CPRE 492 WEEKLY REPORT 14

Project Molecule

8 – 17 January 2017

May1739
may1739@iastate.edu

Dr. Arun Somani

Ryan Wade – Team Leader

Nathan Volkert – Communications Lead

Daniel Griffen – Key Concept Holder

Alex Berns – Webmaster & Scribe

1 CONTENTS

2 WEEKLY SUMMARY

This week we set up our weekly meetings as a group and with our advisor. We brainstormed a synchronization algorithm and created a state transition diagram for it. We continued work on particle packaging and network layer API. We had a successful first meeting with our advisor to discuss the synchronization algorithm and received feedback for how to improve it.

3 PAST WEEK ACCOMPLISHMENTS

All Members:

- Determine algorithm for synchronization.

Ryan Wade:

- Created Transition Diagram for Synchronization Algorithm
- Administrative (set meeting times, coordinate with advisor)

Nathan Volkert:

- Worked on packaging

Daniel Griffen:

- Redesign of network API using asynchronous I/O

Alex Berns:

- Research Mocha

4 INDIVIDUAL CONTRIBUTIONS

NAME	Hours	Semester Total	Cumulative
Ryan Wade	11	11	131
Nathan Volkert	7	7	109
Daniel Griffen	14	14	148
Alex Berns	8	8	107

5 COMMENTS AND EXTENDED DISCUSSION

5.1 SYNCHRONIZATION

State is shared between all nodes in the system and is synchronized in real time. We considered the following use cases when designing the algorithm:

1. A Divided Network
2. Loss of Network
3. A Single Bad Particle Actor

We determined that data will primarily be propogated from single nodes to all other nodes. As such our Fault tolerance is focused on resolving Network or Hardware Failure.

5.2 ASYNC NETWORKING

Previous network API used synchronous I/O. This meant that any I/O operations would block the thread they were running on. Atomic layer implementations using this API would either have to deal with the blocking, or spawn a new thread for each incoming connection. Spawning a new thread for each connection was determined to be too costly and blocking on I/O was not desired if service throughput was to be maintained.

The solution to the problem was to reimplement the network API to utilize asynchronous I/O. The new API does not block the current thread during an I/O operation. Instead other work is done until the I/O operation is completed. This design maximizes throughput without the overhead of threading.

6 PLAN FOR COMING WEEK

We will continue to refine the synchronization algorithm and define the permissions sub-system of the Atomic Layer.

7 SUMMARY OF WEEKLY ADVISOR MEETING

This week, as a group we attempted to solve the unsolvable problems of byzantine fault. Initially we were attempting to create an algorithm to manage the propagation of information that also was protected against bad actors. But after discussing the algorithm with our advisor we determined that we cannot guarantee that this algorithm will not fail. Originally the permissions systems was the primary check for bad actors and the propagation algorithm would be a secondary check. But due to the issues discussed we will only use the permissions systems.

8 SYNCHRONIZATION TRANSITION DIAGRAM