Homework 1

Christopher Canaday

September 21, 2024

Problem 1: Let the strong speedup of a problem of size n that is computationally solved using P processors by a parallel algorithm in time $T(n, P)$ be defined by:

$$S(p) = \frac{T(n,1)}{T(n,P)} \tag{1}$$

and the efficiency of the same parallel algorithm be defined by

$$E(p) = \frac{S(p)}{P} \tag{2}$$

Suppose a parallel algorithm has a parallel runtime complexity of:

$$T(n,p) = \Theta(\frac{n^2}{P} + \sqrt{n}) \qquad \text{for } P \leq n^2 \tag{3}$$

i. What is the execution time of this parallel algorithm on a single processor (we assume a single process is mapped to a single processor)?

   (a) The execution time of the parallel algorithm with a single processor would be:
   $$T(n,1) = \Theta(n^2 + \sqrt{n}) \tag{4}$$

ii. In terms of n, at what processor count, $P_{\max}$, will the algorithm achieve maximum parallel speedup?

   (a) The algorithm will achieve maximum parallel speedup when $P_{\max} = n^2$. The runtime complexity would be:

$$\begin{aligned} T(n, P_{\max}) &= \Theta(\frac{n^2}{P_{\max}} + \sqrt{n}) \qquad \text{for } P_{\max} = n^2 \\ &= \Theta(\frac{n^2}{n^2} + \sqrt{n}) \\ &= \Theta(1 + \sqrt{n}) \\ &= \Theta(\sqrt{n}) \end{aligned}$$

The speedup $S(P_{\max})$ would be:

$$\begin{aligned}
S(P_{\max}) &= \frac{T(n,1)}{T(n,P_{\max})} \\
&= \frac{\Theta(n^2+\sqrt{n})}{\Theta(\sqrt{n})} \\
&= \Theta(\frac{n^2+\sqrt{n}}{\sqrt{n}}) \\
&= \Theta(\frac{n^2+\sqrt{n}}{\sqrt{n}} * \frac{\sqrt{n}}{\sqrt{n}}) \\
&= \Theta(\frac{n^2\sqrt{n}+n}{n}) \\
&= \Theta(n\sqrt{n}+1) \\
&= \Theta(n\sqrt{n})
\end{aligned}$$

iii. What is the parallel efficiency achieved when $P_{\max}$ processors are used?

(a) The parallel efficiency is:

$$\begin{aligned}
E(P_{\max}) &= \frac{S(P_{\max})}{P_{\max}} \\
&= \frac{n\sqrt{n}}{n^2} \\
&= \frac{\sqrt{n}}{n}
\end{aligned}$$

iv. Maximum parallel efficiency is achieved when $E(P) \approx 1$. At what processor count will this algorithm achieve its maximum parallel efficiency?

(a) The maximum efficiency would be with 1 processor.

$$\begin{aligned}
E(P) &= \frac{S(P)}{P} \qquad \text{for } S(P) = \frac{T(n,1)}{T(n,P)} \text{ and } E(P) \approx 1 \\
1 &\approx \frac{\frac{T(n,1)}{T(n,P)}}{P} \\
1 &\approx \frac{\frac{n^2+\sqrt{n}}{\frac{n^2}{p}+\sqrt{n}}}{P} \\
1 &\approx \frac{n^2+\sqrt{n}}{\frac{n^2}{P}+\sqrt{n}} * \frac{1}{P} \\
1 &\approx \frac{n^2+\sqrt{n}}{n^2+P\sqrt{n}} \\
n^2+P\sqrt{n} &\approx n^2+\sqrt{n} \\
P\sqrt{n} &\approx \sqrt{n} \\
P &\approx 1
\end{aligned}$$

Problem 2: The following is a polynomial of a single unknown variable x and degree n−1 where a_0, a_1, $\cdots$, a_{n-1} are known constants.

