Ignacio Deleon

February 13, 2024

Assignment 1 Problem 3

A. If we try to parallelize the `for i` loop (the outer loop), which variables should be private and which should be shared?

'I' needs to be private because each thread should have its own index. The other variables (a, n, temp, and count) should be shared because all threads read and modify them.

B. If we parallelize the `for i` loop using the scoping you specified in the previous part, are there any loop-carried dependences? Explain your answer.

Each iteration of the loop works independently with its own count value and doesn't use any other iterations.

C. Can we parallelize the call to `memcpy`? Can we modify the code so that this part of the function will be parallelizable?

It is difficult to paralyze the memcpy function because you have to write to the same destination array. Direct paralyzation of a memory copy is often tricky.

D. Write a C program that includes a parallel implementation of `Count sort`.

E. How does the performance of your parallelization of `Count sort` compare to serial `Count sort`? How does it compare to the serial `qsort` library function?

The performance of count sour depends on multiple things like the size of the input array, the amount of threads, and the overall architecture. The performance of may not always scale with the amount of threads.