Lab 4 Report

Andres Calderon

acald013@ucr.edu

November 28, 2016

1. Benchmarking of a sequential program reveals that 95 percent 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?
Answer
According to Amdahl's Law:
$$\psi \leq \frac{1}{f + \frac{1-f}{p}}$$
As $p = 10$ and f is the sequential part equal to 5%, the formula becomes:
$$\psi \leq \frac{1}{0.05 + \frac{1-0.05}{10}}$$
$$\psi \leq \frac{1}{0.05 + 0.095}$$
$$\psi \leq \frac{1}{0.145} = 6.896552$$
So the the maximum speedup it can be expected is 6.89.

2. For a problem size of interest, 6 percent 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 a speedup of 10?
Answer
Similarly, following Amdahl's Law:
$$\psi \leq \frac{1}{f + \frac{1-f}{p}}$$
We have $f = 0.6$ and we want $\psi = 10$, so the formula becomes:
$$10 \leq \frac{1}{0.06 + \frac{1-0.06}{p}}$$
$$0.06 + \frac{1-0.06}{p} \leq \frac{1}{10}$$
$$\frac{1-0.06}{p} \leq \frac{1}{10} - 0.06$$
$$\frac{1-0.06}{\frac{1}{10} - 0.06} \leq p$$
$$p \geq \frac{0.94}{0.1 - 0.06}$$
$$p \geq \frac{0.94}{0.04}$$
$$p \geq 23.5$$
The minimum number of processors to achieve a speedup of 10 is 24.

3. What is the maximum fraction of the computation that are inherently sequential if a parallel application is to achieve a speedup of 50 over its sequential counterpart?
Answer
To answer this question we need to follow Amdahl's Law to find f when $p \to \infty$:

$$50 \leq \frac{1}{f + \frac{1-f}{\infty}}$$

As the expression $\frac{1-f}{\infty}$ becomes 0 we have:

$$50 \leq \frac{1}{f}$$

$$f \leq \frac{1}{50}$$

$$f \leq 0.02$$

The maximum fraction of the computation that are inherently sequential cannot be greater than 2%.

4. Shauna's parallel program achieves a speedup of 9 on 10 processors. What is the maximum fraction of the computation that may consist of inherently sequential operations?
Answer
Similarly to the last question, we have $\psi = 9$ and $p = 10$. Replacing on Amdahl's Law we have:

$$9 \leq \frac{1}{f + \frac{1-f}{10}}$$

$$f + \frac{1-f}{10} \leq \frac{1}{9}$$

$$\frac{10f + 1 - f}{10} \leq \frac{1}{9}$$

$$9f + 1 \leq \frac{10}{9}$$

$$9f \leq 1.11 - 1$$

$$f \leq \frac{0.11}{9}$$

$$f \leq 0.0123$$

We conclude that the maximum fraction of the computation that may consist of inherently sequential operations cannot be greater than 1.23%.

5. Brandon's parallel program executes in 242 seconds on 16 processors. Through benchmarking he determines that 9 seconds is spent performing initializations 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?
Answer
According to Gustafson-Barsis' Law we have:

$$\psi \leq p + (1-p)s$$

As $p = 16$ and $s = \frac{9}{242} = 0.037$ the formula becomes:

$$\psi \leq 16 + (1-16)0.037$$

$$\psi \leq 16 - 0.55$$

$$\psi \leq 15.44$$

The scaled speedup which can be achieved is 15.44.

6. Courtney benchmarks one of her parallel programs executing on 40 processors. She discovers that it spends 99 percent of its time inside parallel code. What is the scaled speedup of her program?
Answer
Similarly, applying Gustafson-Barsis' Law we have $p = 40$ and $s = 0.01$:

$$\psi \leq 40 + (1 - 40)0.01$$

$$\psi \leq 40 - 0.39$$

$$\psi \leq 39.61$$

The scaled speedup which can be achieved is 39.61.

7. If a parallel program achieves a speedup of 9 with 10 processors, is it possible to achieve a speedup of 90 with 100 processors if it is ran with the same problem size on the same parallel platform? Why?
Answer
According to the Karp-Flatt metric:

$$e = \frac{\frac{1}{\psi} - \frac{1}{p}}{1 - \frac{1}{p}}$$

As $\psi = 9$ and $p = 10$ we have:

$$e = \frac{\frac{1}{9} - \frac{1}{10}}{1 - \frac{1}{10}}$$

$$e = \frac{0.011}{0.9} = 0.012$$

If we would like to achieve a speedup of 90 ($\psi = 90$) with 100 processors ($p = 10$), the Karp-Flatt metric should be:

$$e = \frac{\frac{1}{90} - \frac{1}{100}}{1 - \frac{1}{100}}$$

$$e = \frac{0.0011}{0.99} = 0.0011$$

It is unlikely that the Karp-Flatt metric decreases when more processors are added, so it is not possible to achieve the above-mentioned assumption.

