Report 4: Loggy homework report

Douglas Hammarstam

September 28, 2022

In this project, a simple web logger has been implemented.

In this seminar, we implemented a logger using the lamnpart time stamp. The lampart timestamp was used in order to maintain the happened before ordering. IE that any related messages maintain a happened before ordering. It wont gurantee the total ordering though.

1 Main problems and solutions

Initially, we have a logger which will print messages in FIFO ordering, but this will not be the case if we introduce more than one worker. Therefore; The main problem in this assignment was to make sure that related messages dont get printed out of sync IE the fist message after the other. Therefore the lampart timestamp system was implemented in order to solve that. The main problem was to keep a hold back queue that kept messages that weren't safe to print yet. The queue therefore held all of the messages that were not safe to print, which meant that they had a higher lampart timestamp than the lowest timestamp that is held in the logger module at the time.

2 Evaluation

In order to evaluate the program, I ran the test supplied with the assignment in order to see that no related messages were printed out of order, and as far as I can see, none of them are printed out of order.

3 Conclusions

The lampart system is a good way to make sure that related messages are not printed out of order. But it does not gurantee complete order of messages. And also, the lampart time system's hold back queue might grow very large if one of the nodes does not send a message for a while. If it does not send a message at all, the system will stop working, since other messages are waiting for one of the nodes to increase the lampart time stand held in the logger module. Using the Vector system, the queue wont really grow that long, since the logger can then determine if two messages are related or not, and then print messages that would not be printed in the lampart timestamp system