkerhet
  - Bättre interoperabilitet mellan olika plattformar
- Utökningsbart protokoll

NFS version 4

- RFC3010 kom december 2000
- I mitten av 2001 upptäckte man ett allvarligt fel: Fillåsningar fungerar inte korrekt vid fork()
- Tillståndsmodellen har nu korrigerats
- Ny RFC beräknas komma vecka 13

NFS version 4

- NFSv4 är ett enda protokoll; inte en svit av protokoll som v2 och v3
  - Inget Mount–protokoll
  - Inget NLM–protokoll (för fillåsning). v4 har inbyggt stöd för fillåsning
  - Ingen status–monitor

XDR & RPC

- NFSv4 bygger precis som tidigare versioner på XDR och RPC
- Men: v4 använder en fast UDP–/TCP–port och kräver ej Portmapper

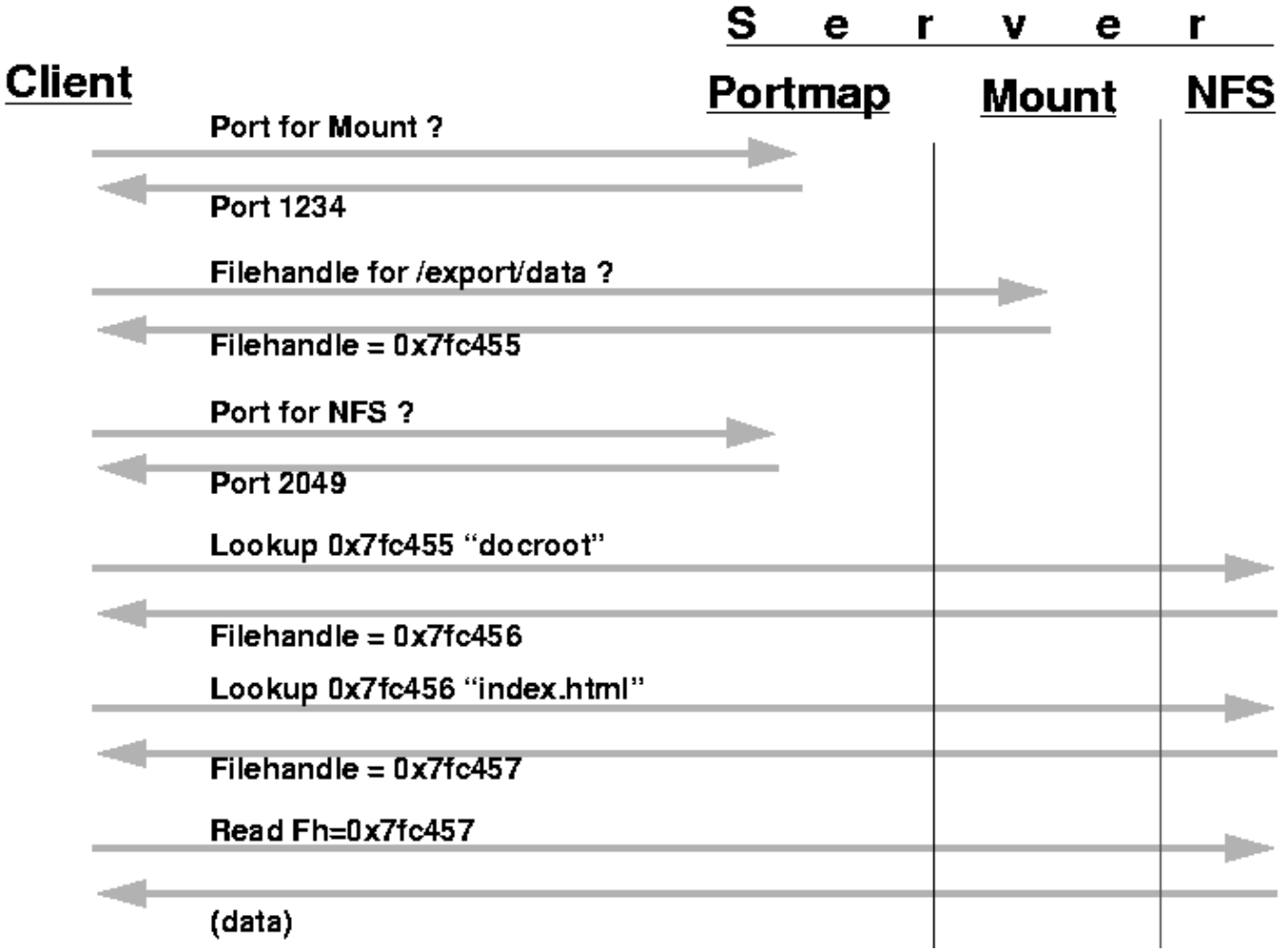
Nivå	Beskrivning	Exempel
7	Applikation	NFSv2/v3/v4
6	Presentation	XDR
5	Session	RPC, RPCSEC_GSS
4	Transport	UDP, TCP
3	Nätverk	IP
2	Länk	LLC
1	Fysisk	Ethernet 10base–T

