CS 8603 - Distributed Systems   III year / VI Sem
CSE
UNIT V

1) Explain in detail about Peer to Peer Computing?
* Peer to Peer systems are used for sharing resources like files, Images etc that are reserved across the computers.
* It makes use of network overlay present in the application level organization.
* Each node in the network can work as both server and client at the same time.
* P2P networks focus on data exchange between wide range of users rather than data exchange between user and central machines.
* So the relation between the nodes in a P2P network will be equal to the data exchange and communication occurring between the participating peers and users.
(eg) Napster, Free Net, Gnutella, Pastry, Chord and CAN.

Features:
1) It is self organizing and makes use of large combined storage, CPU power and resources.
2) It has distributed control which performs faster search for machines and data objects.
3) It is scalable and offer role symmetry for nodes.
4) It has anonymity due to the efficiency in managing achum.
5) It naming mechanism selects geographically close servers.
6) It provides security, authentication and trust due to the redundancy in storage and paths.
The most popular P2P system is Napster

Structured overlay:-
They use few organizational Principles on the basis of Properties of P2P for storing and searching algorithms.

Unstructured overlay:- They use loose guidelines for storing object. Since, there is no proper structure defined for the overlay graph the search mechanisms are becoming even more difficult. ~~&~~ It uses some typical strategies for searching the objects. such as flooding or random walk strategies.
~~&~~ Therefore it can be said that the search strategie are directly related to the overlay structure and data organization mechanism.

Steps used for searching content and to find the node using the content has to be downloaded:-

1. A meta server serves the client's query by connecting it to the less loaded server from group servers.
2. As soon as the client is connected to the assigned server it sends its query along with the identity.
3. The server sends the response message which contain the information about others users connected and also it contains the information about the data that is being shared among the users.
4. The client then picks one of the other user and downloads the required file from it. The availablity of P2P connection between the selected user and the client is enabled by the server

2) Explain the various data Indexing methods, Also discuss in brief about various data overlays present in P2P network?

Data Indexing:- The data Indexing is used to identify the data present in the peer to peer network. It supports physical data independence. It provides protection to the data from user applications.

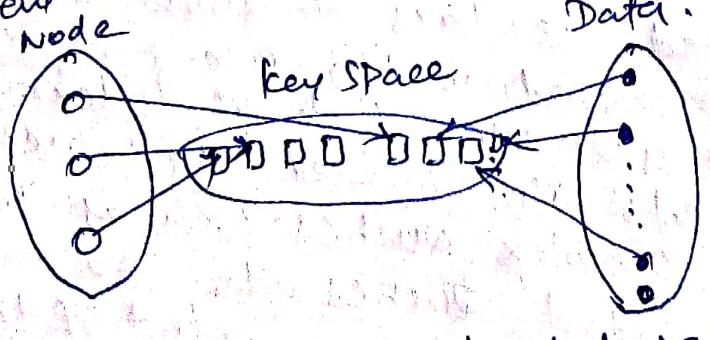
3 data Indexing mechanisms:- 1) Centralized Indexing. 2) Distributed Indexing 3) Local Indexing

1. Centralized Indexing makes use of more than one central servers for storing the indexes of data present on many servers. It maintains a central directory lookup which is used by the DNS lookup and Napster.

2) Distributed Indexing uses indexes for various objects present at different peers in the peer to peer networks. A structure is used in the P2P overlay for accessing these indexes. one of the mechanism used in distributed indexing is distributed hash table. There exists DHT schemes where in every scheme differs in hash mapping, search algorithms, fault tolerance etc.

It uses key space to map the network nodes and data values.

In order to map an address with a logical identifier a consistent has functions is used in the key space.



mapping from node to data value.

Local Indexing:- This Indexing uses every Peer to Index all the objects which are used for Searching. This Indexing is mostly used in unstructured overlays along with flooding search (or) random walk search. Gnutella is one of the mechanism which uses local Indexing.

Semantic Indexing Mechanism:- This Indexing Mechanism uses semantic Indexes which are human readable and understandable. The semantic Indexes can be document name, key word or database key. These Indexes support key word searches, range searches and approximate searches.

Semantic Free Indexing mechanism:- This Indexing Mechanism uses Indexes which cannot be under stood and read by the humans. These Indexes does not support key word searches, range searches and approximate searches. It is based on the Index that obtained by a hashing mechanism

Structured overlay in P2P networks consists of a definite structure. The files are stored in the network. This mapping algorithmic mapping in the accurate lookup inorder to supports fast and data. The mapping solve the Queries for the hashing which helps to used in this context is retrieve the files easily.

