
# OpenAFS Roadmap

Russ Allbery

Jeffrey Altman

Derrick Brashear

# Wouldn't it be nice if...

- Everyone has their pet project.
  - Simon just told you how to get in on the action.
- Some projects have been integrated more quickly than others.
  - The usual barrier is how easily can the code be added and vetted, or added and proven to be able to be turned off.
  - Having a userbase known to be using the actual contributed code helps.
  - If it's being tested with unstable, even better.

# How CVS has held us back


- CVS doesn’t support distributed branches.
  - It could be done, by why bother?
- OpenAFS deltas are mostly analogous to Transarc (IBM) deltas.
  - However, the tools are crude.
- CVS was the obvious choice for an underlying repository in 2000.

# Why git isn't done yet


- Simon finally picked up the ball and ran with it.
  - Meeting with Shawn Pearce earlier this week pointed out some changes we want before switching "finally" to git.
- Different model for handing branches.
  - "pullups" no longer make sense.
- Changes to allow gerrit use.
  - Gerrit doesn't yet deal with branches.

# The good news

- 1.4.11 will be done from CVS.
- 1.5.61 will be done from git.
  - 1.5 branch will die in favor of HEAD.
- Future stable releases will be done from git

# Works in Progress - 1

- Windows native installable filesystem.
- “libosi” portability interface.
  - This may be very ambitious given the scope of the code.
  - More likely, we will start with part of libosi.
  - MCAS atomic operations will be included.
- Extended Callback interface.

# Works in Progress - 2

- Rx Kerberos 5 support.
  - Should be able to start pulling in pieces
- Rx OSD (pending integration)
  - Lack of access to suitable hardware for testing has been a second roadblock here.
    - DESY is providing a persistent test system.
- AFSCommander (pending integration)
  - This will be integrated when we move to git, so Claudio Bisegni can be delegated permission to continue working on it directly.

# Works in Progress - 3

- Byte range locking.
  - Unix clients.
    - Initially as local upgrades to whole-file locks.
  - Servers.
    - A protocol extension will be needed.
- Lock Delegation.
  - Give one client ownership of a range or a whole file for a fixed period of time.
  - All lock requests must be brokered through that client.

# Projects seeking support

- Improving performance on 10gbit networks and implementing support for IPv6.
  - Rx/TCP.
  - bulk data transfer.
- Removing directory limitations:
  - More than 64000 “blobs”.
  - POSIX Extended attributes.
  - Internationalization (Unicode).
  - Better hashing.

# More projects seeking support


- HostAFSd - local filesystem re-exporter
- Non-coding
  - Web site replacement
  - Toolsmithing (we need git tools)
  - Documentation (though we're much better off here now)
- Many more ...

# How can you help?

- Today
  - Contribute code.
  - Contribute non-code.
  - Works for hire.
- Tomorrow
  - That leads in to the OpenAFS governance discussion.

# OpenAFS Futures


Questions?