**Header**

CS6240 Homework 1 Name : Zhongheng Yang NUID:001819307
email: yang.zho@husky.neu.edu

**Weather Data Source Code is in my CCIS Github ( name: ccismichaelyang )**

I have made you (professor and TAs ) as collaborators so that I believe you can see a "multi_thread" repository. In case you cannot see it, please do not hesitate to contact me since I have just begun to use github.

Tests With Greater computational cost( Fib ):

| Multi thread | seq | No lock | Coarse lock | Fine lock | No-sharing |
|---|---|---|---|---|---|
| max | 13229 | 4687 | 8535 | 5489 | 5115 |
| min | 8339 | 4130 | 6295 | 4298 | 4453 |
| mean | 9232.9 | 4282.5 | 7108.5 | 4682.1 | 4782.1 |

Tests Without Greater computational cost:

| Multi thread | seq | No lock | Coarse lock | Fine lock | No-sharing |
|---|---|---|---|---|---|
| Max | 3814 | 1988 | 2008 | 1719 | 2006 |
| min | 2656 | 1306 | 1314 | 1347 | 1316 |
| mean | 2828.9 | 1394.725 | 1460.75 | 1465.85 | 1419.225 |

for tests without greater computational cost, here I run 40 loops for each version in order to have stable results.

1. I expect that no Lock will finish fastest. this is due to the lack of mechanism to ensure correctness.

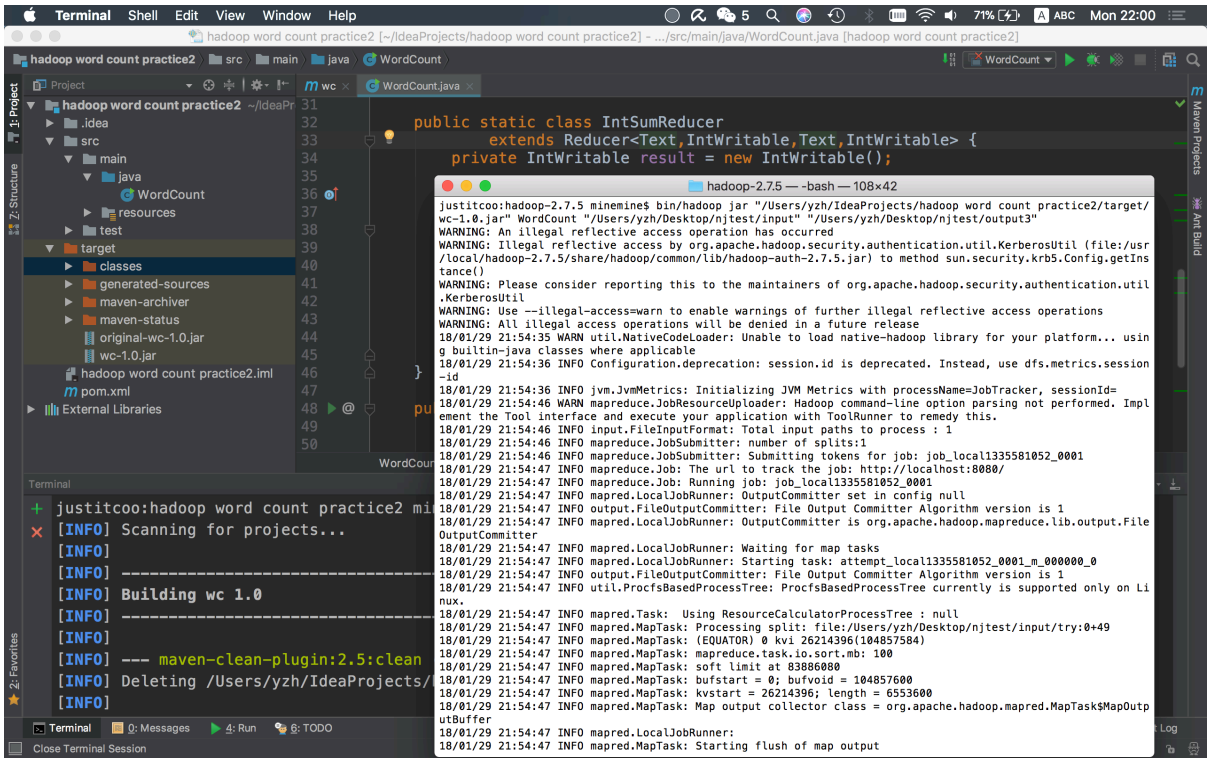
2.I expect that SEQ will finish slowest, since others take advantage of multi-threads. The experiment result is same

3. No-lock is incorrect( have some small difference from SEQ version) and sometimes gives exception when running.

4. Coarse-lock is faster than SEQ version. Although the accumulation data structure in Coarse-lock doesn't run concurrently, the "filtering TMAX" steps run concurrently.

5. In Fib test, coarse lock becomes extremely slow, while in another coarse lock is faster than fine lock. This is because coarse-lock locks the whole hashmap ( accumulation data structure) while fine lock doesn't lock on unnecessary things.

## Word Count Local Execution




The Project directory structure and the console output are both shown in the screenshot.
IDE: intellij

## Word Count AWS Execution