11CS2701 DISTRIBUTED SYSTEMS

Unit 5

PEER TO PEER AND DISTRIBUTED SHARED MEMORY

Peer-to-peer computing and overlay graphs : Introduction - Data indexing and overlays - Tapestry; Distributed shared memay', Absraction and advantages - memory consistency models - Lamport's Bakery Algorithm

* P2P Networks

→ P2P Network systems use an application-level organization of the network overlay for flexibly sharing resources a cross network wide computers.

→ Any node in a P2P network can act as a server to others, and at the same time, as a client

→ Communication and exchange of information is performed directly between participating peers & the relationships between nodes in the network are equal

→ P2P networks tend to share data from a large number of end users rather than one or more central machines

→ P2P file sharing can be done with Napster, Gnutella, Freeneh, Pastry, Chord, CAN

* Desirable Features of P2P Networks

1. self-organizing
2. distributed control
3. role symmetry for nodes
4. anonymity
5. naming mechanism
6. security, authentication, trust

* Performance Features of P2P systems

1. Large, combined storage, CPU power, and resources
2. Fast search for machines and data objects
3. scalable
4. efficient management churn - ongoing entry and exit of nodes, dynamic insertion & deletion of objects
5. Selection of geographically-close server
6. Redundancy in storage and paths

* Application Layer Overlays

→ A core mechanism in P2P networks is searching for data, and this mechanism depends on how the data & the network are organized.

→ P2P search search uses the P2P overlay, which is an application layer overlay, which is a logical graph among the peers that is used for object search & storage

→ The P2P overlay can be:

(i) Structured : hypercubes, meshes, butterfly networks, de Bruijn graphs

→ use rigid organizational principles based on the properties of the P2P overlay graph structure

(ii) Unstructured : no particular graph structure is used.

→ As there is no definite structure to the overlay graph, the search mechanisms are more ad-hoc, and use some form of flooding / random walk.

* Data Indexing and Overlays

→ The data in a P2P network can be identified by using indexing.

→ Data indexing allows the physical data independence from the applications

→ Indexing mechanisms include:

(i) Centralized

(ii) Distributed

(iii) Local

A. Centralized Indexing

→ entails the use of one or a few central servers to store indexes (references) to the data on many peers.

→ Some early P2P networks such as Napster used a central directory lookup.

B. Distributed Indexing

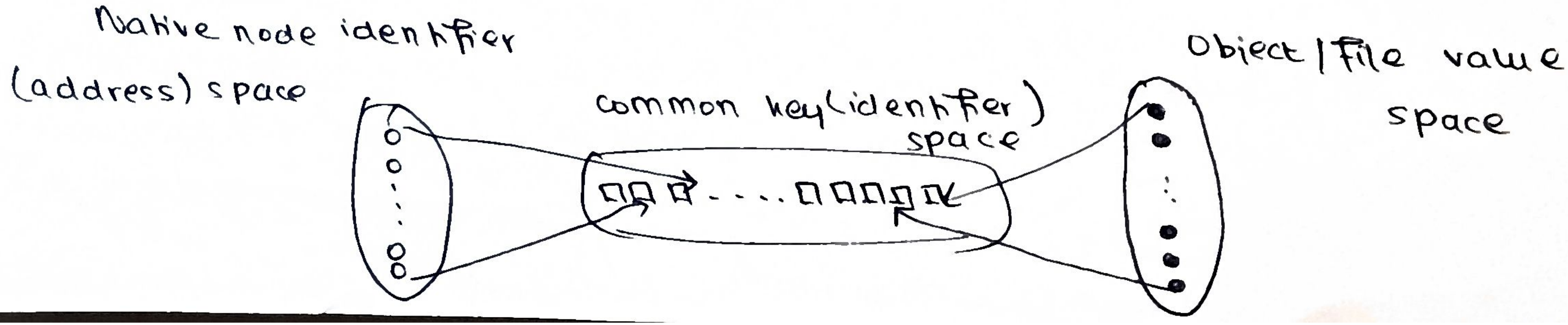
→ Involves the indexes to the objects at various peers being scattered across different peers throughout the P2P network.

→ In order to access the indexes, a structure is used in the P2P overlay to access the indexes.

→ Distributed indexing is a challenging scheme - and the most notable algorithm used is the distributed hash table (DHT).

→ Various DHT schemes differ in hash mapping, search algorithms, diameter for lookup, search diameter, fault - tolerance, ~~resistan~~ resilience to churn.

→ In DHT, the node address is mapped to a logical identifier using a consistent hash function. This is done with a key space.


Native node identifier (address) space
common key (identifier) space
Object / File value space

* ~~Alternative / way to classify indexing / mechanisms~~

C. Local Indexing

→ requires each peer to index only the local data objects and remote objects need to be searched for.

→ This form of indexing is typically used in unstructured overlays in conjunction with flooding search or random walk search.

→ Gnutella uses local indexing

* Alternative way to classify indexing mechanisms

① Semantic Index Mechanism – human readable, eg. a doc name, keyword or database key

– supports keyword searches, range searches & approximate searches

② Semantic Free Index Mechanism – not human readable, corresponds to the index obtained by a hash mechanism. (DHT schemes)

→ searches not supported

* Structured and Unstructured Overlays for Distributed Indexing

① Structured Overlays

→ P2P network topology has a definite structure, and the placement of files and data in this network is highly deterministic

→ allows for a very fast & deterministic lookup for queries

→ called lookup systems - use a hash table - maps key to values

→ Mapping is based on a single characteristic of the file (name/length/predetermined fn. computed on the file)

→ However, range queries, attribute queries & exact keyword queries cannot be handled directly

→ Because of the tight coupling of the regular overlay & the rigid mapping - file insertions, deletions may incur some overhead.

② Unstructured Overlays

→ P2P network topology does not have any particular controlled topology - no control over where files/data is placed.

→ Each peer indexes only its local data objects.

→ Search for a file may entail high msg. overhead and high delays.

→ Complex queries are supported, since criteria for search can be arbitrary

→ Some topologies which may emerge are : Power law random graph, normal random graph

* Properties of Unstructured Overlays

1. Queries may take a long time to find

2. Msg. overhead of a query search may be very high

3. However, can perform exact keyword queries, range queries, complex queries - search can capture the semantics of the data

4. Can accommodate high churn (rapid joining & departure of nodes)

The advantages exist only if certain conditions are satisfied:

(i) some degree of data duplication in the network

(ii) users are satisfied with a best-effort search.

(iii) The network is not so large that it leads to scalabili problems during the search process

* Gnutella

→ uses a fully decentralized architecture

→ nodes index only their local content.

→ overly topology can be arbitrary as nodes join & leave.

→ A node joins the Gnutella network by forming a connection to some nodes found in standard Gnutella - directory-like databases

→ Users communicate with each other, performing the role of both server and client (called servant)

Message types used by Gnutella

search

(TTL)
Flood +

Random

walk

(i) Ping - used to discover hosts

- allows a new host to introduce itself

(ii) Pong - response to pings

- has the port, IP address of the responder + some info. about the amount of data (no. and size of files) that node can make available

(iii) Query - contain a search string & the minimum download speed required of the potential responder

- these msgs are flooded in the network

(iv) Query Hit - sent as response if a node receiving a Query detects a local match in response to a query

- Query Hit contains the port and address (IP), speed, no. of files etc

- The path traced by a Query is recorded in the msg, Query Hit follows the same path in reverse.