Assignment 2's Report

Steven Zaayter

June 24, 2023

1 Answers:

Note:A large number was assigned to "nb_steps" in the code to increase the execution time so it can be studied and compared.

1.1

1. If I hadn't used critical or atomic, multiple threads would have access simultaneously on the code section, which might lead to a wrong answer.
   The usage of critical or atomic prevents such behaviour and only one active thread is allowed to update the specific memory location.

Execution Time				
Versions \ Threads	1	2	4	8
Critical Section	2	2	5	11
Atomic Construct	1	0	1	1

2. Unexpectedly the Execution time increase with the increase of the numbers of threads.

3. Yes, as seen in the previous data frame the differences in performance between both version is clear. Apparently the **Critical section** and **Atomic Construct** differ in their synchronization approach.
   The "**Critical section**" ensures that only one thread can execute the code block at a time, on the other hand the "**Atomic Construct**" allows it but with ensuring that the operations are executed atomically, which lead to a relatively better performance than the latter when the shared variable is accessed frequently by multiple threads.

1.2

Execution Time				
Scheduling \ Threads	1	2	4	8
Static	0.7	2	2.9	5.8
Dynamic	5	10	10.2	20
Guided	0.7	1.5	3	6

- In our case the **Static** scheduling and the **Guided** scheduling seems to have nearly identical execution time. Which leads me to believe that in this case there is no best One scheduling algorithm that provided the best performance.
  In that case I only can try to explain why might the **Dynamic** scheduling performed the slowest, and that might have been caused by the uneven distribution of "workload" between threads thus forcing threads with shorter tasks to wait for the other to terminate their tasks to proceed to what is next in the code.