OpenAFS Back to the Future

Derrick Brashear and Jeffrey Altman
The OpenAFS Project
13 Sep 2010

In the beginning

- *"The network is the computer"* - John Gage
- In the beginning, storage was expensive.
  - Centralized storage with a network filesystem was a path to bring more data to every machine.
  - AFS – CMU era, 1985-1990

As the mix changed

- Computers moved to places where network was more expensive.
  - Caching for offline usage became interesting.
  - AFS – Transarc era, 1991-

Full circle

- Storage is cheap.
  - Managing storage is more expensive.
  - Wide access to data is still critical.
  - Today and into the future.

The AFS model

- Originally, storage was concentrated on a few central nodes.
- Networks were fast and reliable, relatively.
- NAT was unheard-of.

Turning the model on its head

- Nearly every computer has a decent chuck of storage.
- Challenges
  - Allowing for greater use of that storage
  - Not sacrificing manageability.

The basics exist

- Network-based
  - Don’t need the storage to be on “this” node.
- Replication
  - Allows use of empty storage as added reliability - less focus on managing copies of data.

Building blocks

- Start by figuring out what will be solved.
    - First-class client filesystem.
        - Locking, Extended Attributes, per-file ACL, no arbitrary limitations (like directory size)

Building blocks

- Performance
  - Make better use of all available network bandwidth.
  - Don’t send data that isn’t needed.
  - Refine locking in the code, especially at the client.

Building blocks

- Best Practices
  - Improve network authentication
  - TCP instead of just UDP for network traffic.

Becoming a First Class File System

- New challenges involve getting resources we need to support everything OpenAFS does, or should
  - Multiple data streams, extended attributes, unlimited directory sizes, etc
  - Diverse platform support requires robust GUI environment support

Finding the missing pieces

- How do we work out what all is needed?
  - AFS3-Standardization: where protocol changes are needed.
  - openafs-devel: an interface with the developers.
  - openafs-info: an interface with the users.

2009 Edinburgh Hackathon

- RxK5
- Rxgk
- Rx/OSD
- Rx UDP enhancements
- AFS3 protocol-wide RPC updates
- “libosi”
- Extended callbacks

Edinburgh Fallout

- Today's talks:
  - rxk5
  - rxgk
  - RX/OSD
- More shortly:
  - Rx UDP enhancements
  - AFS3 protocol-wide RPC updates
  - "libosi"
  - Extended callbacks

Rx/UDP improvements

- Rx Path MTU discovery
  - Shipping
  - Some embarrassment due to a broken network in Illinois led to the code for this finally becoming top priority.
  - Dummy payload addition to Rx PING ACKs used to discover MTU.
  - Allows a new look at jumbograms.

Rx/UDP Improvements

- Window size negotiation - shipping.
- Rx option negotiation including max calls.
  - Requires additional PING-ACK payload field.

Rx Fixes

- And then there's everything that was broken:
  - Fixed as of 1.5.71:
    - Idle data connection processing could timeout if the send window filled and took longer than the idle data timeout period for the transmit window to re-open.
    - The transmit queue could be emptied prematurely. A required check for the queue being in use was forgotten.
    - The function that is supposed to implement a wait for the transmit queue to cease being busy failed to wait.

More Rx Fixes

- Fixed as of 1.5.74:
  - lock contention avoidance between rx_NewCall and rx_EndCall.
  - races due to inconsistent use of rx_connection conn_data_lock to protect the flags field.
  - inconsistent use of RX_CALL_TQ_WAIT which could result in deadlocks.
  - Must signal transmit queue waiters when flushing. Otherwise, deadlocks can occur.
  - Prevent rx_rpc_stats global lock from being a bottleneck (1.5.75)

AFS3 RPC refresh

- Intent is to future-proof any new RPCs.
  - 64 bit (100ns granular) time support.
  - 64 bit FIDs (AFS vnode identifiers).
  - Per-cell UUID.
  - Server UUID in RPCs.
  - Larger status objects to support large volumes (including quotas).

