OpenAFS
Status Report

European OpenAFS Workshop 2008
Graz, Austria

Derrick Brashear & Jeffrey Altman
24 Sep 2008

First things first

- If we’re going to have a live debug session, we need to know what to debug.
- Because after all, OpenAFS is bug-free, right?

This all sounds familiar

- New stable release coming. (1.4.8)
- New development releases continue apace. (1.5.53)

Recently resolved issues

- A packet leak in Rx
  - Fix targeted for 1.5.53 and 1.4.8 as of this writing
  - More on this later …

New in 1.4.8

- Support for partitions over 2tb.
- AIX 6
- FreeBSD 7 and current.

1.4.8 open issues

- Crash in afs_Analyze on RHEL4 (at least).
- Page mapping on Linux can succeed when the pages are not readable to a user, causing corruption to other accessors.
- A directory lookup issue caused by the fix to *linux-fakestat-avoid-mtpt-fillin-issue-20080415* (which is otherwise a legitimate fix).

New in 1.5.53

- Read-write disconnected AFS support.
  - Thanks to Dragos Tatulea, who did much of this work as part of the Google Summer of Code.
- Cache bypass support.
  - Contributed by Matt Benjamin.
  - Ported to and in use with Rx OSD by Hartmut Reuter.

1.5.53 open issues

- Demand Attach Fileserver bugs (more on that later)

Platform round-up

- AIX 5 and **6**
- **FreeBSD 7 and current**
- HP-UX 11.0, 11i v1 and v2
- Irix 6.5
- Linux 2.2, 2.4, 2.6 (ia32, ia64, x86_64, ppc, ppc64, **arm**, sparc, sparc64)
- MacOS 10.3, 10.4, 10.5.
- Solaris 2.6, 7, 8, 9, 10, 11 (and OpenSolaris)
- Microsoft Windows 2000 SP4 through Server 2008 (32-bit and 64-bit)

MacOS X

- Most of the issues with 10.4 were resolved for 10.5 with help from Apple.
- However, getting tokens at login is (now) “hard”.
- AFSCommander tool available, integration coming.
- Sadly, no kexts on the iPhone.

Linux

- iget() is dead.
  - Cache manager opens files by path, as in OSX, to deal.
- Usual AFS write-on-close semantics were restored in 1.4.7, where possible.
- ARM port finally available (to the rest of you).

ARM Linux

- Actually (I’ve) been kicking (it) around for years.
- empeg a.k.a. RioCar runs ARM Linux 2.4.
  - AFS in your car is sometimes useful.
  - Even (especially?) with no laptop.
- Nokia n810 (ARM Linux 2.6) was impetus for updating and integrating changes.

AIX

- A LAM plugin for Kerberos 5 based aklog is available and works with CDE Screenlock.
- The client properly supports AFSDB.
- AIX 6 support was contributed by Niklas Edmundsson.

BSD

- Previous work on FreeBSD had always stumbled on locking issues.
- Matt Benjamin revisited, fixed, and updated the previous work. He found the locking issues still lurked.
- Further work on OpenBSD and NetBSD has also been done but clients are not ready.

Unix/Linux Clients

- Actually not much exciting on clients.
- Numerous interaction issues with GUI environments have been addressed.
  - Notable exception: Finder (more later)
- Far fewer resources are leaked during client operation. We’re not perfect yet.
- No more gratuitous token disappearance.

Fileservers

- The salvager won’t corrupt directories anymore on 64 bit hosts thanks to good sleuthing by Rainer Toebbicke.
- Your larger-than-2tb partitions are now good to go.
  - Old “vos” clients may report odd numbers for empty partitions.

Playing nicely with other children

- Quotas enforced on TellMeAboutYourself /WhoAreYou calls to clients will preclude resource hogging.
- No more assert()s when a volume is found in an unexpected state.
- The server will never keep clients waiting forever for an answer (nor is the client that patient anymore) starting with 1.4.8.

And who were you, again?

- Client tracking turns out to be hard when clients lie (unbeknownst to themselves).
- Just because an address is reused does not mean it’s the same client.
- The fileserver now takes client address information with a lump of salt.

