Using Range Locking for Concurrent Shared File I/O in NOVA on Manycore Servers

June-hyung Kim, Jangwoong Kim, Hyeongu Kang, Chang-Gyu Lee, Sungyong Park, Youngjae Kim
Department of Computer Science and Engineering
Sogang University, Seoul, Republic of Korea
{junehyung , ski6812, hyeongu, changgyu, parksy, youkim}@sogang.ac.kr

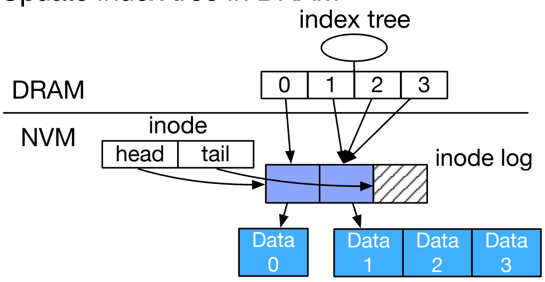
Systor 2019 June 3 - 5 Haifa, Israel

Introduction

- NOVA is a log-structured file system for DRAM and NVM main memories.
- NOVA manages per-CPU data structures to provide scalability: NVM free page list in DRAM and inode table and inode log in NVM.
- NOVA maintains index tree in DRAM where each node points to the log entry for efficient search.
- Read and write operations in NOVA are performed by accessing per-inode log which guarantees file system consistency.
- However, per-inode coarse-grained locks prevent concurrent shared file I/O performance.

Read/Write I/O Operation Flow in NOVA

- Read operation in NOVA
  1. Access index tree and get the address of log entry which is being read
  2. Find data by using data address in the log entry
  3. Copy data into user buffer
- Write operation in NOVA
  1. Allocate data pages and copy user data
  2. Write log entry in the per-inode log
  3. Update tail pointer of per-inode log
  4. Update index tree in DRAM

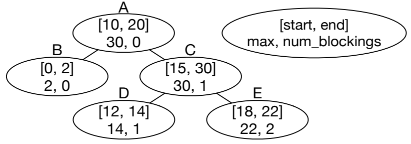


Concurrent Shared I/O Problems in NOVA

- When NOVA performs read or write operations, it requires a lock for inode.
- For reads, NOVA allows concurrent reads on the shared file, thus multiple threads can concurrently access the file.
- However, for writes, NOVA does not allow concurrent writes on the shared file due to inode mutex lock.
- If we apply range locking to NOVA, it can perform multiple write operations with disjoint ranges in the shared file.

Interval Tree-based Range Locking

- Range locking is a technique that can acquire a lock to a portion of a file.
- It can increase the concurrency of threads when accessing the shared file.
- There is a patch of interval tree-based range lock in the Linux community.
- In the interval tree-based range lock, each range lock is represented as a tree node.



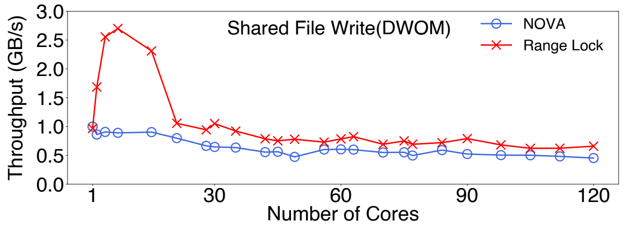
<Interval Tree-based Range Lock>

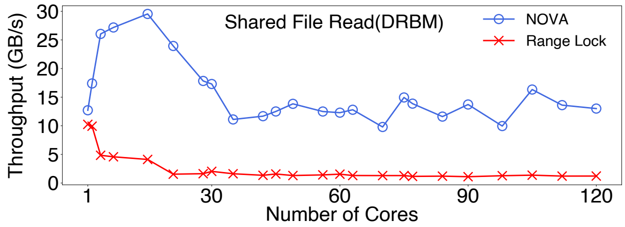
- We modified NOVA to use the interval tree-based range lock instead of inode mutex lock.

Preliminary Evaluation

Experimental Setup

- We evaluate the proposed approach on the testbed equipped with 120 cores and 968GB DRAM.
- We emulate 32GB of DRAM as NVM.
- We use shared file write (DWOM) and shared file read (DRBM) workload in FxMark.
- Each workload performs I/O on different offsets with multiple threads.





Results

- In DWOM, range lock scales up to 7 cores since it benefits from fine-grained range locking.
- In DRBM, range lock does not scale since the interval tree requires a mutex lock when a node is inserted or deleted to/from the tree.
- Performance degradation after 15 cores in both workloads is attributed to lock contention.

Summary

- To improve shared file I/O performance, we adopt an interval tree-based range lock to baseline NOVA.
- Our preliminary results show range locking is effective only for write only workloads, but read only workloads.
- Concurrent shared file reads using the range locking approach are not effective because they require a mutex lock on the entire tree whenever changes are required to add and delete nodes.

Acknowledgments This work was supported by Institute for Information & communications Technology Promotion(IITP) grant funded by the Korea government(MSIT) (No. 2014-0-00035, Research on High Performance and Scalable Manycore Operating System).