Jonathan Frame

CS **491/591**: High Performance Computing Project 4

Performance Analysis of Parallel Programs

Due at 11:59 PM on April 2nd, 2019

For this project we will be mostly using speedup and efficiency of parallel programs. Speedup is defined as as the ratio of the serial runtime of the best sequential algorithm for solving a problem to the time taken by the parallel algorithm to solve the same problem on p processors.

$$\psi(n,p) \leq \frac{\sigma(n)+\phi(n)}{\sigma(n)+\frac{\phi(n)}{p}+\kappa(n,p)}$$

Where $\sigma(n)$ is the inherently sequential computation and $\phi(n)$ is the computation amenable to parallelization. We will also be taking advantage of the fact that the numerator of the speedup equation should sum to 1, and that the two components can generally be expressed as functions of the other:

$$\phi(n) = (1 - \sigma(n))$$

$$\sigma(n) = (1 - \phi(n))$$

Efficiency, ε, is the ratio of speedup to the number of cores.

$$\varepsilon(n,p) \leq \frac{\psi}{p}\ ;\ \varepsilon(n,p) \leq \frac{\sigma(n)+\phi(n)}{p\sigma(n)+\phi(n)+p\kappa(n,p)}$$

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?

Speedup is defined as as the ratio of the serial runtime of the best sequential algorithm for solving a problem to the time taken by the parallel algorithm to solve the same problem on p processors.

$$\sigma = 0.95\ +\ 0.05\ ;\ \phi =\ 0.95\tfrac{1}{10} + 0.05$$

$$\psi\ =\ \frac{0.95 + 0.05}{0.95\frac{1}{10} + 0.05}\ ;\ \psi\ =\ \frac{(0.95 + 0.05)}{(\frac{0.95}{10} + 0.05)}$$

From here we can get to Amdahl's Law:

$\psi \leq \frac{1}{f+\frac{(1-f)}{p}}$ :: $\psi \leq \frac{1}{0.05+\frac{0.95}{10}}$

$\psi \leq \frac{1}{(0.095+0.05)}$; $\psi \leq \frac{1}{0.145}$; $S_{max} \approx 6.9$

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?

$\psi = \frac{T_s}{T_p} = 10$;

$\sigma = 0.94 + 0.06$; $\phi = 0.94\frac{1}{p} + 0.06$

Getting to Amdahl's law: $10 = \frac{1}{\frac{0.94}{p} + 0.06}$

$\frac{1}{10} = \frac{0.94}{p} + 0.06$; $0.1 - 0.06 = \frac{0.94}{p}$; $p = \frac{0.94}{0.04}$; $p = 24$

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?

$\psi = 50$;

$T_s = \sigma + \phi$; $T_p = \sigma + \frac{\phi}{p}$;

Where C_s is the inherently sequential computation and C_p is the computation amenable to parallelization.

$\psi = \frac{(\sigma+\phi)}{(\sigma+\frac{\phi}{p})}$

Amdahl's Law can be expressed with fixed values for n changing p:

$\psi \leq \frac{1}{f+\frac{(1-f)}{p}}$; $f = \frac{\sigma(n)}{\sigma(n)+\phi(n)}$

Above f is the fraction of computation that is inherently sequential.

$$50 = \frac{1}{f + \frac{(1-f)}{p}};$$

$$\frac{1}{50} = f + \frac{(1-f)}{p}$$

Take the limit as p goes to infinity:

$$\lim_{p \to \infty} f + \frac{(1-f)}{p} = f$$

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

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?

$$\psi(n,p) \le \frac{\sigma(n)+\varphi(n)}{\sigma(n)+\frac{\varphi(n)}{p}} = 9$$

Where $\sigma(n)$ is the inherently sequential computation and $\phi(n)$ is the computation amenable to parallelization.

To solve for the ratio $Assume: \ \sigma + \phi = 1; \ and\ replace\ \sigma\ with\ f\ so\ that\ \phi = 1 - f$

$$9 = \frac{1}{\sigma + \frac{\phi}{10}};$$

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

$$f = \frac{1}{81}$$

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?

$$\psi = \frac{T_s}{T_p}$$

Where Ts is the time of the serial program and Tp is the time of the parallel program.

