**Midterm Report**

**Client: Dr. Xuguang** **Chen**

**Role: Team Manager/Network Administrator/Software Engineer**

Currently, I am working on the networking part of the project, which is very essential for this project to succeed.

**Responsibilities:**

* Spin the node webserver on raspberry pi cluster and deploy the website.
* Schedule appointments with Dr. Chen
* Tasking team mates

**Purpose of the project:**

Build a Raspberry Pi cluster Web Server in order to deploy a website that will be used for teaching purposes as per Dr. Chen, to the entire Computer Science students at Saint Martin’s University.

**Plan to complete it:**

This project consists of two parts:

* **Senior Project 1**

This part of the project will be focused on the parallel programming and load balancing. Main page and 13 more dummy link pages have been created which will be deployed on the Cluster Webserver and they will feature load balancing and parallel programming.

* **Senior Project 2**

In this portion of the project, we will be coding in C and html. This website is a tutorial for C programming language. In this portion of the project, we will be working on how to make the tutorial more efficient and more user friendly. Entire time we will be focusing on the website.

**Obstacles:**

Port Forwarding is not being too nice. My haproxy, which is also working as a load balancer, is on the top layer of the server Layer (3 server cluster Raspberry Pi Web Servers). My servers are listening to port 8000 and haproxy is listening to port 80. Haproxy is feeding traffic through port 8000 to my servers but my router is not allowing the “external ip: 80” address from the internet (WAN).

As a result, I can browse the website from the Local Area Network but unfortunately, the work is still in progress on the Wide Area Network side.

**Main Technological Features:**

* **Parallel Programming (MPICH)**

All 3 servers are saved as a one instance into a machine file. Main objective is to call that machine file when the website is populated on to the browser to work on much larger problem size with the faster computation.

* **Load Balancing (Round Robin, leastconn)**

Round robin is a default load balancer and it just switches from one server to another, turn by turn. Instead, if we use leastconn balancer, the server with the least connection will be chosen. It is best to use leatconn balancer when there is too much traffic flowing.

* **Remote Desktop is installed**

Remote Desktop allows us to remotely log in to the local network which enables us to connect to our local servers.