8. Assume a parallel program takes 1000 seconds to finish when using 1 processor and 500 seconds to finish when using 4 processors. What is the minimum time to finish the program when using 16 processors? Assume the problem size is fixed.
Answer
We know that the speedup (ψ) is the ratio between sequential execution time and parallel execution time:

$$\psi = \frac{Sequential\ execution\ time}{Parallel\ execution\ time} \tag{1}$$

So, the speedup using 4 processors is:

$$\psi = \frac{1000}{500} = 2$$

Since efficiency is equal to speedup divided by p:

$$\varepsilon = \frac{\psi}{p} \tag{2}$$

$$\varepsilon = \frac{2}{4} = 0.5$$

We also know that, for a problem of fixed size, the efficiency of a parallel computation typically decreases as the number of processors increases. So, we can estimate the speedup of a system with 16 processors using (2):

$$0.5 \geq \frac{\psi}{16}$$

$$\psi \leq 8$$

If the maximum speedup is 8, the minimum parallel execution time (x) with 16 processors (replacing in (1)):

$$8 = \frac{1000}{x}$$

$$x = \frac{1000}{8}$$

$$x = 125$$

9. Let $n \geqslant f(p)$ 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 to rank the parallel systems shown below from most scalable to least scalable.
Answer
According the scalability function $\frac{M(f(p))}{p}$ we can solve each system as follows:

a) $f(p) = Cp$ and $M(n) = n^2$:

$$\frac{M(f(p))}{p} = \frac{M(Cp)}{p} = \frac{C^2p^2}{p} = C^2p$$

b) $f(p) = C\sqrt{p}\log p$ and $M(n) = n^2$:

$$\frac{M(f(p))}{p} = \frac{M(C\sqrt{p}\log p)}{p} = \frac{C^2p\log^2 p}{p} = C^2\log^2 p$$

c) $f(p) = C\sqrt{p}$ and $M(n) = n^2$:

$$\frac{M(f(p))}{p} = \frac{M(C\sqrt{p})}{p} = \frac{C^2p}{p} = C^2$$

d) $f(p) = Cp\log p$ and $M(n) = n^2$:

$$\frac{M(f(p))}{p} = \frac{M(Cp\log p)}{p} = \frac{C^2p^2\log^2 p}{p} = C^2p\log^2 p$$

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

$$\frac{M(f(p))}{p} = \frac{M(Cp)}{p} = \frac{Cp}{p} = C$$

f) $f(p) = p^C$ and $M(n) = n$:

$$\frac{M(f(p))}{p} = \frac{M(p^C)}{p} = \frac{p^C}{p} = p^{C-1}$$

As $1 < C < 2$ we can assume $\mathcal{O}(p)$.

g) $f(p) = p^C$ and $M(n) = n$:

$$\frac{M(f(p))}{p} = \frac{M(p^C)}{p} = \frac{p^C}{p} = p^{C-1}$$

As $2 < C < 3$ we can assume $\mathcal{O}(p^2)$.

So the rank of the parallel systems from most to least scalable is:

1^{st} System e

2^{nd} System c

3^{rd} System b

4^{th} System f

5^{th} System a

6^{th} System d

7^{th} System g

10. Assume the computation time for your sequential matrix-matrix multiplication program is $2n^3$, and, in the parallel version of the program, the communication time is $16n^2 \log_2 p$. For a problem size n, the total memory needed for the algorithm is $24n^2$ bytes. If your parallel computer has 1024 cores with 1 Gbytes DRAM per core, what is the maximum speed up you can achieve on your computer? If you want to achieve a speed up of 256, what is the minimum problem size you need to run your program with?

Answer

We have that the sequential time complexity is:

$$T(n,1) = \mathcal{O}(2n^3) \tag{3}$$

We assume that there is no sequential part in the parallel program and all the processor take part of the communication step. So, the parallel overhead is:

$$T_0(n,p) = \mathcal{O}(p16n^2 \log_2 p) \tag{4}$$

We know that:

$$\varepsilon(n,p) \leq \frac{1}{1 + \frac{T_0(n,p)}{T(n,1)}}$$

Since speedup equals efficiency times p, we have:

$$\psi(n,p) \leq \frac{p}{1 + \frac{T_0(n,p)}{T(n,1)}}$$

Replacing (3) and (4) we get:

$$\psi(n,p) \leq \frac{p}{1 + \frac{p16n^2 \log_2 p}{2n^3}}$$

$$\psi(n,p) \leq \frac{p}{1 + \frac{8p \log_2 p}{n}} \tag{5}$$

From the statement we know that the total memory available of the system is 1024×1024^3 bytes. Also, for a problem of size n, the total memory needed is $24n^2$ bytes. So, the maximum n which can be computed is:

$$24n^2 = 1024 \times 1024^3$$

$$n^2 = \frac{1024^4}{24}$$

$$n = \sqrt{\frac{1024^4}{24}}$$

$$n = 214039$$

Replacing $p = 1024$ and $n = 214039$ in (5) we get:

$$\psi(n,p) \leq \frac{1024}{1 + \frac{8 \times 1024 \times \log_2 1024}{214039}}$$

$$\psi(n,p) \leq 809.3$$

So, the maximum speed up the program can achieve is 809.3.
For the second question, we will find n in (5) knowing that $\psi = 256$ and $p = 1024$:

$$256 \leq \frac{1024}{1 + \frac{8 \times 1024 \times \log_2 1024}{n}}$$

$$1 + \frac{8 \times 1024 \times 10}{n} \leq \frac{1024}{256}$$

$$\frac{8 \times 1024 \times 10}{n} \leq 4 - 1$$

$$\frac{8 \times 1024 \times 10}{3} \leq n$$

$$n \geq 27306$$

So, the minimum problem size to keep an efficiency of 256 is 27306.