النموذج قبل التدقيق

Our Application is hosted on App Engine by using the App Engine backend. Google Cloud Endpoints consists of tools, libraries and capabilities that generate APIs and client libraries from an App Engine application. In developing mobile applications, Endpoints provides a simple way to develop a shared web backend and because the API backend is an App Engine, the app can use all of the services and features available in App Engine, such as Datastore, Google Cloud Storage, Mail, Url Fetch, Task Queues, and so forth [15]. In our application, we used (GCM) service to inform user about which filter executes at this moment in progress bar. The user will be notified if the filter finished and the next one starts.

++

result :

Figure 4-1 (A) shows the execution time of a low processing case under different input sizes and different network status. In all cases, the value of cloud is more than that of the smartphone, and the gap between them expands as input grows. In the first test, the difference between cloud-based and locally-based is a little bit, while with the second test when the input size value 3MB is almost doubled. Finally, the last test is almost three times the perfect time with Worst network connection. The reason lies in the additional communication cost between cloud and smartphone, which surpasses the processing cost. Therefore, whatever the state of the network in the low processing option, the app should be run on locally.

We measured the execution time of medium processing filters and studied how the execution time in medium case depended on application input parameters and different networks situation. The running on the cloud does not always consume more time than running locally, as shown in Figure 4-1 (B). The reason is that the execution time for medium processing on the mobile will cost time and will grow as we increase the image size. Moreover, the gap between running on the cloud and locally becomes smaller and is reflected for the cloud. We can see from the figure that the execution time in cases of medium or large image sizes and with a good connection will cost less in the cloud. For this option, if the input size more than 3MB and there is an ideal network is available the app should be run remotely.

Figure 4-1 (C) shows the execution time of the high processing case under different input sizes and different network status. In all cases, the execution time values on the cloud are less than those of the smartphone, and the gap between them expands, as the input grows, and reaches up to half of the time with large input sizes. In the first test, the difference between cloud-based and local-based is a little bit, while with the second test is less with high-speed network and a little bit more with slow network connectivity. Finally, the last test when an input size more than 6 MB running application on the cloud take almost half time compared with local based on the perfect network case but with worst network connection take less time than local. Therefore, whatever the state of the network in the high processing case is, the app should be run remotely on the cloud.

A. execution time	B. execution time	**C.** execution time
A. Energy consumption	B. **Energy consumption**	**C. Energy consumption**

Figure 4-1

++

The results of the power consumption were shown in the second row from figure 4-1. As we can see from Figure 4-1 (A) the difference in energy consumption between the mobile and the cloud, will vary depending on the image size and the network status. In the case of small image size, the difference in energy consumption between the mobile and the cloud is very small, but for a slow network, the difference increased for local. The same situation with medium image size, but the difference has increased with slow network status. Finally, in large image size, the energy consumed in the cloud is less than that in mobile. Also, more energy was consumed in the cloud if the network in the worst case. Therefore, depending on the state of the network in the low processing option, the app should be run on locally or remotely.

Figure 4-1 (B) illustrates the energy consumption of medium processing under different input size. We see that only in the first test, with small input size that equal "50 KB”, the energy consumed is smaller on the local device. In other cases, the energy consumed is smaller in the cloud. And for medium processing option running the application in the cloud for any network status will save the energy consumed. As we can see in (C) for any input size or any network status, the energy consumed will be less if the app is running in the cloud. With high computing applications, running the application on the cloud costs the mobile a little energy compared to running it locally. The reason because high intensive processing in mobile device consumes resources and requires much RAM, memory, and energy.

++

The objective of this section is to analyze the performance of running our application remotely under different network situation. We tested our application in different latencies, bandwidth and various response times. The effect of bandwidth will lead to delay as well as to packet loss. The result of this section was the offloading decision should constantly measure the network condition and estimates the bandwidth and latency.

4.2.1

We make our test in three-network status. The first network has a bandwidth value 5.88 Mbps and latency is almost 20ms. The second network has more latency about 35ms and bandwidth available was 2.50 Mbps. The third network that has worst situation with low bandwidth 1.05 Mbps and lower delay roughly 18ms

A.	B.

Figure 4-2-1

The execution time consists of three parts as shown Upload time, Processing time and Download time. Figure 4-2-1(A) presents the changes in execution time with different network status. We can see in that the best performance was in the first network. The second network has overhead in processing time due to delay rate and also middle upload ratio. Finally, the third network has the worst performance although it has lowest delay ,but with low bandwidth the upload and download cost a lot of time.

Energy consumption of the application is running in the cloud is influenced by bandwidth and delay in different networks as shown in figure 4-2-1(B). An Upload energy, Processing energy and Download energy composes the total energy. In the first network, the power consumption on the cloud is decreasing with bandwidth increasing and decrease delays. The second network has more power consumption than the first one due to small bandwidth and delay rate (35ms). Finally, the third network has the biggest energy consumption value because the bandwidth decreases until (1.05 Mbps).

