UNIVERSITY OF SOUTHERN CALIFORNIA

CS402 Assignment 01 – Read/Write Locking

Michael Gerow

February 22, 2013

1 Read Write Locking

Read write locking is implemented similarly to coarse grained locking in that we only have on global lock for the entire table, but instaed of using a pthread_mutex_t we instead use a pthread_rwlock_t. This allows us to use the functions pthread_rwlock_rdlock in order to get a read lock and pthread_rwlock_wrlock in order to get a write lock.

1.1 Testing

We test the correctness of the read write lock similarly to the corarse grained lock, using the test that adds a lot of elements to the database and then removes them, which should leave us with an empty database. This can, like with coarse, be done simply by running:

```
./server_rw < test/250_add_remove | tail -n 900 | less
```

As before, you can simply run the test.py script to run the above command and verify it for all the locking methods.
Of course, we should also expect an increase in speed for read write locking. It turns out that this is only true for certain types of accesses of the database. Performance testing will be covered in the performance document.