Bh-less ordered data mode

Mark Fasheh
Oracle

How It Works

- Attach bh's to pages
- Bh's written out at commit time
- Doesn't use pages directly
  - Journal transaction lock / page lock ordering
  - Possible metadata / data life time issues?
    - Probably not, `b_committed_data` saves us

Problems With This Approach

- Ocfs2 blocksize/clustersize/pagesize trifecta
  - Using blocksize in aops causes many extra lines of code
- Overhead of using bh's for logical/physical mapping of data
  - Memory overhead
  - Per-block mappings harmful to extents

Building Blocks

- `write_begin`/`write_end` allow us to reorder page lock
- `page_mkwrite` allows us to allocate before `writepage` is called
- Journaled data
  - Would this still be a problem for ext3?
  - Ocfs2 doesn't care.

Proposed Solution

- Let FS handle accounting of ordered data
  - logical->physical map via internal extent map
  - FS provides replacement for `journal_dirty_data`
    - Possibly requires some support from JBD
  - FS handles truncate. This looks easy enough.
    - Famous last words
- JBD uses callback in `journal_commit_transaction`
  - Default behavior would stay same, calls `journal_submit_data_buffers`
- Ext3 would still want a page-lockless writepage?
  - Ocfs2 wants it for dealloc anyway