Operationer / procedurer

- v4 har endast två RPC–procedurer (”remote procedures”): NULL och COMPOUND
- v4 defineras som ett antal *operationer*. En v4–operation motsvarar en RPC–procedur i v2/v3
- COMPOUND utför ett godtyckligt antal operationer. Fördel: kan minska antalet paket över nätet

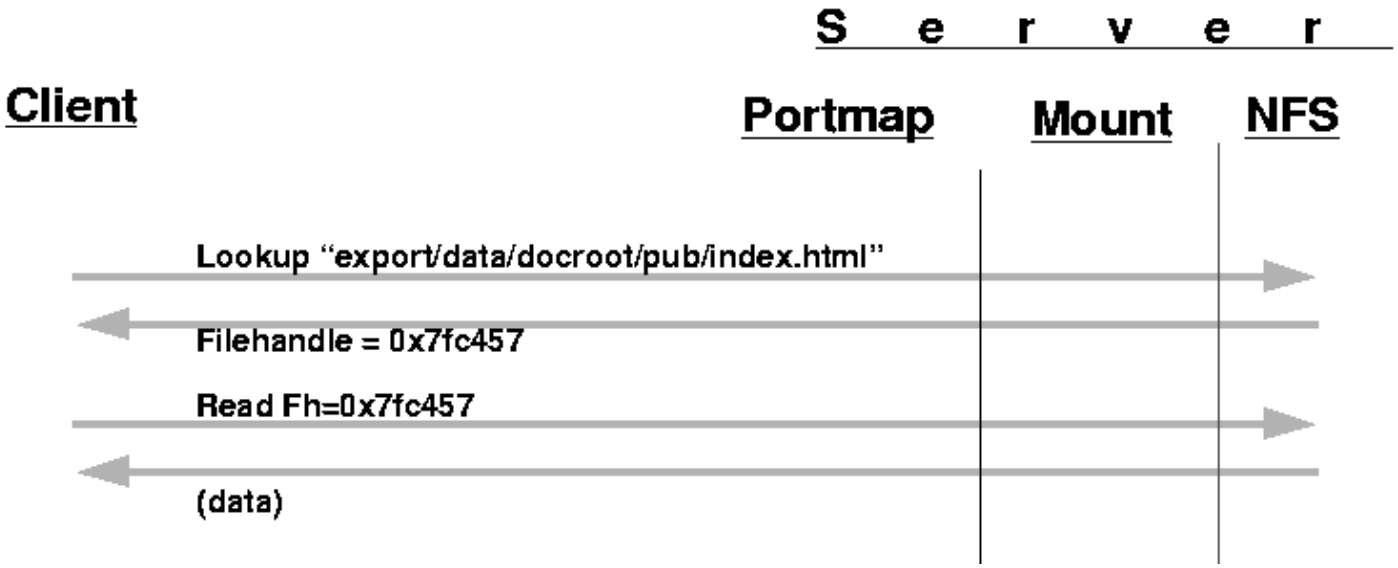
Compound

Läsa /export/data/docroot/index.html via v3:



Compound

Läsa /export/data/docroot/index.html via WebNFS:



- Med v4 & COMPOUND behövs (i teorin) endast en roundtrip
- Men: De flesta UNIX–VFS jobbar på komponentnivå => kan ej dra nytta av multi–component–lookup

Operationer

- v4 har 36 operationer
  - Låsning: LOCK LOCKT LOCKU RENEW
  - Katalog: LOOKUP LOOKUPP READDIR REMOVE RENAME ACCESS CREATE LINK
  - Filhandtag: PUTFH PUTPUBFH PUTROOTFH RESTOREFH SAVEFH GETFH
  - Filoperationer: OPEN OPEN_CONFIRM OPEN_DOWNGRADE READ READLINK CLOSE COMMIT WRITE
  - Attribut: SETATTR OPENATTR VERIFY NVERIFY
  - Special: SECINFO SETCLIENTID SETCLIENTID_CONFIRM DELEGPURGE DELEGRETURN

Tillstånd

- Tidigare NFS–versioner var, i teorin, tillståndslösa (”*stateless”)*
- Men: Låsning, t.ex. med NLM, kräver tillstånd
- I v4 är fillåsning inbyggt, vilket ger v4 tillstånd
- v4 har en OPEN–operation

Nyheter i v4

- Internet–anpassning
- Bättre säkerhet
- Fillåsning med Windowsstöd
- Delegering
- Plattformsstöd
- Filattribut
- Migrering/replikering
- Pseudo–fs

