Report on Implementation
(Offloading Decision Engine app for Mobile Cloud Applications)

Aditya Khune

December 7, 2014

Contents

Abstract

In this document I have explained the Offloading Decision Engine smartphone app that I have created for an Android device. This kind of decision engine is required to help the code offloading process work efficiently. In [1] the authors have proposed a fuzzy decision engine for code offloading, that considers both mobile and cloud variables. I have implemented a similar engine with relevant parameters and with slightly different rules. I have also demonstrated the working of the app. Java code that was developed for the app is shown in the end. I have used Android Studio platform to develop this app for the Google Nexus 4 smartphone on my Linux machine.

1 Introduction

Mobile cloud applications are considered as the next generation of mobile applications, due to their promise of linked and elastic computational cloud functionality. In the offloading model, a mobile application is analyzed so that the most computational expensive operations can and offloaded for remote processing. Offloading is studied extensively for the development of mobile cloud applications. Authors in [1] propose some strategies to enrich the offloading decision process with help of a fuzzy logic engine which will decide whether to offload a processing of certain application or not depending upon device's contextual parameters and cloud parameters.

I have created a similar app which takes input from the user manually for the required parameters and outputs the decision whether to offload or not, and whether to offload on local servers or remote servers.

2 Implementation

In this section I have described the main steps of the implementation. Also all the rules considered for the fuzzy logic and the outcome of the decision engine can be found in the attached excel sheet named 'Offloading Rules and Outputs'.

2.1 Mobile offloading logic

At the mobile platform level, the device uses a decision engine based on fuzzy logic, which is utilized to combine n number of variables, which are to be obtained from the overall mobile cloud architecture. In the current implementation these parameters are entered manually by the user. Following are the Fuzzy sets and Some of the rules considered:

Fuzzy sets considered

- Bandwidth = Speed_Low, Speed_Normal, Speed_High
- WiFi = available, not available
- Data transfered = Data_Small, Data_Medium, Data_Big
- CPU instance = CPU_Low, CPU_Normal, CPU_High

Some of the Rules considered

- Remote Processing = Speed_High AND Data_Small AND CPU_Normal
- Remote Processing = Speed_Low AND Data_Small AND CPU_High
- Remote Processing = Normal_Low AND Data_Small AND CPU_High
- Local Processing = Speed_High AND Data_Small AND CPU_Low
- Local Processing = Speed_Low AND Data_Medium AND CPU_Normal
- Offload on Local Servers = Remote Processing AND WiFi ON
- Offload on Remote Servers = Remote Processing AND WiFi OFF

Let us see what these parameters define. Bandwidth available to user device can be Low, Normal or High. Data that is used by the application can affect the decision of offloading if the Data is too big the offloading can be expensive. CPU instance required by the application can be Low, Normal and High depending upon the computational requirements of a particular application. If the WiFi is available to the user then it makes more sense to offload on the local servers rather than the remote servers. So here I am assuming there are multiple locations available with the user where he can offload his application processing and data. After defining these parameters I have assigned grade of truth values to each of the set considered which fuzzy logic uses to classify the outputs.

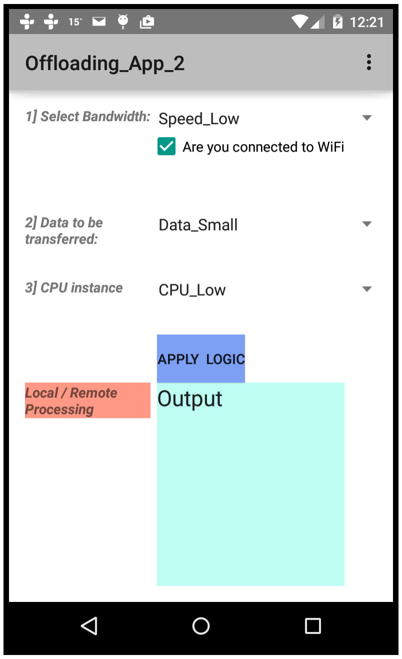
Now let us have a look on the user interface of our offloading app in Figure 1.