+++

4.2.2

We tested the application in three networks status. The first network has a bandwidth value 5.88 Mbps and latency is almost 20ms. The second network that has worst situation with latency value 50ms and bandwidth available was 2.00 Mbps. The third network that has an adequate situation with low bandwidth 2.50 Mbps and lowest delay roughly 30ms.

A.	B.

Figure 4-2-2 Execution time and Energy consumption for Medium processing under different networks

Figure 4-2-2 (A) describes how bandwidth and delay affects in run medium processing apps over cloud under different network status. We can see from figure the execution time was very fast in first network compared to other networks. The application has achieved the best performance due to the high bandwidth and low latency. The application has achieved the worst performance with a second network due to a high delay that effect in processing time in cloud and also middle upload ratio. Finally, the application in the third network has acceptable performance with middle delay, but middle bandwidth, the upload cost a bit of time.

The energy consumed by the medium processing application on the cloud is also increased or decreased depending on available bandwidth and delay ratio in different networks as shown in figure 4-2-2 (B). As we see the energy has been saved significantly in the first network. The second network has the largest value of the energy consumption duo to small bandwidth and high delay. Finally, the third network has the acceptable energy consumption value compared with other networks status

+++

4.2.3

We make our test in three-network status. The first network has a lower bandwidth value about 1.00 Mbps and latency is almost 35ms. The second network has more latency about 25ms and bandwidth available was 5.00 Mbps. The third network with high bandwidth 5.05 Mbps and delay near 30ms.

A.	B.

Figure 4-2-3 Execution time and Energy consumption for High processing under different networks

In most cases in high processing apps as proved before, the value of execution time on the cloud is always less than that on a smartphone. Therefore, the total time as shown in the figure will affect by network status that has different bandwidth and variable delay. In the case of high processing, large image, the effect of input size has a great impact in the offloading process because that involve large amounts of data transfer between mobile and cloud. Also, the complexity of computing in high intensive processing filters requires offload computing to the cloud server.

These types of applications obviously require more bandwidth, low latency, because the response time will affect by both factors. The power consumption on a cloud is decreasing with high processing applications for most networks, but also the total value affected by the current situation of the network.

+++

Core function :

If the difference between the energy consumption for offloading and executing on the device is high, the need for offloading is high and if it’s low, the Energy factor is low. For those values where the difference is neither low nor high, the energy level left in the device is used for evaluating the offloading factor.

Table خطأ! لا يوجد نص من النمط المعين في المستند.-1: Decision Making for Time Factor.

Energy- BiggerValue	Elevel	Ediff	Efactor

Table 3-1 gives the values for quantifying the offloading factor. In the table below, we explain the fuzzy logic rules that used to calculate the energy factor and we assign an Efactor level to make the decision more accuracy. The table contains the values that affect the decision. We started by Energy “Bigger value “that represents the place that consumes more energy to execute the process and it could be a cloud or mobile device. Then we have Energy level that represents Energy available in the mobile device at the moment and we assign have three levels (low, medium and high). E_{diff} represent the difference between local and cloud. For calculating the level of E_{diff} we use a function that returns the level whether low, medium, and high, given an integer number and the maximum scale.

The number is compared with the scale, if the number is within the first third of the scale, the level is low. If the number is within the two-thirds of the scale, the level is medium, else the level is high. The Last value it represents the decision for Efactor and its level.

Time factor measures the time for offloading the execution to the Cloud. This time factor for offloading is related to the execution time (T_{rt}) for offload and retrieve results from the cloud and the current received signal strength (RSS_{net}) of the wireless medium. If the execution time is high, medium or low and the signal strength is low, the time factor for offloading is low, but if the response time is not high and the signal strength falls in the medium range, the time factor is quantified as a medium. Also, if the execution time is high and the signal strength is medium, the time factor for offloading is high. Table 3-2 gives the values for quantifying the Time factor. This quantification along with the offloading factor values help in deciding if the decision is to offload or not.

Table 3-2: Decision Making for Time Factor.

ExecutionTime- BiggerValue	RSS	ETdiff	Tfactor

We explained the fuzzy logic rules on table above to calculate the time factor and also assign Tfactor level to make exact decision. The table contains the values that affect decision start by Execution Time "Bigger value " that represents the place that takes more time to execute the process and it could be a cloud or mobile device. Then we have RSS that represents received signal strength and it have three levels (low, medium and high). ETdiff represent the difference Execution Time between local and cloud. And we assign three levels (low, medium and high). We calculate these levels as before in Ediff . The last value it represents the decision for Tfactor and its level

Table 3-3: Decision making for offloading.

Efactor Deccision	Efactor Decision level	Tfactor Decision level	Decision

Table 3-3 gives the decision for offloading based on the Energy factor and Time factor values. The service is offloaded when the need for offloading is high and when the Time for offloading is low. In the case where the Energy factor is neither high nor low and Time factor is low, the execution is offloaded to cloud or not offloaded depend on the level of the factors.

