Report-3
Choosing between available Network (3G, 4G and WLAN) efficiently to Optimize Offloading performance of the Smartphone, while transporting data, computing on Cloud

Aditya Khune

August 28, 2015

Contents

Abstract

Offloading has been widely considered for saving Energy and increasing responsiveness of the mobile devices. Important research works in this area propose an Offloading Decision Engine based on some strategy to help smartphones give accurate offloading decision logic. A consistent network performance is critical for a Stable offloading decision engine. However, such consistency is difficult to achieve because of unstable network quality and frequent user movements.

With recent advent of advanced network technologies such as 4G LTE networks, there has been increased interest in the offloading domain. But Research shows that LTE is less power efficient compared to WiFi, and drains battery faster than 3G [1].

In this paper we have studied the problem of Network Inconsistency while Smartphone computation offloading. We have proposed a Reinforcement Learning(RL) based offloading system to choose between 3G, 4G and WiFi efficiently. we have created a prototype app and tested it with varying network available (AT & T's 3G, 4G and Comcast's WiFi) on 3 different Smartphones (Amazon Fire phone, Samsung S3 and LG Nexus). Our results indicate that our RL-based offloading system gives more accurate decision engine than other prior works.

Chapter 1

Network Inconsistency

Many prior works in Offloading have proposed an offloading decision engine which will consider the parameters on the device and on cloud to make a correct offloading decision. An Offloading Decision Engine requires a consistent network performance for offloading. However, such consistency is difficult to achieve because of frequent mobile user movements and unstable network quality which varies depending upon the Location of the device, load on the network. The power consumed by the network radio interface is known to contribute a considerable fraction of the total device power.

With recent advent of 4G LTE networks, there has been increased interest in the offloading domain, but Research shows that LTE is as much as 23 times less power efficient compared to WiFi, and even less power efficient than 3G [1]. 4G phones are supposed to be even faster, but that's not always the case.

1.1 Need to choose right network while Offloading

With the advances in networking technology we have 4G available to us, but it is seen that 4G consumes more energy than 3G and WiFi.

In general, anything involving transferring large amounts of data gets a big boost from 4G. If you live in an area that doesn't have 4G coverage, there's no advantage to a 4G phone. In fact, our experiments show that there are serious battery life problems if you buy an LTE phone and don't disable 4G LTE, as the radio's search for a non-existent signal will drain your battery quickly.

Level of connection on 4g will greatly affect your battery life. Meaning, if you have a weak signal your device will be using more power to get data sent and received to and from the network, which will eat your battery up. A strong 4g signal will of course use less battery, the biggest problem is the constant switching from 4G to 3G and back again.

To counter the Network Inconsistency on the device and to optimize the offloading experience we have proposed a novel offloading technique based on Machine Learning which helps a device choose between the available networks with varying conditions. In the next Section we have described this system in detail.

Chapter 2

Smart Offloading Decision Engines to Counter Network Inconsistency

2.1 Reinforcement Learning(RL) Decision Engine

In this section we have explored Reinforcement Learning to create a decision engine for the offloading process. Reinforcement learning is learning by interacting with an environment. Reinforcement Learning (RL) differs from standard supervised learning in that correct input/output pairs are never presented.

Here we are using RL for a Single-step decision problem; RL can be used for Multi-Step decision problems. I have explained how the offloading decision process can benefit from Reinforcement Learning and also presented different scenarios where it can used to improve the overall offloading decision process.

State-Action Value Function

The state-action value function is a function of both state and action and its value is a prediction of the expected sum of future reinforcements.

Set of Possible State Values

In our Algorithm State Values are discrete values.

- Location = Home, Office, Traveling
- Data Transferred = Data_Small, Data_Medium, Data_Large
- Time = Morning, Afternoon, Evening, Night

The offloading system extracts the above parameters (such as Location, Time) from the contextual information of the Smartphone device.

Set of Action Values

Values of Possible Actions are also discrete values.

- Offload using 3G
- Offload using 4G
- Offload using Wi-Fi

Representing the Q Table with Penalty values

Say we are in state S_t at time t. Upon taking action a_t from that state we observe the one step reinforcement p. After this we choose another action a_{t+1} in the same state, this continues until we have explored all the Actions. The cost or Penalty of an action taken from a state is the reinforcement for that action from that state. Use the returned Penalty value to choose best Action, where we want to minimize the Penalty. When there is a change in User's Location for instance User moves from Home to Office, We continue the same steps mentioned above.