As I have said in my current implementation all these parameters are to be filled by user manually. I have created a single screen app where the user can choose from the available options, the Bandwidth available, Data to be transferred, CPU instance requirement of the app, is WiFi available with the user, etc. After giving all these inputs we click on the 'APPLY LOGIC' button which can be seen in blue colour. The decision of the engine will be seen in the Big Output Box on the right bottom corner.

In Figure 2 we can see the options to choose from drop down box for the available parameters. In Figure 3 we can see the decision engine output whether to do a Local Processing or Remote Processing, and if we are offloading then is it on Local Servers or Remote Servers.

3 Conclusion and Next Steps

I have created an app similar to the prototype presented by authors of [1] with some added parameters. All the parameters that are used by the engine are provided manually. In my next steps I will try to up come with an app which takes these parameters automatically from the contextual information of the device and the computing requirements of the app. I would also like to use machine learning techniques such as Clustering or Linear Discriminant analysis to classify the offloading possibilities in order to make an adaptive offloading engine.



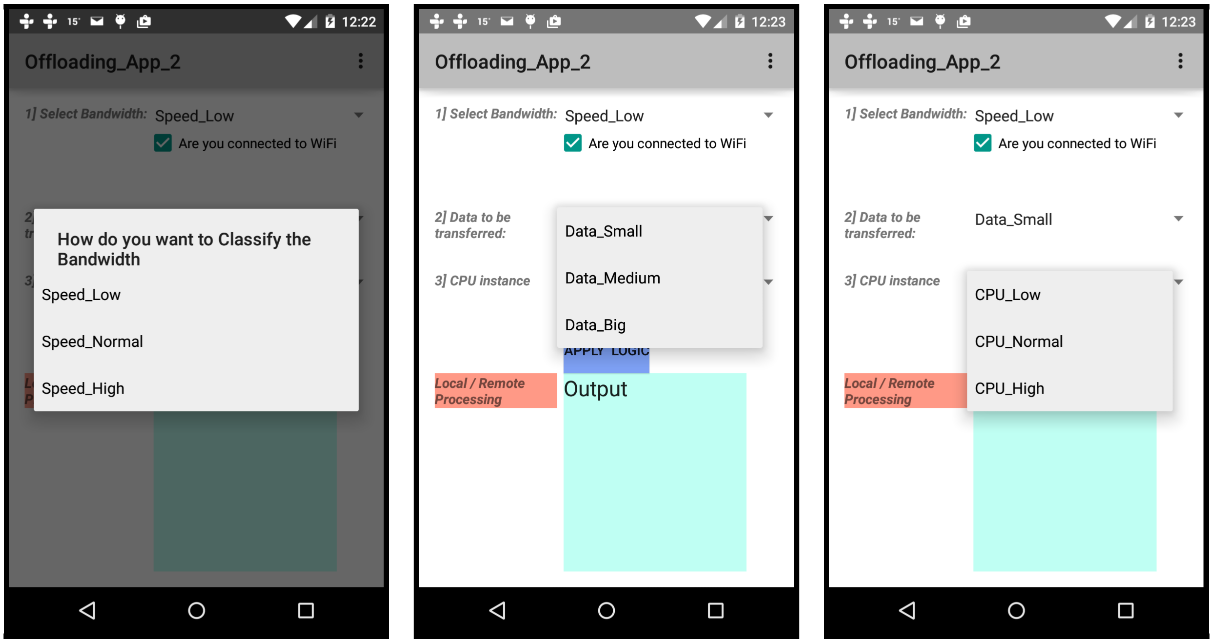
1] Select Bandwidth: Speed_Low

☑ Are you connected to WiFi

2] Data to be transferred: Data_Small

3] CPU instance CPU_Low

Figure 1: Offloading App Activity



1] Select Bandwidth: Speed_Low

☑ Are you connected to WiFi

How do you want to Classify the Bandwidth

- Speed_Low
- Speed_Normal
- Speed_High

1] Select Bandwidth: Speed_Low

☑ Are you connected to WiFi

2] Data to be transferred:

