Tutorial on Client-Server Architecture
SEEM3430 Information Systems Analysis and Design

Pengfei Liu

Department of Systems Engineering and Engineering Management
The Chinese University of Hong Kong

March 19, 2015
Outline

1 Objectives
2 Mainframe
3 Client-Server Model
   - Terms
   - Overview of Client-Server Model
   - Layered Architecture
   - Tiered Distribution
4 Cloud Computing
5 Summary
Find answers to these questions:

1. What is client-server architecture?
2. What are the differences between N-layered architecture and N-tiered distribution?
3. Can you describe the computing paradigm shift?
Outline

1. Objectives
2. Mainframe
3. Client-Server Model
   - Terms
   - Overview of Client-Server Model
   - Layered Architecture
   - Tiered Distribution
4. Cloud Computing
5. Summary
Mainframe computers are computers used primarily by corporate and governmental organizations for critical applications.

- industry and consumer statistics
- enterprise resource planning
- bank transaction processing

Mainframe originally referred to the large cabinets that housed the central processing unit and main memory of early computers.

---

Figure 1: IBM System/360 Model 65 Computer

http://www.computerhistory.org/revolution/mainframe-computers/7/intro/565
Terminal-Mainframe Architecture

Figure 2: Terminal-Mainframe Architecture

Source: http://contentdeliverance.com/
Outline

1. Objectives
2. Mainframe
3. Client-Server Model
   - Terms
   - Overview of Client-Server Model
   - Layered Architecture
   - Tiered Distribution
4. Cloud Computing
5. Summary
Layer vs. Tier

**Definition (Layer)**

Layers describe the logical groupings of the functionality and components in an application. A layer is a logical structuring mechanism for the components that make up the software solution.

**Definition (Tier)**

Tiers describe the physical distribution of the functionality and components on separate servers, computers, networks, or remote locations. A tier is a physical structuring mechanism for the system infrastructure.
Client-Server Model

Definition

The client/server model is a computing model that acts as a distributed application which partitions tasks or workloads between the providers of a resource or service, called servers, and service requesters, called clients. \(^a\)

\(^a\)http://en.wikipedia.org/wiki/Client-server_model

1. A server is a host that is running one or more server programs which share their resources with clients.

2. A client does not share any of its resources, but requests a server’s content or service function.
Figure 3: Client-Server Model

Source: http://contentdeliverance.com/
Two major parts in this tutorial:

1. Logical Layered Architecture
2. Physical Tiered Distribution
Layered Architecture

Splitting application functionality into layers with distinct roles and functionalities

1. Help you maximize the maintainability of the application
2. Optimize the way the application works when deployed in different ways

Figure 4: Layers

Three-layered architecture

Figure 5: The logical architecture view of a layered system

Three-layered architecture

1. Presentation Layer: contains the user oriented functionality responsible for managing user interaction with the system, and generally consists of components that provide a common bridge into the core business logic encapsulated in the business layer.

2. Business Layer: implements the core functionality of the system, and encapsulates the relevant business logic.

3. Data Layer: provides access to data hosted within the boundaries of the system, and data exposed by other networked systems; perhaps accessed through services.
Distributed System

A distributed system consists of multiple autonomous computers that communicate with each other through a computer network in order to achieve a common goal.

N-tiered Distribution

- Distributing the layers over multiple physical tiers
- In order to improve scalability, fault-tolerance, and performance.
Example: an order-processing application

- The example is an order-processing application, and only considers one use case: Process Order.
- This use case is responsible for allowing a customer service representative to enter an order into the system.
Single-tiered distribution

Figure 6: Single-tiered distribution

Two-tiered distribution

Figure 7: Two-tiered distribution

Three-tiered distribution

Figure 8: Three-tiered distribution

Client Tier

- Thin Client: uses a browser to provide the execution environment for the application’s user interface (HTML defined).
- Thick Client: also known as Fat Client/Rich Client, involve compiled code artifacts (components, assemblies, etc.) running on the client machine to provide the application’s user interface and client-side logic.
- Smart Client: represent a best-of-both-worlds approach.
Comparison of Thick, Smart and Thin Client

Figure 9: Comparison of Thick, Smart and Thin Client

http://www.develop-one.net/blog/2006/01/10/DefinitionForSmartClient.aspx
Four-tiered distribution

Figure 10: Four-tiered distribution

Criteria for structuring your solution’s tiers

- Tiers *allow servers and client computers to be optimized for specific tasks*, such as interacting with users, serving Web pages, and hosting databases.
- Tiers *separate servers that had different security profiles, scalability and fault-tolerance requirements*.
- Tiers *reduce the administration and deployment overhead required of distributed applications*.
- However, each tier involved in processing a client request *degrades performance and adds application and system management overhead*. 
Some Notes on N-layered Architecture and N-tiered Distribution

1. Layers are concerned with the logical division of components and functionality, and do not take into account the physical location of components.

2. Layers can be located on different tiers, or they may reside on the same tier.

3. Although both layers and tiers use the same set of names (presentation, business, services, and data), remember that only tiers imply a physical separation.

4. Not all N-layered applications should be distributed as N-tiered, but all N-tiered distribution must internally be designed as N-layered architecture.
Outline

1. Objectives
2. Mainframe
3. Client-Server Model
   - Terms
   - Overview of Client-Server Model
   - Layered Architecture
   - Tiered Distribution
4. Cloud Computing
5. Summary
Definition of Cloud Computing

Definition

Cloud computing is a model for enabling ubiquitous, convenient, on-demand network access to a shared pool of configurable computing resources (e.g., networks, servers, storage, applications, and services) that can be rapidly provisioned and released with minimal management effort or service provider interaction. 

Figure 11: Cloud computing logical diagram

http://en.wikipedia.org/wiki/Cloud_computing
Service Models of Cloud Computing

- **Software as a Service (SaaS):** The capability provided to the consumer is to use the provider’s applications running on a cloud infrastructure.

- **Platform as a Service (PaaS):** The capability provided to the consumer is to deploy onto the cloud infrastructure consumer-created or acquired applications created using programming languages, libraries, services, and tools supported by the provider.

- **Infrastructure as a Service (IaaS):** The capability provided to the consumer is to provision processing, storage, networks, and other fundamental computing resources.
Deployment Models of Cloud Computing

- **Private cloud**: The cloud infrastructure is provisioned for exclusive use by a single organization comprising multiple consumers (e.g., business units). It may be owned, managed, and operated by the organization, a third party, or some combination of them, and it may exist on or off premises.

- **Public cloud**: The cloud infrastructure is provisioned for open use by the general public. It may be owned, managed, and operated by a business, academic, or government organization, or some combination of them. It exists on the premises of the cloud provider.

- **Hybrid cloud**: The cloud infrastructure is a composition of two or more distinct cloud infrastructures (private, community, or public) that remain unique entities, but are bound together by standardized or proprietary technology that enables data and application portability (e.g., cloud bursting for load balancing between clouds).
Computing Paradigm Shift

Figure 12: Computing paradigm shift

Source: BofA Merrill Lynch Global Research.
Outline

1. Objectives
2. Mainframe
3. Client-Server Model
   - Terms
   - Overview of Client-Server Model
   - Layered Architecture
   - Tiered Distribution
4. Cloud Computing
5. Summary
Summary

1. Objectives
2. Mainframe
3. Client-Server Model
   - Terms
   - Overview of Client-Server Model
   - Layered Architecture
   - Tiered Distribution
4. Cloud Computing
5. Summary
Thank You!