NFSv4.2 extension for POSIX ACLs

Proposed new NFSv4.2 attributes
These attributes support POSIX draft ACLs

4 new attributes
FATTR4_ACL_TRUEFORM
FATTR4_ACL_TRUEFORM_SCOPE
FATTR4_POSIX_DEFAULT_ACL
FATTR4_POSIX_ACCESS_ACL

Rationale

When a file server's True Form is POSIX draft ACL, the server must perform an inexact mapping between the NFSv4 ACL<->POSIX draft ACL

- This inexact mapping causes confusion and difficulties using ACLs on these NFSv4 servers
- Since ACLs are useful security tools, their use should be encouraged by providing a method for exact handling of the ACL

- True Form is the type of ACL stored for a file object and used for access permission checking
- The True Form Scope is per-File, per-Filesystem of per-Server as applied to True Form
- When the True Form is ACL_MODEL_POSIX_DRAFT the attributes FATTR4_POSIX_DEFAULT_ACL and FATTR4_POSIX_ACCESS_ACL define the ACL(s) used for the file object

The semantics of POSIX draft ACLs are described in a USENIX paper by Grunbacher and are not described in this document.

An ACE in a POSIX draft ACL consists of:

- tag (user_obj, user, group_obj, group, mask, other)
- Perm (rwx)
- For user and group, a uid or gid (which is handled by a “who” string for this extension)

Issues

- For VERIFY/NVERIFY the ACLs must be compared. The current draft states that a NFSv4.2 server may either reply NFS4ERR_INVAL or do the comparison

  To do the comparison, it recommends:

- ACLs with the same set of ACEs but in a different order be considered the same
- If possible, the comparison should be done by the server after the “who”->uid/gid mapping

Issues (continued)

When POSIX draft ACLs are in use, the group mode bits represent the Mask.

- This conflicts with the definition of the group mode bits in the NFSv4 RFCs
- This already occurs for servers (such as the Linux knfsd) that use POSIX draft ACLs as their true form
- A suggestion of having two sets of group mode bits seems awkward and unnecessary to me?