UNIT 2
CLOCK SYNCHRONIZATION

Introduction

- A distributed system consist of multiple concurrent processes.
- It is economical to share the system resources among these processes.
- Sharing may be:
  - Competitive
  - Co-operative

- Tape drive cannot be used simultaneously , A process must wait if another process is using it.
- Two process Client-Server relationship, for file access operation they cooperate each other.
  - Sharing requires certain rules of behavior that guarantee that correct interaction occurs .The rules for enforcing correct interaction are implemented in the form of synchronization mechanism

SYNCHRONIZATION

- Synchronization-related issues
  - Clock synchronization
  - Event ordering
  - Mutual exclusion
  - Deadlock
  - Election algorithms

ClOCK SYNCHRONIZATION

- Every computer needs a computer clock to keep track of computer time and calculating the time spent by a process in execution on processor.
- An application may have many processes that concurrently running on different nodes of the system.
- Every node need clock that should be synchronized.
- Example, Air-line Reservation System
- If the clock of these nodes are not synchronized ,it is difficult to achieve the correct result. That problem is called the clock synchronization problem.
- Need some suitable ways to properly synchronizing the clocks.

ClOCK SYNCHRONIZATION

- How Computer Clocks are Implemented?
- Three components are

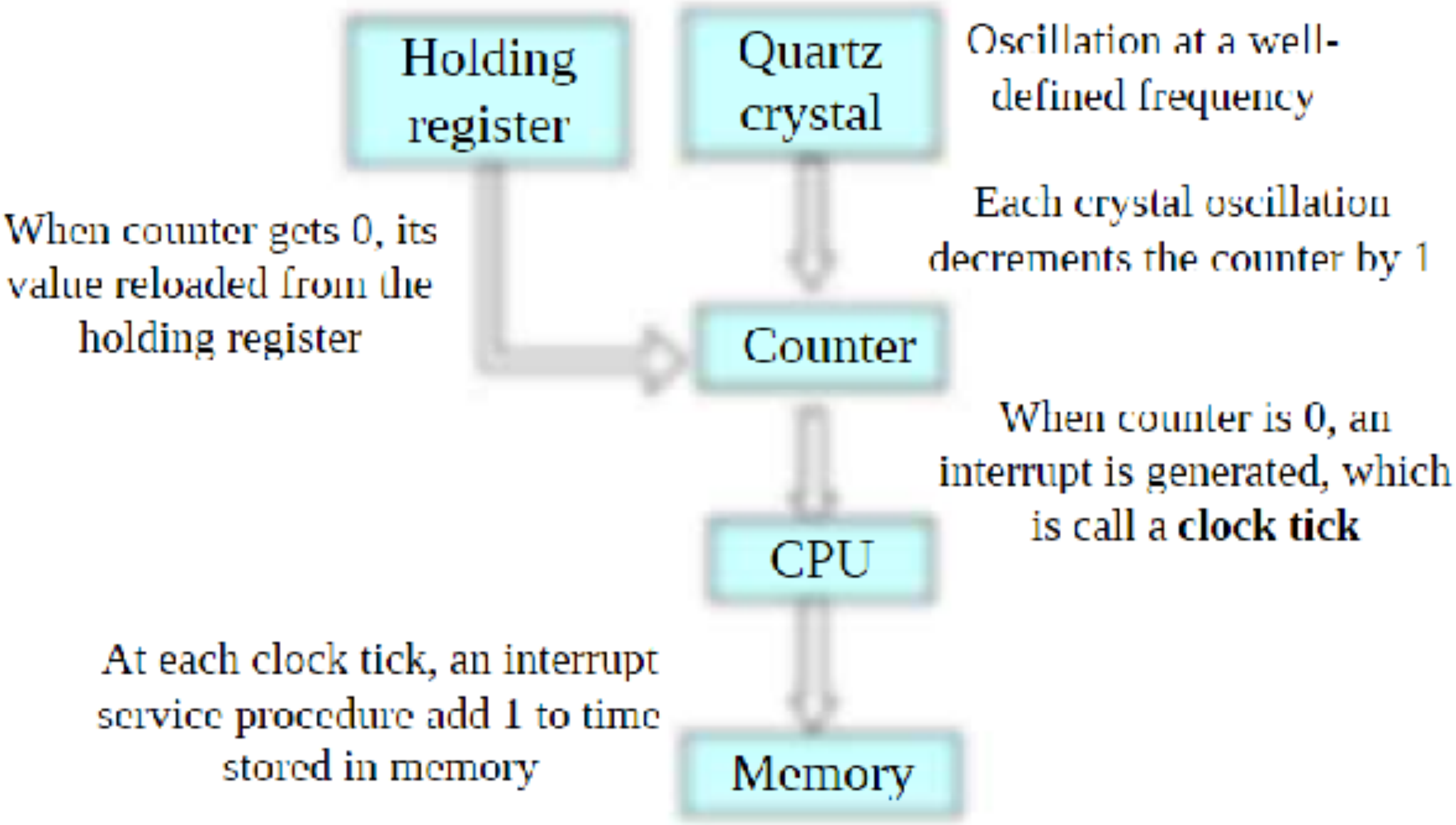
i. Quartz Crystal- Oscillates at well defined frequency