Internet–anpassning

- COMPOUND–proceduren löser problemet med fördröjning: många operationer kan skickas i samma paket
- TCP är obligatoriskt
- Enklare att köra v4 genom brandväggar: Mount– och Portmapperprotokollet borta
- Strängbaserade användar– och gruppid
- Fungerande säkerhetsmodell

Säkerhet

- Säkerheten i NFS bygger på säkerheten i RPC
- RPC definerar ett antal *security flavors*, t.ex. AUTH_SYS
- v4–implementationer måste stödja RPCSEC_GSS (”Secure RPC”), med mekanismerna Kerberos, LIPKEY och SPKM–3
- v4 assisterar RPCSEC via operationen SECINFO

Fillåsning

- De tidigare NLM–protokollen för v2/v3 har stora brister
- v4 stödjer alla typer av fillåsning för POSIX och Windows
- OPEN–operationen stödjer atomiska ”share reservations” för Windows/CIFS
- Fillåsningar baseras på lån (”leases”)
  - Låsningar måste förnyas med jämna mellanrum
  - Implicit förnyelse finns

Delegering

- Servern kan dela ut *delegeringar*: Klienten kan själv hantera OPEN/CLOSE/etc för en fil, utan att kontakta servern
- Klienten kan cacha data, även vid skrivning
- Servern kan återkalla delegering via ett RPC–anrop till klienten
- Liknar CIFS *oplocks*

Cross–platform

- v2/v3 var tänkt att vara plattformsoberende, men var det i praktiken inte
- v4 är bättre
  - Dock: Anpassningarna till UNIX och Windows
- Access Control Lists ingår (baserade på Windows modell)

Filattribut

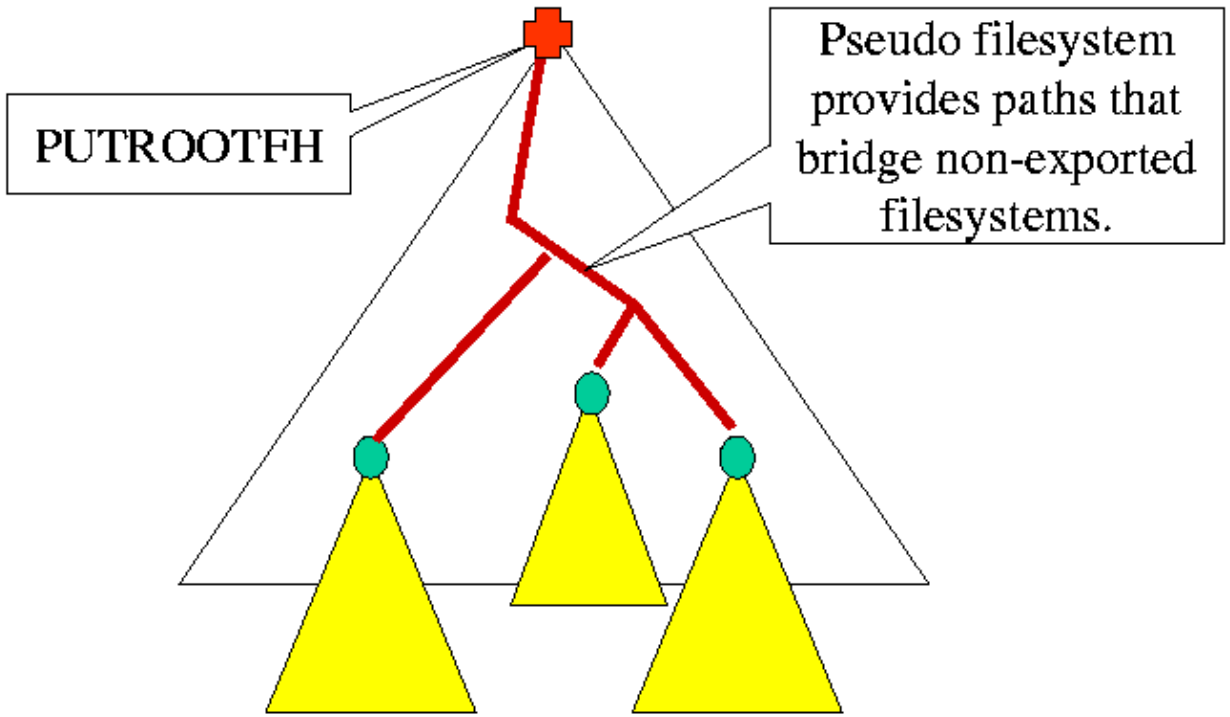
- v4 har tre klasser av filattribut: ”mandatory”, ”recommended” och ”named”
- ”mandatory” och ”recommended” har fasta positioner i en bitvektor
- ”named” möjliggör godtyckliga, namngivna attribut. Hanteras som filer