	Offload Using 3G	Offload using 4G	Offload using Wi-Fi
P_b	P_{b3G}	P_{b4G}	P_{bWIFI}
P_t	P_{t3G}	P_{t4G}	P_{tWIFI}

$$p = P_{b3G} * x + P_{t3G} * y$$

P_b - Penalty for Battery units consumed
P_t - Penalty for time required to offload
P_{b3G} - Penalty for Battery units consumed while offloading with 3G
P_{t3G} - Penalty for time to do the offloading operation with 3G
P_{b4G} - Penalty for Battery units consumed while offloading with 4G
P_{t4G} - Penalty for time to do the offloading operation with 4G
P_{bWIFI} - Penalty for Battery units consumed while offloading with Wi-Fi
P_{tWIFI} - Penalty for time to do the offloading operation with Wi-Fi

Figure 2.1: PenaltyEquation

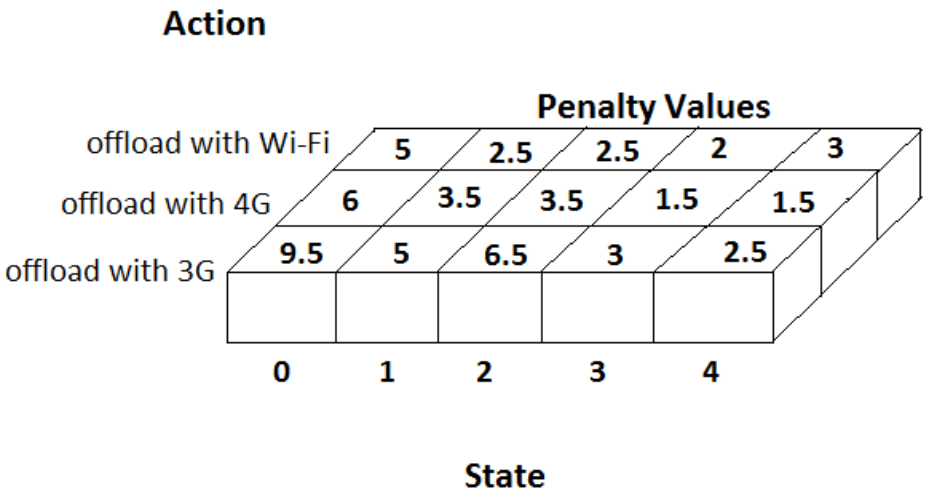

Figure 2.2: Representing the Q Table with Penalty values

Algorithm of our Offloading system

We have developed a partially working prototype of the proposed system as the proof of concept.The prototype implements the basic functionality needed to show the feasibility of the proposed mobile cloud architecture. Below is the algorithm:

1. Detect change in the Device's contextual information (Parameters such as Location (Home, Office, etc) and Time-Period (Morning, Afternoon, Evening and Night))
2. Activate 3G radio interface of the device.
3. Download a file ($data\_size = data\_small$) from the Cloud. Measure the Battery and time consumed for the operation.
4. Upload same file on the cloud. Measure the Battery and time consumed for the operation.
5. Calculate the penalty p with the help of equation in figure 2.1
6. form a Key-Value pair as follows:

   {Location-TimePeriod-data_size:penalty} where

   Key = Location-TimePeriod-data_size

   Value = Calculated penalty p
7. Update the Q-table with the calculated penalty values as shown in figure 2.2.
8. Repeat steps 2-7 above for ($data\_size = data\_mediumanddata\_large$)
9. Repeat steps 2-8 above for 4G and Wi-Fi connection if available.

The application with offloading mechanism will refer to the updated Q-table to make a right decision of choosing the network for offloading the required computing and data on the cloud. The application will look for the minimum penalty values available in the Q-table. The values of constants are $x = 0.5$ and $y = 0.5$ for both optimized battery and performance time, if user prefers better battery performance than elapsed processing time of the application then we change the values to $x = 0.9$ and $y = 0.1$. For example when the user is traveling he might prefer to save his battery power than worrying about the processing time; whereas when the battery charge isn't a sudden problem for the user then he might choose for optimized performance time, in that case we need to change the constant value to $x = 0.1$ and $y = 0.9$.

The Network Inconsistency is also taken care in our Algorithm as we have included the Location parameters in the state values to train our $Q - function$.