A common theme

- We’re (always) looking for volunteers.
- Sometimes we are better about asking than others.
- Today I will reiterate: please help us test 1.5!

Things to test in 1.5

- Cache bypass (Linux-only, new in 1.5.53)
  - Support for additional platforms is also needed.
- Split cache (dedicated portion for read-write data).
  - Early versions of this had corruption issues likely related to *writedcache-enforce-xdcache-writelock-20071208* ; thanks to Stephan Weisand for working with me on that (for quite a while).

It slices, it dices...

- Linux NFS translator continues to be updated as the Linux kernel does.
- Mountpointless volume addressing (/afs/.:mount/cell:volumeid/) is available.
  - Originally done for the Linux NFS translator.
  - On Windows, \\AFS\<cell><type><volume>\
- An extension allows any vnode to be used. (/afs/.:mount/cell:volumeid:vnodeid:uniquifier/)

But wait, there's more

- Read-write disconnected AFS support.
- Multiple (more than 2) local realms.
  - Requires username space to be the same. ("shadow" in any realm is the same person)
  - List realms one per line in /usr/afs/etc/krb.conf or equivalent.

Act now to receive this free gift

- Cache read-ahead. (Configurable window size, defaults to off)
    - Tunable with “*fs precache*”.
- Demand attach fileserver.

An aside on Demand Attach

- Known issues:
  - Volume headers for non-existent data on disk can remain in memory in a way that does not allow them to be purged.
  - Using current 1.5 without Demand Attach has some volume management bugs.
- Please share other issues if you have them!

Pending integration

- Rx connection “bundling” to allow more than 4 in-flight RPCs on a connection.
  - Some further tuning needed.
- Rx OSD.
  - More in the Roadmap.

More pending work

- Extended callback messages to optimize away unneeded traffic.
  - Both change “ranges” when data is stored, and metadata bundling when other things cause the callback.
  - Also possible to get finer-than-whole-volume notifications on releases.

Near term undertakings

- Locking enhancements for Unix clients (finally).
- Large payload (non-jumbogram) Rx packets.
- A fix for the MacOS “Finder cross-volume drag” issue.
  - A userspace helper and the ad-hoc “reference any vnode” semantics make this simple to solve.

And on the horizon

- Multiple volume snapshots.
- RxTCP.
- Directory object changes (Unicode, typed streams, more files, better hashing).
- Full Kerberos 5 support via rxk5.

Windows Client: New Features in Last 12 Months

- Vista SP1 and Server 2008 Certification
- Performance Improvements
  - Hash tables, Lock management redesign, Interlocked operations for reference counts
  - The client service has been profiled and bottlenecks removed. Up to 63 MB/sec data transfers on 64-bit Vista SP1; 54 MB/sec on 32-bit XP SP3
- Failover Improvements
  - Rxkad errors and Idle Data Timeouts
- Unicode character set support

Windows Client: More Improvements

- Directory Searches
  - B+ trees and local directory modifications
- Token management improvements
  - Try home realm first
  - No longer destroy token after RXKAD errors, instead fail over to the next server
- Volume Status Tracking
  - Volume Notification Plug-in Interface
- Constant time Server Probes
  - over 300 servers can be probed simultaneously
  - “fs checkservers –all” returns in just a few seconds

Windows Client: Even More Improvements

- Volume Group Management
- FollowBackupPath registry option
- .readonly Volume CB Optimizations
- Data Version optimizations
- cmdebug –cellservdb
- Out of Quota error reporting
- fs <command> –literal
- Rx Hot Threads

Windows Client: Quality Assurance Efforts

- Test Tools
  - Microsoft Windows Application Verifier
  - MIT File System Stress Test (Workshop ‘06)
  - Ziff Davis WinBench
  - Real World AFS Cell Access
    - ~100 cells, ~100,000 volumes, and millions of directories and files are accessed (~150,000 / day)
- Microsoft Windows Error Reports
  - Mini dumps provided from submissions the world over
- Microsoft File System Plug Fest
- Run-time State Validation

