Chapter 7

Network Computing
7.1 Computer Networks Basics

• Network Performance
  – Gilder’s law
    • George Gilder projected that the total bandwidth of communication systems triples every 12 months.
  – Metcalfe’s law
    • Robert Metcalfe projected that the value of a network is proportional to the square of the number of nodes.
7.1 Computer Networks Basics

• Internet

  – Internet is the collection of networks and routers that form a single cooperative virtual network, which spans the entire globe.

  – The majority of Internet traffic is carried using TCP/IP packets.

  – The explosive usage of the Internet has stimulated rapid growth of interest in electronic publishing, browsing and distributed computing.

    • In the United States alone, 71% of the population use the Internet
7.1 Computer Networks Basics

• Other network technologies
  – Fast Ethernet and Gigabit Ethernet.
  – The Fiber Distributed Data Interface (FDDI).
  – High-Performance Parallel Interface (HIPPI).
  – Asynchronous Transfer Mode (ATM).
  – Scalable Coherent Interface (SCI).
7.1 Computer Networks Basics

A representation of network technologies
7.2 Client/Server Systems

- **Sockets**
  - Sockets are used to provide the capability of making connections from one application running on one machine to another running on a different machine.
  - Once a socket is created, it can be used to wait for an incoming connection (passive socket) or can be used to initiate connection (active socket).
7.2 Client/Server Systems

• Remote Procedure Call (RPC)
  – RPC is the basis of most client/server systems.
  – When the procedure is called, its parameters are sent via the network to the remote computer, and then the remote computer executes the procedure, returns the results, and continues on its way.
  – RPC can be constructed on top of sockets. The socket mechanism can be used to pass parameters.
7.2 Client/Server Systems

- **Remote Procedure Call (RPC)**
  - RPC can be:
    - Blocking: the program that places the call is stopped in its track while waiting for a reply.
    - Nonblocking: the calling program is allowed to continue without waiting for a reply.
7.2 Client/Server Systems

- **Middleware**
  - Is an important intermediate layer of software for the following reasons:
    - It makes it possible for new systems to coexist with legacy systems.
    - It solves a number of interoperability problems because it can simultaneously converts formats and gain access without code rewriting.
    - It isolates system components so that changes in one component have little effect on other components.
    - It lowers effort and time to develop and deploy systems because programmers don’t need to know network and distributed programming details.
7.2 Client/Server Systems

- A client server framework for parallel applications
  - Client (supervisor)
    - Clients create an array of sockets and input/output data streams with all the servers.
    - Clients pass control to the client body, which contains the code specific to the application being executed in parallel. Mainly, it divides the main task into smaller portions and passes one small portion of the task to each server. It then waits for all servers to send back the result of their smaller computations. Finally it merges the results of each server and computes the final solution to the big problem.
    - Client closes all the streams and sockets with all the servers.
7.2 Client/Server Systems

- A client server framework for parallel applications
  - Server (worker)
    - Server creates a server socket on an unused port number.
    - Server waits for connections on that port. Once it gets a request from a client, it accepts that connection.
    - Server creates input and output data streams for that socket.
    - Server passes control to the server body, which contains the code specific to the application executed in parallel. The main server would accept the connection from the client, create a socket, and invoke the server body thread to handle that client.
    - Server goes back and waits for another connection from a client.
7.2 Client/Server Systems

- A client server framework for parallel applications
7.3 Clusters

- A cluster is a collection of stand-alone computers connected using some interconnection network.
- A node in a cluster is an autonomous computer that may be engaged in its own private activities while at the same time cooperating with other units in the context of computational task.
- When all the nodes in a cluster have the same architecture and run the same operating system, the cluster is called homogenous. Otherwise, it is heterogeneous.
7.3 Clusters

A Cluster made of homogenous single-processor computers
7.3 Clusters

• Threads
  – Modern operating systems support threads within processes.
  – Concurrency among processes can be exploited because threads in different processes may execute concurrently.
  – Multiple threads within the same process can be assigned to different processors.
7.4 Interconnection Networks

• Ethernet
  – A packet-switched LAN technology.
  – All hosts connected to an Ethernet receive every transmission, making it possible to broadcast a packet to all hosts at the same time.
  – Ethernet uses a distributed access control scheme called Carrier Sense Multiple Access with Collision Detect (CSMA/CD).
  – Each computer connected to an Ethernet network is assigned a unique 48-bit address known as its Ethernet address, also called the media access control address, (MAC).
7.4 Interconnection Networks