Chapter 3

Prior Related Works

3.1 Fuzzy Logic Decision Engine

In [2] the authors have proposed fuzzy decision engine for code offloading, that considers both mobile and cloud variables. 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. Fuzzy Logic Decision Engine works in three steps namely: Fuzzyfication, Inference and Defuzzification.

Let us see these steps in detail: 1) In Fuzzification input data is converted into linguistic variables, which are assigned to a specific membership function. 2) A reasoning engine is applied to the variables, which makes an inference based on a set of rules. Finally 3) The output from reasoning engine are mapped to linguistic variable sets again(aka defuzzyification).

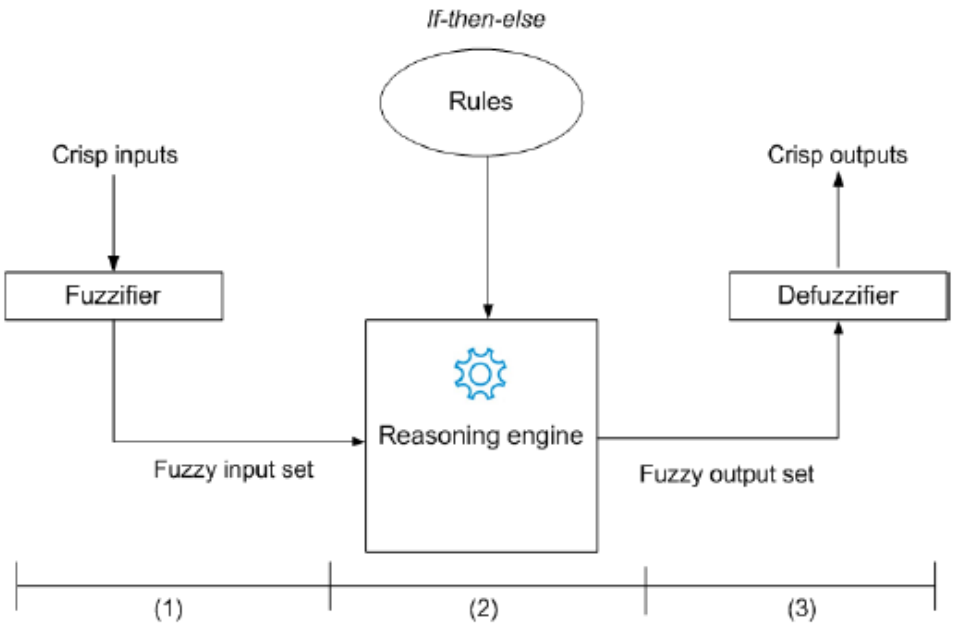

Figure 3.1: Offloading Engine based on Fuzzy Logic

3.2 Problem with this approach

The distribution of different technologies around the world varies significantly. In India, a country with limited broadband infrastructure, 2G remains in active use, while the U.S. and Mexico lean heavily on

Wi-Fi connections. In figure 3.2 we have segmented bandwidth into four different buckets: 0 - 150, 150 - 550, 550 - 2000, and 2000+ kbps. We have mapped them into one of the four Connection Classes Poor, Moderate, Good, and Excellent. Fuzzy Logic is based on simple if-else statements, which is like a hardcoded logic. Because of such variations in the different technologies around the world, it is difficult to rely on the fuzzy logic decision engine presented in [2]. This is because the app developers will have to customize the decision engines depending upon which part of the world the device lies.

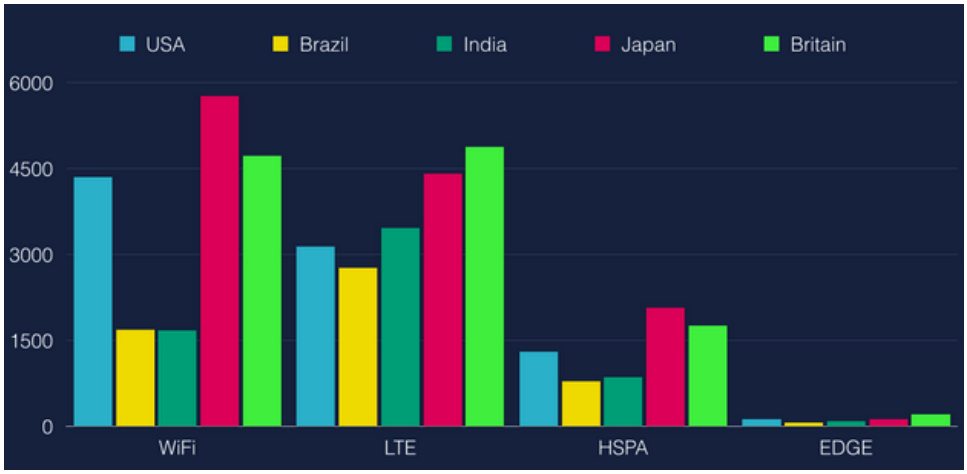

Figure 3.2: Varying Internet Speed across different Countries (Source: Facebook code Community)

