18-842: Distributed Systems **Spring 2015**

Lab 1: Clocks

Demo Scheme

Demonstrations: Students will need to demonstrate an application program that exercises their ClockService class in both Logical and Vector modes. Students also need to show the Logger application running with Vector clocks, probably in a different application. At a minimum, students should be able to:

1. Discuss the system architecture in some detail. It would be very helpful to show up to the demo with an architectural diagram showing the components of your solution.
2. Logical clocks are generated properly, always incrementing for each time stamp generated. When a timestamp is received from elsewhere and given to the ClockService, the local timestamp must be advanced to max {local timestamp, remote timestamp} +1 (as illustrated in Figure 14.6 from the text). Timestamps can be compared properly -- test for cases of a → b, b → a and a || b. Messages can be sent between third party processes to advance the clock.
3. Vector clocks are generated properly, always incrementing for each time stamp generated. When a timestamp is received from elsewhere and given to the ClockService, the local timestamp must be advanced to max {local timestamp, remote timestamp} for each element of the vector, except for the local element of the timestamp which must be incremented (as illustrated in Figure 14.7 from the text). Timestamps can be compared properly -- test for cases of a → b, b → a and a || b.
4. Messages sent and received with MessagePasser are properly integrated with the ClockService (i.e. new timestamp on sent messages, timestamps potentially advanced with received messages). The application doesn't do anything special to get a timestamp on a message.
5. Events other than messages can be timestamped.
6. Show a logging facility. This should be a different application, using MessagePasser, to receive messages and track their order in a → fashion. Students should be comparing the timestamp objects, not actually inventing their own way of doing the comparison. The Logger properly detects and displays concurrent events.
7. Discuss performance in the presence of errors. Students should be able to talk about failure modes of the network (i.e. drop/delay/duplicate) as well as the processes.