6240 – Parallel Data Processing with Map-Reduce
SECTION 2 HW 1 AKASH KADAM

Program Execution Time without delay

Program Type	Min Time(ms)	Avg Time(ms)	Max Time(ms)
Sequential	738	950	2773
No-Lock	405	422	456
Coarse-Lock	1518	1646	2064
Fine-Lock	432	456	570
No-Sharing	761	777	813

Program Execution Time with Fibonacci delay

Program Type	Min Time(ms)	Avg Time(ms)	Max Time(ms)
Sequential	795	1099	3748
No-Lock	418	441	481
Coarse-Lock	1598	1673	1793
Fine-Lock	456	476	525
No-Sharing	563	612	635

Speed Up

Number of Worker Threads = 2

Speed Up without delay:
No-Lock: 2.2511
Coarse-Lock: 0.5771
Fine-Lock: 2.0833
No-Sharing: 1.2226

Speed Up with delay:
No-Lock: 2.4920
Coarse-Lock: 0.6569
Fine-Lock: 2.3088
No-Sharing: 1.7957

Answer 1)
I expected NO-LOCK program to finish fast but with incorrect output. This is because of multi-threading without synchronization i.e we are not holding locks on any objects. The experiments did confirm my expectation.

Answer 2)
I expected COARSE-LOCK program to finish slow as we are using a single lock on the data structure, which we are using to store the stationID and its relevant information. In this case a thread can obtain a lock on the single shared data structure object and do the computation. As a result, other threads in the pool has to wait for till the thread releases the lock.

Answer 3)
On comparing the temperature values after running all the different type of program, I observed that the final values of the NO-LOCK program are inaccurate and the reason being there is no synchronization between threads i.e. all one thread can interfere with the values updated by other thread.

Answer 4)
My observation is based on both the delayed as well as without delay programs. I observe that the SEQ program execution time is fast as compared to the COARSE-LOCK program execution time. The reason being in COARSE-LOCK we are using a single lock on the entire shared data-structure which delays the execution as all the threads in the pool have to wait to obtain the lock if they want to write access the data structure.

Answer 5)
The execution time is slightly increased (10 – 30ms avg. time) in both FINE-LOCK and COARSE-LOCK due to the extra computation on executing the Fibonacci series. I don't observe any much difference between FINE-LOCK and COARSE-LOCK execution with delay.