MICHAEL ORINA
SCII/00825/2019

DISTRIBUTED SYSTEMS ASSIGNMENT 2

1. Discuss clock syncronization.

Clock synchronization is an essential aspect of distributed systems, where multiple computers or devices work together to achieve a common goal. In a distributed system, each device has its own clock, which may be slightly different from other devices' clocks due to clock drift, network delays, or other factors. This difference in clocks can create problems when different devices need to work together, as they must agree on a common timeline to ensure proper coordination and synchronization.

In distributed systems, there are two types of clock synchronization: physical clock synchronization and logical clock synchronization.

Overall, clock synchronization is an important aspect of distributed systems that enables different devices to work together in a coordinated and synchronized manner. By ensuring that devices have a common timeline and are synchronized in their actions, clock synchronization helps to improve the reliability and efficiency of distributed systems.

2. Physical and Logical clocks

Physical clock synchronization involves ensuring that the clocks of different devices are closely aligned with each other. The most common way to achieve physical clock synchronization is through a protocol called the Network Time Protocol (NTP). NTP enables devices to synchronize their clocks by exchanging timestamped messages over the network. When a device receives a timestamped message, it can use the information in the message to adjust its clock to align with the sender's clock.

Logical clock synchronization, on the other hand, focuses on ensuring that different events that occur on different devices are ordered in a consistent manner. Logical clocks are often used in distributed systems to establish a partial ordering of events, which can be used to resolve conflicts and ensure proper coordination. One popular example of logical clocks is the Lamport timestamp, which assigns a unique timestamp to each event based on the order in which they occur.

In addition to NTP and logical clocks, there are several other clock synchronization techniques used in distributed systems, such as Cristian's algorithm, Berkeley algorithm, and GPS-based clock synchronization. Each technique has its own advantages and disadvantages, and the choice of technique depends on the specific requirements of the distributed system.

3. **Internal vs External**

Internal and external clock synchronization are two different approaches to synchronizing clocks in a distributed system.

Internal clock synchronization involves using clock synchronization algorithms that are built into the system to keep the clocks of different devices synchronized. These algorithms may use techniques like clock skew detection and adjustment, or logical clock synchronization to maintain a consistent and accurate timeline across devices.

External clock synchronization, on the other hand, involves relying on an external source to synchronize the clocks of different devices. The most common approach to external clock synchronization is to use a dedicated time server, which provides a reliable and accurate time reference for all devices on the network. This time server may use a protocol like NTP or GPS to provide accurate time information to the devices.

The choice between internal and external clock synchronization depends on various factors, such as the size and complexity of the system, the level of accuracy required, and the availability of external time sources. In general, internal clock synchronization is more common in smaller or simpler systems where external time sources may not be available or necessary. External clock synchronization is typically used in larger or more complex systems where accurate time synchronization is critical, such as in financial trading systems or networked control systems.

Overall, both internal and external clock synchronization approaches have their own advantages and disadvantages, and the choice between them depends on the specific requirements and constraints of the system being designed or used.