# EXT2 Futures


## Theodore Ts'o
## VA Linux Systems

## tytso@valinux.com




# Agenda

- Historical Review of Linux Filesystems
  - Minix, Ext, Ext2, Xiafs
- Ext2 Philosophy
- Planned New Features
- Ext2 and Other Filesystems
- Conclusion

# In The Beginning....

- Minix filesystem
  - Design taken from Minix, implementation by Linux Torvalds
  - 14 character filenames
  - 64k limits: # of blocks, inodes
  - indirect block scheme: 7+1+1
- Ext filesystem
  - Implementation and Design by Remy Card
  - variable length filenames
  - linked list of free blocks and inodes
  - indirect block scheme: 8+1+1



# Enter EXT2

- Update of the Ext filesystem
- Used block, inode allocation bitmaps instead of freelists
- Added triple indirection block
- Added "cylinder groups" aka block groups
- Added fast symbolic links
- Smarter block, inode allocation policies
  - Directory balancing across block groups
  - Block preallocation

# Historical Sidenote: Xiafs

- Written by Frank Xia, circa 1993
- Simple extension of Minix
  - Added variable-length filenames
  - Retained 7+1+1 indirect block scheme
  - Retained contiguous inode table, allocation bitmaps
- Simpler than ext2, so code become stable faster
- Author proposed renaming xiafs to linuxfs



# General Ext2 philosophy

- Trailing edge technology --- done well
- Little in ext2fs which is new and innovative; but we knew what worked well and what didn't.
  - Example: e2fsck speed optimizations
- Standard Unix filesystem semantics
- Has become filesystem of choice within Linux community
  - Conservative changes to retain stability
  - Forwards/backwards compatibility of filesystem format

# Planned New Features

- Ext2 Journaling
- Ext2 B Trees
  - Directories
  - Extents
- Compression
- ACL
- Support for Posix Privileges



# Ext2 Journaling

- Implementation by Stephen Tweedie, Beta quality
- Goal: High Availability
  - Fast recovery in the event of a system failure
- Described in more detail by Stephen at this conference

# Ext2 and B-trees

- Currently being implemented by me
  - Progress has been slow
    - Design not finalized
    - One of the reasons why I started working for VA Linux Systems
- Key B-Tree features
  - Exterior B-trees
  - Hashed key (for filenames)
  - Pre-emptive splitting



# B-Trees and Directories

- Needed for applications that want to put large number of files in one directory
  - Historical side note: Bell Laboratories assumed that applications would use nested directories if this was an issue:
    - /home/t/y/t/tytso
- Challenge: POSIX readdir(), telldir(), and seekdir() semantics

# B-Trees and Extents

- Indirect blocks are a performance problem
  - Large files with double and triple indirection blocks require reference to the indirect blocks. Indirect blocks are often flushed from the cache by the time they are needed.
  - Especially a problem with 1k blocks
  - 90-95% of files in a typical ext2 filesystem are contiguous
- Solution: Store block mapping information as extents in the inode



# B-Trees and Extents, II

- What does this have to do with B-Trees?
- Problem: What happens if the number of extents overflows the inode?
  - Random-access writes causing fragmented block allocation
  - Insertion into sorted list problematic
- Solution: use a B-tree to store the extent information, indexed by logical block number

# B-Trees and Robustness

- B-tree structures have many block pointers
- This makes recovery in the face of filesystem corruption to be challenging
  - Corruption will happen
    - Kernel bugs
    - Power failures and other unclean shutdowns
- Possible soslutions
  - Journalling
  - Magic numbers and checksums in B-tree directories to help in the recovery process
  - More investigation needed!



# Ext2 Compression

- Patches maintained by Peter Moulder
- File-level compression
  - Blocks are ganged together into clusters
  - Clusters are compressed
- Importance given cost of disks?

# Ext2 ACL's

- Implementation by Remy Card; recently picked up by a group of students
- POSIX ACL semantics
  - per-file ACL's
  - default directory creation ACL's
- Efficiency concerns
  - ACL's are shared across files when possible
  - Reference counting



# POSIX Priviliges

- Design from (abandoned) POSIX working group draft
  - Split superuser privilege into an array of fine-grained privileges (ala VMS)
  - Executables have two privilege bitstrings
    - Allowed
    - Forced
- Problems
  - System administration much more complicated
  - Unfamiliar (un-Unix-like) model

# Ext2 and Privileges

- Rough first cut implementation done by Andrew Morgan
- General Design
  - Implement ability to associate attribute data with inodes
  - Small data sets only; kernel-only access
- Needs polishing before integration into ext2 mainline



# Ext2 and Other Filesystems

- Advantage of Ext2
  - Large installed base
  - Users can take advantage of new features without dump/restore
- Disadvantages of Ext2fs
  - Basically, in design a FFS derivative
  - The statically allocated inode table can't really be changed without retooling everything
  - Bacause of the large user base, filesystem format changes have to be made very conservatively

# Conclusion

- Ext2 as a long line of Linux's filesystems
- New features are still being planned and worked on
- In the future some other filesystem may supplant ext2 filesystem. (Just as in the future Linux may be supplanted by something else new.)