Design and Analysis of Parallel Algorithms

Class Test

B142302

Q1

Recall Amdahl's Law for strong scaling.

(a) Imagine a parallel program which exhibits perfectly linear strong scaling. What can you infer about the structure of the code, and the communication pattern between processes? Explain your answer. [3 marks]

1. Some fraction of the code is serial. 2. The run time of the serial fraction is constant. 3. The parallel fraction is perfectly parallelisable. 4. The is no communication cost between every two processes. Hence, there exhibits perfectly linear strong scaling.

(b) Consider the below strong scaling results. State the formula for parallel speedup and parallel efficiency. Calculate the parallel speedup and parallel efficiency for each value of the number of nodes. You may use a spreadsheet program or calculator. [5 marks]

Speedup = 1/(s+p/N)
Efficiency = Speedup/N = 1/(s*N+p)
where *s* is the proportion of execution time spent on the serial part, *p* is the proportion of execution time spent on the part that can be parallelized, and *N* is the number of processors.

Number of Nodes	Runtime (s)	Parallel Speedup	Parallel Efficiency
1	961.931	1.000	1.000
2	590.758	1.628	0.814
4	407.409	2.361	0.590
8	314.037	3.063	0.383
16	267.802	3.592	0.224
24	252.318	3.812	0.159
32	242.888	3.960	0.124

(c) Use this data to estimate the serial fraction of the code. Show your working. [2 marks]
When the processor increased to 2, the parallel speedup increased to 1.628.

1/(s+(1-s)/2) = 1.628
s = 23%

(d) How many nodes would you recommend running this code on? Explain your choice. [2 marks]

I recommend using 4 nodes. The Efficiency is not low with optimistic speedup

Q2.
One of the simplest ways to create a list of all prime numbers up to some value B is the Sieve of Eratosthenes. A serial algorithm for the Sieve of Eratosthenes is presented in Algorithm 1.
(e) What is the asymptotic time complexity of this algorithm? Explain your result. [3 marks]

O(Blog(B)). The first for loop cost $\theta(B)$, the second nested for loop cost $\theta(B)*O(\log B)$ as the inner limit is a logthrimistic as we assign $j \leftarrow 2i$, the outer loop have limit 2 to B-1. So in total it is $\theta(B) + \theta(B)*O(\log(B)) = O(B\log(B))$.

(f) Algorithm 2 does not contain all the information required to calculate the asymptotic time complexity. What information is missing? [3 marks]

The number of processes. The choice of PRAM variant can affect the asymptotically achievable performance.

(g) Choose values for the missing details, then use them to calculate the asymptotic time complexity. Be sure to state your choices. [4 marks]

Choose the number of processors equal to B/2. Because the job is split in half each iteration.
The complexity is O(log(B)).

(h) Is the parallel algorithm cost optimal? Explain your result. [3 marks]

Cost_seq = O(Blog(B)).
Cost_par = O(log(B)) * O(B) = O(Blog(B)).
Cost_seq = Cost_par
Yes it is.