RPC refresh applied

- Not yet standardized
  - But serves as a guideline for what should be accomodated as new RPCs have been defined.

libosi

- Intended to provide OS-agnostic interfaces to common tasks.
- Previously done to support an aborted trace framework.
- Will be refactored and integrated beginning with its own support framework.
    - Compiler/environment detection.

Extended callbacks

- Pulls in libosi as a requirement.
    - Worked around this with “miniosi”, a stripped subset of the code.
- Draft published with the IETF
    - Asynchronous callback coalescing removed pending definition of AFS3 semantics.

New Work

- Byte range locking.
  - Draft available.
  - More in Matt's talk later.
- PTS alternate auth name support.
  - Draft available.
- RxTCP.

Release schedules

- The Edinburgh hackathon included discussion of priorities.
    - Branch for 1.6 candidate with stable DAFS was to begin near-term.
    - Changes for next-stable branch will be revisited after 1.6 release.
    - 1.4.12 released 2009.

Planning for the future

- In the past OpenAFS major releases have been feature driven
  - … even when that was not the intent
- Moving forward OpenAFS major releases will try to be time based
  - Features ready will be shipped
  - Those that aren’t will not be included
- A major/minor release every six to twelve months
  - Depending on the quantity and quality of submissions

A Real Road Map

openafs.org/roadmap.html

- **1.6**
  - The 1.6 series will become the new "Stable"
  - The 1.6 series will include significant improvements to source code quality and one major feature change:
    - Demand Attach File Service
  - Last release without a Windows IFS
  - Pre-release testing for 1.6 is expected to begin real soon now.

A Real Road Map

openafs.org/roadmap.html

- **1.7**
    - The 1.7 series will replace the 1.5 series as the experimental release series.
    - Releases will begin shortly after the 1.6 series enters pre-release testing.
    - The Windows IFS implementation will be integrated into 1.7 releases in preparation for the 1.8 stable release.
    - The 1.7 release series will track the 1.6 series with merge commits.
    - Sept 2010.

A Real Road Map

openafs.org/roadmap.html

- **1.8**
  - The 1.8 series will become the first stable release of OpenAFS to include the Windows IFS implementation.
  - No other new features will be added to 1.8.
  - December 2010.

A Real Road Map

openafs.org/roadmap.html

- **1.9**
  - The 1.9 series will replace the 1.5 series as the experimental release series.
  - Major new features will be integrated into 1.9 releases in preparation for the 1.10 stable release.
  - In Progress.

A Real Road Map

openafs.org/roadmap.html

- **1.10**
  - The 1.10 series will replace the 1.8 series as the stable release series for UNIX and Microsoft Windows.
  - The 1.10 series are scheduled to include:
    - rxk5 security class
    - object storage,
    - RxUDP performance improvements
    - PTS authentication name extensions
    - extended callbacks
    - hcrypto
    - Other …
  - Pre-release testing for 1.10 is expected to begin First Quarter 2011.

A Real Road Map

openafs.org/roadmap.html

- **1.11**
  - The 1.11 series will replace the 1.9 series as the experimental release series.
  - releases will begin shortly after the 1.10 series enters pre-release testing.
  - Major new features will be integrated into 1.11 releases in preparation for the 2.0 stable release.
  - First Quarter 2011.

A Real Road Map

openafs.org/roadmap.html

- **2.0**
  - The 2.0 series will replace the 1.10 series as the current stable series for UNIX and Microsoft Windows.
  - The 2.0 series will include:
    - rxgk security class including Kerberos v5, X.509 and SCRAM
    - protection of anonymous connections
    - protection of the server to client callback connection
    - server coordinated byte range locking
    - Other …
  - Pre-release testing for 2.0 is expected to begin in Third Quarter 2011.

Release schedules (Reality)

- 1.6 branch release meeting reviewed extant changes.
  - Branch for 1.6 candidate exists.
  - 1.7/1.8 for Windows Redirector.
  - Changes not ready for 1.6 continue to be merged into the HEAD for 1.9.