$T_s = 16 \times 233 + 9$; $T_p = 233 + 9$

$S = \frac{3737}{233}$; $S \approx 16$

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?

$S = \frac{T_s}{T_p}$

$\sigma = 0.99 + 0.01$; $\phi = 0.99\frac{1}{40} + 0.01$

$\psi = \frac{0.99 + 0.01}{0.99\frac{1}{40} + 0.01}$; $\psi = \frac{1}{\frac{0.99}{40} + 0.01}$; $S = 28.777$

Using Gustafson-Barsis's Law to predict scaled speedup:

$\psi \le p + (1-p)s$, $where\ s = \frac{\sigma(n)}{(\sigma(n)+\frac{\varphi(n)}{p})}$

$s = \frac{0.01}{\frac{0.99}{40} + 0.01}$; $s = 0.2878$

$\psi \le 40 + (1-40)\ \frac{0.01}{\frac{0.99}{40}+0.01}$

$\psi \le 40 - 39 \times 0.28777$; $\psi \le 40 - 11.21$; $\psi \le 28.7769$

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?

Ok, let's go back to Shauna's example to find out:

$\psi = 9$;

$T_s = \sigma + \phi$; $T_p = \sigma + \frac{\phi}{10}$;

$9 = \frac{\sigma+\phi}{\sigma+\frac{\phi}{10}}$; $9 = \frac{1}{f+\frac{1-f}{100}}$; $f = \frac{1}{81}$

To solve for the ratio $Assume:\ \sigma + \phi = 1;\ and\ replace\ \sigma\ with\ f\ so\ that\ \phi = 1 - f$

And obviously use the new speedup (90) and processors (100) values.

$$90 = \frac{1}{f + \frac{1-f}{100}}$$

$$f = \frac{1}{891}$$

The answer is No. Because in theory the fraction of inherently sequential computation would have to decrease in time from $\frac{1}{81}$ to $\frac{1}{891}$, which could not happen.

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.

$$T_s = \sigma + \phi = 1000;$$

$$T_{p4} = \sigma + \frac{\phi}{4} = 500;$$

$$\psi_4 = \frac{1000}{500}; \psi_4 = 2$$

$$\psi_{16} = 2; \ \psi_{16} = \frac{1000}{T_{p16}}$$

$$Assume\ \sigma + \phi = 1\ and\ solve\ for\ \phi:$$

$$1000((1-\phi) + \frac{\phi}{4}) = 500$$

$$(1-\phi) + \frac{\phi}{4} = \frac{1}{2}$$

$$1 - \phi + \frac{\phi}{4} = \frac{1}{2}$$

$$-\phi + \frac{\phi}{4} =- \frac{1}{2}$$

$$\phi - \frac{\phi}{4} = \frac{1}{2}$$

$$\phi(1 - \frac{1}{4}) = \frac{1}{2}$$

$$\phi(\frac{3}{4}) = \frac{1}{2}$$

$$\phi = \frac{2}{3}$$

$$\sigma = \frac{1}{3}$$

Now Solve for p=16

$T_{p16} = 1000(\sigma + \frac{\phi}{16})$

$T_{p16} = 1000(\frac{1}{3} + \frac{2}{3} \cdot \frac{1}{16})$

$T_{p16} = 375$

9. Let $n \geq f(p)$ 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.
   a. $f(p) = Cp \text{ and } M(n) = n^2;$ $\quad \frac{(Cp)^2}{p} = C^2p$
   b. $f(p) = C\sqrt{p}\, log(p) \text{ and } M(n) = n^2;$ $\quad \frac{(C\sqrt{p}\, log(p))^2}{p} = (C\, log(p))^2 = C^2 log(p)^2$
   c. $f(p) = C\sqrt{p} \text{ and } M(n) = n^2;$ $\quad \frac{(C\sqrt{p})^2}{p} = C^2$
   d. $f(p) = Cp\, log(p) \text{ and } M(n) = n^2;$ $\quad \frac{(Cp\, log(p))^2}{p} = (Cp\, log(p))^2 = C^2p^2 log(p)^2$
   e. $f(p) = Cp \text{ and } M(n) = n$ $\quad \frac{Cp}{p} = C$
   f. $f(p) = p^c \text{ and } M(n) = n.\ Assume\ 1 < C < 2.;$ $\quad \frac{p^{(1<C<2)}}{p} = p^{(0<C<1)}$
   g. $f(p) = p^c \text{ and } M(n) = n.\ Assume\ C > 2.;$ $\quad \frac{p^{(C>2)}}{p} = p^{(C>1)}$

The scalability function shows how memory usage per processor must grow to maintain efficiency. We call

$\frac{M(f(p))}{P}$

the scalability function.

