Chapter 18: Peer-to-peer Computing and Overlay Graphs

Ajay Kshemkalyani and Mukesh Singhal

Distributed Computing: Principles, Algorithms, and Systems

Cambridge University Press

Characteristics

- P2P network: application-level organization of the network to flexibly share resources
- All nodes are equal; communication directly between peers (no client-server)
- Allow location of arbitrary objects; no DNS servers required
- Large combined storage, CPU power, other resources, without scalability costs
- Dynamic insertion and deletion of nodes, as well as of resources, at low cost

Features	Performance
self-organizing	large combined storage, CPU power, and resources
distributed control	fast search for machines and data objects
role symmetry for nodes	scalable
anonymity	efficient management of churn
naming mechanism	selection of geographically close servers
security, authentication, trust	redundancy in storage and paths

Table: Desirable characteristics and performance features of P2P systems.

Napster

Central server maintains a table with the following information of each registered client: (i) the client's address (IP) and port, and offered bandwidth, and (ii) information about the files that the client can allow to share.

- A client connects to a meta-server that assigns a lightly-loaded server.
- The client connects to the assigned server and forwards its query and identity.
- The server responds to the client with information about the users connected to it and the files they are sharing.
- On receiving the response from the server, the client chooses one of the users from whom to download a desired file. The address to enable the P2P connection between the client and the selected user is provided by the server to the client.

Users are generally anonymous to each other. The directory serves to provide the mapping from a particular host that contains the required content, to the IP address needed to download from it.

Structured and Unstructured Overlays

- Search for data and placement of data depends on P2P overlay (which can be thought of as being below the application level overlay)
- Search is data-centric, not host-centric
- Structured P2P overlays:
  - E.g., hypercube, mesh, de Bruijn graphs
  - rigid organizational principles for object storage and object search
- Unstructured P2P overlays:
  - Loose guidelines for object search and storage
  - Search mechanisms are ad-hoc, variants of flooding and random walk
- Object storage and search strategies are intricately linked to the overlay structure as well as to the data organization mechanisms.

Data indexing

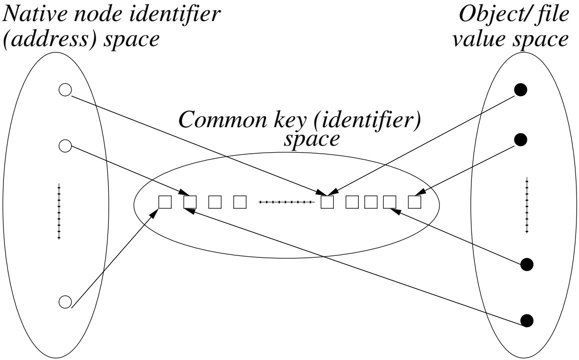
Data identified by indexing, which allows physical data independence from apps.

- Centralized indexing, e.g., versions of Napster, DNS
- Distributed indexing. Indexes to data scattered across peers. Access data through mechanisms such as Distributed Hash Tables (DHT). These differ in hash mapping, search algorithms, diameter for lookup, fault tolerance, churn resilience.
- Local indexing. Each peer indexes only the local objects. Remote objects need to be searched for. Typical DHT uses flat key space. Used commonly in unstructured overlays (E.g., Gnutella) along with flooding search or random walk search.

Another classification

- Semantic indexing - human readable, e.g., filename, keyword, database key. Supports keyword searches, range searches, approximate searches.
- Semantic-free indexing. Not human readable. Corresponds to index obtained by use of hash function.

Simple Distributed Hash Table scheme


Native node identifier (address) space
Object/ file value space
Common key (identifier) space

Mappings from node address space and object space in a simple DHT.

- Highly deterministic placement of files/data allows fast lookup.
- But file insertions/deletions under churn incurs some cost.
- Attribute search, range search, keyword search etc. not possible.