Lecture 02: Architectural Models

Distributed Systems
Behzad Bordbar
School of Computer Science, University of Birmingham, UK

Recap
- Definition of DS
- Examples of DS
- Challenges of the design
  - Heterogeneity
  - Openness
  - Security
  - Scalability
  - Failure handling…

Outline
- Why models?
- models of e-business systems
- various architectures, client-server, p2p
  Variations
- Multiple servers
- Proxy servers
- Mobile code
  Design requirements

Why models?
L1 \times w1 = L2 \times w2

???(more complex models required)
Abstraction to better understand, never real world!

Example: e-Business application
Architecture: structure in terms of separate specified component.

Presentation Layer
Facilitate interaction with the user (human or software)
- Service for price checking
- currency converter
user of the presentation layer submits operations and get responses.
The boundary between P layer and client can be very thin. For example, Java Applet
Before the delivery of the result some data processing is required. Processing is the implementation of the information required by the client of the P. layer ALL are all those programs and module involved in processing the operation Example: Bank Cash Machine Application Logic also called Business process, Business logic

Single Tier: all three into a single component

- Dumb Terminal

Mostly in Mainframes from the days that: 640K ought to be enough for anybody.
- Microsoft Chairman Bill Gates, 1981
- IBM Chairman Thomas Watson, 1943

2-tier Architecture

Birth of PC:
- Small computer (PC)
- Large computers (Server, Mainframe)

Fat Client VS. Thin Client
- The idea of API ☺

Remote Procedure Call

2-tier Architecture

Three Tier

Infrastructure that supports the development of ALL is called a middleware
- Sun RPC
- Java RMI (Remote Method Inv.)
- SOAP (which is RPC based)
- CORBA
- JMS (Java Messaging Services)
- Data Layer resulted in better interfaces
- ODBC (Open DB Connectivity)
- JDBC (Java …)

N Tier

Many Distributed Systems (3-Tiers) interacting.
- Notice: phrase Tier can also imply physical separation of components, i.e. on various hardware (Physical Tier vs. Logical Tier)

Now,
- Question: what the underlying model?
- Study of architecture to ensure the system meets preset and future demand

Basic Client/Server model

Global Time
**Architectural models**

Define
1. software components (processes, objects)
2. ways in which components interact
3. mapping of components onto the underlying network

Why needed?
- to handle varying environments and usage
- to guarantee performance

---

**System architectures**

Main
- Client-server
- Peer to peer (P2P)

Variations
- Multiple servers
- Proxy servers
- Mobile code
- Mobile agent (can have serious security problems)
- Network Computer
- Thin client (you have seen this)

---

**Client server**

Server1 acts as client for Server2

---

**Peer processes**

p2p file sharing

---

**Multiple servers**

Servers may interact

---

**Proxy servers for caching**


Mobile code
Client requests results, applet code is downloaded:

Client interacts with the applet:

Design Requirements for DSs
Judging how good the architecture is...
Performance
- how fast will it respond?
Quality of Service
- are video frames and sound synchronised?
Dependability
- does it work correctly?

Performance
Responsiveness
- fast interactive response delayed by remote requests
- use of caching, replication
Throughput
- dependent on speed of server and data transfer
Load balancing
- use of applets, multiple servers

Quality of Service (QoS)
Non-functional properties experienced by users:
Deadline properties
- hard deadlines (must be met within T time units)
- soft deadlines (there is a 90% chance that the video frame will be delivered within T time units)
- multimedia traffic, video/sound synchronisation
- depend on availability of sufficient resources
Adaptability
- ability to adapt to changing system configuration

Correctness
- correct behaviour wrt specification
- e.g. use of verification
Fault-tolerance
- ability to tolerate/recover from faults
- e.g. use of redundancy
Security
- ability to withstand malicious attack
- e.g. use of encryption, etc

Summary
why models?
e.g. example of Tired systems. What is the underlying model?
Basic client server
Multiple servers, Proxy servers, P2P
The main requirements of DS are
Performance, QoS, Dependability
Further reading: pages 29-47
2.1 Describe and illustrate the client-server architecture of one or more major Internet applications (for example, the Web, email or netnews).

2.2 For the applications discussed in Exercise 2.1 state how the servers cooperate in providing a service.

2.4 A search engine is a web server that responds to client requests to search in its stored indexes and (concurrently) runs several web crawler tasks to build and update the indexes. What are the requirements for synchronization between these concurrent activities?

2.9 Distinguish between buffering and caching.