CS211: High Performance Computing Project 4

Performance Analysis of Parallel Programs

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?
Solution:

$$Maximum\ Speedup = \frac{1}{5\% + 95\% \div 10} = 6.90$$

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?
Solution:
Assuming there are N processors, in order for the parallel program to exhibit a speedup of 10, we have the inequality:

$$Speedup = \frac{1}{6\% + 94\% \div N} \geq 10 \Rightarrow N \geq 94\% \div (10\% - 6\%) = 23.5 \Rightarrow N \geq 24$$

Therefore the minimum number of processors needed in order for the parallel program to exhibit a speed up of 10 is 24.

3. What is the maximum fraction of execution time that can be spent performing inherently sequential operations if a parallel application is to achieve a speedup of 50 over its sequential counterpart?
Solution:
Assuming the fraction of execution time that can be spent performing inherently sequential operations is $x\%$ and $speedup = \frac{1}{x\% + (1 - x\%) \div N}$. When N tends to infinity, speedup tends to $\frac{1}{x\%}$. Therefore in order to achieve a speedup of 50 over its sequential counterpart, there should have $\frac{1}{x\%} \geq 50 \Rightarrow x \leq 2$.
The maximum fraction is 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?
Solution:

$$Speedup = \frac{1}{x\% + (1 - x\%) \div 10} = 9 \Rightarrow x\% = \frac{1}{81} = 1.23\%$$

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?
Solution:

$$Scaled\ Speedup = \frac{9+233\times16}{9+233} = 15.44$$

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?
Solution:

$$Scaled\ Speedup = \frac{0.01+0.99\times40}{0.01+0.99} = 39.61$$

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?
Solution:

According to Amdahl's Law: $\psi = \frac{1}{f+(1-f)/p}$, when $p=10$ and $\psi=9$, there should be $f=\frac{1}{81}$. And due to running with the same problem size on the same parallel platform, $f=\frac{\sigma(n)}{\sigma(n)+\varphi(n)}$ is constant. So when $p=100$,

$\psi \le \frac{1}{f+(1-f)/p} = \frac{1}{1/81+(1-1/81)/100} = 45$. Therefore, it's impossible to achieve a speedup of 90 with 100 processors.

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.
Solution:
When using 4 processors, the speedup is 1000/500 = 2. According to Amdahl's Law: $\psi = \frac{1}{f+(1-f)/p}$, when $p=4$ and $\psi=2$, there should be $f=\frac{1}{3}$. And due

to problem size fixed, $f = \frac{\sigma(n)}{\sigma(n)+\varphi(n)}$ is constant. So when $p = 16$,

$\psi \le \frac{1}{f+(1-f)/p} = \frac{1}{1/3+(1-1/3)/16} = 2.67$. Therefore, the minimum time to finish the program when using 16 processors is 1000/2.67 = 375 seconds.

9. Solution:

a. $Scalability\ function = M(f(p))/p = (Cp)^2/p = C^2 p$

b. $Scalability\ function = M(f(p))/p = (C\sqrt{p}\log p)^2/p = C^2\log^2 p$

c. $Scalability\ function = M(f(p))/p = (C\sqrt{p})^2/p = C^2$

d. $Scalability\ function = M(f(p))/p = (Cp\log p)^2/p = C^2 p\log^2 p$

e. $Scalability\ function = M(f(p))/p = (Cp)/p = C$

f. $Scalability\ function = M(f(p))/p = p^C/p = p^{C-1}(1 < C < 2)$

g. $Scalability\ function = M(f(p))/p = p^C/p = p^{C-1}(C > 2)$

From most scalable to least scalable is c = e > b > f > a > d > g.

10. Solution:

$Speedup \le \frac{2n^3}{2n^3/p+16n^2\log_2 p} = \frac{1}{1/p+8\log_2 p/n}$, so if we want to achieve higher speedup, we should increase n. Because each core has 1G bytes DRAM and the total memory needed for the algorithm is $24n^2$ bytes, so there should be

$24n^2 \le 1G \times p \Rightarrow n \le \sqrt{1024\times1024^3 \div 24} = 214039$. Therefore the maximum speedup is $\frac{1}{1/p+8\log_2 p/n} = \frac{1}{1/1024+8\log_2 1024/214039} = 740.56$.

If we want to achieve a speedup of 256, then $256 \le \frac{1}{1/p+8\log_2 p/n} \Rightarrow n \ge 27307$.

Therefore, the minimum problem size to achieve speedup of 256 is 27307.