ii. Constant Register- To store the constant value that is decided based on the frequency of oscillation of Quartz crystal

iii. Counter Register-To keep track of oscillation of Quartz crystal

Quarts crystal (vibrating crystal create an electrical signal with a precise frequency)

How Clocks Work in Computer



Holding register

Quartz crystal

Oscillation at a well-defined frequency

Each crystal oscillation decrements the counter by 1

When counter gets 0, its value reloaded from the holding register

Counter

When counter is 0, an interrupt is generated, which is call a **clock tick**

CPU

At each clock tick, an interrupt service procedure add 1 to time stored in memory

Memory

ClOCK SYNCHRONIZATION

- To make the computer clock function as an ordinary clock life, the following things are done:

1 . The value in the constant register is chosen so that 60 clock ticks occur in a second.

2. The computer clock is synchronized with real time (external clock).

For this, two more values are stored in the system

- a fixed starting date and time

- the number of ticks.

Drifting of Clocks

- A clock always runs at a constant rate because its quartz crystal oscillates at a well- defined frequency
- Due to differences in crystals ,two clock rates different from each other.
- Difference is small but observable , no matter how accurately initialized.
- The difference accumulated over many oscillations leads to an observable difference in the times of the two clocks
- For every 1,000,000 second there may be drift of 1 sec.(After 11.6 days of 1 second)
- Therefore, Passage of time, a computer clock drifts from the real-time clock.
- Hence, Computer clock must be periodically synchronized with the real-time clock called non-faulty.
- Even non-faulty clocks do not always maintain perfect time.
- A clock is called non-faulty if there is bound on the amount of drift.

DRIFTING OF CLOCKS

Suppose that when the real time is t, the time value of a clock p is Cp(t).