3.3 Smartphone Energizer (SE)

Smartphone Energizer [3] is a supervised learning-based technique for energy efficient computation offloading. In this work authors propose a adaptive, and context-aware offloading technique that uses Support Vector Regression (SVR) and several contextual features to predict the remote execution time and energy consumption then takes the offloading decision. The decision is taken so that offloading is guaranteed to optimize both the response time and energy consumption.

Smartphone Energizer Client (SEC) initially starts in the learning mode, in which it extracts the different network, device, and application features and stores them after each service invocation. After each local service invocation, the Smartphone Energizers profiler stores the context parameters which include the service identifier, input size, and output size along with the consumed energy and time during service execution. When the number of local service invocations exceeds the local learning capacity, the SEC switches to the remote execution by checking if there's a reachable offloading server, then the service will be installed on the server (for the first time only) and will be executed remotely, otherwise it will be executed locally.

3.4 Our work in comparison to the previous works

Reinforcement Learning(RL) is a Unsupervised Learning method, our Algorithm is simplistic in comparison to other works like Smartphone Energizer (SE) [3] which is a supervised learning method. Our results show that we can save upto 20%-30% battery power in the proposed Offloading system while we compare it with prior works ([3], [2]).

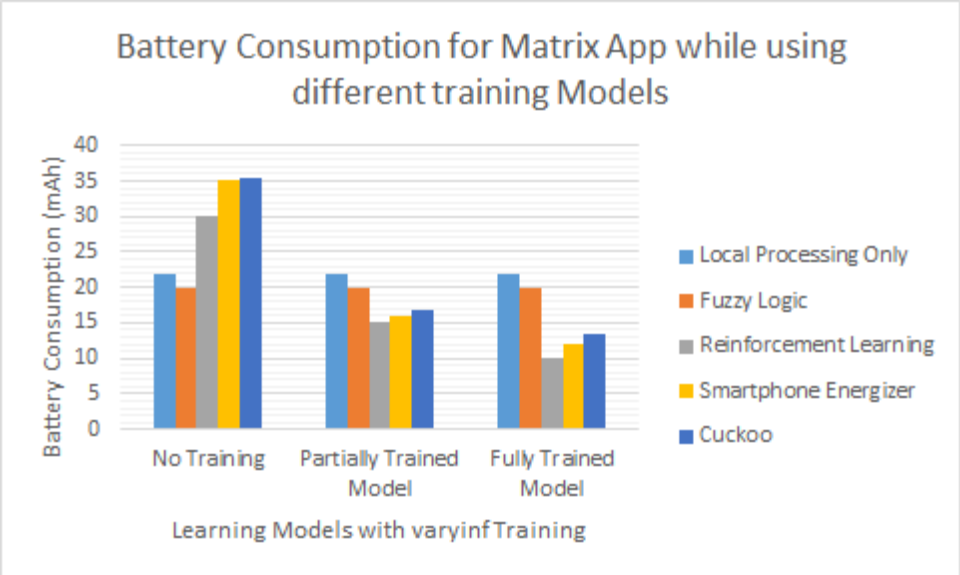

Figure 3.3: Energy Consumption by Different Models with Varying percentage of Training for Matrix Operation Benchmark App

Bibliography

[1] J. Huang, F. Qian, A. Gerber, Z. M. Mao, S. Sen, and O. Spatscheck, "A close examination of performance and power characteristics of 4g lte networks," in *Proceedings of the 10th international conference on Mobile systems, applications, and services*, pp. 225–238, ACM, 2012.

[2] 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.

[3] A. Khairy, H. H. Ammar, and R. Bahgat, "Smartphone energizer: Extending smartphone's battery life with smart offloading," in *Wireless Communications and Mobile Computing Conference (IWCMC), 2013 9th International*, pp. 329–336, IEEE, 2013.