Windows Client: Known Bugs in 1.5.52 to be fixed in 1.5.53

- Lock Hierarchy violations that can result in deadlocks
- Race condition when recycling status cache object that can result in a panic due to a reference undercount
- Random access denied errors when multiple requests require a lock on the same directory object at the same time.
- Directory entries with trailing garbage

Windows Client: More Known Bugs

- Volume Move failover error
- Various memory leaks
- “fs flushXXX” does not destroy B+ trees
- Local directory corruption due to mixing of file server pages and locally modified pages
- Heap corruption during check server operations if new servers are discovered midstream
- File server lock synchronization not properly enforced during NTCreateX and NTTranCreate

Bugs in the Rx RPC Stack

- Multiple initialization of mutex objects
- rx_packet leaks
  - Never noticed; “rxdebug -rxstats” does not report the number of allocated packets
  - rx_call queues re-initialized when objects were still present
  - While reading, rx_packets would be lost
  - Calls reset while transmitting (due to #define error)
- Errors in the computation of the number of allocated packets combined with thread local free packet queues resulted in ever increasing packet allocations
- Rx NoJumbo did not disable use of Jumbo grams

Windows Client: 2008 Plans

- Release 1.5.53 as soon as possible
- Native File System Client
  - SMB interface and Loopback Adapter no longer required
  - “AFS” UNC path and drive letter access
  - Windows Cache and Memory Manager serves data directly from the AFS Cache paging file
  - Mount points reported as Reparse Points
  - Symlinks to Microsoft Dfs paths
  - Separate 64-bit and 32-bit SysNames
  - Public availability by the end of 2008
  - Supported Platforms XP SP2, 2003 SP1, Vista, 2008 (32/64)

Windows Client: Potential 2009 Projects

- Phase out SMB interface support
- Support for DOS and Extended Attributes
- New user interfaces
  - Improved Explorer Shell Extensions
  - Cache Manager and AFS Cell Management Consoles
  - New AFS Control Panel
    - PTS group management
    - AFS token acquisition / configuration (NetIdMgr AFS Provider)
- Read/Write Disconnected Operation Support
- AFS Servers on Microsoft Windows
  - Broken by UNIX Demand Attach File Server functionality
- Process Authentication Groups (native file system only)

Windows Client: OpenAFS and 64-bit Windows

- Here are some reasons to consider 64-bit Windows over 32-bit Windows
  - 64-bit Windows has been supported since XP 64 (April 2006)
  - Maximum Cache Size is ~1TB instead of ~1GB
  - 18% faster AFS cache access
  - Windows 7 will be the last 32-bit version

On version control

- git is coming.
- It should be much easier to track upstream as we work with you on integrating your changes.
- And it should be easier for us to merge them, too.
- More about this in the Roadmap.

Fresh developer blood

- Google Summer of Code accepted us.
  - Summaries available on the OpenAFS website.
  - More in a moment.
- The UIUC Capstone project is working with OpenAFS for the 2008-2009 academic year.
  - Their target project is an improved Windows Server Manager application.

Google Summer of Code

- Read/Write Disconnected AFS, with Simon Wilkinson.
  - Substantially complete. Support shipped in 1.5.53.
  - Cache contents pinning work ongoing by the original contributor, Dragos Tatulea.

Google Summer of Code

- Read/Write Replication, with Derrick Brashear.
  - Partial implementation in OpenAFS RT.
  - Missing pieces are master election, recovery handling, and slave lookup in master and clients.
  - The original contributor, Vishal Powar, cannot continue to work with us due to a new job.

Google Summer of Code

- Linux kAFS client updates, with David Howells.
  - OpenAFS pioctl support partially completed.
  - Contribution to linux-kernel is pending.
  - Jacob Thebault-Spieker intends to continue working with us, likely next on our web site.

In other news

- Lots more to come.
- At least I hope so: Apparently you get to listen to us many more times.
- Oh yeah, got anything for us to debug?

If your cell phone rang, you owe me a beer.

Fermented bubbly rice-water doesn't count.

Luckily here Budweiser is probably actual beer, yes?

Contact Information

Jeffrey Altman

jaltman *at* openafs *dot* org

Derrick Brashear

shadow *at* openafs *dot* org