Assignment 3: Distributed Operating System Principles (COP5615)

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

The communication among the machines in the distributed cluster happens through sockets, here are some important operations involved in communication between two machines:

1) client/Server Operations:

socket: It is responsible for creating the communication endpoints among the communicating machines
bind: this operation binds the local address to the socket
listen defines what could be the maximum number of pending requests
accept: this operation clocks caller until a connection request comes.

connect: operation which tries to establish connection with server
send: This operation is responsible for sending the message (data)
receive: This operation is responsible for receive the message (data)
close: it is responsible to close the connection once the communication completes.

2) message passing interface :

Message passing interface is a communication protocol for communication among different machines in a distributed and parallel computing system. Message passing interface is not a programming language but a interface that means it is a specification allows to send Async messages among the machines. This specifications defines clearly the set of routine which could be implemented by different end systems. These end systems can build upon these low level routines and can come up with high level routines for distributed communication environment. This interface allows users to call the routines from different programming languages. Message passing interface is highly portable because it has been implemented for almost all distributed memory architecture. The other aadvantage of message passing interface is speed because it has been optimized for almost all hardware system on which it runs.

Advantages of MPI:

1) Portability
2) Speed
3) good performance
4) Flexibility

3) Message Queueing model:

Message queues smoothens the communication between the sender and receiver. Message queues allow to save the received messages and process them in FIFO bassets is especially very important when sender is too fast to send the message requests in that case the receiver might get overwhelmed and might not process the messages at the same rate they are received hence message queues can help the receiver to store these messages in queue temporarily till they are processed by the receiver.TCP/IP model also uses queues.

Four combinations of loosely coupled communication:
1) Sender is running, receiver is passive (Ex. Emails) - message queues are supposed to be the persistent message queues
2) sender is passive, receiver is running ()
3) sender is passive and receiver is passive
4) sender is running, receiver is running

Message queueing Model Primitives:
1) Put : append a message in the queue
2) Get : remove the first message but block untill the queue is empty
3) Poll : remove the first message but never block
4) notify : call a handler is a new message is put in the queue

Message Broker:
Message brokers handle the differences or heterogenety in a message queueing system.following are the tasks of a message broker :
1) conversion of incoming messages to the target format.
2) acts a sapplication gateway.
3) providing routing capabilities.
Example: IBM webshephere MQ

Communication in the distributed system:

1) Flooding:
In a distributed network if a machine wants to send a message to all other machines in the network and the limitation is that not all the machines are connected to each other then in that case one simple way to spread the message is via flooding, in this mechanism a machine send a message to all its neighbours and all those recipients then further spred the message to their neighbours and so on.But this can cause problems as the messages being transmitted wil never die and might come back to the origin machine.Without TTL attached to messages flooding is a big problem.

2) Tree based Communication: In this type of communication the message system has tree based hierarchy , the top most node send the message to its children and then children propate the message further down aong their chidlren.

3) Gosspip Communication: Gossip Protocol is a communication protocol, it is process computer to computer communication that works on the same principle as how information is shared on social networks.It works in similar fashion as rumours are spred from one person to another.

Three main types of Gosspi comuunication protocols are :
1) Dissemination Protocols : This is a rumour mongering protocol, In this protocol the information is spread in the network using rumours, this protocol can flood the network with the gossip messages which can produce the worst case load in the network.

2) antientropy propagation model : In distributed systems which are not strongly coordinated ,this model can facilitate consistency and synchronization. This protocol is used to repair the data and also for missing records.

3) Protocols that can compute aggregates: These protocolas can sample information from few nodes of the network to compute an aggregate of the whole network.These local values are combined to obtain system wide values.
example : Push Sum protocol

Naming:
In a distributed system Names are used to denote entities .An entity in a distributed system could be anything. Entities are accessed at an access point. Access points are named by means of an address.if an address can be reassigned to a different entity, we cannot use an address as an identifier.
Example:
1) A telephone is an access point for a person, the telephone number becomes the address of the person.But a person can have multiple phone numbers.

Identifiers:
Since the entities in a distributed system can have multiple access points hence it becomes difficult to use address of access points as regular names for the entites.hence an identifier is used to uniqely identify an entity.An identifier can refer to at most one entity also the identifier always refers to the same entity.
Example : Like a person always has the same social security number (SSN)

identifiers could be conveniently used to uniquely represent entities. However many times identifiers are just randomly generated bit strings, these bit strings are unstructured, or flat names. they do not contain any information regarding location of access points of associated entity.hence these are called flat names.

How to identify and entity using flatnames:

1) Broad casting : This method works only with a distributed system in a local area network, a message containing the identifier of the entity is broadcast to each machine.then each machine checks whether it has the entity or not. machines which have the access point information for the entity reply back with the information.

2) Forwarding pointers : when an entity moves from one node to another, it leaves behind its new location reference. when an entity is located, for example by using a traditional naming service, By following the chain of forwarding pointers the a client can look up the current address.

Home based Approaches:

broadcasting and forwarding pointers methods have scalability problems. these approaches are difficult to implement efficiently in largescale networks ,also forwarding pointers can create performance problems and can rsult into broken links.A popular approach to supporting mobile entities in large-scale networks is to introduce a home location, which keeps track of the current location of an entity.This approach is robust and Special techniques may be applied for network or process failures.In this approach Entity's home address is registered at a naming service then the home registers the foreign address of the entity , client contacts the home first, and then continues with foreign location.

Problems with home based appriaches:

1) Home address needs to be supported for entity's lifetime
2) Home address is fixed
3) Poor scalability

2) Distributed ahsh tables:

A distributed hash table (DHT) is a distributed system that provides a lookup service similar to a hash table: In DHT key-value pairs are saved, and any node in the distributed system can retriev the value associated with that key form DHT.

Refrences:

1) https://www.cs.cornell.edu/johannes/papers/2003/focs2003-gossip.pdf
2) http://se.inf.ethz.ch/people/eugster/papers/components.pdf
3) https://www.geeksforgeeks.org/mpi-distributed-computing-made-easy/
4) https://matheusportela.com/message-queues-for-distributed-systems
5) https://people.engr.tamu.edu/bettati/Courses/662/Generic/Slides/Handouts/Naming.pdf
6) http://csis.pace.edu/~marchese/CS865/Lectures/Chap5/Chapter5.htm