++

Vertical handover

The (HDE) engine starts when the application starts execution. It monitors the network status and detects any change. The engine also takes necessary handoff decision to connect with a different network, in case the connection with the current is at stake, in order to improve its energy efficiency and reduce latency.

Based upon the number of interaction data transmitted per transaction, and the current network status after applying handoff, the decision-making algorithm decides whether to place the execution locally or remotely over the Cloud. The engine works in application background and takes necessary handoff. Also, updates the wireless network info and informs (ODE) with new information to take accurate decision.

In android platform, Automatic handover between 3G and WLAN networks is typically done when the current network link is going down. When the Android device connects to a Wi-Fi network, the platform automatically closes the 3G data connection. In contrast, when the Wi-Fi network is unavailable (or the user moves the device disconnects from the Wi-Fi network), the platform reactivates the 3Gdata connection.

Execution time is measured on Android mobile device, and Google's App Engine servers. Also, Energy consumed is measured by comparing energy consumed when running the app locally and when running it remotely. These measurements will provide a means to analyze the viability of mobile cloud computing and evaluate whether executing code remotely on more powerful servers is advantageous. Time needed to communicate with remote servers will also be measured in order to analyze the added communication cost of remote execution. Furthermore, with a level and complexity of processing, measuring time is important in terms of user experience and the application performance We evaluate both energy consumption and execution time of three types of processing, as shown in table 4-1, with respect to many factors. We will examine how these factors will affect energy consumption and execution time of the applications. For a specified factor, we evaluate its influence on both energy consumption and execution time of three applications under different ranges. We will provide the results obtained from our experiments with a focus on execution time and power consumption.

++

result :

In this section, we introduce the result of our decision engine with the effect of three factors, including input size, bandwidth and CPU processor speed. Some factors related to application, wireless status and mobile device specifications.

4.3.1:

Figure 4-3-1 shows the execution time and power consumption of low CPU-intensive processing under different input sizes. In the figure, the red line represents the execution time and power consumption depend on our decision engine. The green line represents running application locally in the smartphone and the purple line represents running application remotely over the cloud.

First, as we can see in " a) case of Low processing" the value of the execution time when app is running on the cloud is more than when it is running locally, and the gap between them expands as input grows. The reason lies in the time cost due to data transmission between cloud and mobile device, which is more than that cost by running the app locally. Therefore, for this kind of processing, the application should run on mobile device. In this case, we see our decision engine can make a wise offloading decision. The energy consumed by low CPU-intensive processing running on the cloud is much more than that on a smartphone. The reason is that the energy consumed by applying the filters on a mobile device is less than that due to data transmission, including sending input data and receiving results. We also see that with a low CPU-intensive processing application, we should always directly run locally.

In medium processing case as part b in figure. we can see our engine take accurate decision. Due to the fact that the execution time of applying filters on the cloud is less than that on a mobile device when the input size is larger than 1MB, we should run the application on cloud in that case. The results for part b that illustrate the energy consumed by medium CPU-intensive processing is interesting. When the input size is smaller than 1MB, the application running on cloud costs more energy than running locally. After that, the power consumption on smartphone is larger than that on the cloud. Whenever the size of the input is increased, the cost of processing increases. So, in this case processing in the cloud became better because the cost of processing is more than the cost of the transmission data.

A. **Execution Time in low processing**	**A.Energy consumption low processing**
B. **Execution Time in medium processing**	B. **Energy consumption in medium processing**
C. **Execution Time in high processing**	C. **Energy consumption in high processing**

Figure 4-3-1: Execution Time and Energy consumption of three applications under different input size

Furthermore, the execution time costs on the cloud is less than that of the smartphone for high processing as is shown part-c from fig. As the input size increases our engine decides to run the app remotely on the cloud especially when the input size is greater than 1MB. However, the power consumption of running high CPU-intensive processing on a mobile device is much more than running it on a cloud, as we can see from Fig part-c. Because of the high complexity of filters on a mobile device, the power consumed due to the processing is much more than transmitting data. We also observe that with this kind of application, the decision should always be offloaded to the cloud. Moreover, we can see that our engine saves much more energy for high CPU-intensive processing under these circumstances.

As we see in all cases, our engine takes the decision to run the app locally or on the cloud based on the best results. In most cases, it takes the same value or value close to the best performance. Our decision engine still makes a wise decision; thus, the user’s experience is improved.

4.3.2 Bh:

The second factor that we introduce is associated with the effect of wireless network status. The available bandwidth is an important factor and directly affects the app performance especially when it varies from one network to another in heterogeneous environments. We study its effect to our engine and our app performance.

