CSE5306- Distributed Systems
Project2-Report

Team Members-
Megha Modi – 1001750561
Nishank Gujar – 1001861756

I have neither given nor received unauthorized assistance on this work
Signed: Megha Modi Date: 30 Oct 2021
Signed: Nishank Gujar Date: 30 Oct 2021

Implementation

We used Python3.8 as the programming language for the implementation, as we are more familiar with the language as well as gain the concepts associated with the project.

Total Ordered Multicasting: The Updated message is timestamped with sender's logical time and then the updated message is multicast including the sender itself. When the message is received, it is put into local queue and ordered according to timestamp and a multicast acknowledgement.

Vector Clock: It generates partial ordering of events in a distributed system and detecting causality violations. Just as in Lamport timestamps, inter-process messages contain the state of the sending process's logical clock. A vector clock of a system of N processes is an array of N logical clocks, one clock per process, a local smallest possible value copy of the global clock-array is kept in each process. We use multiple processes to emulate multiple nodes in Distributed Systems and assume that all nodes are initiated After completing a local operation, each process sends its updated vector clock to all other processes and then follows the Vector-Clock algorithm.

Locking Scheme: Different nodes in a distributed system are emulated by different processes running simultaneously. Coordinator node regulates the shared file-access. When a process acquires the lock, it simply opens the file, increments a counter in the file, and closes the file. Assume that all processes keep requesting the lock until successfully acquiring the lock.

What we learned

We learnt a lot of concepts taught in class come into life while doing this project. Apart from what we implemented which is Total Ordered Multicasting, Vector Clock & Locking Schema, one of the biggest lessons we learnt was, inter-processing or system-to-system communication and synchronization is non-trivial. We also learnt how to work with threads and establish a multicast between processes.

Issues we encountered

We came across a lot of issues/challenges while implementing this project, but the most challenging part was establishing a multicast communication. We discovered that this is nontrivial when working with processes to emulate multiple systems. We also ran into issues related to using multiple threads and managing global &local data in the threads.

Conclusion

The project was challenging but gave a lot of practical knowledge while implementation. The concepts taught in the class were put to work while implementing this project.