Outline

- Federated file systems requirements draft
  - Overview
  - Summary of draft-ellard-federated-fs-01.txt
  - Status
- SRV-based mechanism for finding namespace components
  - Overview
  - Summary of draft-everhart-nfsv4-namespace-via-dns-srv-01.txt
  - Status

Federated File Systems Requirements Draft

Daniel Ellard
Network Appliance, Inc

Overview

- We want a global, federated namespace
  - Global: namespace looks the same to equivalent observers
  - Federated:
    - Different parts of the namespace have different admins
    - As little centralized authority as possible
    - Minimal cross-member admin privs required
  - Generalized namespace structure
    - The path from the root to some object may traverse any number of federation members along the way
- We want a “minimal” set of requirements
  - Many suggested features left out

Implementing a Global Namespace via NFSv4.x

- The NFSv4.x “referral” mechanism provides the necessary machinery.
- “Refers” the client to a different location of an instance of the object that the client is trying to access.
  - Original purpose: migration (ERR_MOVED)
  - The client can ask for the location(s) of any filesystem
- Can also be used to knit together a multi-server namespace
  - Example:
    - Client does a lookup of /a/b/c on server foo
    - /a/b is on server foo, but /a/b/c is actually on server bar
    - Server foo refers the client to server bar

Requirements – a quick sketch

Junctions: used to link together namespaces.

- Example: we want any access to directory x:/a/b/c to be referred to the directory currently located at y:/d/e.
- Hardwired approach:
  - Put the info “y:/d/e” into junction x:/a/b/c.
- Why is that not enough?
  - The junction shouldn’t hardcode “y:”.
  - The junction shouldn’t hardcode “/d/e”.
- We need an extra layer of abstraction.
  - We want to link to an object, not the location of some arbitrary instance of that object at some arbitrary moment in time.

Junctions and resolution

- A junction contains:
  - An FSN: an identifier (UUID) for the target fileset.
  - An NSDB location: an identifier for a service that can resolve the FSN into a set of fs_locations.
- Junctions don't need updates
  - Update the NSDB, not the junction.
- When a client accesses a junction, the server uses the FSN and the NSDB location to find where to refer the client.
  - Simple approach: server asks the NSDB location about the FSN
  - Other approaches possible (proxies, etc)

Related requirements

To create a junction from x:/a/b to the object located at y:/c/d such that if a client accesses x:/a/b, she'll be referred to that object:

- Server x must be able to ***find*** the necessary info:
  - the FSN for the object at y:/c/d
  - the NSDB location responsible for that FSN
- Server y must be able to ***create*** this info:
  - The NSDB location responsible for that FSN
  - Create (or assign) an FSN to y:/c/d
  - Tell the NSDB location of the instance(s) of this FSN
- Admins must be able to ***update*** this info:
  - Add/delete a location
  - Add/delete/modify other metadata

Status

- Requirements draft seems to have converged
  - More requirements about metadata management, security, etc., that I didn't talk about today
- Next steps:
  - Draft a protocol that satisfies the requirements
    - Current plan: extend Glamour, a protocol from IBM Almaden
  - Submission of the protocol for review and comment
  - Reference implementation of the protocol
  - Always looking for collaborators:
    - federated-fs@sdsc.edu
    - ellard@netapp.com

Using DNS SRV to Specify a Global File Namespace

Craig Everhart
Network Appliance, Inc

Overview

- Goal: a simple AFS-like global namespace
- Paths in the namespace look like: /prefix/domain/etc…
  - The root is implemented locally on the client, and identified by the prefix.
  - The “domain” designates a root of a domain namespace
  - The rest of the path is interpreted by that domain.
- Examples:
  - /nfs4/mycompany.com/users/ellard
  - /nfs4/someuniversity.edu/courses/cs101

Resolution: connecting to the root

1. Client recognizes that the path needs to be resolved using the protocol, because it begins with (or contains) a designated directory.
2. The path component following the designated directory is the domain name where the rest of the path is implemented.
3. Client makes a DNS SVR RR query for the domain name.
4. Client gets back info about the root server(s) for the domain.
   - SVR records provide a level of indirection
   - There may be more than one root server
   - The name of the root server may be different than the domain
   - Root server set for a client may depend on who/where they are
5. Client chooses a root server, accesses the rest of the path.

Extensions and Subtleties

- Convention for root: /nfs4/
  - Convention, not hardwired.
  - In theory, could be anywhere in the client namespace
- Convention for r/w: /nfs4/.example.net
  - Like AFS: “.” for r/w
- DNS SRV RR can specify the port, but not much else
  - DNS SRV is not very expressive
  - Not a big problem: mount the root of the domain, query its fs_locations_info, and then mount the subdirectories appropriately
- Could be done in the server

Status

- IETF draft
- Looking for feedback