Other work not on the Road Map

- Servers for Microsoft Windows
- Process Authentication Groups for Windows
- Integrate .backup volume with Windows Volume Snapshot Service
- Integrate AFS quotas with Windows Quota Service
- Construct Windows Object IDs from AFS cell and FID and then implement the Windows Link Tracking Service
- Windows Management Instrumentation
- Growl-like UI to monitor AFS activity via WMI events

1.6 new features

- Demand-Attach Fileserver
- Universal mountpoint-less volume addressing (/afs/.:mount/cell:volumeid/) is available.
  - Originally done for the Linux NFS translator.
- An extension allows any vnode to be used. (/afs/.:mountcell:volumeid:vnodeid:uniquifier/)
  - Needed to help GUI environment issues.

1.6 new features

- Disconnected AFS
  - Supports read-write operation.
  - No “vnode pinning” yet.
  - Cached writes do not currently persist across client restart/reboot.
- Tunable cache readahead (“*fs precache*”)

Demand Attach Fileserver

- As of 1.5.76, and for 1.6, installed as "dafileserver", "davolserver", "dasalvager" in parallel with old-style fileserver.
- Configure fs or dafs bnode.

A status report

- Since growing for the future requires a solid foundation.

Unix platform summary

- AIX 5 and 6 (though 6.3)
- FreeBSD 7, 8 and current
- HP-UX 11.0, 11i v1 and v2
- Irix 6.5
- Linux 2.2, 2.4, 2.6 (ia32, ia64, amd64, ppc, ppc64, arm, sparc, sparc64)

- MacOS 10.3, 10.4, 10.5, 10.6 (ppc, i386, amd64).
- OpenBSD 4.4, 4.5, 4.6, 4.7.
- Solaris 2.6, 7, 8, 9, 10, 11 (and OpenSolaris)

Ongoing Platform support: Linux

- Linux kernel symbols continue to be removed from our view.
  - Aside from the NFS translator this has not yet been an issue for basic functionality.
- Dynamic sizing for AFS client vnode pool needed to deal with lack of inotify() symbols.

Ongoing Platform support: Linux

- Keyrings now authoritative for PAGs
- Cache bypass (Linux-only, new since 1.5.53)
- 1.5 series features tuning to better utilize the Linux kernel VFS interface.
    - Performance.
    - Correctness.