Rank in order from scalable to least scalable:
If C = 1, then c. = e. Otherwise e. < c.

1. e. = C
2. c. = C^2
3. f. = $p^{(0<C<1)}$
4. b. = $C^2 log(p)^2$
5. a. = C^2p
6. g. = $p^{(C>1)}$
7. d. = $C^2p^2 log(p)^2$

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?

$Sequential\ computation\ time = 2n^3$;

Assume zero computation is inherently sequential, so all can be parallelized.

$$\sigma = 0$$

$$\phi = 2n^3 - \sigma; \ \phi = 2n^3$$

We can solve for n with the knowledge that the total memory needed is 24n2 and the computer has 1024 cores with 1GB of ram:

$$Total\ memory = \#cores \times 1GB$$

$$24n^2 = 1024 \times 10^9$$

$$n = \sqrt{\frac{1024 \times 10^9}{24}}$$

$$n = 205548$$

$$\phi = 2(205548)^3$$

$$\phi = 2(205548)^3$$

$$p = 1024$$

$$\psi(n,p) \leq \frac{\sigma(n)+\phi(n)}{\sigma(n)+\frac{\phi(n)}{p}+\kappa(n,p)} = \frac{0+\phi}{0+\frac{\phi}{1024}+16(205{,}548)^2 log_2(1024)}$$

$$16(205548)^2 log_2(1024) = 16 \times 1048576 \times 10 = 167772160$$

Efficiency, ε, is the ratio of speedup to the number of cores.

$$\varepsilon(n,p) \leq \frac{\psi}{p}$$

$$\varepsilon(n,p) \leq \frac{\sigma(n)+\varphi(n)}{p\sigma(n)+\varphi(n)+p\kappa(n,p)}$$

So

$$\varepsilon(n,p) \le \frac{0+\phi(n)}{\phi(n)+(1024)16(205548)^2 log_2(1024)}$$

Isoefficiency

$$\varepsilon = \frac{1}{1+\frac{(p-1)\sigma(n,p)+p\kappa(n,p)}{\sigma(n)+\phi(n)}}$$

We are assuming $\boldsymbol{\sigma}$ = 0.

$$\varepsilon = \frac{1}{1+\frac{p16n^2 log_2(p)}{\phi(n)}}$$

$$T_0(n,p) = (p-1)\sigma(n) + p\kappa(n,p)$$
$$T(n,1) = \sigma(n) + \phi(n)$$
$$\varepsilon \le \frac{1}{1+\frac{T_0(n,p)}{T(n,1)}}$$

$$\varepsilon = \frac{1}{1+\frac{p16n^2 log_2(p)}{\phi(n)}}$$

Assume efficiency is constant to get:

$$\phi(n) \ge \frac{\varepsilon}{1-\varepsilon} p16n^2 log_2(p)$$

And plug in the definition of efficiency (speedup/processors), then solve for speedup:

$$\phi(n) \ge \frac{\frac{\psi}{p}}{1-\frac{\psi}{p}} p16n^2 log_2(p)$$

Divide by $\boldsymbol{\varphi}$ and set $\frac{p16n^2 log_2(p)}{\phi}$ as a constant, C, for now

$$1 \approx \frac{\frac{\psi}{p}}{1-\frac{\psi}{p}} C$$

$$\psi \approx \frac{p}{C+1}$$

$$\psi \approx \frac{p}{\frac{p16n^2 log_2(p)}{\phi}+1}$$

$$\psi_{max} \approx 732$$

So then to answer the second questions, plug in 256 for speedup and solve for n:

$$256 \approx \frac{p}{\frac{p16n^2 log_2(p)}{\phi}+1} \; ; \; n_{min\ for\ \psi=256} \approx 27306$$