

CSCI 251: Concepts of Parallel and Distributed Systems

Alvin Lin

September 6th, 2017

Topics

- Speedup, Efficiency, Scalability
- Matrix-Vector Operations
- Matrix-Matrix Operations
- Solving Systems of Linear Equations

Speedup, Efficiency, Scalability

These equations model parallel addition:

$$T_p = \frac{N}{P} + 2 \log P$$

$2 \log P$ represents the cost of communication.

$$\text{Speedup} = \frac{N}{\frac{N}{P} + 2 \log_2 P}$$

$$\text{Efficiency} = \frac{\text{Speedup}}{P} = \frac{1}{1 + \frac{2P \log_2 P}{N}}$$

Examples

Suppose we are adding 64 data units with 4 processors:

$$E = \frac{1}{1 + \frac{(2)(4)(2)}{64}} = \frac{1}{1.25} = 0.8$$

If we keep the data size constant but 16 processors:

$$E = \frac{1}{1 + \frac{(2)(16)(4)}{64}} = \frac{1}{3} = 0.333$$

Efficiency does not increase proportionally to the number of processors if the data size remains the same. Suppose $P = 4$ and $N = 128$:

$$E = \frac{1}{1 + \frac{(2)(4)(2)}{128}} = \frac{1}{1.125} = 0.8888$$

To find the optimal number of processors for the optimal T_p :

$$\frac{dT_p}{dP} = 0$$

$$P = \frac{N}{2}$$

Matrix-Vector Operations

Let A be a matrix of size $N \times N$, x is a vector of size N , and y be the resultant vector of size N .

$$Ax = y$$

$$\begin{bmatrix} a_{11} & a_{12} & a_{13} & \dots & a_{1N} \\ a_{21} & a_{22} & a_{23} & \dots & a_{2N} \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ a_{N1} & a_{N2} & a_{N3} & \dots & a_{NN} \end{bmatrix} \times \begin{bmatrix} x_1 \\ x_1 \\ \vdots \\ x_N \end{bmatrix} = \begin{bmatrix} y_1 \\ y_1 \\ \vdots \\ y_N \end{bmatrix}$$

$$y_i = \sum_{j=1}^N (a_{ij})(x_j)$$

$$\begin{aligned}
a_{11}x_1 + a_{12}x_2 + a_{13}x_3 + \cdots + a_{1N}x_N &= y_1 && \Theta(N) \text{ multiplication} \\
\dots &= y_2 \\
\dots &= y_2 \\
&\vdots \\
\dots &= y_N
\end{aligned}$$

This entire process is $\Theta(N^2)$. To compute each y_i , we have $\Theta(N)$ multiplications and $N - 1$ additions, but we will not be taking into account the $N - 1$ additions because they are outshadowed by the computation time of the multiplications (which are simply repeated additions). To compute the vector y , we need $\Theta(N^2)$ multiplications.

Example

If we consider the case where $N = P$. We assign one row of A to one processor.

$$Row_i = P_i$$

Each P_i will compute their corresponding y_i . All P_i 's will work in parallel here, making $T_p = \Theta(N)$ and speedup = $\frac{\Theta(N^2)}{\Theta(N)} = \Theta(N)$ This does not take into account the cost of communication. For this example, each processor needs to have a copy of vector x . It needs to be broadcast to all other processes.

Example

Suppose however, that N is greater than P . Suppose we have 16 rows and 4 processors, P_1 will have $\frac{N}{P}$ vectors to compute, making this process $\Theta(N\frac{N}{P})$.

Matrix-Matrix Operations

$$[A] \times [B] = [C]$$

Let A and B be matrices of size $N \times N$

$$C_{ij} = \sum_{i,j=0}^N a_{ij}b_{ji}$$

This is a computation that is $\Theta(N^3)$. The matrix B must be broadcast to all processors.

Reminders

Professor Mohan Kumar:
mjkvcs@rit.edu
<https://cs.rit.edu/~mjk>

Rahul Dashora (TA):
rd5476@mail.rit.edu

You can find all my notes at <http://omgimanerd.tech/notes>. If you have any questions, comments, or concerns, please contact me at alvin@omgimanerd.tech