The first column in figure 4-3-2 shows the relationship between the execution time of three applications and bandwidth. And the second column represents Energy consumption of three applications under different bandwidth. In the first case part-a, low processing case, the running on the cloud always takes more time than running locally. The execution time decreases gradually as the available bandwidth increases. Therefore, our decision engine can make a wise offloading decision for this kind of processing by running on mobile device. In the same case the value of power consumption on the cloud is always higher than that on a mobile device. So, for this kind of the apps should always be run locally. The explanation for this is the energy consumed by processing computation on a mobile device is less than that consumed by data transmission. However, energy consumption remotely and locally is getting closer and closer as the bandwidth is increased.

A.Execution Time in low processing	**A.Energy consumption low processing**
B. **Execution Time in medium processing**	B. **Energy consumption in medium processing**
C. **Execution Time in high processing**	C. **Energy consumption in high processing**

Figure 4-3-2: Execution Time and Energy consumption of three applications under different bandwidth

In medium processing case, as in figure part-b we see that the results are little different. The execution time on the cloud is less than that on the mobile device when bandwidth is more than 3.5MB/s, while the opposite occurs when bandwidth is smaller. Considering the energy consumption of app on cloud and mobile device, our decision engine offloads processing to a cloud and it makes the right decision again. CPU-intensive processing spends more time on the mobile device than on cloud as shown in figure part-c.

Furthermore, the power consumed on a mobile device is kept the same, and we can also see this in Figure 4-3-2 (b) and (c) for medium and high CPU-intensive processing running under such circumstances. The power consumption on a mobile device is much more than that on the cloud. Additionally, the power consumption on cloud decreases as the bandwidth is increased. So, we should always offload apps to the cloud and this will save much energy for users. Our engine runs apps locally or in the cloud and makes the right decision to get the best results.

4.3.3

The last factor is related to the mobile device capabilities. The smartphone specifications have an important factor directly affecting the performance of applications for support contact with any wireless networks. Figure 4-3-3 shows the relationship between execution time and energy consumption of the three applications and CPU processor speed.

B. **Execution Time in low processing**	**A.Energy consumption low processing**
B. **Execution Time in medium processing**	B. **Energy consumption in medium processing**
C. **Execution Time in high processing**	C. **Energy consumption in high processing**

Figure 4-3-3: Execution Time and Energy consumption of three applications under CPU processor speed

In the first case part-a, which represents a low processing, the running on the cloud always takes less time than running locally. The execution time in the cloud almost has a constant value because the cloud capabilities did not change or affect by smartphone specifications. The execution time on local decreases gradually as the CPU processor speed is increased until it reaches a limiting value with fast processors. Therefore, our decision engine can make a wise offloading decision for this kind of processing by offloading to the cloud.

Also for medium and high processing, we have the same result as shown in fig part b and c. The execution time on the cloud is less than that on a mobile device and the gap between them decreases as the CPU processor speed is increased. Therefore, our decision engine can make a wise offloading decision for by offloading to cloud for all cases. As we see for all cases as shown in the figure, our engine always offloads apps to the cloud. The value of power consumption on the cloud is always smaller than that on a mobile device, and energy consumption of running on a cloud almost does not change. That is to say, for this kind of apps, the decision should always be run remotely

++

In conclusion, this research work shows obvious advantages of mobile cloud computing technology. And by applying engines to decide offloading and vertical handover we introduce an improvement in both the application execution time and the energy consumed in mobile device. These results prove that cloud computing is very probable, and that offloading computations to the cloud server is a viable, timesaving option. As long as network speeds are suitable, it is advantageous to offload computationally intensive applications to a more powerful server. Not only is it advantageous, but also necessary in some situations, as the mobile device is unable to even run certain applications due to memory restrictions or for limited mobile specifications. In general, the most cloud platforms show the advantage of offloading applications to the cloud resources in the context of providing a SaaS. By outsourcing computation offloading to the backend servers, the simple mobile device becomes more powerful. However, there is no best or simple implementation of mobile cloud computing. Options include dynamic vs. static code offload, method vs. OS migration, and various connections protocols. Different applications have different resource requirements affecting the best possible connection to the cloud. Ultimately, MCC application should be built to adapt intelligently to different changes in the surrounding networks, device capabilities and application requirement. To make the device decides which is the best for their particular application.

النموذج بعد التدقيق

This application is hosted on App Engine by using the App Engine backend. Google Cloud Endpoints consist of tools, libraries and capabilities. They generate APIs and client libraries from an App Engine application. The Endpoints provide a simple way to develop a shared web backend, which are used in developing the mobile applications. Because the API backend is an App Engine, the app can uses all of the services and features available in the App Engine, such as Datastore, Google Cloud Storage, Mail, Url Fetch, Task Queues, and so forth [15]. The application uses (GCM) service to inform the user about which filter is being executed at that moment in the progress bar. The user will also be notified when a filter is finished or started.

