Stacking

Erez Zadok & Jan Blunck

LSF 2008

Topics

1. Page cache consumption
2. Stack space pressure
3. Persistent inode numbers

4. Whiteouts
5. Readdir/telldir/seekdir

(1) Page Cache Consumption

Problem:

- each layer maintains its own objects (dentries, inodes, pages, etc.)
- want address_space (or vm_operations)
  - for mmap to work, executing binaries, etc.
- pages are "duplicated" at each layer, increasing memory pressure
  - Even worse when stacking on tmpfs
- some stackable file systems don't change data b/t layers (e.g., unionfs)
  - others do change data (e.g., eCryptfs)

Problem (cont.)

- no way to share pages among layers
  - page→mapping→host points to one inode structure
- also: address_space ops are too complex for what stackable f/s want
  - want: just pass op to lower
  - must deal with: page locking state, page flags, AOP_WRITEPAGE_ACTIVE…

Past attempts included

- copy to lower page, then try to release lower page after lower op is done (e.g., PG_reclaim)
  - Reduces average memory pressure
  - still doesn't relieve the pressure, esp. under stress
- temporarily set upper_page→mapping→host to lower_inode, pass op to lower layer, then fix →host up
  - Only keeping upper pages
  - Ugly to “fix up” a pointer, racy

Past attempts (cont.)

- implement vm_operations→fault op (e.g., aufs)
  - Simplifies code (no address_space ops)
  - Set “vma→vm_file = lower_file”, then call lower →fault op
  - Still needs to “fixup” a pointer
- Other suggestions?

(2) Stack Space Pressure

Problem:

- each layer adds more to the stack
- layers on top of layers possible today

Suggestions/Solutions:

- short term: build kernel with larger stacks
- long term:
  - linked list of ops: foofs_op → barfs_op → lowerfs_op
  - VFS iterates through list?
    - Similar to Windows' I/O Manager

(3) Persistent Inode Numbers

Problem:

- Exporting f/s and some userland tool need persistent inums
- Stackable f/s don’t have a persistent store
  - Rely on lower f/s for that; use iunique()
    - Non-persistent
- Solutions/Suggestions:
  - Inherit lower_inode->I_ino
    - Can’t cross into other lower f/s
  - Store pathname->number persistently
    - Can store in extra file, or small partition
    - Hard-links difficult to support

VFS Stacking Issues

For Union Mounts, UnionFS, etc.

Jan Blunck <jblunck@suse.de>, Erez Zadok
<ezk@cs.sunysb.edu>

Whiteouts - The Missing Filetype

We need to remove whiteouts of logical empty directories

- Call readdir() with specialized handler
- Worked well except that we don’t know the struct vfsmount
- Otherwise add code to all FS that want to support whiteouts
- Otherwise let the userspace handle it (racy)
- Otherwise make readdir() an inode operation
- Introduce opendir(),releasedir(), etc.

POSIX wants us to support directory seeking

Duplicate removal and whiteouts are hard in kernel-space

- seeking makes it even harder
- watch out for DoS by create()/unlink()/seekdir()

Implement in user-space

- need to export visibility of whiteouts and stacking
- need to be able to access a directory on a specific layer