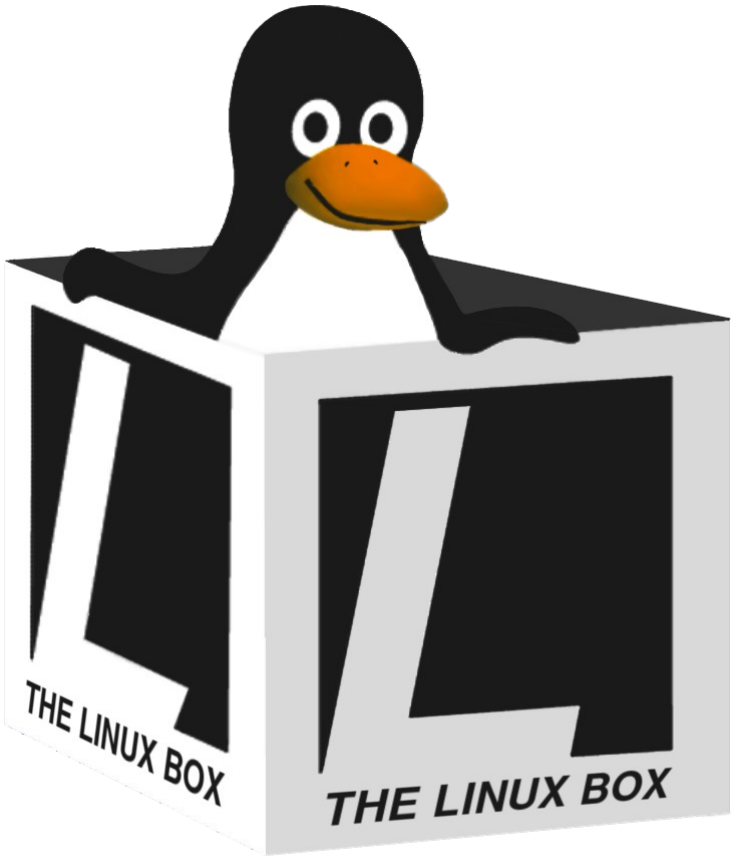
AFS: File Locking 2010



Matt Benjamin
<matt@linuxbox.com>

AFS: File Locking 2010

Background

* Brief history of AFS file locking
* Brief history of OpenAFS file locking efforts
* Brief overview of Extended Callback Information

AFS: File Locking 2010

Problems to be Solved

- Require efficient conflict detection, and also POSIX-equivalent deadlock detection.
- Must support mixed Unix and Windows file locking models.
- Since Unix has at least 3 file locking models itself, must accommodate these cleanly
- Need a robust recovery model

AFS: File Locking 2010

Protocol Concepts

The draft specification provides for a wire-protocol byte-range lock data type AFSByteRange lock, operations in the afsint and afscbint interfaces in the file server.

The already-specified and implemented XCB protocol extension provides async issue and state change support.

AFS: File Locking 2010

Share Reservations

A key new concept is that of share reservation. A bit different than the same-named concept in NFSv4.

In AFS locking, a new lock type, of whole-file granularity, used to negotiate specific locking and file operation semantics. Clients wishing to use specific sharing semantics, such as Windows conflict detection rules or mandatory enforcement, acquire a corresponding share reservation ahead of time (typically, as part of open processing).

AFS: File Locking 2010

System Architecture

A portable locking kernel provides a common subset of locking services to client and server implementations, through a general afs_lock_manager, supported by various callback specializations and flags. A key concept is the lock manager's transaction interface. A server implementation uses the transaction interface to coordinate state change notifications and a lock recovery journal. The client uses transactions to assert lock operations at the file server.

AFS: File Locking 2010

Locking Client

- construct appropriate lock attribute values, esp. Offset and Length, Type, Owner and Uniq attributes that correctly identify a specific lock
- Perform local conflict detection
- Assert locks at the file server
- Synch/flush byte ranges when write locks on those ranges are released
- Coordinate wait operations for POSIX locks with wait semantics (F_SETLKW)
- On some platforms, provide for release of locks outstanding at file close (eg., Linux)
- Various arbage collection and cleanup obligations

AFS: File Locking 2010

Viced

- perform authoritative state management and conflict detection, using lock request information provided by clients
- coordinate deferred lock issue and related state notification
- maintain lock recovery journal, replaying it if necessary on unplanned restarts
- coordinate asynchronous lock issue in response to lock state changes (primarily via 4.4BSD-based lock state graph)
- conditionally perform mandatory enforcement when I/O operations intersect a locked byte range
- various garbage collection and cleanup obligations

AFS: File Locking 2010

Some Viced Architecture Details

- MCAS-based lock manager dictionary (avoid increased coupling with legacy host package, which has locking issues)
- Optimize for concurrency of file operations threads, without compromising consistency of the recovery journal.
- Maximize journal update performance, since it is the bottleneck (implementation's strategy for this is to delegate journal I/O to a dedicated thread whose operations on the underlying SQLite database can then be lock-free, using the corresponding restricted interface, lock state change event messages are dispatched through a fifo queue which, while pthreaded today, is planned ultimately to be a lock-free fifo queue)

AFS: File Locking 2010

Some Code Highlights

- Viced lock manager specialization
- MCAS lock-manager dictionary (flock_dict)
- XCB msg delivery refactoring
- Vice_Sqlite
- CM lock manager specialization
- Range sync (afs_StoreSegmentsRange and osi_FlushSegmentsRange)
- Async issue processing

AFS: File Locking 2010

Status and Availability

- The current locking specification--draft 11?--is available. This draft has minimal deficiencies, and depends on the latest Extended Callback Information draft, with minor changes (introduces unicast XCBs and updated bulk result format). Another draft will be published soon.
- It's only 95% ready, but the design is fully expressed, and "simple locking" works on at least 6 CM platforms.
- In progress draft available soon, for the adventurous. There are defects in the code relevant to deadlock detection and deferred lock processing, but the implementation reflects the final design. These should be addressed in the next few weeks.
- To get access to pre-release code, email matt@linuxbox.com.

AFS: File Locking 2010

Background

* Brief history of AFS file locking
* Brief history of OpenAFS file locking efforts
* Brief overview of Extended Callback Information

AFS: File Locking 2010

Thanks

- To Jeff Altman and Your File System, Inc. for support

- To Derrick Brashear, Tom Keiser, Jeffrey Hutzelman, and others for XCB and locking draft review and feedback

AFS: File Locking 2010

Questions?