++

Results:

Figure 4-1 (A) shows the execution time of a low processing case under different input sizes and network status. In all cases, the value of cloud is more than the value of smartphone. The gap between them expands as the input grows. In the first test, the difference between the cloud-based and locally-based is a little bit. However, in the second test, the input size value is 3MB, which is almost the doubled value. Finally, the last test is almost three times the perfect time with the worst network connection. The reason lies in the additional communication cost of the cloud and smartphone, which surpasses the processing cost. Therefore, the app should run locally whatever the status of network in the low processing option.

The execution time of medium processing filters is measured. In addition, this work studies how the execution time in medium case depends on the application input parameters and different networks situation. It is clear that running the application on the cloud does not always consume more time than running it locally. That is shown in Figure 4-1 (B). The reason is the execution time for medium processing on the mobile will cost time. It will grow as the image size is increased. Moreover, the gap between running the application on the cloud and locally becomes smaller. In addition, it is reflected for the cloud. It is obvious from the Figure that the execution time in cases of medium or large image sizes, with a good connection, will cost less in the cloud. Regarding this option, if the input size is more than 3MB, with an ideal network available, the app should always run remotely.

Figure 4-1 (C) shows the execution time of high processing case under different input sizes and network status. In all cases, the execution time values on the cloud are less than the execution time values on the smartphone. The gap between them expands, as the input grows. The gab reaches up to half time with the large input sizes. In the first test, the difference between the cloud-based and locally based is a little bit. In the second test, the difference is less with the high-speed network, and a little bit more with the slow network connectivity. Finally, in the last test, when the input size is more than 6 MB, running the application on the cloud will takes almost half time. That is compared to the locally based on the perfect network case. However, with the worst network connection, it takes less time than locally. Therefore, the app should always run remotely on the cloud, whatever the status of network in the high processing case.

A. Execution time	B. Execution time	**C.** Execution time
A. Energy consumption	B. **Energy consumption**	**C. Energy consumption**

Figure 4-1

++

The results of power consumption are shown in the second row of the Figure 4-1. It is clear in the Figure 4-

1 (A) that the difference in energy consumption between the mobile and cloud will vary. Since it depends on the image size and network status. In the case of small image size, the difference in energy consumption between the mobile and cloud is very small. However, the difference is increased in the locally based for the slow network. The same situation happens with the medium image size. But the difference is increased in the slow network status. Finally, the energy consumed in the cloud is less than the energy consumed in the mobile in the large image size. In addition, the cloud will consume more energy, if the network is in its worst case. Therefore, the app should run locally and remotely depending on the status of network in the low processing option.

Figure 4-1 (B) illustrates the energy consumption of medium processing under different input size. In the first test, it is obvious that the energy consumed is smaller in the locally based device, with the small input sizes only that are equal to "50 KB". In other cases, the energy consumed is smaller on the cloud-based device. Regarding the medium processing option, running the application on the cloud will save the energy consumed whatever the network status. It is noticeable in (C) that the energy consumed will be less if the app is running on the cloud whatever the input size and network status. However, with the high computing applications, running the application on the cloud will cost the mobile a little energy compared to running it locally. That is because the high intensive processing in the mobile device consumes resources. In addition, it requires much RAM, memory, and energy.

+++

The objective of this section is to analyze the performance of running the application remotely under different network situation. The application is tested in different latencies, bandwidth available and response times. The effect of bandwidth available will lead to delay and packet loss. The results of this section is that the offloading decision should measure the network condition constantly as well as estimate the bandwidth available and latency.

4.2.1

The test is performed on three-network status. The first network bandwidth available is 5.88 Mbps. Moreover, its latency is almost 20ms. The second network has more latency, about 35ms. In addition, its bandwidth available is 2.50 Mbps. The third network has the worst situation. It has a low bandwidth, which is 1.05 Mbps. Also, it has a low delay, which is 18ms approximately.

A.	B.

Figure 4-2-1

The execution time consists of three parts as shown: Upload time, Processing time and Download time. The Figure 4-2-1(A) presents the changes in execution time with the different network status. It is clear that the best performance is in the first network. The second network has overhead in processing time due to delay rate. In addition, it has a medium upload ratio. Finally, the third network has the worst performance, although it has the lowest delay. However, with a low bandwidth, the upload and download cost a lot of time.

The energy consumption of the application, which is running on the cloud is influenced by the bandwidth available and delay in different networks, as shown in the Figure 4-2-1(B). An Upload energy, Processing energy, and Download energy compose the total energy. In the first network, the power consumption on the cloud is decreased because the bandwidth available is increased and the delays is decreased. The second network has more power consumption than the first one due to the small bandwidth available and delay rate (35ms). Finally, the third network has the biggest energy consumption value, because the bandwidth available is decreased until (1.05 Mbps).

