Chapter 7 Exercises

B06901145 李子筠

December 4, 2020

Exercise 1

Using the definition of speedup presented in Section 7.2, prove that there exists a p_0 such that $p > p_0 \implies \psi(n,p) < \psi(n,p_0)$. Assume $\kappa(n,p) = C \log p$.

$$\begin{aligned}
\psi(n,p) < \psi(n,p_0) &\iff \frac{\sigma(n)+\varphi(n)}{\sigma(n)+\varphi(n)/p+\kappa(n,p)} < \frac{\sigma(n)+\varphi(n)}{\sigma(n)+\varphi(n)/p_0+\kappa(n,p_0)} \text{ (by definition)} \\
&\iff \frac{\sigma(n)+\varphi(n)}{\sigma(n)+f(p)} < \frac{\sigma(n)+\varphi(n)}{\sigma(n)+f(p_0)}, \text{ let } f(p) = \varphi(n)/p + \kappa(n,p) \\
&\iff \sigma(n)+f(p_0) < \sigma(n)+f(p) \\
&\iff f(p_0) < f(p)
\end{aligned}$$

$$\begin{aligned}
f(p) &= \varphi(n)/p + \kappa(n,p) = \varphi(n)/p + C\log p \\
f'(p) &= -\varphi(n)/p^2 + C/(p \ln 2) \\
f'(p) &= 0 \text{ when } p = (\varphi(n)\ln 2)/C
\end{aligned}$$

Let $p_0 = (\varphi(n)\ln 2)/C$,
for $p > p_0$, $f'(p) > 0$, $f(p)$ is increasing, so $f(p) > f(p_0)$, $\psi(n,p) < \psi(n,p_0)$ (derived above).
Therefore, there exists a $p_0 = (\varphi(n)\ln 2)/C$ such that $p > p_0 \implies \psi(n,p) < \psi(n,p_0)$.

Exercise 2

Starting with the definition of efficiency presented in Section 7.2, prove that $p' > p \implies \varepsilon(n,p') \le \varepsilon(n,p)$.

$$\begin{aligned}
p' > p &\implies p'\sigma(n) + p'\kappa(n,p') \ge p\sigma(n) + p\kappa(n,p) && (\kappa(n,p) \text{ is non-decreasing}) \\
&\implies \frac{\sigma(n)+\varphi(n)}{\varphi(n)+p'\sigma(n)+p'\kappa(n,p')} \le \frac{\sigma(n)+\varphi(n)}{\varphi(n)+p\sigma(n)+p\kappa(n,p)} \\
&\implies \varepsilon(n,p') \le \varepsilon(n,p) && \text{(by definition)}
\end{aligned}$$

Exercise 3

Estimate the speedup achievable by the parallel reduction algorithm developed in Section 3.5 on 1, 2, ..., 16 processors. Assume $n = 1,000,000$, $\chi = 10$ nanoseconds and $\lambda = 100\ \mu$ sec.

Sequential execution time: $\chi(n-1)$

Parallel execution time: $\chi\left(\left\lceil \frac{n}{p} \right\rceil - 1\right) + \lceil \log p \rceil (\lambda + \chi)$

Speedup: $\dfrac{\chi(n-1)}{\chi\left(\left\lceil \frac{n}{p} \right\rceil - 1\right) + \lceil \log p \rceil (\lambda + \chi)}$

For $n = 1,000,000$:

p	1	2	3	4	5	6	7	8
Speedup	1.00	1.96	2.83	3.70	4.35	5.08	5.79	6.45

p	9	10	11	12	13	14	15	16
Speedup	6.62	7.14	7.64	8.11	8.55	8.97	9.37	9.76

Exercise 4

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?

By Amdahl's Law,

$$\begin{aligned} \psi &\leq \frac{1}{f + (1-f)/p} \\ \psi &\leq \frac{1}{5\% + 95\%/10} \\ \psi &\leq 6.896 \end{aligned}$$

The maximum speedup achievable is 6.90.

Exercise 5

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?

By Amdahl's Law,

$$\begin{aligned} 10 &\leq \frac{1}{6\% + (1-6\%)/p} \\ 10(6\% + 94\%/p) &\leq 1 \\ 940\%/p &\leq 1 - 60\% \\ p &\geq 940\%/40\% \\ p &\geq 23.5 \end{aligned}$$

At least 24 processors are needed for the parallel program to exhibit a speedup of 10.

Exercise 6

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?

