Parallel Programming Notes

Ayush Raina

January 30, 2025

Lecture 1

Classification of Architectures - Flynns Classification

- SISD: Single Instruction Single Data, for example, serial computers.
- SIMD: Single Instruction Multiple Data, for example, vector processors and processor arrays.
- MISD: Multiple Instruction Single Data, for example, trying different ways to decrypt a message.
- MIMD: Multiple Instruction Multiple Data, for example, multi-core processors.

Classification based on Memory

- **Shared Memory:** All processors share a common memory. Communication is done using this shared address space. This is further classified into UMA (Uniform Memory Access) and NUMA (Non-Uniform Memory Access).
  - UMA: All processors have equal access time to all memory locations. Memory access is slow and has limited bandwidth. UMA has single memory controller.
  - NUMA: Processors have variable access to memory locations. Memory access is faster and has higher bandwidth than UMA. NUMA has multiple memory controllers.
- **Distributed Memory:** Each processor has its own memory.

Shared memory could itself be distributed among processor nodes. Each processor might have some portion of the memory that is physically close to it and therefore accessible in less time.

Cache Coherence Problem

If each processor in a shared memory multiprocessor machine has a data cache, then the problem of cache coherence arises. Our objective is that processes should not read **stale** data.

Cache Coherence Protocols

- **Write Invalidate:** When a processor writes to its cache, it sends invalidate signal to all other caches that have a copy of that memory location. This means all other cache locations must discard their copy of the memory location and fetch it again from the main memory if needed.
- **Write Update:** When a processor writes to its cache, it sends the updated data to other caches that have a copy of that memory location. This keeps all caches up to date.

False Sharing

If two processors access different variables that are located within same cache line but are not related to each other. In this case any write to one variable will cause cache line to be invalidated and other processor might have to reload the data from main memory. This is called false sharing.

In next lecture we will discuss about interconnecting networks.

Lecture 2

Interconnection Networks

They are used in both shared memory to connect processors to memory and in distributed memory to connect different processors to each other. Components for interconenction networks are interfaces such as Peripheral Component Interconnect (PCI) or PCI - Express used for connecting processors to network and a network link which connected to communication network.

Communication Network

It consists of switching elements to which processors are connected through ports. **Switching elements** receive data from one point and send it to another point. **Switch** refers to collection of these switching elements. **Network topology** is specific pattern of connections in which these switching elements are connected.
In shared memory systems processors as well as memory units are connected to communication network.

Different Kinds of Network Topologies

1. **Bus:** All processors are connected to a single bus. It is simple and cheap but has limited bandwidth.
2. **Crossbar Switch:** It consists of 2D grid of switching elements, where each switching element consists of two input and two output ports. Input Ports are connected to output ports through a switching logic.

In previous case of Crossbar Switch, we require nm switching elements where n is number of processors and m is number of memory units in case of shared memory architecture or m is the number of processors in distributed memory architectures. To reduce switching complexity, we can use **Multistage Network - Omega Network**.

3. **Omega Network:** It consists of $logP$ stages each consisting of $P/2$ switching elements.

Consider Distributed Memory Architecture. In Crossbar switch, there is dedicated path for any processor to communicate with any other processor without contention but in Omega Network, there is contention if 2 processors wants to communicate to 2 different processors, they might have to take same path through some stage of the network.

If P_i and P_j wants to communicate in Omega Network, first convert $ID(P_i)$ and $ID(P_j)$ to binary and then keep comparing most significant bits and follow **cut through routing** or **pass through routing** to reach destination.

Some commonly used network topologies used in distributed memory architectures are Mesh, Torus, hypercubes and Fat tree.

1. **Mesh:** Grid of switching elements where each switching element is connected to 4 directional neighbours.
2. **Torus:** Mesh with wrap around connections. In this $T(i,1)$ is connected to $T(i,n) \forall i$. Similarly $T(1,j)$ is connected to $T(m,j) \forall j$.
3. **Hypercube:** Keep Joining binary n cubes to get hypercube. In hypercubes, distance between any two processors is bit difference between their binary representation.
4. **Fat Tree:** It is a tree like structure where each node is a switch and each switch has multiple ports. Leaves are processors. As we go up the tree, number of ports in switch increases. This is Non Blocking Network means no contention because there is unique path between any two processors. Fat Tree has a special property which makes all this happen and that is **Number of Links from node to a children = Number of Links from node to parent**.

bandwidth: maximum amount of data that can be transferred over the network in given time, usually measured in bits per second. (bps)

Evaluating Interconnection Topologies

1. **Diameter:** Maximum distance between any two processing nodes.Smaller diameter means lower latencies.
2. **Connectivity:** Number of Paths between two nodes. It can also be defined as minimum number of links that need to be removed to disconnect the network. Higher connectivity improves fault tolerance.
3. **Fault Tolerance:** Ability of network to operate correctly even if some links or switches fail.
4. **Bisection Width:** Minimum number of links that need to be removed to divide the network into two equal halves.
5. **Channel Width:** Number of bits that can be simultaneously communicated over a link i.e number of physical wires between two nodes. Higher Channel width increases data transfer capacity.
6. **Channel Rate:** Performance of single physical wire i.e The speed at which single physical wire transmits the data and is generally measured in bits per second(bps).
7. **Channel bandwidth:** The total data transfer capacity of a link. It is calculated as Channel Width * Channel Rate. Higher Channel bandwidth allows faster communication.
8. **Bisection bandwidth:** This is defined as maximum volume of communication between two halves of network or in other words maximum data transfer capacity between two halves of network and is given by bisection width * channel bandwidth. Higher bisection bandwidth means better performance under heavy traffic.

Questions / Doubts

1. How fat tree network has constant bisection bandwidth?