Homework 5 - Solutions

April 14 , 2023

1. A parallel program has 1% serial component (i.e., one percent of the work is serial). The program is executed on 1000 cores. What is the maximum speedup? What is the maximum efficiency?

   Solution: Given that the one percent of the work is serial.

   We know that the Serial component $f = \frac{T_{ser}}{W}$ determines the fraction of the work that is serial.

   Therefore, in our case, $f = 0.01$

   Also, in every computation, there is part of computation which must be done serially and a part that can be speedup into p processors.

$$\therefore W = T_{ser} + T_{par}$$
$$T_p = T_{ser} + \frac{T_{par}}{p}$$
$$T_p = T_{ser} + \frac{W - T_{ser}}{T_{par}}$$

   Substituting f we get,

$$T_p = f * W + \frac{W - f * W}{p}$$
$$\frac{T_p}{W} = f + \frac{1-f}{p}$$
$$\frac{1}{S} = f + \frac{1-f}{p}$$
$$\frac{1}{S} = 0.01 + \frac{1-0.01}{1000}$$
$$\frac{1}{S} = 0.01 + 0.99 * 10^{-3}$$
$$\therefore \text{Maximum Speedup } S = 90.99$$

   $\therefore$ Maximum Efficiency $E = \frac{S}{p} = \frac{90.99}{1000} = 0.09$

2. The serial runtime of a program is $t_c.n$ (where tc is the time for a basic operation). The parallel runtime for the same program is $\frac{t_c.n}{p} + \log p$. If the program achieves efficiency of 0.75 on 16 cores for n = 10M , what will its efficiency be for the same n at 1024 cores?

   Solution: Given: Serial Runtime $T_s = t_c * n$

   Parallel Runtime $T_p = \frac{t_c.n}{p} + \log p$

   Program has efficiency = 0.75, p=16, n = 10^7

$$\begin{aligned}
\text{Efficiency} E &= \frac{S}{p} = \frac{T_s}{p * T_p} \\
E &= \frac{t_c * n}{p(\frac{t_c * n}{p} + \log p)} \\
E &= \frac{t_c * n}{t_c * n + p \log p} \\
t_c &= \frac{E * p \log p}{n * (1 - E)} \\
t_c &= \frac{0.75 \times 16 \times \log 16}{10^7 \times 0.25} \\
\therefore t_c &= 192 \times 10^{-7}
\end{aligned}$$

   Now, finding the efficiency for the same n at 1024 cores

$$\begin{aligned}
E &= \frac{t_c * n}{t_c * n + p \log p} \\
E &= \frac{192 \times 10^{-7} \times 10^7}{192 \times 10^{-7} \times 10^7 + 1024 \times \log 1024} \\
E &= \frac{1}{54} \\
E &= 0.018
\end{aligned}$$

   Therefore, the Efficiency for same n at 1024 cores is 0.018

3. The serial runtime of a program is $t_c.n$ (where tc is the time for a basic operation). The parallel runtime for the same program is $\frac{t_c.n}{p} + \log p$. f the program achieves efficiency of 0.75 on 16 cores for n = 10M , what should n be to achieve the same efficiency on 1024 cores?

Solution: Given: Serial Runtime $T_s = t_c * n$

Parallel Runtime $T_p = \frac{t_c.n}{p} + \log p$

Program has efficiency = 0.75, p=16, n = 10^7

$$\begin{aligned}
\text{Efficiency} E &= \frac{S}{p} = \frac{T_s}{p * T_p} \\
E &= \frac{t_c * n}{p(\frac{t_c * n}{p} + \log p)} \\
E &= \frac{t_c * n}{t_c * n + p \log p} \\
t_c &= \frac{E * p \log p}{n * (1 - E)} \\
t_c &= \frac{0.75 \times 16 \times \log 16}{10^7 \times 0.25} \\
\therefore t_c &= 192 \times 10^{-7}
\end{aligned}$$

