LECTURE NOTE 1 [CSC 421]

INTRODUCTION TO DISTRIBUTED COMPUTING

1.1 Distributed System

A Distributed System is a system whose components are located on different networked computers, which communicate and coordinate their actions by passing messages to one another from any system to appear as a single system to the end user.

The computers that are in a distributed system can be physically together and connected by a local network, or they can be geographically distant and connected by a wide area network.

A distributed system can consist of any number of possible components such as mainframes, personal computers, workstations, minicomputers, and so on. Common use cases of distributed systems are electronic banking systems, massively multiplayer online games, and sensor networks.

1.2 Functionality

Two general ways distributed systems function:

- Each component of the system works to achieve a common goal and the end-user views results as one combined unit.
- Each component has its end-user and the distributed system facilitates sharing resources or communication services.

1.3 Architectural models

Distributed systems generally consist of four different basic architectural models:

- ✓ Client-server — Clients contact the server for data, then format it and display it to the end-user.
- ✓ Three-tier — Information about the client is stored in a middle tier rather than on the client, to simplify application deployment.
- ✓ n-tier — Generally used when the server needs to forward requests to additional enterprise services on the network.
- ✓ Peer-to-peer — There are no additional nodes used to provide services or manage resources. Responsibilities are uniformly distributed among components in the system, known as peers, which can serve as either client or server.

1.4 Distributed Computing

Distributed computing is computing over distributed autonomous computers that communicate only over a network. Distributed computing systems are usually treated differently from parallel computing systems or shared-memory systems where multiple computers share a common memory pool that is used for communication between the processors.

Distributed memory systems use multiple computers to solve a common problem, with computation distributed among the connected computers (nodes) and using message passing to communicate between the nodes.

An example of distributed computing is grid computing where the nodes may belong to different administrative domains.

Distributed computing can include heterogeneous computations where some nodes may perform a lot more computation, some perform very little computation and a few others may perform specialized functionality (like processing visual graphics).

One of the ***main advantages of using distributed computing*** is that efficient scalable programs can be designed so that independent processes are scheduled on different nodes and they communicate only occasionally to exchange results – as opposed to working out of a shared memory with multiple simultaneous accesses to a common memory.

Cloud computing is also a specialized form of distributed computing, where distributed Software as a Service (SaaS) applications utilize thin clients (such as browsers) that offload computation to cloud-hosted servers (and services).

Distributed computing, virtualization, service orientation, and Web 2.0 form the core technologies enabling the provisioning of cloud services from anywhere on the globe.

1.5 Web 2.0 Technologies

Web 2.0 technologies constitute the interface through which cloud computing services are delivered, managed, and provisioned. Besides the interaction with rich interfaces through the Web browser, Web services have become the primary access point to cloud computing systems.

1.6 Service Orientations

Service orientation is the underlying paradigm that defines the architecture of a cloud computing system. Cloud computing is often summarized with the acronym XaaS meaning, Everything-as-a-Service - that underlines the central role of service orientation.

Infrastructure-as-a-service solutions provide the capabilities to add and remove resources, but it is up to those who deploy systems on this scalable infrastructure to make use of such opportunities with wisdom and effectiveness.

Platform-as-a-service solutions embed into their core offering algorithms and rules that control the provisioning process and the lease of resources. These can be either completely transparent to developers or subject to fine control.

1.7 Virtualization

It is a technology that serves as a core feature of the infrastructure used by cloud providers. Virtual environments could be abstractions of virtual hardware or a runtime environment.