HW7

Wednesday, March 10, 2021 6:12 PM

1. $\left.\begin{matrix}\psi(n,p)\\ \text{speedup}\end{matrix}\right\} \leq \frac{T_s}{T_p(p)}$

$\left.\begin{matrix}\text{sequential time}\\ T_s\end{matrix}\right\} = s(n) + f(n)$ $\left.\begin{matrix}\text{parallel time}\\ T_p(p)\end{matrix}\right\} = s(n) + \frac{f(n)}{p} + \kappa(n,p)$

s(n) denotes the sequential time that cannot be parallelized
f(n) denotes the portion of the program that can be parallelized
$\kappa(n,p)$ denoted the communication cost and parallel overhead

Let $\kappa(n,p) = 0$
$s(n) = 0.85$ $f(n) = 0.15$
Then,

$$\begin{matrix}\psi(n,p)\\ \text{speedup}\end{matrix} \leq \frac{0.15+0.85}{0.15+\frac{0.85}{p}+0} = \frac{p}{0.15p+0.85} \longleftarrow (1)$$

2. $\left.\begin{matrix}\epsilon(n,p)\\ \text{efficiency}\end{matrix}\right\} \leq \frac{T_s}{p.\,T_p(p)} = \frac{\sigma(n)+\varphi(n)}{p\left[\sigma(n)+\frac{\sigma(n)}{p}+\kappa(n,p)\right]}$

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

$$\epsilon(n,p) \leq \frac{1}{0.15p+0.85} \longleftarrow (2)$$

P	2	4	8	16	Inf
Speed up	1.7391	2.7586	3.9024	4.9231	6.6667
Efficiency	0.8697	0.6896	0.4878	0.3080	0.0000

1) 2)

$f = 0.15$
$s = 0.85$

3) $s = 0.15$
4) $\psi(n,p) \leq p + (1-p)s$ $\epsilon(n,p) \leq \frac{p+(1-p)s}{p}$

P	2	4	8	16
Speed up	1.85	3.55	6.95	13.75
Efficiency	0.925	0.8875	0.8686	0.8594

5.)
$$\epsilon(n,p) = \frac{T_S(n)}{p \cdot T_p(n)} = \frac{s+f}{p\left[s+\frac{f}{p}+k(n,p)\right]}$$

$$0.8 = \frac{1}{p(s)+(1-s)}$$

$$800s + 0.8(1-s) = 1$$
$$799.2 s = 0.2$$
$$s = \frac{0.2}{799.2} = 0.00025$$

Here
$s+f=1$
$p=1000$

$$f = 0.99975$$

$$S = \frac{25}{100000} = \frac{1}{4000}$$

$$f = \frac{3999}{4000}$$

6) Program 1

Speedup has doubled everytime the # of processes is doubled and it seems it is not bounded by the serial portion of the initial program.
This is in accordance with Gustafson-Barsis
Scaling is good because the serial portion of a parallel execution is constant.

Program 2

Speedup is being flattened towards the end of the table.
It seems like close to 1/6 th of the program is serial
Therefore the maximum speedup that can be achieved is limited by the serial portion of the program, according to the Amdhal's law.