Migrering/replikering

- Protokollet möjliggör att servern signalerar att en fil har flyttats och vart, via attributet fs_locations
- fs_locations kan också användas för lastbalansering
- Hur migrering mellan servrar ska ske anges ej av protokollet. Ett separat protokoll kommer att utvecklas för detta

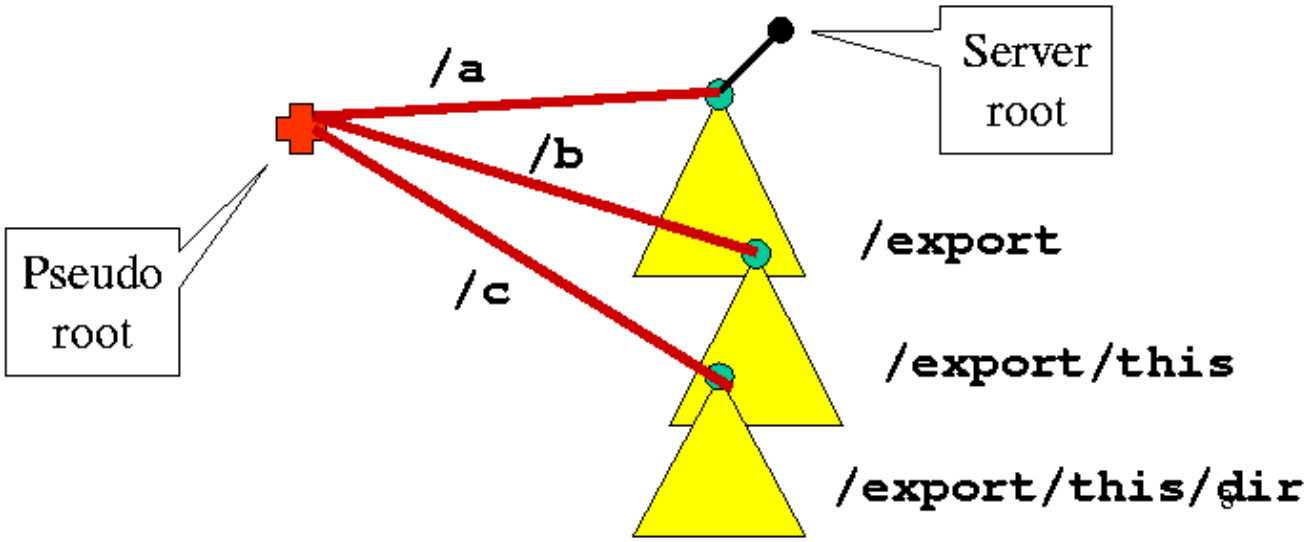
Pseudo–fs exports

- Mount–protokollet borta
- Utgår alltid ifrån serverns root



Pseudo–fs exports

- Linux är extra flexibelt och kan exportera godtycklig namnrymd
- Det exporterade trädet behöver ej matcha lokala sökvägar



Diverse

- UTF–8 används för filnamn etc
- v4 är utökningsbart: Regler för hur nya operationer, attribut etc ska hanteras
- Det finns ett valfritt fält, *tag*, som kan utnyttjas vid felsökning med sniffers etc

Nackdelar med v4

- Komplicerat protokoll
- Återhämtning efter krasch/omstarter komplicerat
- Stödjer ej *disconnected operation*, som t.ex. CODA gör
- I vissa fall långsammare än v2/v3

v4 jämfört med CIFS

- v4 ett betydligt renare protokoll, CIFS har byggts på lager på lager
- NFS är ”multiuser”, samma anslutning kan användas för ett helt system. Med CIFS måste varje användare ha en eget TCP–koppel
- CIFS har mycket overhead: varje paket har en 36–bytes header
- CIFS fungerar dåligt i POSIX–miljöer

Implementationer

- Hummingbird
  - Klient och server för Windows
- Sun
  - Klient och server i Solaris
- NetApp
- EMC
- Python
- Linux

Implementationer

- Python–implementationen (min)
  - RPC–kompilator
  - NFSv4–bibliotek
  - Testverktyg (nfs4st)
  - Kommandoradsklient

Implementationer

- Linux–implementationen
  - Utvecklas av CITI/University of Michigan
  - Sponsras av Sun och NetApp
  - Baserad på v3–kodbasen
  - Stödjer RPCSEC_GSS och delegeringar
  - Finns f.n. som patch till 2.4.4.
  - Klarar 97% av mina tester

Länkar

- NFSv4:
  http://www.nfsv4.org
- Linuximplementationen:
  http://www.citi.umich.edu/projects/nfsv4
- Min Pythonimplementation:
  http://www.cendio.se/~peter/pynfs/