Now, finding n to achieve the same efficiency on 1024 cores

$$\begin{aligned}
E &= \frac{t_c * n}{t_c * n + p \log p} \\
n &= \frac{E * p \log p}{t_c * (1 - E)} \\
n &= \frac{0.75 \times 1024 \times \log 1024}{192 \times 10^{-7} \times (1 - 0.75)} \\
n &= 160 \times 10^7
\end{aligned}$$

Therefore, to achieve the same efficiency on 1024 cores, the n has to be 1600M

4. The serial runtime of a program is n^2. The parallel runtime of the program is $\frac{n^2}{p} + np\log p$. What is the cost ($p * T_p$ product) of the parallel system? Is the system cost optimal? What is the largest number of processors for which the system is cost optimal?

Solution: Given: Serial Runtime $T_s = n^2$

Parallel Runtime $T_p = \frac{n^2}{p} + np\log p$

Cost of the parallel system is:

$$p * T_p = p * (\frac{n^2}{p} + np\log p)$$
$$p * T_p = n^2 + np^2\log p$$

The system would be cost optimal if $p * T_p \sim O(T_s)$

$n^2 + np^2\log p \sim O(n^2)$ The system will not be cost optimal if $p \sim O(n)$, or any asymptotically higher function of n, because $n^2 + np^2\log p \sim O(n^3\log n)$, which is asymptotically higher than $O(n^2)$

Therefore for system to be cost optimal:

$$np^2\log p \sim n^2$$
$$p^2\log p \sim n$$
$$p^2 \sim \frac{n}{\log p}$$
$$p^2 \sim \frac{n}{\log n}$$
$$\therefore p \sim \sqrt{\frac{n}{\log n}}$$

Therefore, the largest number of processors for which the system is cost optimal is $p \sim O(\sqrt{\frac{n}{\log n}})$

5. The serial runtime of a program is n^2. The parallel runtime of the program is $\frac{n^2}{p} + np\log p$. What is the isoefficiency of the system?

Solution: Given: Serial Runtime $T_s = n^2$

Parallel Runtime $T_p = \frac{n^2}{p} + np\log p$

For isoefficiency, $W \sim T_o$, that us as long as useful work is same as total overhead, we will get constant efficiency.

$$T_o = p * T_p - T_s$$
$$T_o = p * (\frac{n^2}{p} + np\log p) - n^2$$
$$T_o = n^2 + np^2\log p - n^2$$
$$\therefore T_o = np^2\log p$$

$$\text{For Isoefficiency, } n^2 \sim W \sim T_o \sim np^2\log p$$
$$n \sim p^2\log p$$
$$\therefore n^2 \sim (p^2\log p)^2$$
$$\therefore n^2 \sim p^4\log^2 p$$

Therefore, the isoefficiency is $p^4\log^2 p$

6. The total overhead associated with a serial program with work W is $\sqrt{W} * p * \log p$.

   What is the isoefficiency of the system?

   Solution: Given: The total overhead associated with serial program with work W is $\sqrt{W} * p * \log p$

   For isoefficiency, $W \sim T_o$, that us as long as useful work is same as total overhead, we will get constant efficiency.

$$
\begin{aligned}
W \sim T_o \sim \sqrt{W} * p * \log p \\
W \sim \sqrt{W} * p * \log p \\
\sqrt{W} \sim p * \log p \\
\therefore W \sim p^2 \log^2 p
\end{aligned}
$$

   Therefore, the isoefficiency is $p^2 \log^2 p$

7. The serial runtime of a program is n^2. The parallel runtime of the program is $\frac{n^2}{p} + np \log p + p^2$. What is the isoefficiency of the system?

