RepCL for IoT

Ishaan Lagwankar

June 2024

1 Reading List

- Time, clocks, and the ordering of events in a distributed system
- Logical time in distributed computing systems
- Replay Clocks
- Timestamps in message-passing systems that preserve the partial ordering
- Achieving Causality with Physical Clocks
- Virtual time and global states of distributed systems
- Network time protocol (NTP)
- Analysis of bounds on hybrid vector clocks
- Ishaan Lagwankar's MS Thesis

2 Implementation README

- Clone the repository: Replay Clocks.
- Switch to branch 'cdes'.
- Run `bash runSimulation.sh` on a UNIX shell. You should see data being collected.
- Understand the structure of how data is being collected by tracing the bash script.
- The core functions of the library are in the src folder. The core simulation functionality resides in simulator.cc. The core RepCl functionality resides in replay-clock.cc.

3 Beginner Tasks

- Read at the least `Ishaan Lagwankar's MS Thesis`. Make sure you understand what the clock does and what its features are.
- Test your knowledge by answering the following questions as a followup to the relevant question bubble in GitHub Projects.
  - What is a Replay Clock?
  - What are the Replay Clock's key components?
  - How does the Send/Local algorithm work?
  - How does the Recv algorithm work?
  - What happens when the parameters of the system are in the infeasible region of the clock?
- The clock uses a bitset$< 64 >$ variable to store the offset and the offset bitmap vectors. Convert this to a vector$< int >$ and rewrite the relevant functions to ensure the function does not change. Change the function signature only if necessary.
- Once done, test the clock using the branch `c++-src-bitmap`. Use the file TestCaseRC.cpp to find the test cases and rewrite them as you wish. Make sure the clock works as expected with those test cases.
- Once this is done, start researching on IoT simulation techniques. We specifically need an IoT simulator that allows us to change the working of the underlying device, and control network parameters. NS-3 is a good option, but if you find something better, let me know!