CMSC416: Introduction to Parallel Computing

Topic: Parallel Networks
Date: April 23, 2024

N-dimensional mesh / torus networks

Mesh and torus networks are commonly used to interconnect processors or nodes with a supercomputer. These networks are typically characterized by their arrangement of nodes and switches, which is essential to minimizing latency and maximizing bandwidth across the network.

N-dimensional Mesh Networks:
Switches are arranged in an N-dimensional grid and each switch typically encompasses a single node or a small group of nodes. The switches are responsible for routing data between nodes across the network, and each switch is connected to 2N other switches, where N is representative of the number of dimensions here. For example, in a 2D mesh, each switch has 4 connections (up, down, left, right). Messages are routed through dimensions one at a time and this network architecture scales linearly as well with the number of nodes.

N-dimensional Torus Networks:
A torus network extends the mesh by adding wraparound links. The edges here form a continuous loop in each dimension. Each switch in an N-dimensional torus is also connected to 2N switches, but this offers an advantage in the form of a reduced network diameter due to the loop-around conditions.

Fat-tree networks

A fat-tree network is a layered, hierarchical framework that has multiple paths between any two nodes to prevent bottlenecks. As you move up the hierarchy, the switches have more bandwidth to accommodate the total traffic across the loyal layers. “Router radix” refers to the number of ports on a router or switch, and in a fat-tree network has a radix of k. The number of nodes connected to each router or switch is typically k / 2, which means that each switch can handle traffic from k / 2 compute nodes. A pod is a subsection of the network consisting of a set of switches that connect a group of compute nodes, and each pod includes k/2 switches at each level within the pod. The maximum number of pods is k, as the “router radix” is k per switch. Pods are designed to be self-sufficient and can handle internal traffic efficiently while also connecting to the larger network. In the fat-tree hierarchy, there are usually three levels of switches:

Level 1 (Edge Level): The switches connect directly to the compute nodes.
Level 2 (Aggregation Level): The switches connect the edge switches to the core layer.
Level 3 (Core Level): The top-level switches interconnect all the pods and handle traffic that needs to travel between pods.

The bandwidth is typically much “fatter” near the top of the tree, which ensures that as multiple paths from the lower levels converge, the network can still handle the increased traffic without becoming a potential bottleneck. In regards to scalability, fat-tree networks can scale by adding more pods or increasing the radix k and due to the presence of multiple redundant paths, the network can continue to function even if some switches or links fail, which makes it a very scalable and robust architecture for HPC environments.

Dragonfly network

The dragonfly network is a two-level hierarchical network composed of groups of routers. Each group at the lower level is interconnected with itself, often in the form of a complete graph of nodes, meaning that every router has a direct connection to every other router within the same group. The groups are then sparsely connected at the higher level. High-radix routers are used, meaning each router has a large number of ports, allowing for a significant number of connections within a group. This in turn enables a low network diameter, as there are less routers for data to pass through before reaching its destination due to the dense architecture of supernodes in this network. A supernode in this context refers to a collection of routers that likely function as a single cohesive unit within the network architecture.

Message life-cycle

The source in this life-cycle is where the message begins, and this could be a computer or a server, for example. The messages are determined by the application generating the message, and destination, how often messages are sent (frequency), and the size of the messages are also sent. The generation of messages can range widely in time as well, from microseconds to tens of seconds. The NIC (Network Interface Card) receives data from the source, and prepares it for transmission by packetizing the data, or breaking the data into packets. This step typically involves some delay involving hundreds of nanoseconds. As packets enter the network, the routers/switches determine the best path for them to reach their destination, and this step incurs a delay of approximately 100 nanoseconds. Packets may temporarily be stored in buffers if they cannot move forward immediately due to network congestion, for example. The packets travel through various links between routers/switches and these links can become congestion points if many packets are trying to simultaneously traverse the same paths, ranging in time from 1 to 50 nanoseconds. Once the packets reach the destination’s local network, they are received by the NIC and correspondingly assembled back into the original message to deliver to the destination application. The time taken for the destination to process the received message can vary from microseconds to tens of seconds.

Network congestion

Network sharing occurs when multiple data flows, originating from different programs or processes, use the same network infrastructure, such as switches, routers, and physical links.

In a network, shared resources typically include the physical transmission media, along with the switches and routers that direct traffic from one point to another. Congestion on these shared links happens when too much data is sent through the network at once, and can cause delays and potential packet loss. For example, congestion can be caused if two programs send data over a network simultaneously, filling the bandwidth of the shared links. The subsequent programs that are trying to communicate through these congested links will then also experience delays, resulting in increased latency.