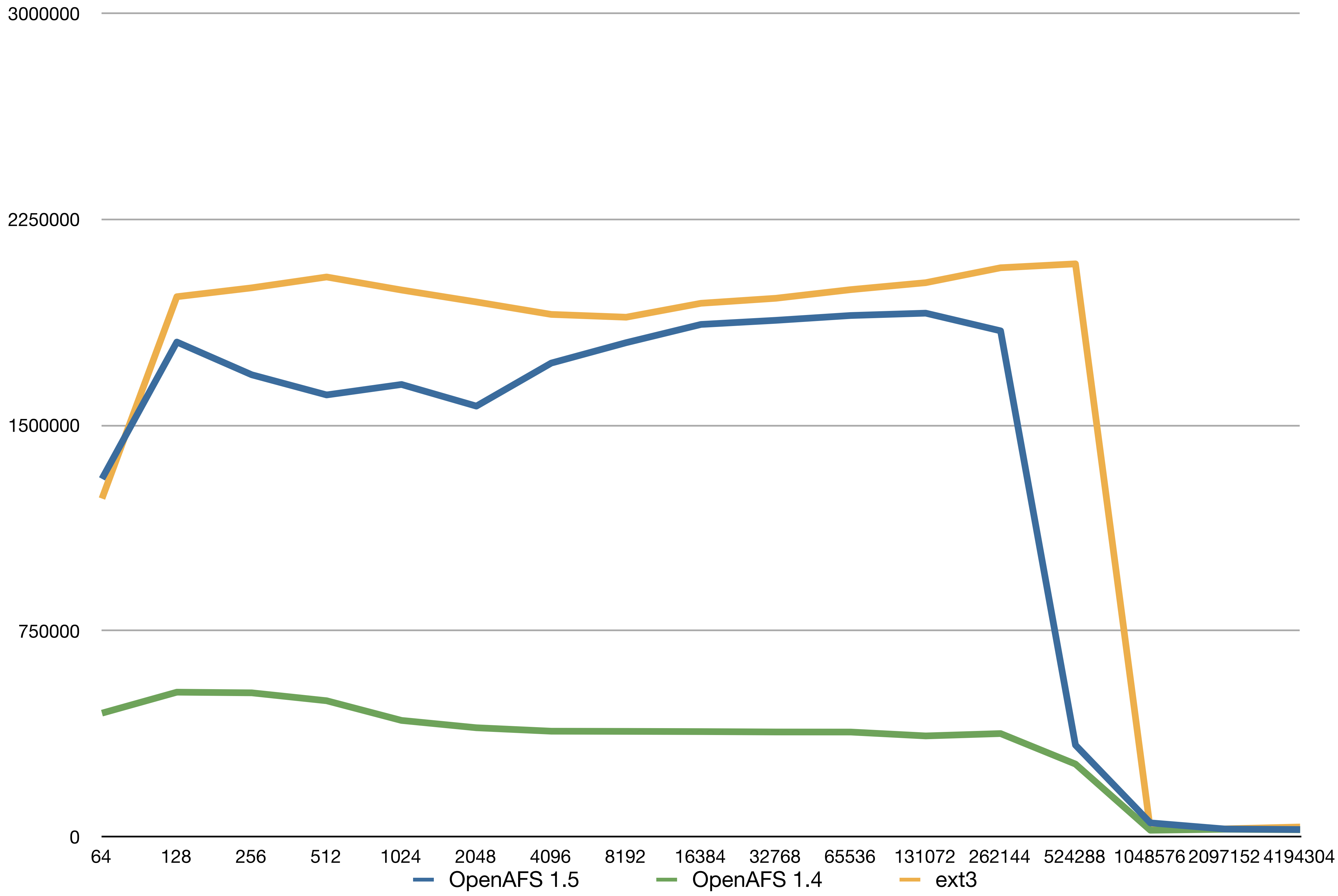
Linux Page Cache Improvements

- Reduce the number of redundant reads by correctly using the page dirty flag
- Enable readahead when filling the page cache from disk
- Remove duplicate writes of pages to disk by telling the kernel what we're doing
- Populate the page cache with a background thread, rather than doing it during requests

Linux: Minimizing Data Copies

- Copying data is expensive
- Minimise the number of copies between the network, the various caches and user space
- Significant improvements made to write-on-close
- Other cases an ongoing project

Linux Cache read performance: AFS should match ext3 below 1GB



Ongoing Platform support: MacOS

- AFSCommander integrated (Preferences Pane).
  - Offers GUI configuration of many aspects of user experience with OpenAFS on Mac.
- MacOS 10.6 Snowleopard includes support for 64 bit.
  - 64 bit kext
  - 64 bit userspace (including on 32 bit kernel)

MacOS issues

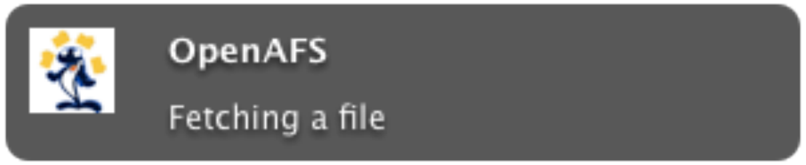
- A fix for the MacOS "Finder cross-volume drag" issue.
  - A userspace helper and the ad-hoc "reference any vnode" semantics are used.
- Still no PAG support.
- Bulkstatus now enabled.
- Finder dropbox (insert file) support.

