Chp 5 - Exercise 1 - Individual

•The best serial algorithm for an input of size n, takes n*log(n) computations.
	•We develop a parallel algorithm that uses 8 processors.
	•For input of size n, each processor does n/8 computations.
	•Additionally, each processor incurs a communication overhead cost of log(n) computations.

1. What would our equation for Speed Up look like?
2. What would our equation for Efficiency look like?
3. If we suddenly had 16 processors would this affect speed up and efficiency?

Grading Rubric
_____ (2 Points) Setup equation for #1
_____ (2 Points) Setup equation for #2
_____ (2 Points) Explained answer correctly

Answer:

1. Speedup Equation:
The speedup (S) is the ratio of the serial algorithm's time to the parallel algorithm's time.

Serial Algorithm Cost:
- Takes n log(n) computations.

Parallel Algorithm Cost with 8 processors:
- Each processor performs (n / 8) computations plus a communication overhead of log(n) computations.
- Therefore, the total parallel cost is (n / 8 + log(n)).

Speedup (S) = n log(n) / (n / 8 + log(n))

2. Efficiency Equation:
Efficiency (E) is the speedup (S) divided by the number of processors (P).

E = S / P
Substituting S, we get:
E = (n log(n) / (n / 8 + log(n))) / 8
Simplifying:
E = (n log(n)) / (n + 8 log(n))

3. Effect of 16 Processors:

Just because increased the number of processors to 16 doesn’t mean we would gain more speed and efficiency. In fact , based on the communication cost above (log(n) between processors would n + log(n) since there are more path ways to communication which leads to more overhead needed and synchronization. In Addition, increase the processors compared to improving the tasks divide and conquer part doesn’t always yeilds benefits in linear fashion. When using 16 processors, each processor performs n / 8 computations instead of n / 16. This is because certain tasks can't be perfectly split, leading to inefficiencies and overlapping work. Additionally, there might be dependencies between computations, causing some processors to wait for others to complete their tasks, further preventing a linear reduction in workload. Thus, increasing processors doesn't always result in a proportional decrease in individual workload due to these factors.