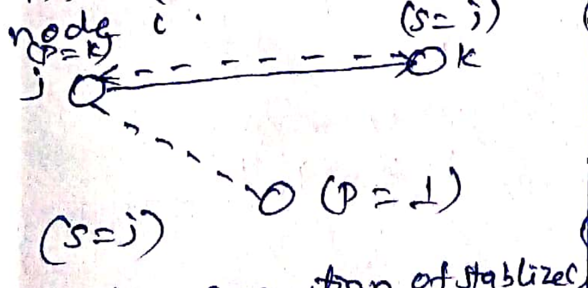
Unstructured overlays doesnot have definite structure. The files stored in this are not controlled. The overlay makes use of local Indexing such that only local objects are Indexed. This overlay can be used for complex Queries such as range Queries attribute based Queries etc. The main disadvantage is that Queries consume a lot of time for searching a file and some times it even fails to search a file even if it is present in the network

(4)

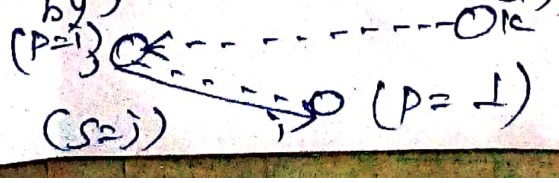
3) Explain about node joins, node failures and departures.

1) Node joins a new ring is created when node i executes "Create - New - Ring". This function creates a new ring with a singleton node.
The function "Join - Ring (j)" is invoked by node i. in order to join a ring that has node j. This node j helps in locating i's successor on the logical name and notifies 'i' about its successor.
1. The successor node i is required to update its predecessor entry to i.
2. The predecessor must revise its successor field to the node i.
3. The node i should identify its predecessor.
4. At node i, there is a need for a finger table.
5. Every node must keep their FINGER tables updated So that the presence of 'i' can be recorded.
These actions are accomplished only when each node invokes the following procedures periodically.
1. stablize ()
2. Check - Predecessor ()
3. Fix fingers ().

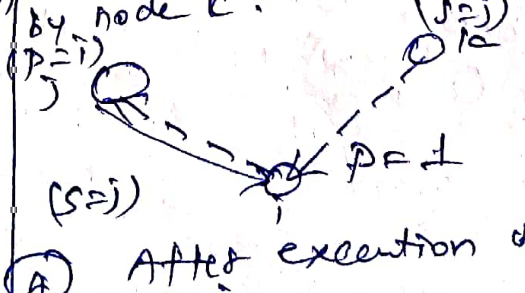
1) After execution of ring (.) by node i.



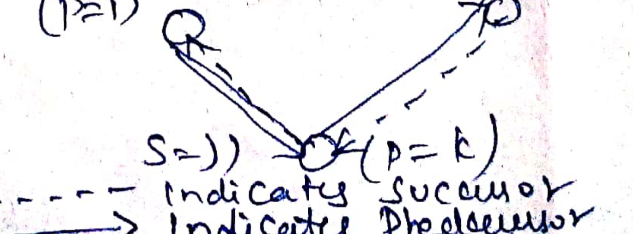
2) At the execution of stablize() by i and execution of notify(i) by j.



3) After the execution of stablize() by node k.



4) After execution of notify(k) by node i.



5) ←---- Indicates successor
⟶ Indicates Predecessor

Node failures and Departures:

If the node j fails suddenly, then its successor node i on ring detects the failure when it is busy with the execution of Check Predecessor() If other node k causes i to execute notify(k) then the process i updates its predecessor field. This happens only when the variable of k's successor is j

For doing so, it requires the predecessor of failed node to identify that its respective successor is failed and thus it gets a new functioning successor.

3) Complexity of Chord Distributed Hash tables:

1. The finger table size is $\log(n) \leq m$
2. In a chord network of n nodes, each node having $(1+\epsilon)k/n$ keys with high probability. Here k is the No. of keys. Since the result validity depends on conflict free mappings and randomness of hash function used, high probability clause is needed.
3. The look up time in average case is $1/2 \log(n)$.
4. In a chord network the search for successor in locate successor with n nodes needs time complexity of $O\log(n)$ with high probablity.

4) Discuss about the overview of CAN and its Initialization?

A CAN indexing method that maps objects to their respective locations in the network. It is scalable efficient and can be used for large scale storage management systems and wide area name resolution service which decouple naming scheme and resolution name.

3 operations namely 1) Insertion of (key, value) tuples, 2) searching of (key, value) tuples and 3) Deletion of (key, value) tuples.

* An Ideal CAN design is generally scalable, fault tolerant distributed and independent of the naming structure be implemented at the application layer.
* It can also be self healing and self organizing.
* It is said to be d-dimensional logical cartesian coordinate
* A CAN is a d-dimensional logical d-torus topology. arranged as a virtual overlay space that was is a d-dimensional
* This topology mesh with wrap around.

