Filesystem I/O Speedups

Marshall Kirk McKusick
<mckusick@mckusick.com>

Cambridge BSD Developers Summit
31st August 2012

University of Cambridge
Cambridge, England

Overview

- Instigated by Scott Long
- Observed that I/O throughput had dropped on every release since 6.0
- Throughput had dropped by a factor of 7.
- Well-intentioned bug-fixes had been responsible

The 30-Second I/O Avalanche

- Historically flushed every filesystem every 30 seconds.

- Create a "dirtied" wheel with 30 slots
  - put a vnode on end of wheel when first dirtied.
  - Once per second advance wheel and flush any files there
  - Files not flushed or removed after 30 seconds get flushed
  - Create a "filesystem syncer vnode" at mount time which cycles around the cleaning wheel every 30-seconds to flush metadata

- During the merge of 4.4BSD the filesystem syncer vnode was changed to flush every vnode which effectively reverted to the old avalanche behavior

Unnecessary Zeroing

- When writing just a part of a newly allocated file block, the remainder of the block must be zeroed
- When writing all of a newly allocated block, there is no need to zero it if the copyin succeeds
- Code changed to always zero the block in case the copyin fails
- Fixed to only zero full-sized writes when the copyin fails.

Avoiding Unnecessary Locking

- Three routines ran every 30-seconds over all 350,000 vnodes
  - vfs_sync to ensure that mmap'ed files flushed modified pages
  - ffs_lazy_sync to ensure access times are updated
  - qsync to ensure that quota changes are flushed
- Observe that 99% of vnodes are inactive and thus cannot be accessed or changed
- Keep a list of the active vnodes and change the above three routines to only inspect this list
- Use inactve list linkage fields in vnode for active list to avoid adding additional member to vnode (saving 3Mb of space)

Future Work

- Direct dispatch in GEOM
- Add a flag settable by each module to say that it is MP-safe and thus can be run by invoking thread rather than using "g-up" and/or "g-down" thread
- Avoids single-threading in GEOM layer
- Care needed to avoid stack overflow