CSCI 251: Concepts of Parallel and Distributed Systems

Alvin Lin

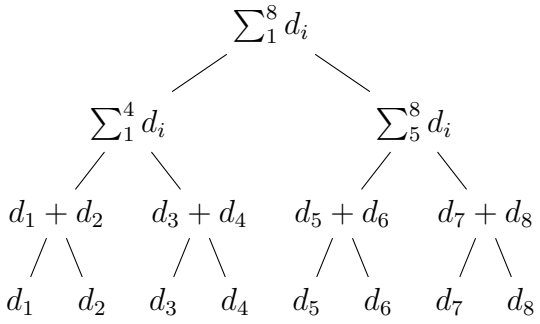
August 30th, 2017

Topics

- Parallel Addition
- Speed up
- Amdahl's Law
- Efficiency
- Scalability
- Superlinear speedup
- Merge Sort

Parallel Addition

Suppose we have N numbers to add. A serial computation would require (N-1) additions, making it an $O(N)$ operation. If we were to add N numbers 2 at a time in a tree structure like manner, then it would only take $\log_2 N$ steps.



Speed Up

The speed up is the time taken to execute a given problem on a single computer over the time take to execute the given problem on a parallel computer. We make several assumptions about the parallel computer:

- each processor runs on the same clock speed
- the bandwidth is the same for all communications
- the memory access times are the same

Message Passing Model

Each of the steps in the tree above requires a communication step as well as the addition operation.

$d_1 \leftarrow$	d_2	$d_3 \leftarrow$	d_4	$d_5 \leftarrow$	d_6	$d_7 \leftarrow$	d_8
P_1	P_2	P_3	P_4	P_5	P_6	P_7	P_8

$d_1 + d_2 \leftarrow$ $d_3 + d_4$ $d_5 + d_6 \leftarrow$ $d_7 + d_8$

$\ldots$

Even numbered processors will pass a message to P_{i-1}.

Shared Memory Model

Alternatively, there can exist a region of shared memory which all processors P_i can access. During the communication step, each processor P_i writes d_i into shared memory. The flow will be as follows:

- memory write
- memory read
- add
- memory write

There needs to be a mechanism (mutex lock/semaphore) to make the shared memory operations atomic.

Modeling the speed up

The speed up is limited by the portion of the program that can be parallelized. Suppose in a given problem, α is the portion that cannot be parallelized. $(1 - \alpha)$ is the portion that can be parallelized. Given that the time taken to execute the algorithm on a single computer is $T_s = 1$, the time taken to execute the same algorithm on a parallel computer with P processors is given by:

$$T_p = \frac{1 - \alpha}{P} + \alpha$$

The speed up S is given by:

$$\begin{aligned} S &= \frac{T_s}{T_p} \\ &= \frac{1}{\frac{1-\alpha}{P} + \alpha} \end{aligned}$$

Amdahl's Law

This speed is bounded by $\frac{1}{\alpha}$, thus you cannot get a speed up greater than $\frac{1}{\alpha}$. Suppose $\alpha = 0.2$, meaning 20% of the program cannot be parallelized. With 2 processors:

$$S = \frac{1}{\frac{0.8}{2} + 0.2} = \frac{1}{0.6} = \frac{5}{3}$$

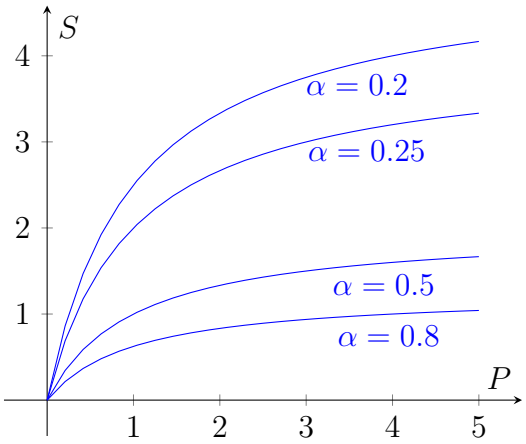
With $P = 4$:

$$S = \frac{1}{\frac{0.8}{4} + 0.2} = \frac{1}{0.4} = 2.5$$

With $P = 8$:

$$S = \frac{1}{0.3} = 3.33$$

Graphing this, we can see the limitations of the speed up in that it is upper bounded by $\frac{1}{\alpha}$:



Efficiency

The efficiency E is equal to the speed up S over the number of processors P:

$$\begin{aligned} E &= \frac{S}{P} \\ &= \left[\frac{1}{\frac{1-\alpha}{P} + \alpha}\right]\frac{1}{P} \end{aligned}$$

Scalability

In the initial addition problem, N was equal to P. In real world situations, N is often far greater than P. For $N = 64$ and $P = 4$, each processor should get $\frac{N}{P}$ data

elements.

$$P_1 \to \sum_{1}^{16} d_i$$
$$P_2 \to \sum_{17}^{32} d_i$$
$$P_3 \to \sum_{33}^{48} d_i$$
$$P_4 \to \sum_{49}^{64} d_i$$

$\frac{N}{P} - 1$ additions on each processor is $O(\frac{N}{P})$ with $\log_2 P$ steps, or $O(\frac{N}{P} + \log(P))$.

Superlinear speedup

Depth-First Search Trees, search time can be halved by dividing the task between two computers.

Reminders

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

Jennifer Burt (Additional Contact):
`jennifer@cs.rit.edu`

Homework

Check MyCourses for sample questions. There will be a small quiz in class next Wednesday.

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