



**N I C
F N O
S D N
U F
S E
T R
R E
Y N
C
E**

Preparing Intel for NFSv4

Travis V Broughton
Sr. Systems Programmer
Intel Corporation
tvb@intel.com



**N I C
F N O
S D N
U F
S E
T R
R E
Y N
C
E**

Background

- Intel's Engineering Computing group
 - Supports Intel's hardware and software design efforts worldwide
 - Many regional organizations supporting dozens of campuses
 - Evolved from loose federation of individual IT groups
 - Different "standards" for data management



**N I C
F N O
S D N
U F
S E
T R
R E
Y N
C E**

Last Year...

- AFS* EOL will require changes
- NFSv4 looks like a good fit for global data sharing
- AFS to NFS migration non-trivial
 - Enabling technologies and tools needed

* Other names and brands may be claimed as the property of others



**N I C
F N O
S D N
U F
S E
T R
R E
Y N
E**

Objectives

- Standardize environments across sites
- Build a global data sharing architecture using off-the-shelf components
- Re-use as much existing infrastructure as possible



**N I C
F N O
S D N
U S F
T R E
R Y R E
N C E**

Global Data Sharing Components

- Filesystem protocol
- Consistent mount table across clients
 - Directory service for filesystem locations
- Directory service for groups
- User credentials



**N I C
F N O
S D N
U F
S E
T R
R E
Y N
C E**

Our Vision

- Filesystem: NFSv4
- Mounting mechanism: An automounter
 - am-utils? autofs?
- Groups: LDAP
- User credentials: Kerberos
 - Active Directory? MIT?



**N I C
F N O
S D N
U F
S E
T R
R E
Y N
C
E**

Other Goals

- Plan for evolution of components
 - Avoid niche solutions that are hard to remove
- Support legacy hardware
 - Continue to utilize NFSv3 servers that can't run NFSv4
 - Support NFSv3 clients as much as possible



**N I C
F N O
S D N
U F
S E
T R
R E
Y N
C
E**

Early Experiences

- First step: Implement consistent “global” namespace standard
 - Consistent automounter-managed filesystem layout
 - /nfs/<site>/..., very similar to AFS
 - Canonical location for all data
 - Encountered automounter issues
 - Native autofs not the same on all platforms
 - Found workarounds (sometimes code changes), and now have widespread deployment



**N I C
F N O
S D N
U F
S E
T R
R E
Y N
C
E**

Early Experiences

- Second Step: Replication/caching of data
 - Reduce dependency on WAN
 - Increase performance
 - User education needed – changes no longer visible as they are made
 - User-level tools needed for “emergency” synchronization



**N I C
F N O
S D N
U F
S E
T R
R E
Y N
C
E**

Early Experiences

- **Current Step: NFSv4 testing**
 - Participating in various beta programs
 - Using CITI Linux client
 - Saw NFSv4 Ops on first day of testing!
 - GSS-API configuration not as simple
 - Thus far, not much user/administrator-visible difference between NFSv3 and NFSv4



**N I C
F N O
S D N
U S F
T R E
R Y R
E N
C E**

Next Steps

- Continue testing NFSv4
 - Bottom out on Kerberos configuration
 - Microsoft vs. MIT KDC?
 - Automounter + Kerberos?
 - Client keytabs?
 - Cross-realm?
 - Need more experience with ACLs
 - Need more client platforms to ensure interoperability
 - “Mixed” exports – Kerberos for remote clients, netgroup-based exports for local legacy v3 clients



**N I C
F N O
S D N
U F
S E
T R
R E
Y N
E**

Community

- Last year's NFS Industry Conference
 - Co-located with Bakeathon
- Interim IETF WG meeting
 - Co-located with Bakeathon
- WG mailing list
- NFSv4 WG seems very receptive customer/community input



**N I C
F N O
S D N
U F
S E
T R
R E
Y N
C
E**

Conclusions

- “Roll-your-own” filesystem is a lot of work, but promises to be more flexible in the long run
- NFSv4 maturing rapidly
 - Optimistic about 2004 production use
- A lot has changed in the 10+ years we’ve been using AFS
 - Time to revisit design based on single one-size-fits-all solution vs. small set of optimized point solutions