Assignment - 1

Distributed Systems

Submitted by:
M. Reshma
CSE - III
710420104032
21/02/23

10
22/2/23

Physical clock Synchronization : NTP.

Clock synchronization is done by two methods:

1. Internal synchronization.
2. External synchronization.

External synchronization.

* External synchronization refers to synchronization of process clock C_i with an authorize external source S. Let $D>0$. be the synchronization bound and S be the source of UTC.

* We say that clocks C_i are accurate within the bound of D.

Internal synchronization.

* Let $D>0$ be the synchronization bound and C_i and C_j are clocks at processes P_i and P_j respectively

* Then $|C_i(t)| < D$ for $i,j = 1, 2, \ldots N$ and for all real times t. we say that clocks C_i, C_j agree within the bound of D.

Clock correctness.

* A hardware clock, H is said to be correct if its drift rate is within a bound $\rho > 0$.

* Interval: $(1-\rho)(t'-t) = H(t) - H(t) = (1+\rho)(t'-t)$
  (where $t' > t$)

* A faulty clock is one that does not obey its correctness condition.

* A clocks crash failure is said to occur when the clock stop ticking altogether.

Synchronization in a synchronous System.

* A synchronous distributed system is one in which the following bounds are defined:
  (i) Time to execute each step of a process.
  (ii) Each message transmitted over a channel.
  (iii) bound:-
  Each process has a local clock whose drift rate from real time has a known bound.

* ...ans = min tmax/2 and receivers clock = t+(min+m

Cristian's Method for synchronizing clocks

The clock requests time and synchronizes only if the round-trip is within a certain

Cristian's Algorithm.

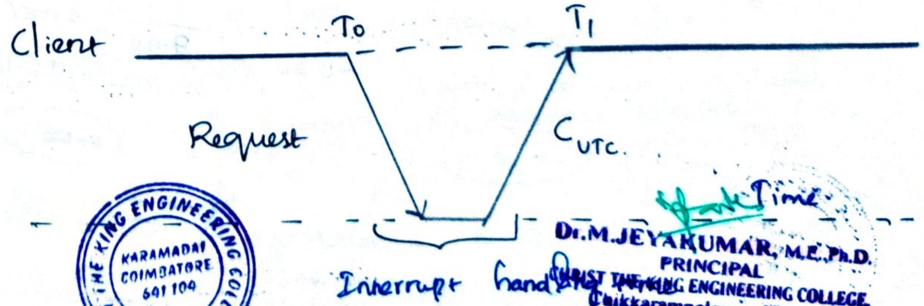
* Christian algorithm is centralized passive time server type algorithm.

* Assumptions: There is a machine with WWW receiver, which receives precise UTC.
  It is called the time server

Algorithm.

* A Machine sends a request to the time server at least every d/2r seconds, where d is the maximum difference allowed between a clock and the UTC.

Current time :-



Time stamp:

$$T_{new} = T_{server} + (T_0 + T_1)/2.$$

Adjust the clock.

* If the clock is faster than the UTC, add less to the time memory

* If the local clock is slower than the UTC, add more to the memory for each clock.

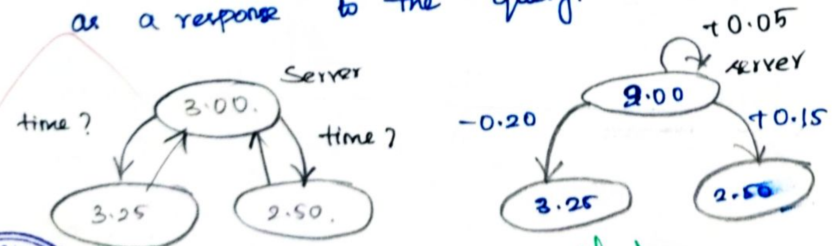
Berkeley Algorithm

Time server is a active machine.

* The server polls each machine periodically, asking it for the time.

* Instead of sending the updated time back to the slaves, which would introduce further uncertainly due to network delays.

* Each of these machines sends a time stamp as a response to the query.



Network Time Protocol.

* Cristians method & the Berkeley algorithm are intended for intranents.

* The Network Time Protocol (NTP) is yet another method for synchronizing clocks that uses a hierarchical architecture. Where the top level of the hierarchy are servers connected to a UTC time source such as a GPS unit.

Network time protocol goals:-

* Multicast on high speed local LANs.

* Procedure call mode

* symmetric mode.

Features of NTP.

1. Multicast
2. Procedure-call.
3. Symmetric mode.

Multicast mode:

* high speed LAN
* LAN.

Procedure mode:-

* Operations of Cristians algorithm.

Symmetric mode:

* Symmetric mode is intended ... by the master servers that supply ... in LANs.

Dr.M.JEYAKUMAR, M.E.,Ph.D.
PRINCIPAL
CHRIST THE KING ENGINEERING COLLEGE,
Chikkarampalayam Village,
Karamadai, Mettupalayam Taluk,
Coimbatore - 641 104.

Snapshot Algorithms for FIFO channels.

Candy-Lamport algorithm.

* Candy-Lamport algorithm records a set of process and channel states such that the combination is a consistent global state. Communication channels assumed to be FIFO.

Assumptions.

* The chandy-Lamport algorithm uses a control message, called a marker whose role in a FIFO system is to seperate messages in the channels.

Algorithm:-

1. Initiator process P_0 records its state locally.
2. Marker sending rule for process P_i:

   After P_i has recorded its state, for each outgoing channel Ch_{ij}, P_i sends one marker message over Ch_{ij}.

3. Marker receiving rule for process P_i:

   Process P_i on receipt of a marker over channel Ch_{ji}.

   If (P_i has not yet recorded its state) it

   Records its process state [illegible]

   Records the state of [illegible]

(3) Message-Passing System versus Shared Memory systems.

Message-Passing

* Two Processes communicate with each other by Passing messages. (Direct and indirect)
* Message Passing used as a method of communication in microkernels.
* Send
* Receive.

Shared Memory

* Shared memory systems are those in which there is common shared address space data variables for synchronization among the Processor.
* Shared memory allows maximum speed and convenince of communication, as it can be done at memory speed
* shared memory is faster than message passing.

Emulating Passing on a shared memory

* Synchronization Primitives like write and read. Operation are controlled by sender/receiver processor.
* A [illegible] message passing can be emulated by [illegible] by P_i to the mail box [illegible] by P_j from the mail[illegible]

starts Recording messages arriving Over other incoming channels.

else (Pi has already recorded its state).

Pi records the state of chji as the set of all messages. it has received Over chji since it saved its state.

Property of the Recorded Global State.

* The recorded goal state may not correspond to any of the global states that occured during the computation.

* Each process records its own state, and the two processes that a channel is incident on. cooperate in recording the channel state.

* On receiving the marker, q records its state, which is D, and records the state of c to be the empty sequence.

* On receiving the marker, p records the state of c' as the sequence consisting of the single message M.'