DC Previous Year Q&A

Comprehensive

Q. What are all the types of indexing in P2P networking? Explain each with an example.

Indexing in P2P networking is carried out through three main approaches: centralised indexing, flooding-based querying, and distributed indexing using DHTs.

Centralised indexing involves a single server maintaining a global index of peers and the content they host. A querying peer sends a lookup request to the central server and receives a list of peers containing the requested file. An example is the original Napster, where the central index server located file metadata while data transfer occurred directly between peers.

Flooding-based indexing works without a central server. Each peer forwards a query to its neighbours, who continue forwarding until the file is located or a hop limit is reached. This method is used in Gnutella, where each node propagates search requests through the overlay network.

Distributed Hash Table (DHT) indexing distributes the index across all nodes using a structured lookup algorithm. Each node stores responsibility for a portion of the keyspace. A peer performs lookups by routing through the overlay to the node responsible for the key. Examples include Chord and Kademlia, where each object key maps to a specific peer.

Q. Explain the concept of Quorum used in Maekawa's algorithm. Why does Maekawa's algorithm use a RELEASE message?

In Maekawa's mutual exclusion algorithm, each node is associated with a quorum, which is a subset of nodes whose permission is required before entering the critical section. Quorums are constructed so that every pair of quorums intersects, ensuring at least one common node between any two quorums. This intersection guarantees that two processes cannot simultaneously obtain all permissions from their respective quorums, thereby enforcing mutual exclusion.

A RELEASE message is required so that nodes in the quorum know when a requesting process has completed its critical section. Since each node grants only one outstanding request at a time, the RELEASE message frees that node to grant permission to queued or future requests. Without explicit release, nodes would remain indefinitely locked, leading to starvation and deadlock.

Q. In a distributed network with eight nodes participating need to come to a common agreement with some expected minimum failures. What kind of algorithm can be implemented to suit the above scenario and how many failures are acceptable? Give justification for the number of acceptable failures and explain appropriate algorithm.

PTO

A system of eight nodes that must reach agreement in the presence of failures can use a Byzantine Agreement algorithm, as it permits nodes to agree on a value even when some nodes behave incorrectly. Byzantine agreement protocols can tolerate up to f failures only if the total number of nodes satisfies $n \geq 3f + 1$. With $n = 8$, the maximum number of tolerable failures is $f = 2$, because $3f + 1 = 7$, which is within the available eight nodes.

An appropriate algorithm is the Byzantine Generals Algorithm, in which each node sends its proposed value to every other node, and nodes perform recursive message exchange across rounds so that faulty nodes are outvoted by the consistent behaviour of correct ones. After collecting messages, each node applies a decision function that selects the majority value among the received messages. With fewer than one-third faulty nodes, all correct nodes reach the same final decision despite malicious or inconsistent behaviour from faulty peers.

Q. Explain CAP theorem and justify the statement “The CAP theorem says that a distributed system can deliver only two of three desired characteristics”.

The CAP theorem states that in the presence of a network partition, a distributed system must choose between consistency and availability, but cannot guarantee both simultaneously. Consistency requires all nodes to return the same and most recent value after an update. Availability requires that every request receives a timely response even if some nodes are unreachable. Partition tolerance requires the system to continue functioning even if communication between parts of the system is disrupted.

When a partition occurs, nodes cannot communicate reliably. If the system prioritises consistency, it must delay responses until nodes synchronise, thereby losing availability. If it prioritises availability, nodes must respond immediately even if they hold divergent data, thereby losing consistency. Partition tolerance cannot be sacrificed in a distributed environment because failures and message loss are unavoidable. Hence, only two properties may be offered at any time, which justifies the CAP theorem’s statement.

Q. What kind of communication and computing technology is best suited for an environment where there is no centralized server nor a dedicated distributed set up also the environment is highly dynamic in nature with no dedicated resource allocation? Give a day-to-day example of the same and explain in detail.

A peer-to-peer (P2P) communication and computing model is suited for an environment lacking a central server, lacking a fixed distributed infrastructure and exhibiting high dynamism. In a P2P system, each node acts both as a client and a server, and resources such as compute, storage or bandwidth are contributed directly by the participating nodes. Since there is no requirement for a stable central coordinator, the system adapts naturally to changing membership, making it appropriate for dynamic environments.

A day-to-day example is file sharing over P2P networks. Nodes join and leave at any time without disrupting the system. Each participant stores a portion of files and responds to queries from others. Requests for content are routed through available peers, and the system scales as more nodes participate. This example illustrates decentralised operation, lack of dedicated infrastructure and reliance on shared resources contributed voluntarily by peers, matching the characteristics described in the question.