Homework 1 report

Gustav Jenner

February 2023

1 Introduction

All tests were ran on AMD Ryzen 5 5500U with Radeon Graphics (6 core)

2 Assignment 2 Quicksort

2.1 Solution

Normal quicksort works by selecting a pivot in this case the last element in the array. Then the array is partitioned by elements lower and higher than the pivot.A new pivot is chosen for the sub arrays and the same partitioning is done (recursively) until all the elements in the array are sorted. In this solution recursive parallelism is used as instead of just calling the quicksort method recursively a separate thread is created. The advantages of this solution is that partition on the different ”section” of the array is done concurrently. Furthermore as tests were done comparing the time for a regular quicksort algorithm vs multithreaded quicksort algorithm that had no limit on the number of threads. This threaded quicksort algorithm was found to be highly inefficent as both the hardware limitation of to many threads and that it is inefficent to create a new thread for small tasks i.e partitioning a subarray with 2 elements for example. Thus a global variable was used for limiting the total number of threads, and locking was used for the critical section of checking and subtracting from the thread counter.

2.2 Result

Number of threads	Multithreaded Quicksort time(milliseconds)
1	448
2	334
4	251
8	228
16	162
32	165
64	149

Table 1: results from multithreaded quicksort code where array length $n = 10^6$ all done on same array.

3 Assignment 3 Compute pi

3.1 Solution

A similar solution to the quicksort with both recursive parallelism and limiting the amount of threads through a global counter and locks. Instead of quicksort however the adaptive quadrature routine is used to calculate the area of a circle with the radius of 1 (pi). As the run method cannot return any value, calculated area was stored in thread attribute called ans.

3.2 Result

Number of threads	Compute Pi time(milliseconds)
1	1297
2	792
4	785
8	480
12	337
32	289
64	355

Table 2: results from compute pi code where $e = 10^{-24}$.