MacOS issues

- Finder uses fsevents to change your view.
  - On authentication changes.
  - Not on callbacks yet.
- coreservicesd and leaked stat info.
- StartupItems->LaunchDaemons

MacOS issues

- Growl notification agent
  - Uses "mariner" logging interface (fetches, stores, creates, deletes, OSX-specific uprintf equivalent for log messages)



OpenAFS

Fetching a file

Windows: Since Roma

- 1.5.77 is the current release
  - 11 releases in last 12 months
  - Over 100 changes

Windows 7 and Server 2008 R2

- Officially supported
- Continuing issue with loss of netbios name resolution when network link status changes
- Microsoft may have fixed it but it is unclear
- The AFS redirector avoids the issue entirely

Windows Security Update MS10-020

- Issued first in April 2010 (KB980232)
- Prevents invalid SMB response from injecting data into applications that execute QuerySecurityInfo() API calls
- Apps that do not check the return code will crash when QuerySecurityInfo() fails
- Fix for OpenAFS in 1.5.75

Windows:
Dynamic Server Preferences

- Windows clients as of 1.5.66 no longer rely on class-based network addressing to determine server preferences
- Server preferences adjust dynamically every ten minutes based on real time RTT measurements

Windows: “fs newcell” updated

- "fs newcell" with no arguments is still accepted in order to maintain compatibility with prior Windows behavior.
- "fs newcell -cell <cell> -dns" instructs the cache manager to add the new cell but obtain the vldb server info from DNS.
- “fs newcell -cell <cell> ... -registry" instructs the cache manager to add the new cell and also save the cell configuration data in the registry for use the next time the service restarts.
- The -vlport and -fsport options are accepted although the -fsport value is currently unsupported by the cache manager.

Windows: New Registry Options

- FreelanceImportCellServDB
  Default is 0 (off)
- NatPingInterval
  Default is 0 seconds (off)
- UnixModeFileDefault / UnixModeDirDefault
  Default is 0777
- RxMaxRecvWinSize (128), RxMaxSendWinSize (128), RxMinPeerTimeout (350ms)
- ReadOnlyVolumeVersioning
  Default is 0 (off)

Google Summer of Code 2009

- Last year:
  - Windows MMC management snapin, Brant Gurganus.
  - Improved OpenAFS server selection, Jake Thebault-Spieker.
  - OpenAFS features in the Linux kAFS client, Wang Lei.

Google Summer of Code 2010

- This year:
  - Encrypted Storage, Sanket Agarwal.
  - A port of OpenAFS to NetBSD, Matt Smith.
  - Userspace interface for the Linux kAFS client, Wang Lei.
  - Extended attributes via AppleDouble files, Kelli Ireland.
  - Implementing Microsoft's Safe String (StrSafe.h) Library for UNIX/Linux, Jonas Sundberg.

GSoC Project: Extended Attributes

- Netatalk code not usable due to license change.
- Apple code not usable due to APSL issues with e.g. Debian.
- Userspace tool to manipulate dot underbar files works.
- Kernel implementation in Gerrit, has bugs to be resolved.

GSoC Project: Encrypted Storage

- Basic code done, not including real cryptography.
- Includes an internal interface in the cache manager
  - Allows decryption/encryption of files between cache and userspace.
- Code in OpenAFS Gerrit.

GSoC Project: NetBSD support

- Further along than before.
- Supports latest NetBSD (5.0.2).
- Still issues due to vnode operation changes.
- Code not yet in OpenAFS Gerrit.

