Distributed Computing

Mahesh C

Design Issues and Challenges - Algorithmic Perspective

- Designing useful execution models and frameworks
- Dynamic distributed graph algorithms and distributed routing algorithms
- Time and global state in a distributed system.
- Synchronization.
- Synchronization/ coordination mechanisms.
- Reliable and fault-tolerant distributed systems
- Real-time scheduling.
- Performance
- Group communication, multicast, and ordered message delivery.
- Monitoring distributed events and predicates.
- Distributed program design and verification tools.
- Data replication, consistency models, and caching.
- World Wide Web design – caching, searching, scheduling.
- Distributed shared memory abstraction.
- Load balancing.

Applications of Distributed Computing

- Mobile systems
- Ubiquitous or pervasive computing
- Publish-subscribe, content distribution, and multimedia.
- Security in distributed systems.
- Sensor networks
- Peer-to-peer computing
- Distributed agents
- Distributed data mining
- Grid computing

A Model of Distributed Computations

- The processors do not share a common global memory and communicate solely by passing messages over the communication network.
- A distributed system consists of a set of processors that are connected by a communication network. The communication delay is finite but unpredictable.
- There is no physical global clock in the system to which processes have instantaneous access.
- Thecommunication medium may deliver messages out of order, messages may be lost, garbled, or duplicated due to timeout and retransmission, processors may fail, and communication links may go down.

A Model of Distributed Computations

The system can be modeled as a directed graph in which vertices represent the processes and edges represent unidirectional communication channels.

A Distributed Program

A distributed program is composed of a set of n asynchronous processes p_1, p_2, ..., p_i, ..., p_n that communicate by message passing over the communication network. Without loss of generality, assume that each process is running on a different processor.

A Distributed Program

- Let C ij denote the channel from process p_i to process p_j and let M_{ij} denote a message sent by p_i to p_j . The communication delay is finite and unpredictable.
- Process execution and message transfer are asynchronous – a process may execute an action spontaneously and a process sending a message does not wait for the delivery of the message to be complete.

A Distributed Program

- The global state of a distributed computation is composed of the states of the processes and the communication channels.
- The state of a process is characterized by the state of its local memory and depends upon the context.
- The state of a channel is characterized by the set of messages in transit in the channel.

A Model of Distributed Executions

- The execution of a process consists of a sequential execution of its actions.
- The actions are atomic and the actions of a process are modeled as three types of events, namely, internal events, messagesend events, and message receive events.

A Model of Distributed Executions