~~CAN~~ DESIGN :-

1. To setup a CAN virtual coordinate space and dividing it between nodes as soon as they join CAN.
2. To perform routing in virtual coordinated space for locating the node that is allocated to the region that consists the point $\vec{p}$.
3. To maintain the CAN, due to the node failures and departures.

Initialization of CAN :-

1) Every CAN has a specific DNS name which is used for mapping IP addresses of one or more bootstrap nodes of that CAN.

This boots strap node is used to trace the list of nodes which are Participating in CAN.

3) CAN can be Joined by the joiner node which Queries a boot strap node through DNS look up. The boot strap node gives response with IP addresses of selected nodes that are Participating in the CAN.

4) The Joiner node selects the Point P randomly in the coordinated space. It forwards a request in the CAN for allocating to one of the nodes in the CAN. Then the receiver routes a region that has P. Then the region that has request to owner old owner (P) of region the P using CAN routing Algorithm.

5) The owner node old owner (P) divides the region into half and allocates one half to the joiner node. This division is performed using the ordering of all dimensions, this method is used for merging regions. The tuples of (k,v) where key k joiner, are also transferred to the node joiner.

6) The joiner node finds the IP addresses of its respective neighbours from the old owner (P). The old owner (P) and a subset of old owner (P) neighbours are considered as neighbours. The old owner updates its neighbours set. The old owner and new joiner informs their neighbours about the space allocation changes such that they have proper information about the neighbours and this routing can be done correctly. Every node must forward an immediate update of its allocated region and then followed by the periodic HEART BEAT refresh message to all the neighbours.

5) Discuss about tapestry also explain about overlay and routing?

Tapestry is a distributed hash table which provides routing of messages depending up on the GUID's of the resources.

It operates same like pastry. Participants in the network can publish objects by periodically routing a publish message

The owners of the resources have the authority to store the resources.

Copies of all the resources hold same GUID as the original one. Hence many copies of the resources are placed near to recent users to reduce latency, fault tolerance, to avoid failures and to balance the load.

Tapestry uses SHA-1 to produce 160 bit identifier which is used to represent objects and nodes.

Each node is assigned to a unique node ID which refers to node or GUID's that represent objects and are used for performing routing operations.

Participants in the network can publish objects by periodically routing a publish message toward the root node.

Each node along the path stores a pointer mapping the object. multiple servers can publish pointers to the same object. The redundant links are prioritized by latency and or locality.

objects are located by routing a message towards the root of the object.

Each node along the path checks the mapping and redirects the request appropriately.

The effect of routing is convergence of near by paths heading to the same destination.

overlay and Routing:- A tapstery uses a common identifier space indicated using m-bit values which is expressed in a hexadecimal notation. (ie base 16)

* This tapstery assigns m to be 160 ie m=160

* Every Identifier is mapped to a group of unique nodes.

* These nodes are present in the network and called as Identifiers rootset, expressed as O_n.

* The $|O_n|$ is a small constant and if $|O_{n_2}| > 1$ then it increases the fault tolerance.

If node v exists then Vid = O_{n_2} such that v is denoted as the root of identifier O_n and if node V is absent then deterministic role is used for identifying other unique nodes that shares the largest common prefix with Identifier O_n and hence acts as a surrogate root.

Algorithm:

Input: Integer table [1 to $\log_b 2^m$, 1 to b].

```
Input: Integer table [1 to log_b 2^m, 1 to b] is
1: The nexthop ie (i, On = d, od2 O.... od log_b m) is
   exceuted at Vid to route the On. Here i is the level of
   the table (1+ length of longest Common Prefix).
2. while table [1, di] = ⊥
   do
3.   di ← (di+1) mod b;
4)   If table [i, di] = V
     then
5)     return (NEXT-HOP (i+1, On))
6    else
     return (Table[i, di])].
```

The algorithm is invoked as NEXT HOP (1, On) at the source node. For determining the hop i of the the route, node U executes the function that holds the Prefix i-1 digits in common with O_n.

6) Disuss in detail about the distributed shared memory abstraction along with the design issues of DSM?

* In a distributed shared memory architecture, the memory is distributed equally across all processors available in the architecture

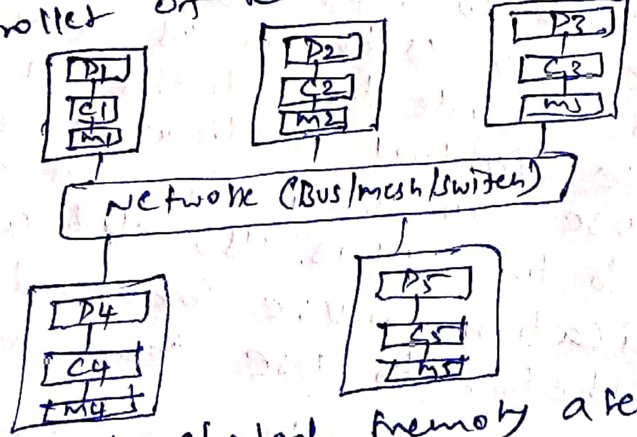
* Each Process is referred to as a node. The nodes are connected to each other via network.

