Abstract: Smart Mobile Phone devices are widely used for current daily usages. However it has the limited Capacity in terms of the parameters as Memory, CPU, Power Consumption and network connectivity. So the main issue is to address the problem of complex and huge applications which run on mobile phones, to decide offloading schedule dynamically, so that to optimize the battery lifetime, memory usage and CPU utilization. In this paper first to extract and estimate the run time parameters in mobile devices with given specification for applications like sorting and image processing techniques. With simulation of task manager and operating system interfaces. Secondly to find out the the optimal values using with the programming parameters in case the remote cloud execution by offloading. In the later case hence we combine these two experiments to identify the suitable configuration and types of applications and complexity of the same. In additionally once decided the offloading decision, using Cloudme tool to authenticate and encrypted the application data before it send to cloud.

Keywords: Mobile Computing, Mobile Cloud Computing, Computation Offloading, Security, MatLab and Cloudme.

I. INTRODUCTION

Smart Mobile phones are mainly used for storing and processing of data parallel, but continuous this process is not possible because of due to limitations of battery, memory and CPU usage. So overcome this problem using the mobile cloud computing (MCC). MCC is the two portions one is mobile computing and another one is cloud computing. Mobile phone works are offloading to the remote, the huge number of resources, so it increases the battery life time, memory usage and CPU utilization. In cloud computing clients works with different application via with internet, it refers the infrastructure for both data storage and data processing. On demand basis MCC is the main useful for minimizing the development and execution cost of mobile applications. In some complex or huge computation task mobile phone is not possible computation to the work, so the work is offload to cloud. Some main parameters are memory, battery and CPU usage.

Fig.1 Mobile Offloading Technique

In this Paper, novel work to extract and estimate the battery usage, memory usage and CPU utilization based on application the run time. Next to find out the optimal values using the integer programming. Based on the optimal values offloading decision is made. Using java code in net beans we extract the values of parameters for the applications as (Quick sort, Face detection and Gaussian blur).We get the optimal values from the Objective function.

Based on the optimal values we decide the mobile application will work on locally or cloud. The optimization model is provides the offloading decision dynamically at run time based on the constraints such as CPU utilization, Memory and energy consumption. Four different types of mobile phones are used in the experimental studies to test the optimization model to improve the application performance additionally used the Cloudme tool for authenticate the data transferring for security purpose.

II. RESEARCH METHODOLOGY

Reader: This is initial module, it is responsible for read all parameters such as size, CPU, Memory and Power. Once all the values of availability in the mobile nodes tried to communicate the estimator module. Here as an example we use four mobile nodes parameters.

Fig.2 Available Capacity of Parameters
\[ \sum_{i=1}^{n} \text{memory}_i \times (1-x_i) + \sum_{i=1}^{n} \text{memory}_\text{Sec}_i \times x_i \leq M_{th} \]  

(2)

In memory two sections are used one execution in local another one is execution in remote. Once it decides made in offload prior it must be encrypted and send.\(M_{th}\) is the total size of memory.

**Estimator:** The Behavior for this module is to calculate the required parameters to computation the application. The values are extracted from the task manager by using through java code using net beans. Once the values are estimated based on the application (Quick sort, Face detection and Gaussian Blur) all are communicate to the Profile Module.

![Fig 3 Required Capacity of Parameters for the Applications](image)

**Profile:** This profile module is responsible for getting the values of Execution time, CPU utilization, Memory and Energy Consumption from the estimator module. Additionally to generate new database for updating each data, this process mainly used for decision maker module.

**Decision Maker:** After This is the core module of the framework, it contains the integer programming model in the runtime with three constraints based on Power Consumption, CPU usage and Memory Usage. Based on the four mobile nodes and objective function produced the optimal values for these nodes. Using the MatLab tool to getting to the optimal values for these constraints. Based on this optimal values we decided the execution run on locally or Remote (Cloud) \(x_i=0\) (local) or \(x_i=1\) (cloud)

The Objective function is

\[ \text{Min}(R_{\text{transfer}} + W_{\text{memory}} + W_{\text{memory}} + R_{\text{CPU}} + W_{\text{CPU}} + R_{\text{power}} + W_{\text{power}}) \in [0,1] \]  

(1)

where

\[ R_{\text{transfer}} = \sum_{i=1}^{n} \text{input}_i \times x_i \]

\[ R_{\text{memory}} = \sum_{i=1}^{n} \text{memory}_i \times (1-x_i) \]

\[ R_{\text{CPU}} = \sum_{i=1}^{n} \text{CPU}_i \times (1-x_i) \]

\[ R_{\text{power}} = \sum_{i=1}^{n} \text{power}_i \times (1-x_i) \]

\[ R_{\text{transfer}}, R_{\text{memory}}, R_{\text{CPU}} \] and \(R_{\text{power}}\) represent the Required for input size, Memory used, CPU utilization and Energy consumption for \(i\) respectively.