• Switches
  – A $n_1 \times n_2$ switch consists of:
    • $n_1$ input ports
    • $n_2$ output ports
    • Links connecting each input to every output
    • Control logic to select a specific connection
    • Internal buffers
  – The connections between input ports and output ports may be:
    • One-to-one (point-to-point)
    • One-to-many (multicast or broadcast)
    • Many-to-one: may cause conflicts at the output ports and needs arbitration.
7.4 Interconnection Networks

• Switches
  – When only one-to-one connections are allowed, the switch is called crossbar.
    • An $n \times n$ crossbar switch can establish $n!$ connections.
  – If we allow both one-to-one as well as one-to-many in an $n \times n$ switch, the number of connections that can be established is $n^n$. 
7.4 Interconnection Networks

- **Switches**
  - Routing can be achieved using 2 mechanisms:
    - **Source-path**: the entire path to the destination is stored in the packet header at the source location.
    - **Table-based**: the switch must have a complete routing table that determines the corresponding port for each destination.

Source-path Routing versus Table-based Routing
7.4 Interconnection Networks

• Myrinet Clos network
  – Myrinet is a high-performance, packet communication and switching technology.
  – Myrinet switches are multiple-port components that route a packet entering on an input channel of a port to the output channel of the port selected by the packet.
7.4 Interconnection Networks

- Myrinet Clos network

Network Spine

Clos “Spreader” Network
Connects Spine (upper 8 switches) to Leaves (16 lower switches)

128-host Clos Network using 16-port Myrinet Switch

128 Hosts
7.4 Interconnection Networks

- Myrinet Clos network

64-host Clos Network using 16-port Myrinet Switch (Each line represents 2 links)
7.4 Interconnection Networks

- Myrinet Clos network

32-host Clos Network using 16-port Myrinet Switch (Each line represents 4 links)
7.4 Interconnection Networks

• The Quadrics network (QsNet)
  – Consists of 2 hardware building blocks
    • A programmable network interface called Elan:
      – connects the Quadrics network to a processing node containing one or more CPUs.
      – Elan provides substantial local processing power to implement high-level message passing protocols (ex: MPI).
    • High-bandwidth, low-latency communication switch called Elite:
      – QsNet connects Elite switches in a quaternary fat-tree topology.
7.4 Interconnection Networks

- The Quadrics network (QsNet)
  - Elite switch of Quadrics network:
    - Elite networks are source routed.
7.5 Cluster Examples

• Berkeley Network of Workstations (NOW)
  – The Berkeley Network of workstations is an important representative of cluster systems.
  – In 1997, it was one of the top 200 fastest supercomputers in the world.
  – The infrastructure included
    • 100 SUN Ultrasparcs and 40 SUN Sparcstations running Solaris,
    • 35 Intel PCs running Windows NT or a PC Unix variant
    • 500-1000 disks
    • All is connected by a Myrinet switched network.
7.5 Cluster Examples

• Berkeley Network of Workstations (NOW)
  • The programming environments used are:
    – Sockets
    – MPI
    – A parallel version of C called Split C.
7.5 Cluster Examples

• The Beowulf Cluster
  – The idea of the Beowulf Cluster was to achieve supercomputer processing power using off-the-shelf commodity machines.
  – The communication between processors in Beowulf has been done through TCP/IP over the Ethernet internal to the cluster.
  – Channel bonding is a technique to connect multiple Ethernets in order to distribute the communication traffic.
  – Two of the early successful Beowulf clusters are Loki and Avalon.
7.5 Cluster Examples

• FlashMob I
  – A FlashMob supercomputer was created by connecting a large number of computers via a high-speed LAN.
  – A FlashMob computer is temporary and organized ad hoc for the purpose of working on a single problem.
7.6 Grid Computing

- Grids consist of multiple systems that work together while maintaining their distinct identities.
- Grid resources, which span the entire globe, include hardware, software, data and instruments.
- An important concept is grids is the virtual organization, which offers a unified view of resources.
- The challenge is to allow meaningful sharing of resources without compromising local autonomy.
- Examples of grid platforms and tools are Globus and TeraGrid.
7.7 Summary

- Distributed platforms may be connected in a variety of ways ranging from geographically dispersed networks to architecture-specific interconnection structures.
- A processing unit in such system is an autonomous computer that may be engaged in its own private activities while at the same time cooperating with other units in the context of some computational task.
- Network computing is concerned with how to use multiple computers to solve single or multiple problems, more or less simultaneously.
- A number of models exist to aggregate the resources of multiple computer engines for large-scale processing tasks.