Ghent University

Parallel and Distributed Software Systems

Homework Assignment 4

Tibo Vanheule

GHENT UNIVERSITY

Academic Year 2023-2024

1 Correctness of multi-threaded code

(Q1.1): pinpoint the first error + explain why the code is incorrect

On line 45, adding a number to a certain variable in memory is not thread-safe as it involves multiple steps. Each of these steps can also use pipelining and hence, the entire operation will require multiple clock instructions. If other threads write to this memory location while another thread is computing an updated value, this will result in data-corruption (incorrect results). The ultimate outcome will depend on the precise order in which threads are writing to this memory location. This is called a data race.

Solved using the second method, a mutex together with a local variable.

(Q1.2): pinpoint the second error + explain why the code is incorrect

threadID is passed as reference[1], this causes incorrect behaviour as the variable could be updated when the thread evaluates it. Solved by passing the lvaue instead of a rvalue.

2 Influence of memory access patterns on software performance

(Q2.1) Report the runtimes and speedup.

# Threads	1	2	4	8	16
Runtime	0.434	0.830	0.940	0.900	0.444
Speedup	1	0.523	0.461	0.481	0.975

Table 1: Results without optimalisation on a hpc node. Speedup is relative to thread one.

(Q2.2) Report the runtimes and speedup.

# Threads	1	2	4	8	16
Runtime	0.365	0.195	0.097	0.050	0.037
Speedup	1	1.875	3.756	7.358	9.760

Table 2: Results with optimalisation on a hpc node. Speedup is relative to thread one.

[1]Using std::ref, which returns a std::reference_wrapper