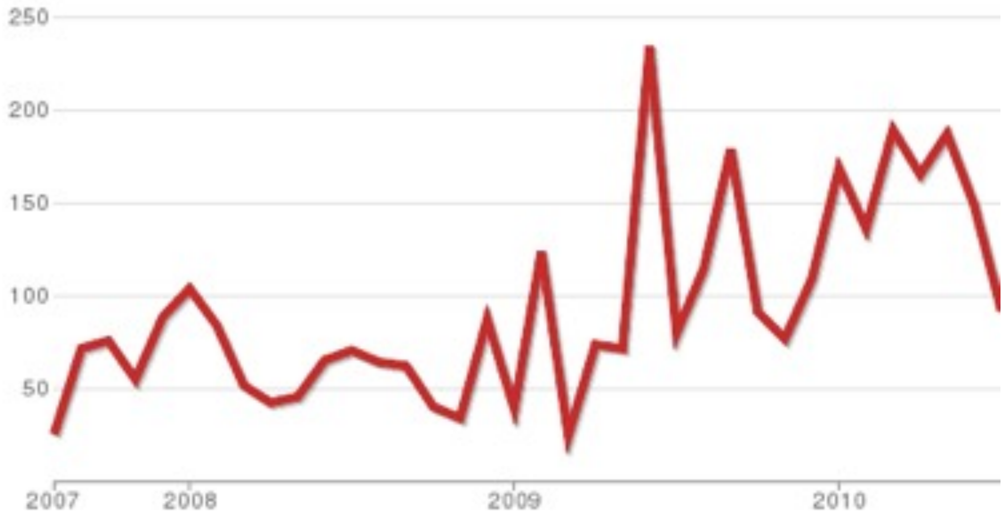
GSoC Project: Microsoft SafeString Library

- libstrsafe is a cross platform implementation of the Safe String Library (StrSafe.h) provided by Microsoft. The library will allow multi platform software to use a single library for safe string operations. This will reduce the risk for buffer overflows and will increase code sharing between code for different platforms.
- Status: Implementation completed
- based on Microsoft documentation and behavior test results http://msdn.microsoft.com/en-us/library/ms647466%28VS.85%29.aspx
- Licensed under the ISC License
- Available from https://www.ohloh.net/p/libstrsafe/

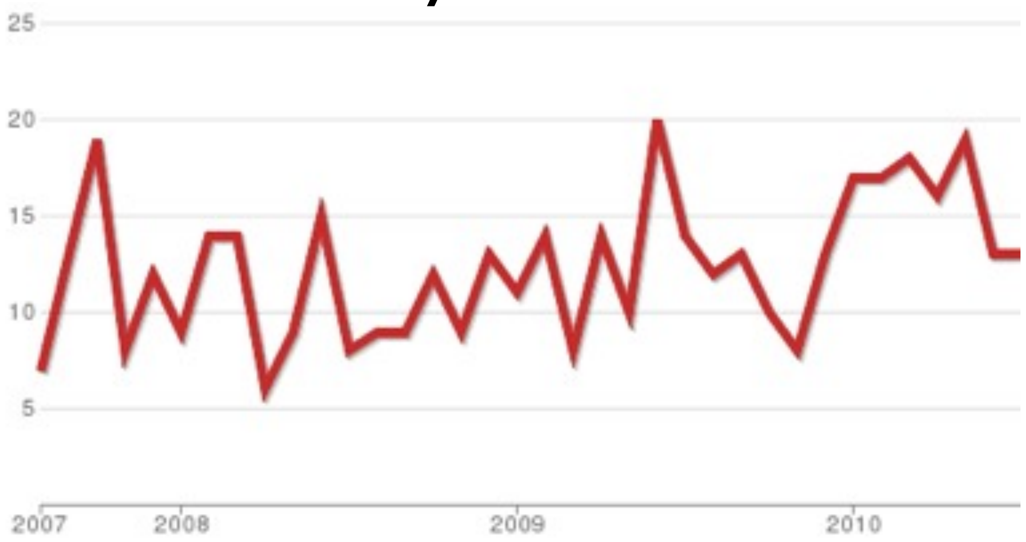
On version control

- git was rolled out last year.
- Huge success story.
- Visit gerrit.openafs.org for more visibility into the contribution process and to help review incoming submissions.
- Nearly 2500 submissions pushed through gerrit.

