Address Space Operations

Steven Whitehouse, Christoph Hellwig
Linux Storage and Filesystem Workshop
San Jose, 2008
GFS2 Update

- Much has happened since last year:
  - Code is smaller (by approx 20k)
  - Rewrite of some AOPs:
    - writepage/writepages to avoid jdata deadlock
  - Now uses page_mkwrite() and the “standard” filemap_fault
  - Faster block map (as of Friday!)
    - Now supports allocation of extents of data blocks
    - Next step is to take advantage of that at a higher level
      - Need to change write_begin/_end
      - Want to zap gfs2_write_alloc_required()
    - Would like to merge “unstuff” but that presents locking issues
      - Not impossible, but needs careful thought
  - Shrinking various data structures (glock & in-core inode)
  - Many other bug fixes....
“Almost duplicate” operations

- readpage/readpages
  - Readpages is generally used more if implemented
  - Lock ordering issue for distributed filesystems (try-lock doesn't work)
    - Uses AOP_TRUNCATED_PAGE
- writepage/writepages/launder_page
  - launder_page is writepage but with no release of the page lock
    - Used only for write out of buffers prior to DIO
  - Lock ordering problems (transaction lock - jdata) with writepage can be avoided by using writepages
  - Currently with writepages, writepage will only be called with WB_SYNC_NONE so can be ignored
Extending writes to multiple pages

Why?
- Current scheme has a lot of overhead due to per page:
  - Locking
  - Transactions
  - Etc.
- Problems:
  - Lock ordering wrt source pages
- Solutions:
  - Nick Piggin's “perform write” work or something similar