Problem set 4

Peter Rotar

October 2023

Parallel Pthreads: The test gave correct results for 1, 2, 4 and 8 threads. For speedup we use $S(p) = T_s/T_p$.

# of threads	time (s)	speedup
sequential	46.652278	–
2	40.906651	1.140457
4	28.038396	1.663871
8	31.622304 !	1.475297 !

It should be noted that the tests were run on CPU with 6 cores. Therefore, for 8 threads it comes to bad side of oversubscrition, making the measured time (in a sense) inaccurate.

OpenMP: Unfortunately, CPU reaches 100% when running OpenMP, therefore the time is not accurate, although the output is. Since I was not able to correctly use Snotra cluster, I could not retrieve the correct time results. Thus, the OpenMP implementation should (in theory) speed up the process.

Question 1: Critical section describes the section of the code where the code segment must not overlap between different threads, i.e. only one thread executes the code segment at a time.

Question 2: We can tell each thread how long does it need to wait before starting execution of critical section, or make threads dependent, i.e. make the thread wait until the other thread wakes it up.

Question 3: Oversubscription describes the situation when multiple threads need to be executed simultaneously. Therefore it can sometimes happen that CPU can not handle them all at the same time. As mentioned in the lectures, sometimes it can be useful, since some threads have to wait before they can be executed. Thus, optimized code can make use of that. On the other hand it can cause latency's and risk of deadlocks, if it is not properly managed.

Question 4: The pseudo-code implements the answers from above.

Algorithm 1 Ordered output

Implement a buffer and gather up all the threads.
Get the rank of the threads $[0, 1, \ldots, n-1]$
for $i \geq 1$, make the ith thread wait until $(i-1)$th thread wakes it up.
ith process runs the critical section, for $i \in [0, 1, \ldots, n-1]$.
if $i \neq (n-1)$: wake up $i+1$.