By Amdahl's Law,

$$\begin{aligned}
50 &\le \frac{1}{f + (1-f)/p} \le \lim_{p\to\infty} \frac{1}{f + (1-f)/p} \\
50 &\le 1/f \\
f &\le 1/50 = 0.02
\end{aligned}$$

At most 2% of execution time that can be spent performing inherently sequential operation for the program to achieve a speedup of 50.

Exercise 7

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?

By Amdahl's Law,

$$\begin{aligned}
9 &\le \frac{1}{f + (1-f)/10} \\
8.1f + 0.9 &\le 1 \\
f &\le 0.0123
\end{aligned}$$

At most 1.23% of computation that may consist of inherently sequential operations.

Exercise 8

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?

By Gustafson-Barsis's Law,

$$\begin{aligned}
\psi &\le p + (1-p)s \\
\psi &\le 16 - 15 \cdot \frac{9}{242} \\
\psi &\le 15.442
\end{aligned}$$

The scaled speedup achieved by Brandon's program is 15.44.

Exercise 9

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?

By Gustafson-Barsis's Law,

$$\psi \leq p + (1 - p)s$$
$$\psi \leq 40 - 39(1 - 99\%)$$
$$\psi \leq 39.61$$

The scaled speedup of Courtney's program is 39.61.

Exercise 10

The execution times of six parallel programs, labeled I-VI, have been benchmarked on 1, 2, ..., 8 processors. The following table presents the speedups achieved by these programs.

	Speedup					
Processors	**I**	**II**	**III**	**IV**	**V**	**VI**
1	1.00	1.00	1.00	1.00	1.00	1.00
2	1.67	1.89	1.89	1.96	1.74	1.94
3	2.14	2.63	2.68	2.88	2.30	2.82
4	2.50	3.23	3.39	3.67	2.74	3.65
5	2.78	3.68	4.03	4.46	3.09	4.42
6	3.00	4.00	4.62	5.22	3.38	5.15
7	3.18	4.22	5.15	5.93	3.62	5.84
8	3.33	4.33	5.63	6.25	3.81	6.50

For each of these programs, choose the statement that best describes its likely performance on 16 processors:

A. The speedup achieved on 16 processors will probably be at least 40 percent higher than the speedup achieved on eight processors.

B. The speedup achieved on 16 processors will probably be less than 40 percent higher than the speedup achieved on eight processors, due to the large serial component of the computation.

C. The speedup achieved on 16 processors will probably be less than 40 percent higher than the speedup achieved on eight processors, due to the increase in overhead as processors are added

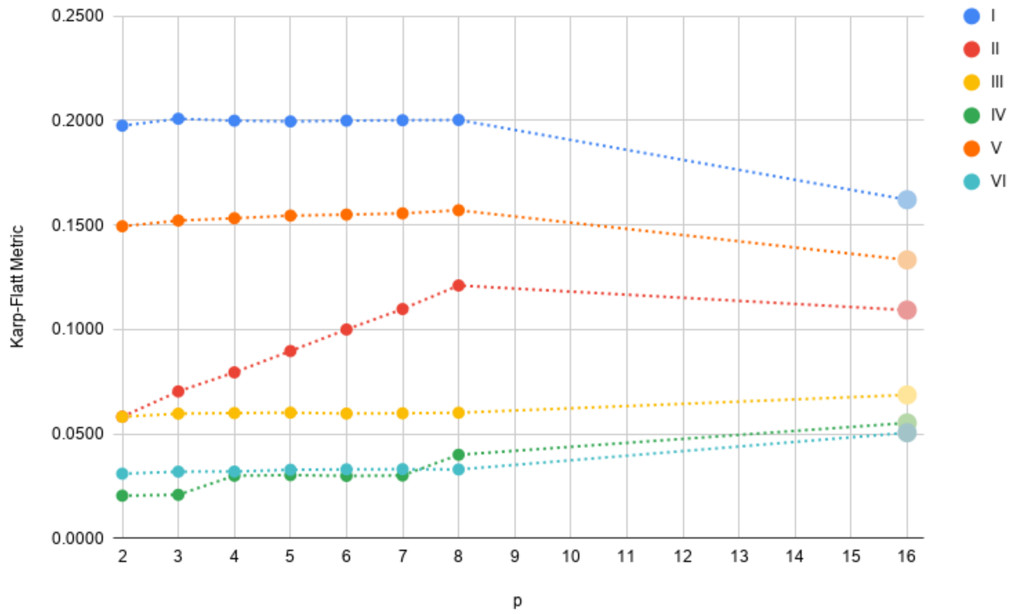

Figure 1: Karp-Flatt Metric of each program
(p=16 is calculated with 140% of speedup at p=8)

Program I: (B), The experimentally defined serial fraction remains constant at about 0.2. The speedup at $p = 16$ when the fraction is 0.2 is 4 ($\frac{1}{0.2 \cdot (1-1/16)+1/16}$), which is less than 140% of speedup at $p = 8$ (140% $\cdot$ 3.33 = 4.66). Because the fraction remains constant, the inability to reach the speedup is due to large serial component.

