Project Planning document
Design Project

Christopher Arredondo-Fallas
2016094916

Tecnológico de Costa Rica

Contents

1 Project name 2

2 Company 2

3 IP Requirements 2

4 Problem Description 2

5 General objective 2

6 Specific objectives 2

6.1 First objective 2

6.2 Second objective 2

6.3 Third objective 3

7 Stakeholders 3

8 Solution description 3

8.1 First layer 3

8.2 Second layer 3

8.3 Third layer 3

9 Deliverables and Criteria of acceptance 4

10 Activities and effort budget 6

11 Risk analysis 7

12 Work schedule 8

1 Project name

Cloud edge computing System using Wind River Linux with Kubernetes

2 Company

Wind River Systems

3 IP Requirements

The intellectual property of Wind River Systems including Wind River Linux cannot be shared with anyone who has not purchased a licence, given that statement, any code generated that uses an open source library will be shareable thanks to the GPL licence.

4 Problem Description

For years, systems have used a client-server type of architecture when thinking about a network focused system. These systems are useful but in general have a really problematic defect, latency. Latency is not that big of a problem is areas where a quick response is needed but in our growing world of IoT(internet of things) this is almost a requisite for a functional and fast system. This problem can be addressed by creating a system with layer in which the higher the layer the farther away the node is. If you need a quick and low latency task to be done by a node, the easiest way to reduce latency is to have the node as close as possible, like for example a cheap embedded system that is replicated around a city inside the area you live. The name of this solution is called Cloud Edge Computing.

5 General objective

Implement a solution using a Cloud Edge Computing Architecture comprised of three layers.

6 Specific objectives

6.1 First objective

Implement three layers of the system using Wind River Linux for the embedded devices.

6.2 Second objective

Generate an example to test the system.

6.3 Third objective

Compare using a simple latency test with a single server architecture.

7 Stakeholders

The stakeholders for this project include my supervisor as the most interested part and the company interested in having an implementation of a cloud edge computing with its products.

8 Solution description

This solution is divided in three layers that are interconnected and redirect traffic if necessary, the entrance of the system is the first layer.

8.1 First layer

The first layer is comprised of an embedded system or multiple embedded systems in which is works as the low latency entrance of the system. This layer is based on Wind River Linux and is used with kubernetes, in this case there are two options, k3s or OverC with kubernetes.

This layer is in charge of processing the request from the app and redirecting it to the next layer if necessary. This layer is also in charge of processing low latency critical tasks.

8.2 Second layer

The second layer is a server running on Linux in which is in charge of doing mid level processing tasks, it is also in charge of redirecting traffic to the third layer in case it is necessary. This layer is replicated less than the first layer but more than the third layer.

8.3 Third layer

This layer is the last layer and it is in charge of high level processing tasks. This layer is only one server in our case and in production it should be a couple more but still less than the second and first layer. This layer in general has the highest latency.

9 Deliverables and Criteria of acceptance

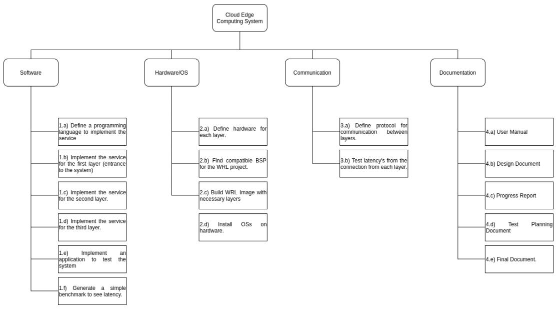
Cloud Edge Computing System
Software
Hardware/OS
Communication
Documentation
1.a) Define a programming language to implement the service
1.b) Implement the service for the first layer (entrance to the system)
1.c) Implement the service for the second layer.
1.d) Implement the service for the third layer.
1.e) Implement an application to test the system
1.f) Generate a simple benchmark to see latency.
2.a) Define hardware for each layer.
2.b) Find compatible BSP for the WRL project.
2.c) Build WRL Image with necessary layers
2.d) Install OSs on hardware.
3.a) Define protocol for communication between layers.
3.b) Test latency's from the connection from each layer.
4.a) User Manual
4.b) Design Document
4.c) Progress Report
4.d) Test Planning Document
4.e) Final Document.

