9 Peer-to-peer systems

Saturday, January 12, 2019 2:04 PM

Describe peer-to-peer systems. Discuss some of the design issues in peer-to-peer systems.

[Coulouris et al. chapter 10]

- Peers are identical
- Peers are both senders and recipients
- Communication is indirect: space uncoupled

- Globally scalable
- Load balancing
- Optimization for local interactions between neighbors

- Routing overlay: distributed algorithm responsible for locating nodes and objects
  - Routes requests to objects
  - Handle new object insertion/deletion
  - Handle node addition / removal
- Replicas of objects may be stored to ensure high availability

- Objects are identified by GUID: unique
- Tracked in DHT: distributed hash table
- New object is inserted: first we compute a GUID, then search for it in DHT, so it's not a dupe
- GUID is not human readable

- Where do we send the message?

Without a central tracker

- Know at least 1 peer (preferably more)
- Addressee: GUID (globally unique identifier); hexadecimal

Algorithm: Pastry

- A peer has a table of some peers

- Needs GUID (rest of the algorithms only need a reference)
- Treats GUID table as a csorted circle

- If objects / nodes leave in the meantime, leaf set that contains the faulty GUID needs to be repaired (updated)
- Nodes send heartbeat messages to neighbouring nodes in their leaf sets, to signal they are alive

Binary search

- Starts in the middle of a sorted list, throws away irrelevant half, repeats for the remainder
- Steps: log(16)n, rounded up

N	Log(16)n
16	1
256	2
4096	3

65536	4
…	…

Algorithm: Flooding

- The peer sends the message to all the peers it knows, and so on
- Breaks down in larger networks

Algorithm: Random walk

- Peer sends the message to a fixed number of randomly selected peers

Algorithm: Gossip

- Peer sends to a fixed percentage of peers

Algorithm: Ultrapeers

- Skype