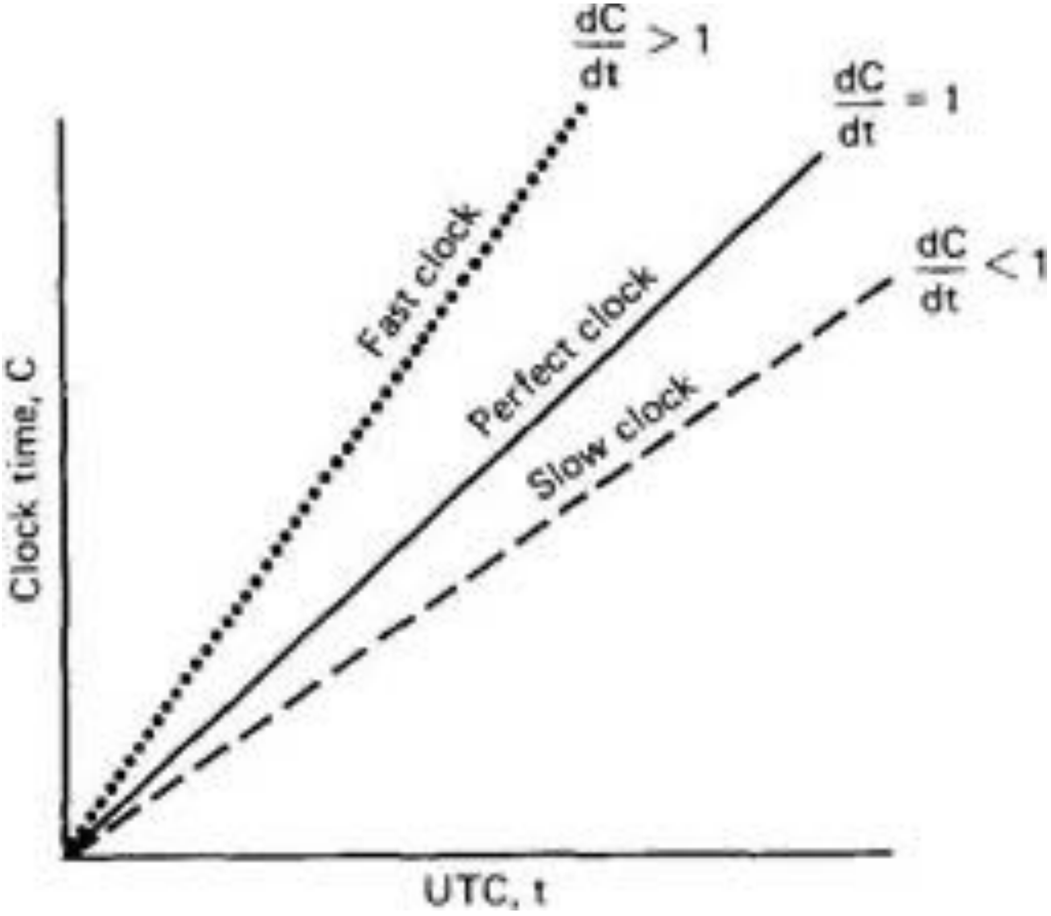
If all clocks in the world were perfectly synchronized, we would have Cp(t) =t for all p and all t.

That is, if C denotes the time value of a clock, in the ideal case dc/dt should be 1.

Max Drift Rate allowable **ρ**

$$1- \rho <= dc/dt <= 1+ \rho$$

Drifting of Clocks cont..



Slow, perfect and fast clocks

DRIFTING OF CLOCKS

- The nodes of a distributed system must periodically resynchronize
- Their local clocks to maintain a global time base across the entire system

Types of Clock Synchronization

Distributed system requires the following types of clock synchronization

1.Synchronization of the computer clocks with real-time (or external) clocks

- Required for real time applications
- Allows the system to exchange information about the timing with other systems and users.
- A external time source used UTC(Coordinated universal time).
- GEOS – Geostationary Operational Environmental Satellites
- Commercial devices (Time Providers) available to receive these signals

2. Mutual (or internal) synchronization of the clocks of different nodes of the system:

-Provide a constant view of time across all the nodes of distributed systems. App that requires Consistent view of time across all nodes of DS - Process Migration

External Synchronized clocks are also internally synchronized converse is not true-Internally synchronized clock may drift arbitrarily far from external time.

CLOCK SYNCHRONIZATION ISSUES

- Two clocks cannot be perfectly synchronized.
- In practically for the synchronization, the difference of time value of two clocks should less than delta.
- The difference in time values of two clocks is called clock skew.
- Clock skew of any two clocks is set less than delta .
- It is necessary of synchronization that each node should read the clock values of other nodes. The mechanism of reading the clock values of other nodes varies in different algorithms.
- Clock synchronization requires each node to read the other nodes' clock values. Errors occur mainly because of unpredictable communication delays
- Time must never run backward because it causes repetition of certain operations, when fast clock is set to slow down.
- Time should not be slowed down or moved backward at once, it should be gradually updated with each interrupt.

Clock Synchronization algorithms

- **Centralized Algorithms**
  - Passive time server
  - Active time server
- **Distributed Algorithms**
  - Global Averaging Algorithms
  - Localized Averaging Algorithms

Centralized Algorithms

- One node has a real-time receiver called time server node.
- Clock time of this node is used to correct the time of all other nodes.
- The goal of this algorithm is to keep the clock of all other nodes synchronized with the clock time of the time server node.
- Depending the role of the server node, centralized algorithms are again of two types,
  - Passive time server
  - Active time server

Passive Time server Algorithm

- Each node periodically sends a message (“ Time = ? “) to the time server.
- Server quickly responds with a message (“Time = T”).
- T is the current time of the server node.
- When client node sends the request of time, its clock time is To .
- When client node receives the time from server, its clock time is T1 .
- Propagation time of the message from server to client node is estimated to be (T1-To) / 2.
- Therefore, Clock of client node is readjusted to T+(T1-To)/2 .
- Due to unpredictable propagation time between two nodes , is not very good estimation.
- Two methods are described below to solve this problem.