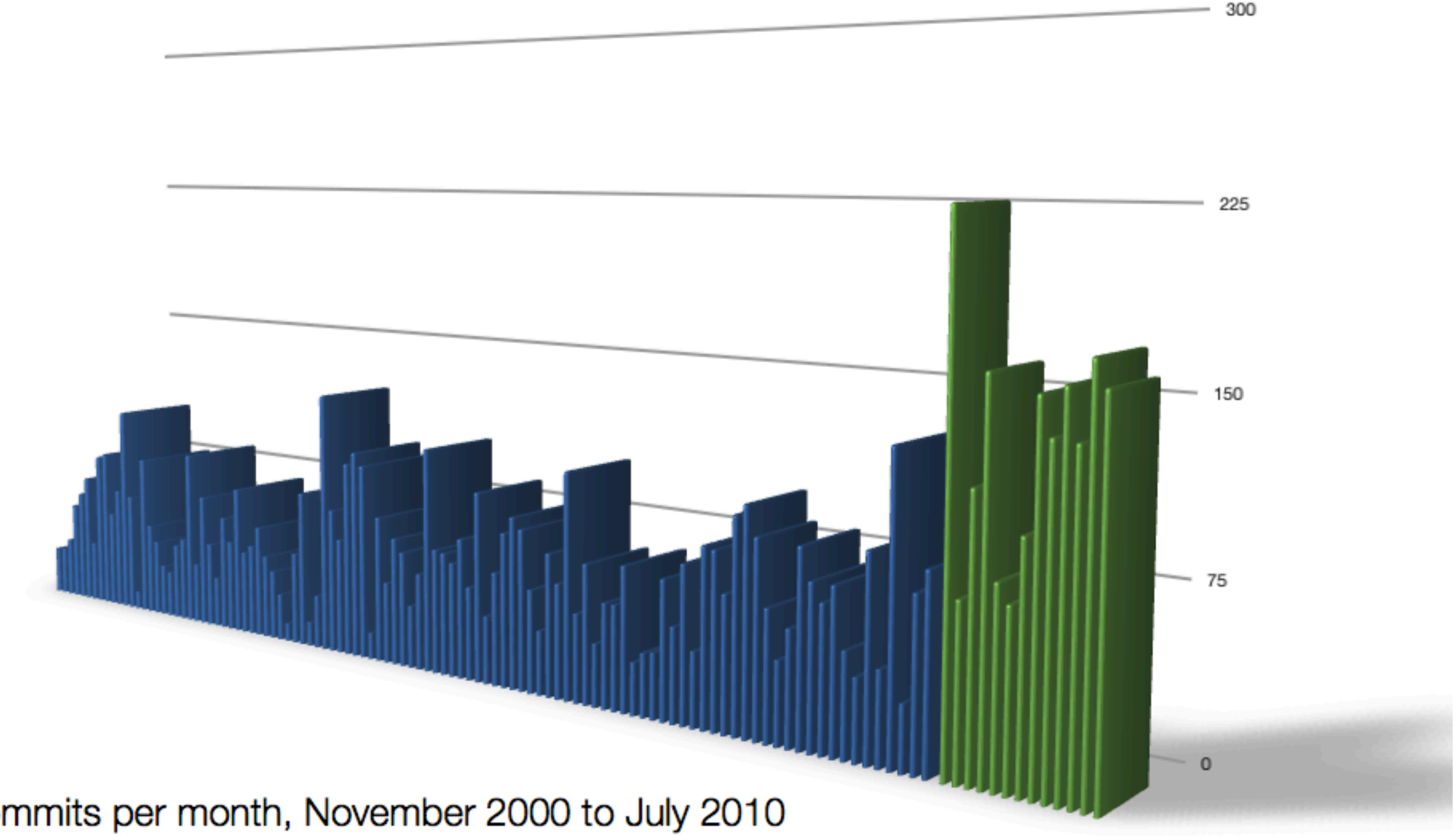
Monthly Commits



Monthly Contributors



Lies, damn lies and statistics



Commits per month, November 2000 to July 2010

Talk back to us

- Mailing lists:
  - Openafs-info http://lists.openafs.org/mailman/listinfo/openafs-info
  - Openafs-devel http://lists.openafs.org/mailman/listinfo/openafs-devel
- IRC chat room: #openafs on freenode
- Jabber developer MUC: openafs@conference.openafs.org