Getting started with AFS code

Perry Ruiter

EAKC2014
March 28

Agenda

- Background
- Environment/initial setup
- Where is the code?
- How do you build from the source?
- How to develop a fix and push it upstream?
- Simple debug techniques

Teaching an old (sled) dog new tricks

SINE NOMINE ASSOCIATES

- Live and work on the we(s)t coast of Canada
- 13 years as a VM systems programmer
  - provincial government
  - large mainframe environment
  - primarily email and relational database
- 15 years as an operating system developer
  - IBM z/VM
  - mostly assembler; a little C
- No prior experience with AFS, Kerberos, Linux

SINE NOMINE
ASSOCIATES

My development environment

- Macbook Pro running CentOS Linux as a VMWare guest
- Linux guest hosts test cell and clone of git source repository
- Lots of memory and an SSD means guest performance is surprisingly good
- SNA maintains a farm of supported servers

Hurdles encountered creating test cell

- Documentation not always correct
  - IBM Quickstart Guide
- Really need a guru for initial setup
  - Linux/Kerberos skills would have helped
- SNA has looked at providing automated cell creation appliance
  - Meffie's openafs-robotest on github

Where is the code?

- Code is maintained in a git repository
- Internally to Sine Nomine we have a mirror that I clone/update from
- Externally you would clone from git.openafs.org:
  - git clone git://git.openafs.org/openafs.git

How to rebuild from the source …

- Install prerequisite programs
  - varies by system
- Then simply standard make regimen
  - ./regen.sh
  - ./configure
  - make
  - make dest

How to rebuild from source

- make is extremely verbose
  - save the output somewhere
  - useful to check compiler options, etc.

- http://wiki.openafs.org/HowToBuildOpenAFSFromSource/
  - More details and recommended configure options

Develop a fix and push it upstream …

SINE NOMINE ASSOCIATES

- Configure git
  - name, email address, options, etc.
- Register with gerrit (code review tool)
  - need an OpenID and public key
  - scp -p -P 29418 gerrit.openafs.org:hooks/commit-msg .git/hooks/
- Ensure your repository is current
  - git pull --rebase
  - alternatively git fetch + git rebase

Develop a fix and push it upstream …

SINE NOMINE ASSOCIATES

- Create a new branch to contain your changes
  - git checkout –b *branchname*
- Now simply edit the source file(s) and make your changes
- git add .
- make, test, rinse, repeat

Develop a fix and push it upstream …

SINE NOMINE ASSOCIATES

- Create commit message
  - git commit
  - *subsystem*: *change summary*
    - keep length under 72 characters
  - blank line and fuller description
  - git commit --amend
- Giving up and starting over
  - git reset --hard

Develop a fix and push it upstream

SINE NOMINE ASSOCIATES

- Finally push to gerrit
  - git push ssh://gerrit.openafs.org/openafs.git HEAD:refs/for/master
- Community prefers several small incremental updates to a single large “big bang” update
- Other useful git commands
  - git status, git log
- http://wiki.openafs.org/GitDevelopers/

Debug tips …

- SELinux
  - disable for simple test environment
- rxdebug
  - lots of useful information
  - rxdebug *host port* [–version]
    - Ports 7000, 7001, 7002, 7003, 7007,…
- packet trace
  - UDP packets!

Debug tips

- fstrace (not to be confused with fs trace)
  - circular buffer of cache manager trace points
- increase process debug levels
  - sudo kill –TSTP $(pgrep vlserver)
  - sudo kill -HUP $(pgrep vlserver)
- http://docs.openafs.org/Reference/

SINE NOMINE ASSOCIATES

Contact: pruiter@sinenomine.net