Peer-to-Peer Middleware

Reference: George Coulouris, Jean Dollimore and Tim Kindberg, "Distributed Systems Concepts and Design", Fifth Edition, Pearson Education, 2012

Peer-to-Peer Middleware

- Peer To Peer Middleware
  - To provide mechanism to access data resources anywhere in network
  - **Functional Requirements :**
    - Simplify construction of services across many hosts in wide network
    - Add and remove resources at will
    - Add and remove new hosts at will
    - Interface to application programmers should be simple and independent of types of distributed resources

Peer-to-Peer Middleware

- Peer To Peer Middleware (contd)
  - **Non-Functional Requirements :**
    - Global Scalability
    - Load Balancing
    - Optimization for local interactions between neighboring peers
    - Accommodation to highly dynamic host availability
    - Security of data in an environment simplify construction of services across many hosts in wide network
    - Anonymity, deniability and resistance to censorship

Peer-to-Peer Middleware

- Peer To Peer Middleware (contd)
  - Global scalability, dynamic host availability and load sharing and balancing across large numbers of computers pose major design challenges.
  - Design of Middleware layer
  - Knowledge of locations of objects must be distributed throughout network
  - Use of replication to achieve this

Routing Overlays

- Routing Overlays
  - Responsible for locating nodes and objects
  - Implements a routing mechanism in the application layer
    - Separate from any other routing mechanisms such as IP routing
  - Ensures that any node can access any object by routing each request through a sequence of nodes
    - Exploits knowledge at each node to locate the destination

Routing Overlays

- GUIDs
  - ‘pure’ names or opaque identifiers
    - Reveal nothing about the locations of the objects
    - Building blocks for routing overlays
  - Computed from all or part of the state of the object using a function that deliver a value that is very likely to be unique. Uniqueness is then checked against all other GUIDs
  - Not human readable

Routing Overlays

- Tasks of a routing overlay
  - **Routing Request to Objects**: Client submits a request including the object GUID, routing overlay routes the request to a node at which a replica of the object resides
  - **Insertion of Objects**: A node introduces a new object by computing its GUID and announces it to the routing overlay
  - **Deletion of Objects**: Clients can remove an object
  - **Node addition and removal**: Nodes may join and leave the service

Routing Overlays

- Types of Routing Overlays
  - DHT – Distributed Hash Tables
  - DHT – GUIDs are stored based on the hash value
    - (128 bit hash value using SHA-1 algorithm)
  - DOLR – Distributed Object Location and Routing
  - DOLR is a layer over the DHT that maps GUIDs and address of nodes at which replicas of objects are located. put() and get() APIs
  - DOLR – GUIDs host address is notified using the Publish() operation

P2P Vs Distributed Processing

Peer to Peer

- Millions of nodes cooperate to achieve a common goal
- Distribution of resources are usually explicit but location not known
- WAN based
- Home rather than enterprise based resource servers
- No overall management – insecure resources
- Intermittent connectivity – probabilistic access
- Application level protocols

Distributed Processing

- Smaller numbers of nodes cooperate
- May provide a single virtual machine concept with transparent distribution
- Mostly LAN based
- Within a single or a few enterprises
- Managed system – resources can be more trusted
- Tries to provide deterministic access to resources
- Middleware protocols supporting application level interaction

No fundamental difference

Summary

- Napster – immutable data, unsophisticated routing
- Current – mutable data, routing overlays, sophisticated algorithms
- Internet or company intranet support
- Distributed Computing (SETI)

Summary

- **Benefits of Peer-to-Peer Systems**
  - Ability to exploit unused resources (storage, processing) in the host computers
  - Scalability to support large numbers of clients and hosts with load balancing of network links and host computer resources
  - Self-organizing properties of the middleware platforms reduces costs

Summary

- **Weaknesses of Peer-to-Peer Systems**
  - Costly for the storage of mutable data compared to trusted, centralized service
  - Can not yet guarantee anonymity to hosts

Thank You