Sabancı University

Faculty of Engineering and Natural Sciences

CS406/CS531
Parallel Computing
Parallel Processing and Algorithms

Spring 2022-2023

Homework #1

Due: 07/04/2023 - 23:59

PLEASE NOTE:

You HAVE TO write down the solutions on your own.
Plagiarism will not be tolerated!

Q1 (15 pts)

Suppose a fraction **r** of the runtime of a serial program is *"perfectly parallelized"*, and the remaining fraction **1 - r** is *"inherently serial"*. Give an upper bound on the speedup one can have.

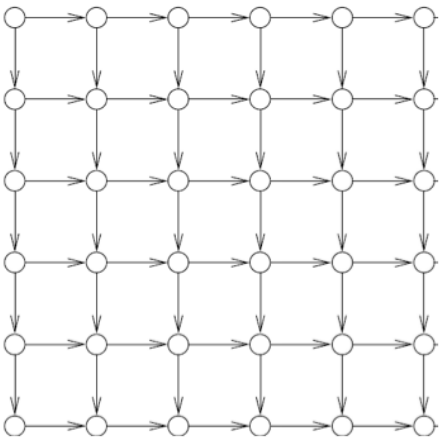
Q2 (15 pts)

Describe the following terms and their differences in **at most two** sentences each:

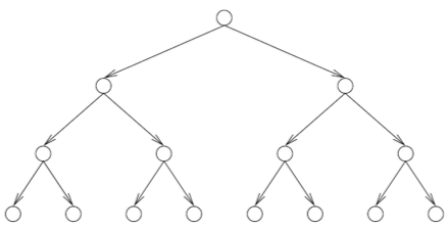
a) (5 pts) Latency **and** bandwidth
b) (5 pts) Spatial locality **and** temporal locality
c) (5 pts) Shared address space **and** distributed adress space

Q3 (15 pts)

Given two sample task dependency graphs as in the figures (3.1) and (3.2) of **N uniform** tasks. **Assume the graphs below are only instances of a family of task dependency graphs following the same pattern**, so **n** and **N** can be different.



(3.1) $N = n^2$



(3.2) $N = 2^n - 1$

a) (5 pts) What is the critical path length for each graph **in terms of n and N**?

b) (5 pts) What is the **maximum** degree of concurrency for each graph **in terms of n and N**?

c) (5 pts) What is the speedup and efficiency one can obtain **in terms n and N** if the number of processing elements is equal to the degree of concurrency? Show all your work.

Q4 (10 pts)

Implement a parallel function with OpenMP to reverse an array in five lines. Try to be as efficient as possible. The function **must be not more than 5 lines and be an in-place function**, i.e., using extra memory is not allowed.

```
#include <iostream>
#include <memory>

using namespace std;

//at most 5 lines including pragma, no memory allocation
void reverse(int* arr, int length){
  //
  //
  //
  //
  //
}


int main(){
  int N = 1000000;
  int* arr = new int[N];

  for(int i = 0; i < N; i++){
    arr[i] = rand();
  }
  reverse(arr, N);
}
```

Q5 (15 pts)

Consider a processor operating in 4GHz connected to a DRAM with a latency of 25ns (no caches). Assume the processor has two multiply-add units and is capable of executing four instructions at each cycle of 0.25 ns.

a) (3pts) What is the peak performance of the processor in FLOPS?

b) (5pts) Consider the problem of computing the dot product of two vectors (for one multiply- add, we need two data items): What is the peak performance in FLOPS for that algorithm? Explain the math you use.

c) (3pts) Assume that you have a single cycle cache and the block size (memory bandwith) is one word. What is the peak performance for dot product computation in this case?

d) (4pts) What is the peak performance again with a single cycle cache if the block size is 8?

Q6 (10 pts)

Suppose that you implemented a function **party()** as shown below:

```
int party(){
 int a = 0;
#pragma omp parallel for num_threads(3)
 {
  for(int i = 0; i < 5; i++){
    a = a + 1;
  }
 }
  return a;
}
```

a) (3 pts) While building the executable, you forgot to use **–fopenmp**. The code still compiles; what are the maximum and minimum values the function party can return?

b) (7 pts) You added **–fopenmp** to build the executable. What are the maximum and minimum possible values the function can return?

What and Where to Submit (PLEASE READ, IMPORTANT)

Your answers must be a **pdf** file (preferably prepared by LaTeX , but MS Word converted pdf's are also OK). It must clearly show your work; mathematical expressions should be written correctly and clearly. For example, you should write an equation as $x = \frac{N}{2^{n-1}}$. Otherwise, your solutions might look ambiguous and you might lose points. You don't need to include questions to your submission, just state the question and list your answers.

The file must be named as **SUCourseUserName_YourLastname_YourName_HWnumber**. Zip your pdf file into a folder of the same name and upload to SuCourse.
You will receive no credits if your compressed folder does not expand or it does not contain the correct files.
Good Luck!

CS406-531 Team (Fatih Taşyaran, Kamer Kaya)