
whamcloud

whamcloud

Optional where
Optional date

New IO Engine

- Jinshan Xiong
  Whamcloud, Inc.
  Jinshan.xiong@whamcloud.com

Why do we need this change

- For new grants
  - Grant used to be allocated by pages, this is good for ldiskfs
  - ZFS uses extent and bigger block size, we should consider extent overhead and allocate grant based on blocks
- Better IO quality
  - We have been using FIFO to cache dirty pages for a long time, the IO quality totally depends on how applications generate data
  - Bad IO kills performance a lot – fragmented IO on obdfilter
  - Merge sequential pages on the client side to make good IO

Why do we need this change – cont.

- Reduce loi_list lock contention
  - Contention is heavier since `fat' node is not fat any more
  - This is a known problem for a long time
  - Come up with a per object lock
- Prioritize caching pages
  - If a write lock is being canceled, the covering pages should be flushed ASAP

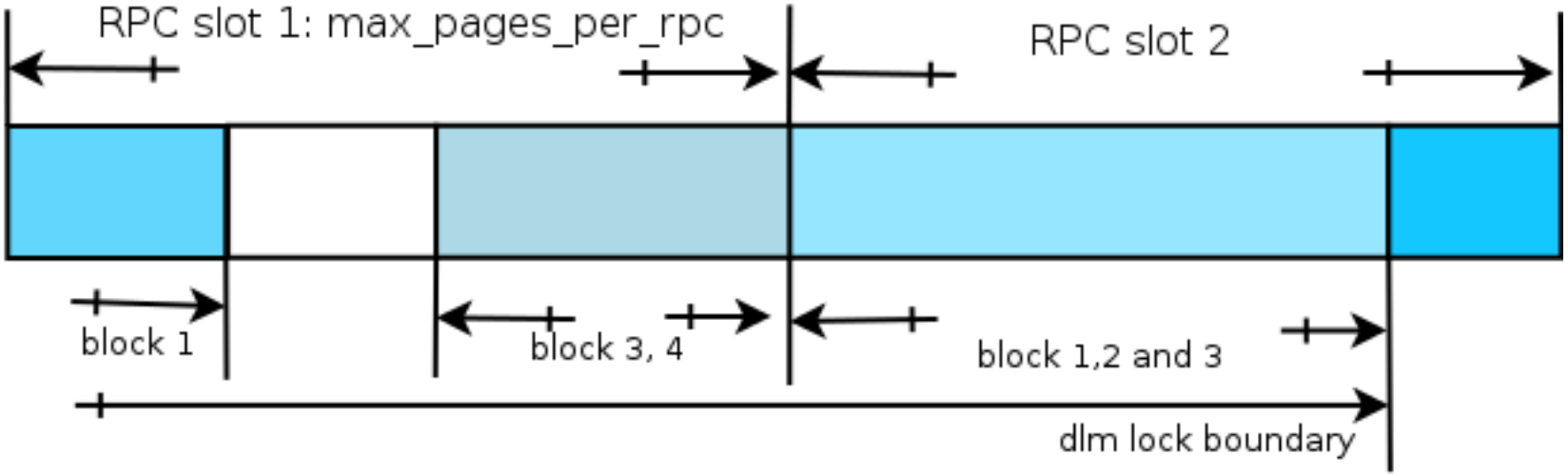
OSC_EXTENT definition

- Osc extent sits at OSC layer
- It represents a set of contiguous blocks
  - The pages in an osc_extent are not required to be contiguous
  - Extents have start and end page index, expand to block boundary if possible
- extents are managed by per object red-black tree
  - Start of extent as keyword

OSC_EXTENT attributes

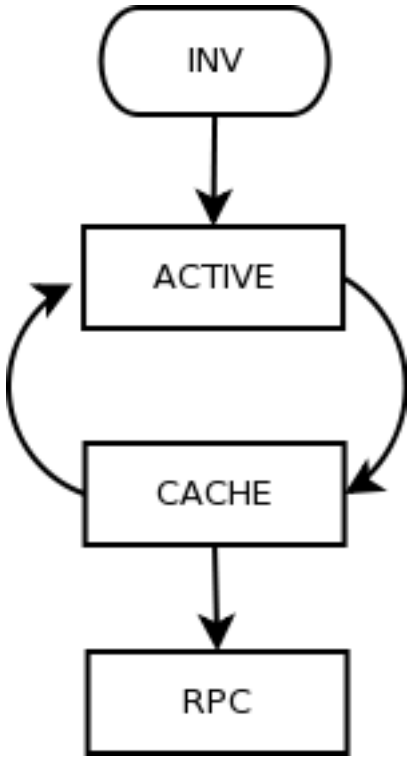
- Every caching dirty page must be covered by one extent exclusively
- One extent must be packed in one RPC, this implies:
  - The # of pages must be less than cl_max_pages_per_rpc, well it is not easy to change cl_max_pages_per_rpc any more
  - Have to be covered by only one cl_lock exclusively
    - Otherwise, it will be too bad if an extent is covered by two cl_locks, and one of them is being canceled
- Extent dies after the covering pages have been flushed
- Pages are added into extent in osc_page_cache_add()

RPC slots, extent and pages



State Change

- On going IO holds an active extent
- active extents can't be selected for RPC
  - This helps form better IO because contiguous pages more likely in the same RPC
- Cache extents can be picked up later by an IO and merged with other extents to compose RPC



Policies to compose RPC

- Definition of HP and urgent extents
  - Lock canceling -> HP extents
  - Page writeback or sync write -> urgent extents
- HP extents first
- Get first urgent extents, and merge next extents in tree for RPC

Design trade-offs

- Should we mix granted pages with non-granted pages?
  - This may make obdfilter complex
  - But, write –EDQUOT pages along with other caching pages can save one RPC
- Should direct IO pages consume grants on the client side?
  - Fairness among clients
  - But, applications are waiting for direct IO write, error will be seen
  - Both are okay

Current Status

- Framework is finished; debug and test are on going
- Remaining work
  - Change max_page_per_rpc is not easy
  - New grant parameters
  - Performance tune

…

34 files changed, 3270 insertions(+), 1094 deletions(-)

Thank You

- Jinshan Xiong
  Whamcloud, Inc.
  Jinshan.xiong@whamcloud.com