University of Advancing Technology
CSC318 - Software Engineering Principles
Brandon Nay

Object-Oriented Programming on a Distributed System

25th July 2020

OVERVIEW

I recently became interested in distributed systems and wanted to use this time to get involved and make something interesting. Half of the planned system sits assembled on my desk, and the rest comes in the mail at the start of the week. I'll be using four Raspberry Pi's wired to a network switch as my platform.

Although the true end product is not yet decided, the basic principle will be the same: object-based messages between nodes, with each Pi besides the primary node acting as a self-driven participant in an ecosystem that will be visualized by the primary node's console.

Since I'm new to distributed systems, how this project will look at the end of this course will be shaped by what I learn in the process of documenting and mapping out the scope - one path may be more achievable than the others. Here are a few potential end products:

1. Each node (Raspberry Pi) controls a different population using a genetic algorithm, all within the same environment. Each population may gain a beneficial feature unique from the others on spawn
2. Free-for-all auto combat using a complex decision engine (Q-learning, TensorFlow Lite) where each node controls and improves its own actors
3. Another system involving TensorFlow Lite

These aren't ideal examples of what distributed computing is for. Because of this there's also a good chance the system as a whole will run slower than using a single CPU due to the overhead produced in a distributed system. Still, it should be fun to develop and certainly stretch my mental muscles.

FEATURES

1. System is able to exchange object-based messages
2. Each node controls its own actions, has its own ‘brain’
3. Primary node keeps each node up-to-date on the actions of the others, informing their decisions
4. (Stretch) Primary node provides a visual of the system’s state

TECHNOLOGIES

1. Project lives on GitHub
2. Making use of GitHub’s project management functionality
3. Majority of code written in Python 3.6 on Raspberry Pi OS (formerly Raspbian OS)
    - Library for cluster control TBD
    - Potential use of TensorFlow Lite
4. System cluster is accessed via SSH
5. Physical hardware:
    - 4x Raspberry Pi Model 4, 4GB variant
    - 4x Micro SD cards flashed with Raspberry Pi OS
    - 4x Network cables - CAT6A, 1 ft each
    - 1x Network switch - 8-port gigabit (possible expansion of nodes post-class)
    - 1x Structure supporting four Raspberry Pi’s with mounted 40mm fans and included heat sinks

MILESTONES

Sprint 1: Feature 1

Build computing cluster, setup cluster control, exchange an object between nodes.

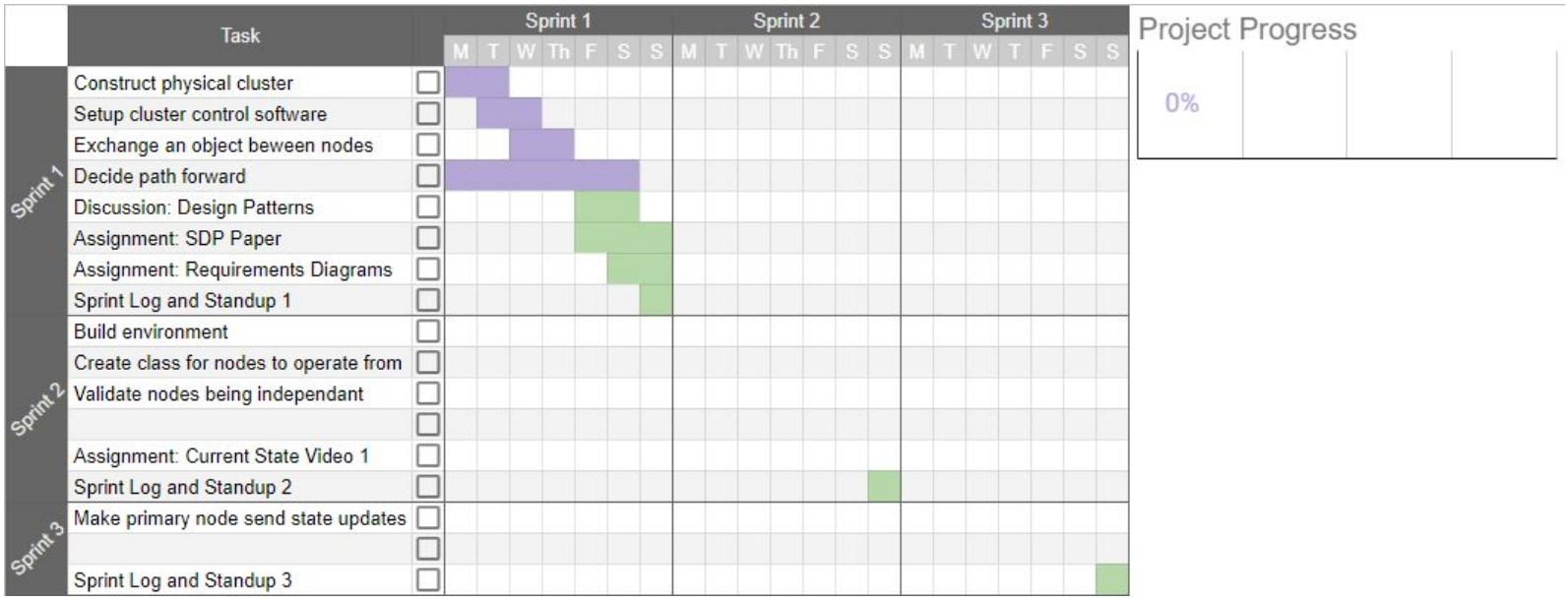
Sprint 2: Feature 2

Based on research during sprint 1, design an environment and create node instructions so they may perform actions within that environment.

Sprint 3: Feature 3

Use the primary node to keep each node up to date with the actions of the others, informing their decisions.

SCHEDULE



Schedule will have tasks added as they become more apparent/as needed. Progress bar updates as tasks are checked off.