* Since memory is distributed across the processors, the controller present in individual node decides whether the memory should be local or remote based on the address.

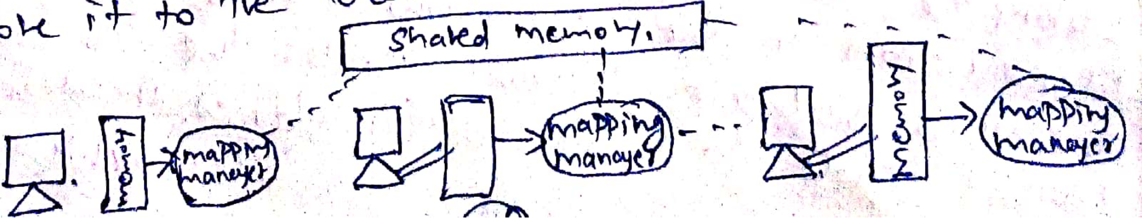
* The local memory is specific to the processor and doesnot require transmission of explicit messages for retrieving the data.

* If the processors want to communicate with a remote memory, then it should explicitly send a message which is similar to a Remote procedure call to the controller of remote memory.

P = Processor
C = cache
M = memory



In distributed shared memory architecture every individual processor has its own independent cache memories due to which there is a possibility of cache coherence problem to be encountered. Despite of being uncatachable the shared data can be made catchable by a software that copies the data from the shared part of the address space and store it to the local private portion of cached address space.



In the context of DSM data can be easily accessed by the programs in the shared memory.

DSM facilitates the system by transferring their data between primary memory and secondary memory and also primary memories of all distinct nodes.

Each node can have its individual data stored in the shared memory. Since that data can be transferred from one node to another node, the ownership of data will change accordingly.

When a program wants to access the data in shared memory, a mapping manager is used to provide mapping between the shared memory address and the physical address physical memory address and the physical address irrespective of its locations.

Issues:

1) It is difficult to decide which semantic is suitable for accessing the shared objects. If the semantics are not specified clearly then the programmer might face difficulties in coding the program.

2) The method used to implement the semantics is the best possible way is known as replication. It is difficult to decide the degree of replication, whether to use partial replication at some of the sites or to use full replication at each site. In addition to this it is also difficult to decide which replication to use for read and write operations. whether to use read replication

3) It is difficult to decide the locations for replication when full replication is not used.

4) It is difficult to decide the location for remote data that is used by the applications.

5. During the implementation of current semantics, if there is reduction in the communications delay and the messages that are present under the covers then it might leads to some difficulties in designing the DSM system

7) Explain in detail about Lamports Bakey Algorithm?.

* The purpose is to find the solution for the critical section problem using the concept of mutual exclusion.

* The Algorithm intends to improve the robustness of the processes that handle multiple threads.

* multiple threads can be accessed simultaneously on the same resource.

* Data sets corrupted if two or more threads perform a write operation on the same memory location or if read operation is performed before write operation is finished.

Hence the concurrent execution of threads can be prevented using lamports bakery algorithm.

* Lamport visualized a bakery in which every customer entering in it is provided with a unique number.

* As the no. of customers increases, the number given to them also increase consecutively.

* A global counter is maintained to display the No. of current customer that is being served

* when the baker finished serving, the next number of the customer that was waiting in a queue is displayed.

Here the lamports analogy is slightly modified

* Here the customers are considered as threads and these threads are identified by the letter 't'.

* It is possible that similar no. of can be provided with more than one thread.

Therefore the difficulty arises in Identifying them
In order to overcome this priority is set to.
each thread.
A lower value of 't' has a higher Priority whereas higher value of 't' has a lower Priority.
Hence these threads with higher Priority are permitted to enter first time in to the critical section.
critical section is a Part of code that must be executed by only one thread at a time. This is similar to the bakery analogy, where other customers wait untill the baker finishes serving the current customer.

<u>Algorithm:</u>

```
    int selection [Procs n] = {false};
    int num [Procs _n] = {0};
    selection [t] = true;
    num [t] = max (num) ++;
    Selection [t] = false;
    for (i=0 ; i< Procs_n ; i++).
    {
    while (selection [i]);
    while (num [i] != 0 && (num [i] < num [t] || num [i]
                                      == num [t] &&
                                           i < t));
    }
    /* critical section */.
    num (t) = 0;
```