Hands-on-4: Valgrind

date: 2018/10/17
ID: 516413990003
Name: Huichuan Hui

---

1. Question1
   The cause of the key missing is originated inside the function pthread_create(&tha[i], NULL, put_thread, (void *) i) (shown below). The argument pointer (void*) i, which is used for array keys's index n in following-called function put_thread(void *xa), is overwritten when for-loop iteration increments. In order to prevent the argument i from being overwritten, add the lock in the function put like the code below.

```
        void put(int key, int value){
            int b = key % NBUCKET;
            int i;
            assert(pthread_mutex_lock(&lock) == 0);
            // Loop up through the entries in the bucket to find an unused
one:
            for (i = 0; i < NENTRY; i++) {
                if (!table[b][i].inuse) {
                table[b][i].key = key;
                table[b][i].value = value;
                table[b][i].inuse = 1;
                assert(pthread_mutex_unlock(&lock) == 0);
                return;
                }
            }
            assert(pthread_mutex_unlock(&lock) == 0);
            assert(0);
        }
```

2. Question2
   Add lock and unlock statements in put, and make sure there is no resource contention of valuable "table[][].inuse", which is originated from (void *) i in pthread_create mentioned above question.

3. Question3
   No, they have almost the same runnning times.

   - one-threaded
     completion time for put phase = 2.982153
     completion time for get phase = 2.940046
   - two-threaded

completion time for put phase = 3.996490

completion time for get phase = 3.733425

4. Question4

   Most likely because of a number of nested loops. And The lock function's overheads affect speed.

5. Question5
   Because The locks' overhead is affecting the speed.
6. Question6
   Because readig from the same index of table, which is caused from race condition, doesn't cause any errors, wheras race condition DOES cause key missing because some keys would not be written.

7. Question7

   Remove the resource contention existed in the pthread_create(&tha[i], NULL, put_thread, (void *) i). No race conditions on Ubuntu.

```
        // Create nthread put threads
        long myid[nthread];
        for(i = 0; i < nthread; i++) {
            //assert(pthread_create(&tha[i], NULL, put_thread, (void *) i) ==
0);
            myid[i]=i;
            assert(pthread_create(&tha[i], NULL, put_thread, &myid[i]) == 0);
        }
```

   And

```
    static void *
    put_thread(void *xa)
    {
        //long n = (long) xa;
        long n = *((long*) xa);
        long i;
        int b = NKEYS/nthread;

        for (i = 0; i < b; i++) {
            put(keys[b*n + i], n);
        }
    }
```