Program II: (C), The experimentally defined serial fraction is linearly increasing. If it keeps growing with this rate, the the fraction would be 0.2 at $p = 16$, which corresponds to a speedup of 4 (less than 140% of speedup at $p = 8$). Because the fraction is linearly growing, the inability to reach the speedup is due to the parallel overhead.

Program III: (A), The experimentally defined serial fraction remains constant at about 0.06. The speedup at $p = 16$ when the fraction is 0.6 is 8.42 ($\frac{1}{0.06 \cdot (1-1/16)+1/16}$), which is greater than 140% of speedup at $p = 8$ (140% $\cdot$ 5.63 = 7.88).

Program IV: (A), The experimentally defined serial fraction grows slowly with p, and it would reach about 0.05 at $p = 16$. The corresponding speedup would be 9.14 ($\frac{1}{0.05 \cdot (1-1/16)+1/16}$), which is greater than 140% of speedup at $p = 8$ (140% $\cdot$ 6.25 = 8.75).

Program V: (B), The experimentally defined serial fraction remains constant at about 0.154. The speedup at $p = 16$ when the fraction is 0.154 is 4.83 ($\frac{1}{0.154 \cdot (1-1/16)+1/16}$), which is less than 140% of speedup at $p = 8$ (140% $\cdot$ 3.81 = 5.33).

Program VI: (A), The experimentally defined serial fraction remains constant at about 0.0324. The speedup at $p = 16$ when the fraction is 0.0324 is 10.76 ($\frac{1}{0.0324 \cdot (1-1/16)+1/16}$), which is greater than 140% of speedup at $p = 8$ (140% $\cdot$ 6.50 = 9.10).

Exercise 11

Both Amdahl's Law and Gustafson-Barsis's Law are derived from the same general speedup formula. However, when increasing the number of processors p, the maximum speedup predicted by Amdahl's Law converges on $1/f$, while the speedup predicted by Gustafson-Barsis's Law increases without bound. Explain why this is so.

The problem size is fixed in Amdahl's Law, but not in Gustafson-Barsis's Law. So when p increases, the speedup is capped at $\frac{1}{f}$ in Amdahl's Law.
In Gustafson-Barsis's Law, $s = \frac{\sigma(n)}{\sigma(n)+\varphi(n)/p}$. If the problem size is fixed, s would approach 1 as p increases. $\psi \le p + (1-p)s = p + 1 - p = 1$, the speedup would get closer and closer to 1 (No speedup). But the problem size is not fixed in Gustafson-Barsis's Law, we can increase the problem size to maintain a fixed s, so the speedup can grow with p.

Exercise 12

Given a problem to be solved and access to all the processors you care to use, can you always solve the problem within a specified time limit? Explain your answer.

No. $\psi \le \frac{\sigma(n)+\varphi(n)}{\sigma(n)+\varphi(n)/p+\kappa(n,p)}$, because $\sigma(n) \ne 0$ and $\kappa(n,p) \ne 0$, the speedup is limited.
There are always some part in program that can not be parallelized (e.g. I/O operation), and there are some parallel overhead (which increases with the number of processors used). So the speedup cannot increase without bound, we can't always solve the problem within a specified time limit.

Exercise 13

Let $n \ge f(p)$ 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. $f(p) = Cp$ 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 $M(n) = n$

f. $f(p) = p^C$ and $M(n) = n$. Assume $1 < C < 2$.

g. $f(p) = p^C$ and $M(n) = n$. Assume $C > 2$.

Scalability function = $M(f(p))/p$

system	a	b	c	d	e	f	g
scalability	C^2p	$C^2\log^2 p$	C^2	$C^2p\log^2 p$	C	p^{C-1}	p^{C-1}

(Most scalable) $e, c < b < f < a < d < g$ (Least scalable)