Assignment(Lecture 9 & 10) : Distributed Operating System Principles (COP5615)

Name: Vikas Chaubey**, UFID:** 3511 5826**, Email:** vikas.chaubey@ufl.edu

1) Hierarchical location services (HLS):
HLS is based on position-based routing, to enable position-based routing a node must be able to discover another node. this is implemented using location service. The main idea behind Hierarchical location services is to build search tree on large scale in which incorporated network is divided into many domains and each node in the tree represent a separate domain.

Hierarchical approaches (Tree Organization): In this approach the root node of the tree has information about all entities and entities address is stored in a intermediate node and also in leaf nodes, it will contain pointer to a child node only if the subtree connected to child contains the address of the entity.

2) Name space: A namespace can be defined as a declarative region in general that can contain certain signs or names which are used to identify objects. In a naming graph the leaf nodes represent entities, and a directory node is a node which refers to other nodes. It contains pointers to refer to other nodes. A directory node can point to multiple child nodes in subtree by having multiple pointers. Hence directory node can work as Name space as it can contain different kind of attributes (address, names, entity's type etc.).

3)Name space implementation: The name space implementation is done by dividing the name resolution process and management among different machines, this could be done by distributing nodes present in the naming graph.

4) Name Resolution: To do resolution of names we need to start with some node and follow the pointers contained within this node. In order to resolve names, we can use following techniques:

a) **Name linking:** Name linking could be done either by hard linking which means that a name will be resolved by following a specific path in the naming graph from one node to another. Other way is to do soft linking in which a node can contain name of other nodes hence in this strategy we can resolve a name by reading the node's content and then in this content we can obtain information regarding the destination node we want to go to.

b) **Mounting:** Mounting is used to merge two different name spaces. In this process a node identifier in the current name space is associated with a node identifier in the foreign name space. The node in the current name space which contains node identifier for foreign name space is called mount point while the connecting node in the foreign name is called mounting point where the name resolution begins.

5) Different systems of Name resolution: Domain Name server (DNS) is a good example of Hierarchical organized name space in which each subtree is a domain and a path to domains root node is domain name.

a) **Iterative Name resolution:** In this system of name resolution the components are resolved one by one that means first domain at top level is resolved and then we move one level down each time and determine and resolve other subdomains on by one.

b) **Recursive Name Resolution:** In this system the root node in the naming graph resolves and pass the results to the next node it finds, and then the next node start resolving the name further this continues until all components are resolved.

6) Directory Services and LDAP: There are many cases where it is much easier and more convenient to name entities and search entities based on their attributes, this is how a directory service functions. However, since this method involves looking up requested attributes against the actual attributes, the operations is very expensive because all entities are inspected in order to identify the matching one.

In order to solve this issue, the directory services could be implemented as database with traditional naming system, to do that Lightweight Directory Access protocol (LDAP) is used. **Lightweight Directory Access protocol (LDAP)** is a collection of all entries, and each record is named uniquely which is called relative distinguished name , this name helps in look up.

Lecture 10 :

1) Clock Synchronization: The purpose of clock synchronization is to keep the deviation between time from two different machines within certain limit or minimum. If clocks are perfectly synchronized in two different machines, then synchronization of two system becomes easy however in reality perfectly synchronized clocks do not exist. Also, it's an expensive affair to create synced clocks at nano seconds of precision. Universal coordinated time (UTC) is determined by the cesium 133 atom clocks which is very accurate,

a) **Physical clock:** Physical clock is a physical process which incorporated methods to measure the process itself in order to calculate the passage of time.
for example: cesium 133 atom clocks for UTC, quartz clock in processors in machines.

b) Logical clocks: Physical clocks are not accurate hence logical clocks are used create a partial or total ordering of events, Logical clocks is a type of mechanism which allows the capture of causal and chronological relationships in a distributed system. In a distributed system a physical clock might not be present in order to synchronize different node operations hence in this case a logical clock is used. In distributed systems a logical clock allows global ordering on events from different processes.

In order to maintain a global view of system behavior, a timestamp is attached to each event which has following properties:
1) within a process if event a happened before event b, then a--> b, then we say C(a) < C(b) where "<" represents happened before relationship.
2) if a sends a message to b(recipient) then also we say C(a)< c(b)
3) The relation is transitive, a < b and b < c => a < c

Lamport Algorithm lets define the distributed system as a set of processes. And each process is modeled as sequence of events. In lamport algorithm in a process when a new event take place the counter of the process is incremented by one. When the process sends a message then the message timestamp is set according to the counter of the process. Also when the message is received by another process then that second process sets its counter to max (second process counter, timestamp) , whichever is maximum. Lamport clocks can guarantee that if a < b then C(a) < C(b). However, it can't guarantee, that if C(a) < C(b) then event a happened before b.

c) Vector Clock: In distributed system vector clock is an algorithm which is used to generate partial ordering of events. The problem with Lamport Timestamps is that they can't tell if events are concurrent or not. This problem is solved by Vector Clocks.

clock synchronization algorithms:
1) Network time protocol (NTP): Network time protocol is used to synchronize computer clock times in a network. In this protocol a client node or machine which wants to correct its clock time sends a time request exchange to NTP server. NTP server responds to this request. Using this response, the client is able to calculate its link delay and it then adjusts the clock time accordingly to match the clock time of server. In order to set the clock correctly many time request exchanges are done between client and server.

2) The Berkeley algorithm: It is an algorithm which is used in a distributed system in order to do synchronization among clocks of different machines. The algorithm assumes that no machine in the distributed system has an accurate time source or access to UTC server. In this algorithm a master node controls the clock synchronization of whole system. A master node periodically sends requests to slave nodes and obtains their clock time. Each slave node responds to the request and send their respective clock times to the master node. After receiving these clock times from slave nodes master node calculates average time difference among all slave clock times including master node's clock time. Then this averaged time difference is added to the master node's clock time and this new time is broadcasted over the network.