- Data_Small
- Data_Medium
- Data_Big

3] CPU instance

1] Select Bandwidth: Speed_Low

☑ Are you connected to WiFi

2] Data to be transferred: Data_Small

3] CPU instance

- CPU_Low
- CPU_Normal
- CPU_High

(a) Select Bandwidth

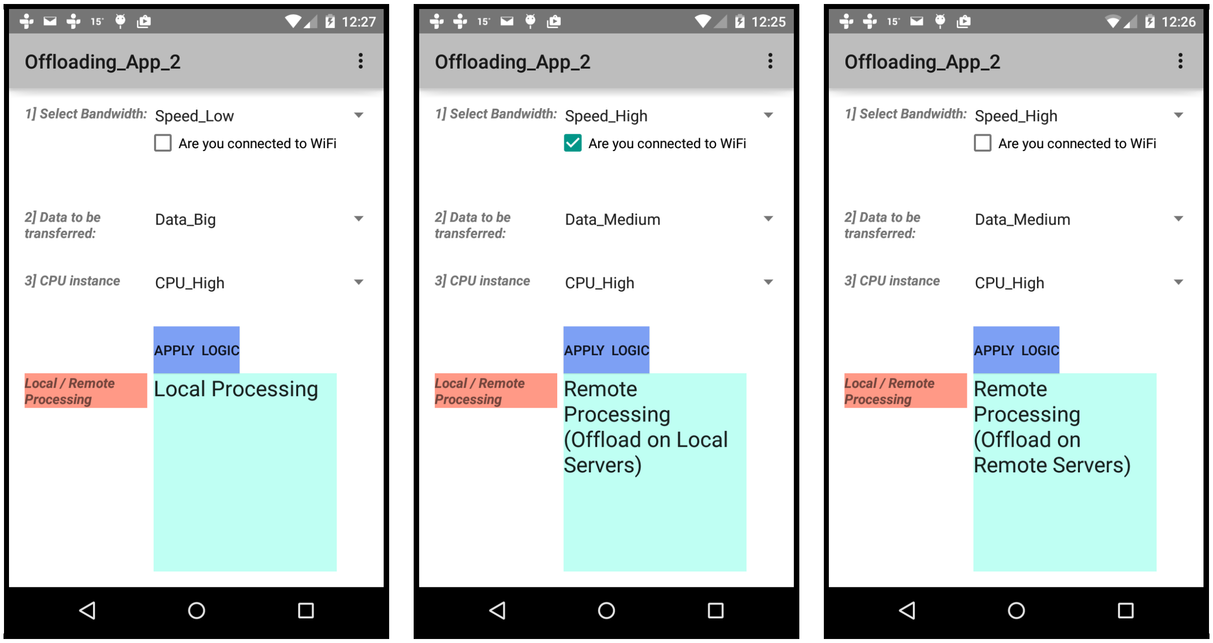
(b) Select Data

(c) Select CPU instance

Figure 2: Select Parameters Manually

References

[1] H. R. Flores Macario and S. Srirama, "Adaptive code offloading for mobile cloud applications: Exploiting fuzzy sets and evidence-based learning," in *Proceeding of the fourth ACM workshop on Mobile cloud computing and services*, pp. 9–16, ACM, 2013.



Offloading_App_2

1] Select Bandwidth: Speed_Low

□ Are you connected to WiFi

2] Data to be transferred: Data_Big

3] CPU instance CPU_High

APPLY LOGIC

Local / Remote Processing: Local Processing

Offloading_App_2

1] Select Bandwidth: Speed_High

☑ Are you connected to WiFi

2] Data to be transferred: Data_Medium

3] CPU instance CPU_High

APPLY LOGIC

Local / Remote Processing: Remote Processing (Offload on Local Servers)

Offloading_App_2

1] Select Bandwidth: Speed_High

□ Are you connected to WiFi

2] Data to be transferred: Data_Medium

3] CPU instance CPU_High

APPLY LOGIC

Local / Remote Processing: Remote Processing (Offload on Remote Servers)

(a) Local Processing (b) Offload on Local Servers (c) Offload on Remote Servers

Figure 3: Output of Decision Engine