+++

4.2.2

The application is tested on three networks status. The first network bandwidth available is 5.88 Mbps, and its latency is almost 20ms. The second network has the worst situation. Its latency value is 50ms, and its bandwidth available is 2.00 Mbps. The third network has the standard situation. I has a low bandwidth available, which is 2.50 Mbps. In addition, it has the lowest delay, which is 30ms approximately.

A.	B.

Figure 4-2-2 Execution time and Energy consumption for Medium processing under different networks

Figure 4-2-2 (A) describes how the bandwidth available and delay effect in running the medium processing apps on the cloud under different network status. It is obvious in the Figure that the execution time is very fast at the first network compared to the other networks. The application achieves the best performance due to the high bandwidth available and the low latency. The application achieves the worst performance with the second network due to the high delay, which effects the processing time in the cloud, in addition to the medium upload ratio. Finally, the application in the third network has an acceptable performance. It has a medium delay and bandwidth. However, the upload costs a bit of time.

On the other hand, the energy consumed by the medium processing application on the cloud is increased or decreased. It depends on the bandwidth available and delay ratio in different networks, as shown in the Figure 4-2-2 (B). It is noticeable that the energy is saved significantly in the first network. The second network has the largest value of the energy consumption due to the small bandwidth available and high delay. Finally, the third network has the acceptable value of energy consumption compared to the other networks status.

+++

4.2.3

The test is performed to three-network status. The first network has a low bandwidth available, which is about 1.00 Mbps. In addition, its latency is almost 35ms. The second network has more latency, which is about 25ms. In addition, its bandwidth available is 5.00 Mbps. The third network has a high bandwidth, which is about 5.05 Mbps. Moreover, its delay is 30ms approximately.

A.	B.

Figure 4-2-3 Execution time and Energy consumption for High processing under different networks

In most cases in the high processing apps, as proved before, the value of execution time on the cloud is always less than value of execution time on the smartphone. Therefore, the total time, as shown in the Figure, is effected by the network status, which has different bandwidth available and delay. In the case of high processing apps with the large image, the effect of input size has a great impact on the offloading process, because of the large amounts of data transfer between the mobile and cloud. In addition, the complexity of computing on the high intensive processing filters requires offload computing on the cloud server.

These types of applications obviously require more bandwidth available and low latency, because the response time is effect by the both factors. The power consumption on the cloud is decreased in the high processing applications for most networks. However, the total value is effected by the current situation of network.

+++

Core function:

If the difference between the energy consumption for offloading and executing to the device is high, the need for offloading is high. If the difference is low, the energy factor is low. The energy level in the device is used for evaluating the offloading factor, if the difference is neither low nor high.

Table خطأ! لا يوجد نص من النمط المعين في المستند.-4: Decision Making for Time Factor.

Energy- BiggerValue	Elevel	Ediff	Efactor

Table 3-1 gives the values for quantifying the offloading factor. The table below explains the fuzzy logic rules that used to calculate the energy factor. Efactor level is assigned to make the decision more accurate. The table contains the values that effect the decision. Energy-BiggerValue represents the place that consumes more energy to execute the process. It could be a cloud or mobile device. The Elevel represents the energy available in the mobile device now. The Elevel has three levels: low, medium and high. E_{diff} represents the difference between the cloud-based and locally based. A function is used for calculating the level of E_{diff}. It returns the level whether low, medium, and high. In addition, it gives an integer number and maximum scale.

The number is compared with the scale. If the number is within the first third of the scale, the level is low. If the number is within the two-thirds of the scale, the level is medium. If the number is within the last third of the scale, the level is high. The last value represents the decision for the Efactor and its level.

Time factor measures the time for offloading the execution to the Cloud. The time factor is related to the execution time (T_{rt}) for offloading and retrieving the results from the cloud and current received signal strength (RSS_{net}) of the wireless medium. The time factor for offloading is low, if the execution time is high, medium or low, and the signal strength is low. However, the time factor is quantified as a medium, if the response time is not high, and the signal strength falls in the medium range. Moreover, the time factor for offloading is high, if the execution time is high, and the signal strength is medium. Table 3-2 gives the values for quantifying the Time factor. This quantification, along with the values of offloading factor help in deciding whether to offload or not.

Table 3-5: Decision Making for Time Factor.

ExecutionTime-BiggerValue	RSS	ETdiff	Tfactor

The fuzzy logic rules are explained in the table above. They are used to calculate the time factor and assign Tfactor level to make the exact decision. The table contains the values that effect the decision. It begins with the ExecutionTime-BiggerValue, which represents the place that takes more time to execute the process; it could be a cloud or mobile device. Then, it includes the RSS that represents the received signal strength, which has three levels: low, medium and high. Moreover, it includes the ETdiff, which represents the difference in the Execution Time between cloud-based and the locally based. The ETdiff has three levels: low, medium and high. The last value of the table represents the decision for the Tfactor and its level.

Table 3-6: Decision Making for Offloading.