Solution: Given: Serial Runtime $T_s = n^2$

Parallel Runtime $T_p = \frac{n^2}{p} + np \log p + p^2$

For isoefficiency, $W \sim T_o$, that us as long as useful work is same as total overhead, we will get constant efficiency.

$$T_o = p * T_p - T_s$$
$$T_o = p * (\frac{n^2}{p} + np \log p + p^2) - n^2$$
$$T_o = n^2 + np^2 \log p + p^3 - n^2$$
$$T_o = np^2 \log p + p^3$$

$$\text{For Isoefficiency, } n^2 \sim W \sim T_o \sim np^2 \log p$$

or

$$n^2 \sim W \sim T_o \sim p^3$$

Solving both the above expressions we get isoefficiency as,

$$n^2 \sim np^2 \log p$$
$$n \sim p^2 \log p$$
$$\therefore n^2 \sim p^4 \log^2 p$$
$$\text{Using 2nd term, we get } n^2 \sim p^3$$

For the isoefficiency, we pick the highest of the values, that is, $n^2 \sim p^4 \log^2 p$, which means that increase the problem size as $p^4 \log^2 p$ to make sure that overhead time T_o does not take the efficiency down, that is, to keep the efficiency constant.

8. The memory complexity of the program in the previous question is n^2. Assume that memory scales linearly in the number of cores. What is the memory constrained speedup of the program in the previous question?

Solution: Given: Serial Runtime $T_s = n^2$

Parallel Runtime $T_p = \frac{n^2}{p} + np\log p + p^2$

Memory complexity of the program is n^2

memory scales linearly in the number of cores, therefore $n^2 \sim p$

Speedup is:

$$S = \frac{T_s}{T_p}$$
$$S = \frac{n^2}{\frac{n^2}{p} + np\log p + p^2}$$

Substitute $n^2 = p$ we get,

$$S = \frac{p}{\frac{p}{p} + p\sqrt{p}\log p + p^2}$$
$$S = \frac{p}{1 + p^{1.5}\log p + p^2}$$
$$S \sim \frac{p}{p^2}$$
$$S \sim p^{-1}$$

Therefore, the memory constrained speedup of the program is $S \sim p^{-1}$

9. What is the time constrained speedup for the same problem?

Solution: Given: Serial Runtime $T_s = n^2$

Parallel Runtime $T_p = \frac{n^2}{p} + np \log p + p^2$

For Time Contraint, $T_p = O(1)$ Therefore, $T_p = \frac{\text{Asymptotic no of operations}}{\text{no of processors}} \sim \frac{n^2}{p} \sim O(1)$

Therefore, $n^2 = p$ Speedup is:

$$S = \frac{T_s}{T_p}$$
$$S = \frac{n^2}{\frac{n^2}{p} + np \log p + p^2}$$

Substitute $n^2 = p$ we get,

$$S = \frac{p}{\frac{p}{p} + p\sqrt{p} \log p + p^2}$$
$$S = \frac{p}{1 + p^{1.5} \log p + p^2}$$
$$S \sim \frac{p}{p^2}$$
$$S \sim p^{-1}$$

Therefore, the time constrained speedup of the program is $S \sim p^{-1}$

10. What is the efficiency under weak scaling for the same problem?

Solution: Given: Serial Runtime $T_s = n^2$

Parallel Runtime $T_p = \frac{n^2}{p} + np\log p + p^2$

In weak scaling, we increase the problem size as $W \sim O(p)$

Therefore, $n^2 \sim p$

Efficiency is:

$$E = \frac{S}{p} = \frac{T_s}{p * T_p}$$
$$E = \frac{n^2}{p * (\frac{n^2}{p} + np\log p + p^2)}$$
$$E = \frac{n^2}{n^2 + np^2\log p + p^3}$$

Substitute $n^2 = p$

$$E = \frac{p}{p * (\frac{p}{p} + p^2\sqrt{p}\log p + p^3)}$$
$$E = \frac{1}{1 + p^{1.5}\log p + p^2}$$
$$E \sim \frac{1}{p^2}$$
$$E \sim p^{-2}$$

Efficiency under weak scaling is $O(p^{-2})$