RAMA UNIVERSITY

www.ramauniversity.ac.in

FACULTY OF EGINEERING AND TECHNOLOGY

Distributed Systems(BCS-701)
LECTURE-04

Dr. Hariom Sharan

Professor & Dean

Computer Science & Engineering

OUTLINE

- System Models
- Architectural Model
- Middleware
- System Architectures
- Client Server
- Peer to Peer
- References

SYSTEM MODELS

INTRODUCTION:

Systems that are intended for use in real-world environments should be designed to function correctly in the widest possible range of circumstances and in the face of many possible difficulties and threats.

Different system models are,

Architectural models describe a system in terms of the computational and communication tasks performed by its computational elements; the computational elements being individual computers or aggregates of them supported by appropriate network interconnections.

Fundamental models take an abstract perspective in order to examine individual aspects of a distributed system. In this chapter we introduce fundamental models that examine three important aspects of distributed systems: interaction models, which consider the structure and sequencing of the communication between the elements of the system; failure models, which consider the ways in which a system may fail to operate correctly and; security models, which consider how the system is protected against attempts to interfere with its correct operation or to steal its data.

ARCHITECTURAL MODEL

The architecture of a system is its structure in terms of separately specified components and their interrelationships. The overall goal is to ensure that the structure will meet present and likely future demands on it. Major concerns are to make the system reliable, manageable, adaptable and cost-effective. The architectural design of a building has similar aspects – it determines not only its appearance but also its general structure and architectural style (gothic, neo-classical, modern) and provides a consistent frame of reference for the design.

Software Layers

- **Software architecture referred to:**
  - The structure of software as layers or modules in a single computer.
  - The services offered and requested between processes located in the same or different computers.
- Software architecture is breaking up the complexity of systems by designing them through layers and services.
  - **Layer:** a group of related functional components.
  - Service: functionality provided to the next layer.

ARCHITECTURAL MODEL

Platform

▪The lowest-level hardware and software layers are often referred to as a platform for distributed systems and applications.

- ➢These low-level layers provide services to the layers above them, which are implemented independently in each computer.
- ➢These low-level layers bring the system's programming interface up to a level that facilitates communication and coordination between processes.

•Common examples of platform are: Intel x86/Windows, Intel x86/Linux Intel x86/Solaris , SPARC/SunOS, PowePC/MacOS

Middleware

It was a layer of software whose purpose is :

- To mask heterogeneity presented in distributed systems and provides interoperability between lower layer and upper layer.
- To provide a convenient programming model to application developers.

Major Examples of middleware are:

- Sun RPC (Remote Procedure Calls)
- OMG CORBA (Common Request Broker Architecture)
- Microsoft D-COM (Distributed Component Object Model)
- Sun Java RMI

System Architectures

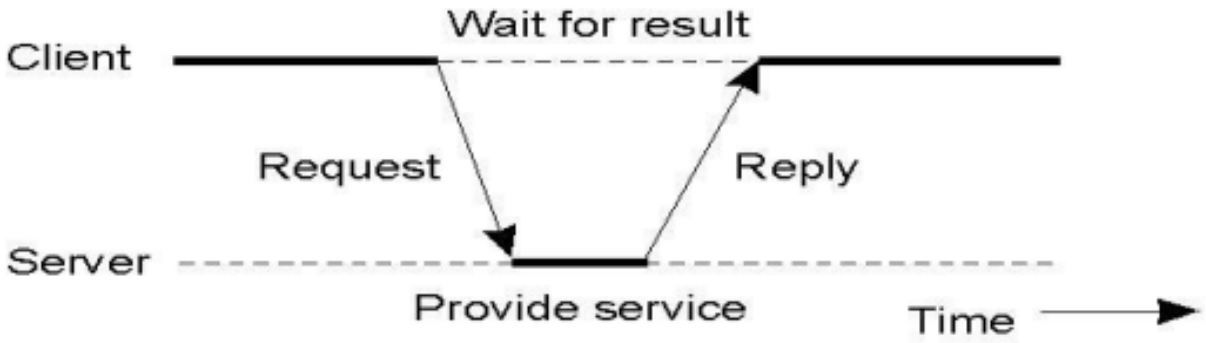
•The most evident aspect of distributed system design is the division of responsibilities between system components (applications, servers, and other processes) and the placement of the components on computers in the network.

It has major implication for:

- •Performance
- •Reliability
- •Security

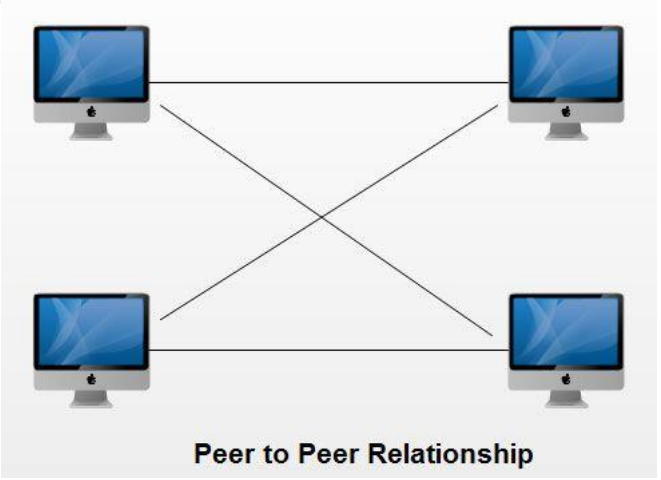
Client Server

This is the architecture that is most often cited when distributed systems are discussed. It is historically the most important and remains the most widely employed. In particular, client processes interact with individual server processes in potentially separate host computers in order to access the shared resources that they manage. Servers may in turn be clients of other servers, as the figure indicates. For example, a web server is often a client of a local file server that manages the files in which the web pages are stored. Web servers and most other Internet services are clients of the DNS service, which translates Internet domain names to network addresses. Another web-related example concerns search engines, which enable users to look up summaries of information available on web pages at sites throughout the Internet.



Peer to Peer

In this architecture all of the processes involved in a task or activity play similar roles, interacting cooperatively as peers without any distinction between client and server processes or the computers on which they run. In practical terms, all participating processes run the same program and offer the same set of interfaces to each other. While the client-server model offers a direct and relatively simple approach to the sharing of data and other resources, it scales poorly. The centralization of service provision and management implied by placing a Service at a single address does not scale well beyond the capacity of the computer that hosts the service and the bandwidth of its network connections.



Peer to Peer Relationship

References

- https://www.javatpoint.com/digital-image-processing-tutorial
- https://www.tutorialpoint.com/
- Distributed Systems, Concepts and Design, George Coulouris, J Dollimore and Tim Kindberg, Pearson Education, 4th Edition,2009.
- Distributed Systems, Principles and paradigms, Andrew S.Tanenbaum, Maarten Van Steen, Second Edition, PHI.
- Distributed Systems, An Algorithm Approach, Sikumar Ghosh, Chapman & Hall/CRC, Taylor & Fransis Group, 2007.