Efactor Deccision	Efactor Decision level	Tfactor Decision level	Decision

Table 3-3 gives the decision for offloading based on the values of energy factor and Time factor. The service is offloaded when the need for offloading is high, and when the time for offloading is low. If the energy factor is neither high nor low, and the time factor is low, the decision of execution offloading to the cloud or not depends on the level of the factors.

++

Vertical handover:

The (HDE) engine starts when the application starts execution. It monitors the network status and detects any change. If the connection with the current network is at stake, in order to improve its energy efficiency and reduce its latency, the engine takes the necessary handoff decision to connect with a different network.

The decision-making algorithm decides whether to place the execution locally or remotely on the Cloud based on the number of interaction data transmitted per transaction and the current network status after applying the handoff. The engine works in the application background to takes necessary handoff. In addition, it updates the info of wireless network. Moreover, it informs the (ODE) with new information to take the accurate decision.

In the android platform, the automatic handover between 3G and WLAN networks is done usually when the current network link is going down. When the Android device connects to the Wi-Fi network, the platform automatically closes the 3G data connection. In contrast, when the Wi-Fi network is unavailable (or the user disconnects the Wi-Fi network from the device), the platform reactivates the 3G data connection.

The execution time is measured on the Android mobile device, and the Servers of Google's App Engine. The energy consumed is measured by comparing the energy consumed when running the app locally, and when running it remotely. These measurements will provide means to analyze the viability of mobile cloud computing, and evaluate whether executing the code remotely on more powerful servers is advantageous or not. The time needed to communicate with the remote servers is measured to analyze the communication added costs of the remote execution. Furthermore, with the level and complexity of processing, the measurement of time is important in the terms of user experience and application performance. The energy consumption and execution time of three types of processing are evaluated; as shown in table 4-1 with respect to many factors. The table examines how these factors effect the energy consumption and execution time of the applications. The table evaluates the influence of each specified factor on both the energy consumption and execution time of three applications under different ranges. The results obtained from the experiments are shown below with the focus on the execution time and power consumption.

++

Results:

This section shows the results of decision engine, with the effect of three factors; including the input size, bandwidth available and CPU processor speed. Some factors related to the application, wireless status, and mobile device specifications.

4.3.1:

Figure 4-3-1 shows the execution time and power consumption of the low CPU-intensive processing under different input sizes. The red line in the Figure represents the execution time and power consumption, which depends on the decision engine. The green line represents running the application locally on the smartphone. The purple line represents running the application remotely on the cloud.

First, it is clear in the case of low processing that the value of the execution time when the app is running on the cloud is more than the value of the execution time when the app is running locally. The gap between them expands on the input grows. Because the time cost due to the data transmission between the cloud and mobile device is more than the time cost of running the app locally. Therefore, regarding this kind of processing, the application should run on the mobile device. In this case, the decision engine can make a wise offloading decision. The energy consumed by the low CPU-intensive processing running on the cloud is much more than running on the smartphone. The reason is that the energy consumed by applying the filters on the mobile device is less than the energy consumed by applying the filters locally due to the data transmission, which includes sending input data and receiving results. It is obvious that with the low CPU-intensive processing application, it is better to run it locally.

In the medium processing case, it is noticeable that the engine takes the accurate decision as shown in Part B of the Figure. Because the execution time of applying filters on the cloud is less than the execution time on a mobile device when the input size is larger than 1MB. So, it is recommended to run the application on the cloud in that case. The results in Part B illustrates that the energy consumed by the medium CPU-intensive processing is increasing when the input size is smaller than 1MB. Since the application that runs on the cloud costs more energy than the application that runs locally. The power consumption on the smartphone is larger than the power consumption on the cloud. Whenever the input size is increased, the cost of processing is increased, too. Therefore, processing in the cloud is better because the cost of processing is more than the cost of transmission data.

A. Execution Time in low processing	**A. Energy consumption low processing**
B. **Execution Time in medium processing**	B. **Energy consumption in medium processing**
C. **Execution Time in high processing**	C. **Energy consumption in high processing**

Figure 4-3-1: Execution Time and Energy consumption of three applications under different input size

Furthermore, the costs of execution time on the cloud are less than the costs of execution time on the smartphone at the high processing as shown in Part C of the Figure. As the input size is increased, the engine decides to run the app remotely on the cloud, especially when the input size is greater than 1MB. However, the power consumption of running high CPU-intensive processing on the mobile device is much more than the power consumption on the cloud as shown in Part C of the Figure. The power consumed due to the processing is much more than transmitting data, because of the high complexity of filters on the mobile device. It is observed that with such kind of the application, the decision should always be offloaded on the cloud. Moreover, it is obvious that the engine saves much more energy at the high CPU-intensive processing under these circumstances.

As it is noticeable in the all cases, the engine takes the decision to run the app locally or on the cloud based on the best results. In most cases, the engine takes the same value or a very close value to the best performance. So, the decision engine makes a wise decision, which improves the user’s experience.

