NFS INDUSTRY CONFERENCE

NFSv4 Open Source Implemetation Update

William A. (Andy) Adamson

CITI

University of Michigan

andros@citi.umich.edu

NFS INDUSTRY CONFERENCE

Outline

- Brief history
- Roadmap and Status
- A scaling issue
- Related work

NFS INDUSTRY CONFERENCE

Brief History

- NFSv4 Open Source Reference Implementation project
- Sponsored by Sun Microsystems
- IETF Reference Implementation
- Linux client and server, FreeBSD Client

NFS INDUSTRY CONFERENCE

Brief History

- Fleshing out protocol spec
- Flushing out protocol bugs
- Complete 2.4 implementation, but isolated from NFSv2/v3

NFS INDUSTRY CONFERENCE

Brief History

- Delivered the critical building blocks in Linux 2.5
  - Complete rewrite
  - Integrated with NFSv2/v3
  - Identical performance
- Some pieces still to come in Linux 2.6
  - As “bug fixes” not new features

NFS INDUSTRY CONFERENCE

Road map for Linux 2.6

- Full featured NFSv4 Client and Server by years end
  - Share, byte-range lock, and delegation state
  - Kerberos v5
  - ACL
  - Reboot recovery

NFS INDUSTRY CONFERENCE

Linux 2.6 Status

- RPCSEC_GSS
  - Multiple mechanism framework
  - Kerberos v5 mechanism
    - Privacy coming soon
  - Kernel GSS context cache and up call
    - Server side in process
  - SPKM3 mechanism (PK based)
    - Submitted soon

NFS INDUSTRY CONFERENCE

Linux 2.6 Status

- Principal to ID kernel cache and ACL's
  - Client and server kernel cache and up call
    - Server side just submitted
  - POSIX ACL mapping implementation
    - Ready for submission, depends on ID mapping

NFS INDUSTRY CONFERENCE

Linux 2.6 Status

- State
  - Server: Open share state, and byte-range locking in 2.6
  - Client: Open share state, and byte-range locking re-write in progress
  - Delegation: client and server implementation coming soon

NFS INDUSTRY CONFERENCE

Linux 2.6 Status

- State Recovery
  - Client reboot recovery
    - Initial framework coded and tested
    - Needs integration with client state re-write
  - Server reboot recovery
    - Coming soon

NFS INDUSTRY CONFERENCE

Open State Scaling

- Whole file locking based on access and deny bits (Windows 'op locks')
- State: Open owner, open stateid
- Client presents open owner at OPEN
- Open stateid returned by server and binds open owner to open file
- READ/WRITE use open stateid

NFS INDUSTRY CONFERENCE

Open State Scaling

- Client: open owner is unit of serialization
  - One rpc in flight with OPEN, OPEN_CONFIRM, OPEN_DOWNGRADE,CLOSE
- Client chooses granularity of open owner
  - One open owner per pid
  - One open owner per credential

NFS INDUSTRY CONFERENCE

Open State Scaling

- Server: open owner is unit of state bookkeeping
  - Creates state for each new open owner
  - Releases state after last CLOSE
- Server: open stateid is bumped on each OPEN on an existing open owner/file tuple
  - Hard-links: READ/WRITE stateid possibly invalidated => client resends

NFS INDUSTRY CONFERENCE

Open State Scaling

- More open owners means less serialization on client, more state on server
  - Smallest number for saleability
  - Large enough number so that open owner serialization does not hurt client performance

NFS INDUSTRY CONFERENCE

Open State: Linux Client

- Pool of open owners
  - First: map one open owner per open file (lots of server state)
  - When OLD_STATEID recovery coded, client can respond to hard link READ/WRITE resends: map one open owner to many open files
- Discover how few open owners is needed

NFS INDUSTRY CONFERENCE

Lock State Scaling

- Same issues as Open State
- Small number of lock owners
  - Would like to use one for the whole client
- Client tests for local POSIX lock conflicts before putting request on the wire
  - If local conflicting lock, no RPC sent

NFS INDUSTRY CONFERENCE

FreeBSD Client

- Rebased from OpenBSD to FreeBSD
  - Target: Mac 10X client
- Vnode ops with Open state
- RPC layer separated from NFS
- Share user daemons with Linux
- Beginning submission process to FreeBSD kernel stream

NFS INDUSTRY CONFERENCE

Related Work

- Principal to ID mapping
  - New nsswitch services being prototyped
  - Secure SASL/GSSAPI LDAP mapping requests implemented
  - Client ACL tool designs being considered

NFS INDUSTRY CONFERENCE

Related Work

- Naming, Migration, and Replication
  - Global name space implementation
    - Extended DNS and Automount daemon
  - Migration and replication implementation
    - Use of the FS_LOCATIONS attribute
  - Mutable replication implementation
    - Server redirect, server to server protocol
- Work by Jiaying Zhang jiayingz@umich.edu

NFS INDUSTRY CONFERENCE

Related Work

- NFSv4 for cluster computers
  - Symmetric NFSv4 servers
    - Parallel access, NFSv4 state sharing
  - Experimenting with protocol extensions
    - For MPIO applications
    - For load balancing

NFS INDUSTRY CONFERENCE

Questions?!

http://www.citi.umich.edu