Paper Review 1

Name: Bharath Venkatesh
Unity ID: 001037220

Paper: Time,Clocks and Ordering of Events in Distributed System

This paper aims at providing a clear understanding of partial ordering and total ordering of events that occur in a distributed environment. It introduces the concept of logical clocks as means of establishing partial ordering and further extends the concept to obtain total ordering of the system. The author derives an algorithm for synchronizing logical clocks to obtain total ordering of the system and further demonstrates the use of this algorithm by solving a classical synchronization problem of two competing processes trying to obtain a shared resource. Although the algorithm works fine under normal circumstances, if the ordering perceived by the user is different from the one obtained the algorithm it leads to anomalous behavior. The algorithm proposed is thus improvised to introduce physical clocks to solve such anomalies in the system. Further an upper bound on how far out of synchrony these clocks can drift is deduced.

In case of distributed systems, it is sometimes impossible to determine which event “happened before” another. Thus it can only represent a partial ordering of the system. One of the primary advantage of the paper is that it extends the concept of partial ordering to deduce a consistent total ordering of the system. In that front the authors claim seems to hold true.

The second advantage of this paper is that it proposes a distributed algorithm which does not need a central synchronizing process or central storage. Thus it can be generalized to implement any desired synchronization for a distributed system.

Finally, the paper was able to demonstrate a solution to handle the problem of clock drift in distributed systems.

Although the algorithm provides a simple and extendable algorithm to handle a distributed environment, it has several shortcomings. Firstly, the paper assumes that all processes will actively participate in the system. Meaning that if a single process fails or halts for some time then this event will halt the entire system from continuing. This poses a need for a solution to make the system understand gracefully handle the failed process and continue without halting other processes. The 'Optimal Clock Synchronization' paper discusses means of setting an upper bound on the delay to provide such flexibility.

Secondly, the required accuracy of a physical clock is dependent on the transmission delay. The paper assumes that the discrete ticks of the clock is fast enough to ensure that it is unique at any point of time which might be faulty at very low transmission delays.

Lastly, the system is completely depending on the working of logical and physical clocks. Also it works on a n assumption that clocks are never set backwards. This poses a question of what happens if any of these clocks fail or if they are set backwards. This will lead to several anomalies in the system. More robust clocking mechanisms like Vector Clocks have been designed to counter the problems arising due to clocks in this paper.

<u>Paper: Distributed Snapshots – Determining Global States of Distributed Systems</u>

This paper proposes an algorithm to determine the global state of a distributed system as a combination of the states of each individual process in the system. It does so by taking a "snapshot" of the current state of the process and the channels associated with it and sending a marker over its channel so that any process receiving the marker will record state if marker is previously not seen and set the channel state to empty, else stop recording and set the state of the channel to the sequence of messages received since the marker was previously seen. The paper further proposes a solution to the stability detection problem using the algorithm previously proposed by stating that if the stable property is true, then it remains true at the end of snapshot and if it is false, then it was false from the beginning of the snapshot.

This paper proposes one of the basic algorithms around which several checkpointing algorithms have been designed in the recent past. One of the primary advantage of this algorithm is that it deduced a consistent global state of the system taking into account the communication delays and clock skew. Having a global view of the system helps largely in solving several problems related to Stable property detection and checkpointing. Examples include Deadlock detection in databases, Phase termination etc.

Secondly, the algorithm provides a clear understanding of the relationships between local process states, global system states and points in a distributed computation thus eliminating any misunderstandings leading to incorrect algorithms in the past.

Finally, the algorithm works in linear time complexity. If we have a distributed system with e edges and a diameter d then recording of one instance of the algorithm takes O(d) time, as the marker needs to reach each node in the system only once.

Although the paper forms a base for several future research, the paper alone had several flaws. Firstly the paper is limited to ordering of events only in FIFO ordering. Meaning that if there exists a need to schedule events based on priority then the paper proves to be insufficient. Recent works of Lai-Yang Algorithm helps resolve this issue.

Second, the paper assumes a complete failure free environment of execution which is impractical. A fail-safe mechanism needs to be designed on top of this algorithm that can handle process crashes and channel failures.

Third, the paper assumes that any one of the participating nodes will spontaneously generate a snapshot and send the marker. However it remains to be seen as to what might happen in case of multiple concurrent snapshots by participating processes. Spezialetti-Kearns algorithms that support bidirectional channels and concurrent initiators helps resolve this issue.

Finally, the paper states all transitions solely based on message exchanges. This might not be relevant for all distributed systems.

Chandy-Lamport algorithm had several tightly bound assumptions causing several overheads. Several of these assumptions are elasticized and new algorithms have been developed. Lai-Yang algorithm, Spezialetti-Kearns algorithm, Acharya-Badrinath algorithm and Alagar-Venkatesan algorithms are few of such instances.