

Lecture 6: February 11

*Lecturer: Horst D. Simon**Scribes: Bor-Yiing Su*

Disclaimer: *These notes have not been subjected to the usual scrutiny reserved for formal publications. They may be distributed outside this class only with the permission of the Instructor.*

Announcement:

1. 02/18: President's day. No Class.
2. 02/20: Discussion Session.
3. 02/22: Lecture No. 8.

6.1 Hybrid Systems

Due to the limitations of shared memory systems, thread-based techniques are unlikely to be the choice of running parallel programs. However, they can be used to optimize the processes in each machine. It is possible that the hybrid systems will be the dominate machines for parallel programs. For hybrid systems, we use thread-based techniques to control processes within each machine, while using message passing techniques on top of the machines to deal with the communication among them.

6.2 Properties of Distributed Memory Architectures

6.2.1 Diameter and Latency

Definition 6.1 *Let V be the set of all nodes in the network. Let v be any one of the nodes in the network. Let $d(v_1, v_2)$ be the length of the shortest path from node v_1 to node v_2 . Diameter is defined as: $\max_{v_1 \in V, v_2 \in V} d(v_1, v_2)$.*

Definition 6.2 *Latency is the delay between sending and receiving times.*

- Latency is limited by hardware and routing.
- Hardware latency might not reflect the real latency, it is the software latency that dominates the overall latency.
- We can use MPI ping-pong test to estimate the latency among processors.
The MPI ping-pong test is that the first processor sends a message to the second processor. After the second processor receives the message, he replies to the first processor. The latency is measured by half the time that the first processor sends out the message minus the time that it receives the response. It is also called the *1-way latency*.
- There is no significant improvements on latency over time.

6.2.2 Bandwidth

Definition 6.3 Bandwidth is defined as the number of wires divided by the time spends by each wire to send a bit.

- Bandwidth is important for applications that use large messages.
- The effective bandwidth is usually lower than the physical link bandwidth.
When sending out a message, it is usually padded by some headers in the front, some tailors at the end, and some error correction code to verify whether the data is corrupted. These additional information consumes some of the physical link bandwidth, thus the effective bandwidth is reduced.
- The bandwidth of a large message is better than that of a small message.

Definition 6.4 Let C be the set of cuts that divide the network into two balanced parts. Let $e(c)$ be the number of edges crossed by the cut c . Bisection Bandwidth is defined as the bandwidth of the cut \hat{c} such that $e(\hat{c}) \leq e(c) \forall c \in C$.

- Bisection Bandwidth is important for applications that each processor will communicate to all other processors. For example, the FFT algorithm needs very efficient bisection bandwidth to achieve better performance.

6.3 Network Topology

Let n be the number of nodes in the network.

- Linear Array
 - Diameter = $n - 1$.
 - Bisection Bandwidth = 1.
- Torus (Ring)
 - Diameter = $n/2$.
 - Average Distance = $n/4$.
 - Bisection Bandwidth = 2.

The torus topology just add one more connection in the linear array topology, but its diameter is shortened and its bandwidth is increased. Moreover, any algorithm that works on linear array will still works on the torus topology.

- 2 – D Mesh
 - Diameter = $2(\sqrt{n} - 1)$.
 - Bisection Bandwidth = \sqrt{n} .
- 2 – D Torus
 - Diameter = \sqrt{n} .

- Bisection Bandwidth = $2\sqrt{n}$.

Higher dimension meshes can be changed to higher dimension torus by the similar concepts. In theory, higher dimension torus will have shorter diameter and larger bandwidth. However, higher dimension torus is difficult to layout. Thus torus with dimension larger than 3 is seldom used.

- Hypercubes For a hypercube with dimension d , the number of nodes inside is $n = 2^d$
 - Diameter = d .
 - Bisection Bandwidth = $n/2$.

The greycode addressing for hypercubes is very useful. Each code differs from any neighbors by only one bit. Such property is very useful for routing algorithms.

- Trees
 - Diameter = $\log n$.
 - Bisection Bandwidth = 1.

The tree topology is very easy to layout. Thus it is the most commonly used topology currently. To increase the bisection bandwidth of a tree layout, some hierarchical fat wires are used. That is, for the edges in higher levels, fatter wires are used to handle larger data flows.

- Butterflies
 - Diameter = $\log n$.
 - Bisection Bandwidth = n .

This topology is natural for FFT algorithms. However, in addition to computation nodes, it needs some switch nodes. Moreover, it needs too many wires to establish the connection. Therefore, this topology is fading away.

6.4 Performance Models

6.4.1 $\alpha - \beta$ Model

The $\alpha - \beta$ model measures the time between sending and receiving a message of length n . The time is measured by:

$$Time = latency + \frac{n}{bandwidth} \quad (6.1)$$

- Often written $Time = \alpha + \beta n$.
- Usually $\alpha \gg \beta \gg$ time per flop.
- One long message has shorter time than many short messages. $\alpha + \beta n \ll n(\alpha + \beta)$.
- The time spent for communication is longer than that for computation. Thus the less communication the better.
- This model matches with the real message exchange time very well.

6.4.2 LogP Parameter

The LogP model measures the end-to-end latency, including the sending overhead from the sender, the transportation time by wires, and the receiving overhead from the receiver. We can overlap the transferring flow for better performance. That is, when the wire is transferring the first message, the sender is processing the second message. While the receiver is processing the first message, the wire is transferring the second message, and the sender is processing the third message, and so on. Let o_{send} be the sending overhead. Let L be the time for the message traveling from the sender to the receiver by wire. Let o_{recv} be the receiving overhead. Let gap be the time between each pair of messages that sent by the sender. Then the end-to-end latency is:

$$(o_{send} + L + o_{recv} - gap) + n \times gap = \alpha + n \times \beta \quad (6.2)$$

- The sending overhead is not improved significantly over time.

6.5 Message Passing Interface(MPI)

- There are a lot of message passing libraries. Among all of them, the MPI is currently the industry standard.
- All communication and synchronization requires subroutine calls. That is, the parallel programmer has to deal with synchronization issues by himself.
- MPI is a library, all operations are performed by routine calls.
- MPI starts with MPI_Init, and ends with MPI_Finalize.
- A group and context together form a communicator.
- A process is identified by its rank in the group associated with a communicator.