Figure 1: Work breakdown structure.

Table 1: Breakdown of work and its acceptance criteria

Code	Description	Criteria of acceptance
1.a	To implement the service of the Cloud edge computing system, it is necessary to define a programming language. The options are C, C++ and Nodejs.	The defined language has a basis to prove its election as the primary language to use in the system.
1.b	Fully Implement the service in the first layer.	The first layer is already accessible and can be used to make tests.
1.c	Fully Implement the service in the second layer.	The second layer is fully functional and accessible and is available to connect from the first layer
1.d	Fully Implement the service in the third layer.	The third layer is fully functional and accessible and is available to connect from the first layer
1.e	Implement the image processing application to test the functionality of the system	Images can be sent, processed and stored on the correct layer. First layer is simple color processing, second layer is face recognition and third layer is storage.
1.f	Using simple calculations, read and document latency's obtained from the first second and third layer	Obtain an average from each layer of the observed latency.
2.a	Define what hardware to use for each layer.	Define and demonstrate why the hardware was chosen for the specific layer.
2.b	Based on the chosen embedded device, search for a compatible BSP.	Obtained and integrated chosen BSP.
2.c	Build the Wind RIver Linux image with the required layers and packages to enable the necessary libraries.	Generate a bootable WRL image.
3.a	Define a protocol to connect the layers with low latency.	Integrate and demonstrate functionality and connectivity.
3.b	Test latency's in between the connection of the layers.	Obtain an average from each layer of the observed latency.
4.a	User manual	Supervisor validated.
4.b	Design document	Supervisor validated.
4.c	Progress report	Supervisor validated.
4.d	Test planning document	Supervisor validated.
4.e	Final document	Supervisor validated.

10 Activities and effort budget

Code	Name	Hours
A-00	Obtain requirements	5
A-01	User manual	2
A-02	Design document	15
A-03	Requirement document	2
A-04	Progress reports	21
A-05	Test planning document	3
A-06	Final Document	14
A-07	Investigate and select programming language	2
A-08	Define hardware for each layer.	2
A-09	Find compatible BSP for the WRL project.	2
A-10	Build WRL Image with necessary layers	10
A-11	Install OSs on hardware.	8
A-11	Define protocol for communication between layers.	2
A-12	Implement the service for the first layer (entrance to the system).	15
A-13	Implement the service for the second layer.	15
A-14	Implement the service for the third layer.	15
A-15	Test latency's from the connection from each layer.	2
A-16	Implement an application to test the system	25

With a total of 160 hours of work and having available around 10 weeks of work time, it is required to do 16 hours of work time per week.

11 Risk analysis

Table 2:

Type	Hazard	Impact	Mitigation
Personal	Sickness	Would have to halt the project.	Work extra hours afterwards
Personal	Supervisor not available	Difficulty obtaining requirements and observations	Work on other objectives until available.
Hardware	Damaged device	Halting the project	Looking for a similar device.
Hardware	Faulty device	Slowing the project	Looking for a similar device.
Connection	No internet	Halting the project	Searching for a location with internet access
Connection	Faulty layer connection protocols	Slowing down the project	Use extra hours to debug faulty connection
Software	Selected programming language generates difficulties integrating with WRL	Slowing down the project	Try using another programming language

12 Work schedule

Table 3: Work schedule per week.

Code	Start	End	Hours
A-00	10/02/2020	14/02/2020	5
A-01	9/05/2020	13/05/2020	2
A-02	16/05/2020	27/05/2020	15
A-03	24/02/2020	28/02/2020	2
A-04	09/03/2020	15/05/2020	21
A-05	18/05/2020	23/05/2020	3
A-06	9/03/2020	17/04/2020	14
A-07	9/03/2020	13/03/2020	2
A-08	9/03/2020	13/03/2020	2
A-09	9/03/2020	13/03/2020	2
A-10	9/03/2020	13/03/2020	10
A-11	16/03/2020	20/03/2020	8
A-11	23/03/2020	27/03/2020	2
A-12	30/03/2020	10/04/2020	15
A-13	13/04/2020	01/05/2020	15
A-14	04/05/2020	08/05/2020	15
A-15	11/05/2020	15/05/2020	2
A-16	18/05/2020	29/05/2020	25