

Adding Extended Attribute Support to NFS

James Morris
Red Hat, Inc.
jmorris@redhat.com

LinuxCon 2009 – Portland, OR

Introduction

Extended Attributes

- Arbitrary file metadata
- Two main types
 - System
 - User
- Non-standard, yet widely implemented

Types of Implementation

- Simple name / value pairs
 - user.foo = “bar”
 - Used by: IRIX, Linux, FreeBSD
- Subfiles / forks
 - May:
 - have file semantics (open, seek etc.)
 - include subdirectories
 - be highly structured
 - Used by Sun, Apple, Microsoft and others

Name/value model

- All name/value xattr implementations differ, have different APIs, and are generally incompatible
- Some OSs use flags for 'system', 'user' types
- Linux uses text namespaces
 - user, system, security, trusted, os2

Linux Implementation

- Simple and effective userland API
 - setxattr, getxattr, listxattr, removexattr
- Probably the most flexible such implementation
- System-level xattr use is common
 - ACLs, SELinux, SMACK, ecryptfs
- User-level xattrs: beagle, ...

NFS Support

- Extended attributes are not supported in the NFS standards
- NFSv3 is closed wrt IETF activity
- NFSv4 includes *Named Attributes*, which are based on the Solaris subfile model and incompatible with name/value schemes
- Several non-standard implementations shipped

Proposal for Linux NFSv3

- Implement simple name/value xattr side-protocol based on the GPL'd IRIX code
- ACL support was added to Linux NFSv3, and this is similar in scope and nature
- Rationale: all major filesystems have xattrs and we should be able to use them over the network!

Proposal for Linux NFSv3 (...)

- Side-protocol does not break existing protocol
- Possibility of interop and establishing a de-facto standard for BSD, legacy IRIX clients etc.
- NFS availability may encourage more xattr adoption (chicken & egg problem)

Linux NFSv3 Status

- Prototype code working!
- In upstream Linux community RFC process
- Initial code
 - Limited to *user* namespace
 - Not wire-compatible with IRIX due to differences in OS level implementation
 - Interoperability certainly possible; need to consider usefulness vs. code complexity

NFSv4

- NFSv4 needs xattr extension to properly support Linux/BSD etc.
 - Thankfully, NFSv4 was designed for extensibility!
- NFSv3 work should be a useful model for this
- IETF process could be lengthy

Discussion

Resources

- <http://namei.org/nfsxattr>
 - Code, documents
- <http://www.linux-nfs.org/>
 - Useful stepping off point
 - Linux NFS mailing list information