Assignment 1: Performance Analysis

Ekkapot Charoenwanit
Course: Parallel Computing

March 26, 2021

Problem 1.1. Amdahl's law and Gustafson's law

1) Suppose you need to run a simulation for your research and your advisor wants you to run it for a fixed problem size and he expects to see the results in a month. Having run the simulation with the same problem size for several times, you are sure that it will take longer than a month. Thus, you are considering parallelizing your simulation kernel to speed up the running time. After profiling the simulation kernel on one processor, you find out that 92% of the simulation kernel has the potential to benefit from multiple processors and 8% of it is sequential.

What speedup can you expect when the kernel is run on 16 processors?

What is the maximum speedup you would expect provided that you had access to an infinite number of processors?

2)[Quinn] Benchmarking of a sequential program reveals 95% of the execution time is spent inside functions that are amenable to parallelization.

What is the maximum speedup we could expect from executing a parallel version of this program on 10 processors?

3)[Quinn] For a problem size of interest, 6% of the operations of a parallel program are inside I/O functions that are executed on a single processor.

What is the minimum number of processors needed in order for the parallel program to exhibit speedup of 10?

4) [Quinn] Brandon's parallel program executes in 242 seconds on 16 processors. Through benchmarking, he determines that 9 seconds is spent performing initialization and cleanup on one processor. During the remaining 233 seconds, all 16 processors are active.

What is the scaled speedup achieved by Brandon's program?

5) Given a sequential program where the sequential fraction has been determined to be 0.20 for a fixed problem size of interest, what is the expected speedup ψ and efficiency ε if a parallel implementation is to execute on 2, 4, 8, 16 and an infinite number of processors?

p	2	4	8	16	∞
ψ					
ε					-

6) Suppose that a simulation kernel makes extensive use of floating-point operations, with 75% of the execution time consumed by the floating-point operations. With a new hardware design, the number of

floating-point units has doubled.

What speedup can you expect from running the simulation kernel on the new hardware?

Problem 1.2. The Karp-Flatt Metric

1) Given the following results of a parallel program, what can you say about the scalability of the parallel program? If the scalability is poor, is it because of Amdahl's law or an increased parallel overhead in the parallel program?

p	2	4	8	16	32	64
ψ	1.67	2.50	3.33	4.00	4.44	4.69
e						

2) Given the following results of a parallel program, what can you say about the scalability of the parallel program? If the scalability is poor, is it because of Amdahl's law or an increased parallel overhead in the parallel program?

p	2	4	8	16	32	64
ψ	1.20	2.40	4.80	9.60	19.2	38.4
e						

3) Given the following results of a parallel program, what can you say about the scalability of the parallel program? If the scalability is poor, is it because of Amdahl's law or an increased parallel overhead in the parallel program?

p	2	4	8	16	32	64
ψ	1.90	3.20	5.00	6.20	6.50	6.90
e						

Problem 1.3. Isoefficiency

1) Suppose that you are implementing a parallel reduction algorithm where the problem size scales with an increasing number of processors. The computing cluster you are using has 4 GB of memory per processor. It is given that parallel execution of the parallel reduction algorithm on 8 processors uses 1 GB of memory per processor. Having done an isoefficiency analysis on the parallel program, you find out that the isoefficiency relation is $n = C_1 p \log_2 p$. Assume that the total memory consumption is $M(n) = C_2 n$.

Can you scale your parallel reduction algorithm to 256 processors with the efficiency remaining constant as in the execution on 8 processors?

2) [Quinn] Let $n = f(p)$ denote the isoefficiency relation of a parallel system and $M(n)$ denote the amount of memory required to store a problem of size n. Use the scalability function $\frac{M(f(p))}{p}$ to rank the parallel systems shown below from most scalable to least scalable.
(a) $f(p) = Cp$ and $M(n) = n^2$
(b) $f(p) = C\sqrt{p}\log p$ and $M(n) = n^2$
(c) $f(p) = C\sqrt{p}$ and $M(n) = n^2$
(d) $f(p) = Cp\log p$ and $M(n) = n^2$

(e) $f(p) = Cp$ and $M(n) = n$