$$P(x) = a_0 + a_1 x + a_2 x^2 + a_3 x^3 + \cdots + a_{n-1} x_{n-1} \tag{5}$$

The goal here is to compute the value of the polynomial at a given value of x, say, x_0, that is, evaluate $P(x_0)$. Describe step-by-step a parallel algorithm to compute $P(x_0)$. Assume that $n = k_1 p$ and $p = 2^{k_2}$ where k_1 and k_2 are both positive integers greater than 1. What are the parallel computation and communication costs of your algorithm.

i. The problem is perfect for Horner's Method. The algorithm steps are:

   (a) Partition a_i in parts and assign them to p processes.
   (b) Broadcast the value of x_0.
   (c) Do parallel prefix with the multiplication operation.
   (d) Locally multiply the partial products with the right coefficient.
   (e) Locally add all the products.
   (f) Do a reduce to add all the local products together.

ii. The computation cost is:

$$T(n,p)_{comp} = \Theta(C_{op} * (\frac{n}{p} + \log_2 p)) \tag{6}$$

Where C_{op} is the cost of performing the multiplication operation. There are $\frac{n}{p}$ computations in each process and during the parallel prefix you will additionally multiply $\log_2 p$ times. There are also $\frac{n}{p}$ additions in each process, but this gets consumed by the Θ.

iii. The communication cost is:

$$T(n,p)_{comm} = C_{reduce} + C_{parallelPrefix} + C_{broadcast} + C_{personalizedSend} \tag{7}$$

Considering that p is guaranteed to be a power of two I am going to assume that the network is in a HyperCube configuration. With a HyperCube Configuration the cost of a reduce would be:

$$C_{reduce}^{HyperCube} = \Theta((\tau + \mu m_0) * \log_2 p) \tag{8}$$

Where m_0 is the message size. In this case the message size would be one number from each process. The cost for the parrallel prefix would be:

$$C_{parallelPrefix}^{HyperCube} = \Theta((\tau + \mu m_0) * \log_2 p) \tag{9}$$

Where m_0 is a number being sent to the process to the right. The $\log_2 p$ comes from the number of times this occurs during the parallel prefix. The cost for the broadcast of x is:

$$C_{broadcast}^{HyperCube} = \Theta((\tau + \mu m_0) * \log_2 p) \tag{10}$$

This operation is the same as reduce, but in reverse so it has the same cost. To send the a's we have to do a personalized send to give processes the correct value. The cost for a personalized send is:

$$C^{HyperCube}_{personalizedSend} = \Theta(\tau \log_2 p + \mu m_1 p) \tag{11}$$

Here m_1 is the size of $\frac{n}{p}$ numbers. In other words.

$$m_1 = \frac{n}{p} * m_0 \tag{12}$$

Thus,

$$C^{HyperCube}_{personalizedSend} = \Theta(\tau \log_2 p + \mu m_0 n) \tag{13}$$

So the communication cost assuming a HyperCube network configuration is:

$$\begin{aligned} T(n,p)^{HyperCube}_{comm} &= \Theta((\tau + \mu m_0) * \log_2 p) \\ &\quad + \Theta((\tau + \mu m_0) * \log_2 p) \\ &\quad + \Theta((\tau + \mu m_0) * \log_2 p) \\ &\quad + \Theta(\tau \log_2 p + \mu m_0 n) \end{aligned}$$

Which simplified is:

$$\begin{aligned} T(n,p)^{HyperCube}_{comm} &= \Theta((\tau + \mu m_0) * \log_2 p + \tau \log_2 p + \mu m_0 n) \\ &= \Theta(\tau \log_2 p + \mu m_0 \log_2 p + \mu m_0 n) \\ &= \Theta(\tau \log_2 p + \mu m_0 (\log_2 p + n)) \end{aligned}$$

So, the total cost is:

$$T(n,p)^{HyperCube}_{total} = \Theta(C_{op} * (\frac{n}{p} + \log_2 p) + \tau \log_2 p + \mu m_0 (\log_2 p + n)) \tag{14}$$