

# Report: Loggy - a logical time logger

Muhammad Azeem ([azee@kth.se](mailto:azee@kth.se))

September 24, 2020

## 1. Introduction

A logger was implemented which used logical time clocks to keep track of order of messages passed by the worker nodes.

Initially Lamport Clock was implemented to track message-times. And in later stages Vector Clock was implemented to keep more precise track of time of workers.

## 2. Main problems and solutions

Initially the main issue was understanding how the sending message was getting logged later than the receiving message. This became more understandable with more trails by changing the jitter times. When jitter was zero, the sending message was logged before the receiving message.

Keeping the Clock functionality abstracted to the 'time' module was another interesting challenge, but it helped keep the code the other classes simply to understand.

Another challenge was to log message by time, this required making and keeping track of the hold-back queue and deciding which messages are safe to be logged was another challenge. This was done by reading the regularly comparing the hold-back queue with the latest time and deciding which messages are safe to log accordingly. It required keeping track of multiple lists and comparing their elements, etc.

## 3. Evaluation

Several tests were done, at first using the simple logger. Later implementing Lamport Clock with the hold-back queue and finally with Vector clock.

At first using the simple logger, the receivers message was being logged before the sender's message. This was clearly because of the jitter added to the receiver

worker. But at jitter=0 & sleep=0, the sender's message finally appeared before the receiver's message.

The Lamport Clock also provided similar results, however, when the hold-back queue was implemented, the sender's message (having less time than receiver's message) started appearing first. Furthermore, all messages were then ordered by time. And lastly Vector Clock with hold-back queue provided mostly similar results to Lamport Clock hold-back queue.

#### **4. Conclusion**

I understood the differences between not considering time at all, using physical time, and using logical time esp. with hold-back queues. Lamport Clock became more understandable. The advantage of using logical clocks in a distributed system were very clear. Furthermore, using Vector Clocks added useful precision (however, possibly not very easily scalable) to the timestamps.