4.3.2 Bh:

The second factor is associated with the effect of wireless network status. The bandwidth available is an important factor. Since it directly affects the performance of app, especially when it varies from one network to another in the heterogeneous environments. This project studies this factor and its effect to the engine and performance of app.

The first column in the Figure 4-3-2 shows the relationship between the execution time of three applications and bandwidth. The second column represents the energy consumption of three applications under different bandwidth. In the first case is a low processing case as shown in Part A. It shows that running the app on the cloud always takes more time than running it locally. The execution time is decreased gradually as the bandwidth available is increased. Therefore, the decision engine can make a wise offloading decision for this kind of processing by running the app on the mobile device. In the same case, the value of power consumption on the cloud is always higher than the value of power consumption on the mobile device. Therefore, it is better to run this kind of apps locally. The explanation for this is that the energy consumed by the processing computation on the mobile device is less than the energy consumed by the data transmission. However, the energy consumption whether remotely or locally is getting closer and closer as the bandwidth available is increased.

A. Execution Time in low processing	**A. Energy consumption low processing**
B. **Execution Time in medium processing**	B. **Energy consumption in medium processing**
C. **Execution Time in high processing**	C. **Energy consumption in high processing**

Figure 4-3-2: Execution Time and Energy consumption of three applications under different bandwidth

In the medium processing case, as in Part B of the Figure, the results are little different. The execution time on the cloud is less than the execution time on the mobile device when the bandwidth available is more than 3.5MB/s. The opposite occurs when the bandwidth available is smaller than 3.5MB/s. Regarding the energy consumption of app on the cloud and mobile device, the decision engine offloads processing to the cloud, and it makes the right decision again. The CPU-intensive processing spends more time on the mobile device than on the cloud as shown in Part C of the Figure.

Furthermore, the power consumed on the mobile device is kept the same. It is clear in the Figure 4-3-2 (b) and (c) that the power consumption on the mobile device is much more than the power consumption on the cloud for the medium and high CPU-intensive processing running under such circumstances. In addition, the power consumption on the cloud is decreased as the bandwidth available is increased. So, it is recommended to offload the apps on the cloud because that saves much energy for users. The engine runs the app locally or on the cloud, and makes the right decision to get the best results.

4.3.3

The last factor is related to the capabilities of mobile device. The smartphone specifications play as an important factor, since they directly effect the performance of applications in supporting the connection with any wireless networks. Figure 4-3-3 shows the relationship between the execution time and energy consumption of the three applications and CPU processor speed.

C. **Execution Time in low processing**	**A. Energy consumption low processing**
B. **Execution Time in medium processing**	B. **Energy consumption in medium processing**
C. **Execution Time in high processing**	C. **Energy consumption in high processing**

Figure 4-3-3: Execution Time and Energy consumption of three applications under CPU processor speed

The first case, Part A, represents a low processing case. In this case, running the app on the cloud always takes less time than running it locally. The execution time on the cloud almost has a constant value, because the cloud capabilities do not change or effect by the smartphone specifications. The execution time on the locally based device is decreased gradually as the CPU processor speed is increased; until it reaches a limiting value with the fast processors. Therefore, the decision engine can make a wise offloading decision for this kind of processing by offloading to the cloud.

The medium and high processing have the same results as shown in Part B and C of the Figure. The execution time on the cloud is less than the execution time on the mobile device. The gap between them is decreased as the CPU processor speed is increased. Therefore, the decision engine can make a wise offloading decision by offloading to the cloud for all the cases. The engine always offloads the apps to the cloud for all the cases as shown in the Figure. The value of power consumption on the cloud is always smaller than the value of power consumption on the mobile device. In addition, the energy consumption of running the apps on the cloud is almost not changed. It is worthy to say that for this kind of apps, the decision should always be running the apps remotely

++

In conclusion, this project shows obvious advantages of the mobile cloud computing technology. In addition, by applying engines to decide the offloading and vertical handover, that will improve both the application execution time and the energy consumed by the mobile device. These results prove that the cloud computing is very probable, and the offloading computation to the cloud server is a viable timesaving option. As long as the network speeds are suitable. It is an advantage to offload the computationally intensive applications to a more powerful server. It is not only an advantage, but it is also necessary in some situations. E.g., it is necessary when the mobile device is unable to run certain applications due to memory restrictions or limited mobile specifications. Generally, the most cloud platforms show the advantage of offloading the applications to the cloud resources in the context of providing a SaaS. By outsourcing computation offloading to the backend servers, the simple mobile device becomes more powerful. However, there is no best or simple implementation of the mobile cloud computing. Options include dynamic vs. static code offload, method vs. OS migration, and various connections protocols. Different applications have different resource requirements effecting the best possible connection to the cloud. Finally, the MCC application should be built to adapt intelligently to different changes in the surrounding networks, device capabilities and application requirements. That is necessary to make the device decides the best particular application for it.