CSC2002S
PARALLEL PROGRAMMING
ASSIGNMENT 1

JFFCAL001

Introduction:

The aim of this report is to outline the speedup that can be achieved by parralelising the execution of an algorithm to calculate the sunlight that trees are exposed to.

The algorithm is designed to be fed parameters for a certain area of a 2D array array corresponding to the dimensions and positions of a single tree. The algorithm then recursively segments this area into smaller areas until a cutoff is reached. Thereafter the area is sequentially iterated through and the sum of exposure returned to the stack.

Methods:

The problem of parallelising the sum of a 2D array is high parallelisable. The formulation used essentially nested the ideas used when parallelising the summing of an array, i.e. split the array in half each recursive step and use threads to calculate each half. For this 2D area, this idea is nested, so a cutoff of an array of arrays is reached (each thread has CUTOFF arrays) and then each array is summed similarly by recursively splitting. This segmenting is performed sequentially for each tree in order to preserve the output of the algorithm, namely the ordered sum of the sunlight exposure each tree receives.

The algorithm was initally validated by using a simple 5x5 input that is easily calculated to ensure the correct output:

5 5

4.5 5.5 4 3.5 4 4.5 5 4.5 4 4 5 5 5.5 4.5 5 4 4.5 5 5 4 4 4 4.5 4 3.5

2

1 1 3

2 3 4

The algorithm can replicate the desired output to satisfaction.

To find results for the speedup of this algorithm, the algorithm was run on 2 different architectures. One was a Windows laptop with a hyper-threaded dual core (4 logical processors) CPU, and the other was a Virtual Machine running Ubuntu with 2 logical processors on the same laptop.

Results and Discussion

The fully sequential implementation of the alogorithm takes a similar amount of time on each architecure. Testing for an input area of 3000x3000 with 1 000 000 trees takes roughly 0.1 seconds, with the Windows machine being slightly faster by about 0.05 seconds at most.

However, with the same input the paralell algorithms take approximately 13 seconds on the Windows machine and 40 seconds on the VM running Ubuntu. This is with a sequential-cutoff of 500, meaning each thread begins to run sequentially once the area they have is smaller than 500x500. This is undesirable and points to a problem in the algorithm. This is likely because the algorithm is sequentially calculating each tree in this way.

Therefore for this input the speedup of the is negative (it takes longer.)

However, we can clearly see that more logical processors on the Windows machine improves the operation of the parallel algorithm significantly (by about 3 times, despite only having 2 more processors).

It was also found that the VM architecture ran into a Memory problem when initially run with about 2 GB of memory. The Memory allocated had to be increased in order for the program to run.

Conclusions

The current implementation of the algorithm that is presently designed is not worth applying to this kind of problem in Java. Possible alterations include parallelisation such that multiple trees are summed concurrently and their summed exposure stored in an array so that the desired outpu can still be achieved.