

6.824 Lab9 : Project

Ying Yin (yingyin@csail.mit.edu)

1 Overview

To improve concurrency in the YFS, read/write locking is added to allow multiple readers to access the same file at the same time. In addition, peer-to-peer data transfer is used to allow clients to write dirty extents directly to the next YFS client(s) that wants to access the extent. This helps to reduce the load on the extent server as more clients are added.

2 Method

To allow sharing, a protocol similar to the MESI protocol for cache coherence is used.

2.1 Lock Server

There are two acquire calls: `acquireExclusive()` and `acquireShared()`. A variable i is used to keep track of each lock state. If $i = 0$, the lock is free. If $i > 0$, it counts the number of shared lock owners. If $i = -1$, there is an exclusive lock owner. The server also keeps track of current owners, read waiters and write waiters for each lock for revoke and retry calls.

Starvation for writers can occur if the system is heavily loaded and there is always at least one reader for the same file. This is because a write (exclusive) lock can only be obtained if all readers release the lock. To prevent this, a new reader will be asked to retry if there is a write waiter for the currently shared lock.

2.2 Lock Client and Extent Client

The lock client keeps the access state (nil, shared, exclusive) of each lock. If the access state is nil, the client does not have the cached lock. A shared lock allows the client to read the associated data from the extent server and cache it. If a client is asked to release its shared lock, it invalidates its cache entry before complying. An exclusive lock allows a client to read or write the associated data and cache it. If a client is asked to release its exclusive lock or downgrade it to a shared lock, it must write back the dirty data before complying. It retains its cache entry if it is downgrading the lock, but invalidates it if releasing the lock.

2.3 Peer-to-Peer Data Transfer

The revoke request from the lock server to the lock client contains the requester(s) for the lock. If the exclusive lock is revoked by another exclusive lock requester, the dirty extent is transferred to the requester without writing back to the extent server because the

content at the extent server would not be valid anyway. If the exclusive lock is downgraded to shared lock, the dirty extent is transferred to all the shared lock requesters as well as the extent server, so new shared lock requesters can obtain the up-to-date extent from the extent server later.

After a dirty extent is transferred to the requester(s) and the exclusive lock is released, the lock server has to ensure that it asks all the requester(s) to retry. In the case of a single exclusive lock requester, the server should grant the lock only to that requester which is the first one on the write waiters list. In the case of shared lock requesters, the server can grant lock to other shared lock acquirers too, because the extent at the server is valid.

3 Evaluation

The new version of YFS is made sure to pass all the existing tests first, namely the `lock_tester`, `rsm_tester` and the file system tests from lab4. An additional test is added to the `lock_tester` to test concurrent acquiring of exclusive and shared locks by 10 clients multiple times.

Additional file system tests are added to compare performance between the new version and the old one. The total number of RPCs at both the lock server and the extent server is used as the comparison criteria. **Test a** tests concurrent 20 reads of the same file by 2 YFS clients. **Test b** tests concurrent 20 writes of the same file by 2 YFS clients. **Test c** tests concurrent writes and reads by 4 YFS clients.

test	# RPCs		reduction
	new version	old version	
a	45	550	92.0%
b	155	200	22.5%
c	228	905	74.8%

Table 1: Comparison of total number of RPC calls.

The results in Table 1 shows that, with sharing, there is a significant improvement in performance for concurrent reading. The slight reduction of RPCs for concurrent writing is due to the avoidance of writing dirty extent to the extent server.

4 Lines of Code

In addition to lab 8, the new code added is about 1900 lines, including the code for testing.