Chapter 2 Architectures

• Software architecture – logical organization of the software components
  – Architectural styles: layered, object-based, event-based, shared data space

• System architecture – the instantiation of a software architecture
  – Centralized, decentralized, hybrid

Layered Architecture

• Each layer uses the layer below to implement new service that is used by the layer above.
• Adopted by networking software.
Object-Based Architecture

- Each object corresponds to a software component
- Components interact via remote procedure calls

Event-Based Architecture

- Components communicate via events
  - Use a publish/subscribe paradigm
  - Components can publish events, subscribe to events
  - Events are delivered to subscriber once published by publisher
  - Communicating processes need not explicitly refer to each other – decoupled in space
  - Communicating processes must both be active – coupled in time
Shared Data Space

- Processes communicate through a common repository
- Publishers post items to shared data space; subscribers access items at a later time
- Processes are decoupled in both space and time

Centralized Architectures

- Basic client-server model
  - Server: a process that offers a service
  - Client: a process that requests a service from a server
Application Layering

• Many client-server applications are organized into three layers
  – User-interface level: contains application’s user interface
  – Processing level: contains the functions of application
  – Data level: contains the data that a client wants to manipulate through the application

Search Engine Example

The simplified organization of an Internet search engine into three different layers.
Physically Two-Tiered Architecture

- Physically distribute a client-server application across two types of machines: client machine and server machine

Physically Three-Tiered Architecture

- Application server also acts as a client
- Very common in web-based applications
Decentralized Architectures

• Peer-to-peer (P2P) systems
  – All processes are equal
  – Each process acts as a client and a server at the same time
• Processes in P2P systems form an overlay network
  – Links represent communication channels between processes
• Two types of P2P systems
  – Structured: the overlay network is constructed using a deterministic procedure
  – Unstructured: the overlay network is constructed using a randomized algorithm

Chord: A Structured P2P System

• Each data item/node is assigned a random key/identifier from a large ID space
• Nodes are logically organized in a ring
• Data item with key $k$ is mapped to the node with smallest identifier $id \geq k$, denoted as $\text{succ}(k)$
• The system provides an operation $\text{LOOKUP}(key)$ that will return the network address of $\text{succ}(k)$. 

![The mapping of data items onto nodes in Chord.](chart)
Chord (Continued)

• When a node wants to join the system
  – Generate a random identifier id
  – Do a lookup on id to get the network address of succ(id)
  – Contact succ(id) and its predecessor, insert itself in the ring
  – Data items for which node id is now responsible are transferred from succ(id)

• When node id wants to leave the system
  – Inform its predecessor and successor
  – Transfer its data items to succ(id)

Unstructured P2P Systems

• Overlay network constructed based on randomized algorithms
  – Each node maintains a list of c randomly chosen neighbors (referred to as a partial view)
  – Each node P periodically selects a node Q from its partial view
  – P and Q exchange members from their respective partial views

• Data items are randomly placed on nodes
  – Flood the network with a search query to locate a data item.

• Membership management
  – To join the network, a node contacts an arbitrary node from a list of well-known access points
  – To leave the network, a node can depart without informing any other node
Superpeers

- Locating data items can be made more efficient by using superpeers
  - A regular peer is connected to a superpeer as a client
  - Superpeers maintain an index of clients’ data items
  - Clients submit queries to their superpeer
- Superpeers should be long-lived processes with high availability and enough capacity to process requests

Hybrid Architectures

- Client-server combined with P2P
- Examples
  - Edge-server systems
  - BitTorrent
Edge-Server Systems

- Often used for Content Delivery Networks (CDNs)
- Servers placed “at the edge” of the Internet
- Edge servers serve content

BitTorrent

- P2P file downloads: a peer downloads chunks of a file from multiple other peers
- User contacts the tracker for the requested file to get a list of active nodes and starts to download chunks from them
- Force altruism: *If P sees Q downloads more than uploads, reduce rate of sending to Q*