1. This method assumes the availability of some additional information.

Two things are known:

  - Approximate time taken by the time server to handle the interrupt .
  - Process time of request message of client (“ Time = ? “) .

- Sum up of both times is the I .
- Best estimate in this case would be (T1-To-I) / 2.
- So , Total time at the client node is readjusted to T+ (T1-To-I) / 2.

2. This methods assumes the no additional information, several measurements of T1-To are made, and threshold is set.

- If the (T1-To) exceeds threshold then values are discarded.
- The average of remaining measurements is calculated and half of this value is added to T.
- Alternative way is, Measurements for which (T1-To) is minimum is considered to be most accurate one. And half of this added to T.

Active Time Server

- In this approach , Time server periodically broadcast its clock time (“ Time = T”) .
- Other nodes receive broadcast message and use the clock time in the message for correcting their own clocks.
- Each node has a priori knowledge of the approximate time (Ta) .
- Ta is the time from the propagation server to client node.
- And Client node is readjusted to T + Ta .

Major Faults:

- Broadcast message reaches too late at the client node due to some communication fault, Clock of this node readjusted to incorrect value.
- Broadcast facility should be supported by network.

Active Time Server (Solutions)

Berkeley Algorithm:

- Time server periodically sends a message (Time= “?”) to all nodes in a group.
- Each node send its clock value to server node.
- Server has prior knowledge of propagation time from client to server node.
- Based on prior knowledge it first readjust the clock values of reply messages.
- Then takes fault tolerant average.
- Time server choses a subset of all clocks values that do not differ from one another by more than specified amount.
- And average is taken only for clocks value for subset.
- And eliminates reading from the unreliable clocks.
- Calculated average is current time which all clocks should be readjusted.

Disadvantages of Centralized Algorithms

- Single point of failure , if time server node failed , clock synchronization operation cannot be performed.
- From scalability point of view , adding more node will put more burden on the single central server.

- Solutions, of these problem are coming next…

Distributed Algorithms

- Externally synchronized clocks are also internally synchronized
- Each node's clock is independently synchronized with real time.
- All clocks of system remain mutually synchronized.
- Each node is equipped with real time receiver so that each node can be independently synchronized.
- Theoretically internal synchronization of clock is not required.
- In practice, due to inherent inaccuracy of real time clocks, different real time clocks produce different time.
- Internal synchronization is performed for better accuracy.
- Types of internal Synchronization:
  - Global averaging
  - Localized Averaging

Global Averaging

- Each node broadcasts its local clock time in the form of a special “resync” message.
- After broadcasting the clock value , the clock process of a node waits for time T (T is calculated by algorithm).
- During this period, clock process collects the resync messages broadcast by other nodes.
- Clock process estimates the skew of its clock with respect to other nodes (when message received) .
- It then computes a fault-tolerant average of the estimated skews and use it to correct the local clock.
- Two commonly algorithms use to calculate the fault-tolerant average.

Global Averaging cont..

- 1st simplest algorithm is to take the average of estimated skews and use it as the correction for the local clock. And only below a threshold are taken into account, and other are set to zero before computing the average.

- 2nd Algorithm, each node limits the impact of faulty clocks by discarding the highest and lowest skews. And then calculating the average of the remaining skews. And use it to the correction for the local clock.

Localized Averaging Algorithm

- Global algorithm was only Suitable for small networks due to broadcast facility.
- Where node has direct communication to every other node (fully connected topology) .
- This algorithm attempt to overcome the drawbacks of the global averaging algorithm.
- Nodes of a distributed system are logically arranged in some kind of pattern such as Ring.
- Periodically each node exchange its clock time with is neighbors in the ring.
- And then sets its clock time by taking Average.
- Average is calculated by , its own clock time and clock times of its neighbors.