**Memory usage Constraints**

**Total Execution Time constraints:** In the processing time execution in the cloud less than the application run in mobile devices, \(S_M\) and \(S_C\) Processor speeds and \(C\) be the number instructions involved in the Methods. \(C_{\text{enc}}\) represents number of instructions encrypted in that prior to send \(D\) network bandwidth and \(B\) time required to transfer this data.

\[ \text{Exe}_{\text{time}\_\text{local}}>\text{Exe}_{\text{time}\_\text{remote}} \]

\[ \text{Exe}_{\text{time}\_\text{local}} = \frac{C}{S_{\text{sec}}} \times x_i \]

\[ \text{Exe}_{\text{time}\_\text{remote}} = (\frac{C}{S_{\text{sec}}} + \frac{C}{S_{\text{enc}}} + \frac{D}{B} + t_{\text{overhead}}) \times x_i \]

**Total energy consumption for application constraints:** \(P_M\) need to consume watts to execute the application in local, \(P_d\) idle time and \(P_{i}\) W for transferring data to the cloud.

\[ \text{Energy}_{\text{con}\_\text{local}}>\text{Energy}_{\text{con}\_\text{remote}} \]

\[ \text{Energy}_{\text{con}\_\text{local}} = P_m + S_{\text{M}} \times x_i \]

\[ \text{Energy}_{\text{con}\_\text{remote}} = (P_{\text{sec}} + S_{\text{M}}) + (P_{\text{sec}} + S_{\text{C}}) + (P_d + B) \times x_i \]

After the optimization model, we decided the execution place, if suppose the execution on cloud, before the values are authenticated through the Cloudme tool and also encrypted. After that the data should be sending to cloud. CloudMe is a file storage services operated by CloudMe AB that offers cloud storage file synchronization. We use separate user name and password. We create space for authenticate the security services.

![Fig 4 CloudMe Tool for Storing and Authenticate the Data Transfer](image)

III. FRAMEWORK EXECUTION FLOW

**Input:** Mobile phones Memory, CPU usage and Energy consumption

**Output:** Decide place for work (Local or Remote)

1. Read Mobile Nodes (M1, M2, M3, M4) //Consider four mobile i.e n=4
2. Each of i do
3. Read \(R_{\text{transfer}}, R_{\text{memory}}, R_{\text{CPU}}, R_{\text{power}}\) Output of Estimator
4. Get the solution from the optimization model to determine the decision variables \(x_1, x_2, x_3, x_4\)
5. if the decision is offloading
   //i.e \(x_i=1\)
6: send the application to cloud for remote execution
else
7: Execute the application locally.
end if

First we read the input parameters, consider four mobile nodes. For each nodes i read one by one. By using the Estimator output required CPU, Memory and Power consumption for applications. Second used the Required parameters to get the solution from optimization model to determine the decision for x1, x2, x3 and x4, if the decision is offloading, xi = 1 or application executed in local.

Fig. 5 Flowchart for Execution Flow

IV. RESULTS AND DISCUSSION

Tabulated for the Co-efficient of the parameters for input, memory and power are shown in table 1 to 3.

Table.1 Mobile nodes parameter for available and required value for quick sort application

<table>
<thead>
<tr>
<th>S.No</th>
<th>Mobile Nodes (M1, M2, M3, M4)</th>
<th>Input Parameters (Availability)</th>
<th>Output Parameters (Requirements)</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>CPU (KB)</td>
<td>Memory (KB)</td>
<td>Power</td>
</tr>
<tr>
<td>01</td>
<td>M1</td>
<td>69%</td>
<td>2090</td>
</tr>
<tr>
<td>02</td>
<td>M2</td>
<td>72%</td>
<td>2218</td>
</tr>
<tr>
<td>03</td>
<td>M3</td>
<td>68%</td>
<td>2199</td>
</tr>
<tr>
<td>04</td>
<td>M4</td>
<td>81%</td>
<td>2002</td>
</tr>
</tbody>
</table>

Table.2 Mobile nodes parameters for available and required value for Face detection application

<table>
<thead>
<tr>
<th>S.No</th>
<th>Mobile Nodes (M1, M2, M3, M4)</th>
<th>Input Parameters (Availability)</th>
<th>Output Parameters (Requirements)</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>CPU (KB)</td>
<td>Memory (KB)</td>
<td>Power</td>
</tr>
<tr>
<td>01</td>
<td>M1</td>
<td>69%</td>
<td>15185</td>
</tr>
<tr>
<td>02</td>
<td>M2</td>
<td>72%</td>
<td>18180</td>
</tr>
<tr>
<td>03</td>
<td>M3</td>
<td>68%</td>
<td>20100</td>
</tr>
<tr>
<td>04</td>
<td>M4</td>
<td>81%</td>
<td>22295</td>
</tr>
</tbody>
</table>

The results of quick sort application are shown in figure 6.

Fig. 6(a) Required CPU utilization

Fig. 6(b) Required Memory(KB)

Fig. 6(c) Required Energy Consumption
The results of face detection application are shown in figure 7.

In the proposed work, integer programming model is applied to the applications. Based on optimal values the execution place is decided locally or Cloud. But in that experimental work two different cases are used to prove it. Case one for application methods executed in locally. Another case for application methods executed in cloud.

In the solution the proposed integer programming model is applied to the constraints based on this to select the decision for these applications (sorting, image processing techniques). In that experimental work two cases are used. One is application executed in Mobile devices and another one is application executed in Cloud. Additionally cloudme used for to protect the data to secured transfer.

---

Fig.6(d) Size of the input

Fig.7(a) Required Memory

Fig.7(b) Required CPU Utilization

Fig.7(c) Required Size

Fig.7(d) Required Power Consumption

In the proposed work, integer programming model is applied to the applications. Based on optimal values the execution place is decided locally or Cloud. But in that experimental work two different cases are used to prove it. Case one for application methods executed in locally. Another case for application methods executed in cloud.

In the solution the proposed integer programming model is applied to the constraints based on this to select the decision for these applications (sorting, image processing techniques). In that experimental work two cases are used. One is application executed in Mobile devices and another one is application executed in Cloud. Additionally cloudme used for to protect the data to secured transfer.
Once the decision is made, to execute the application in cloud, using the Cloudme tool for storing and authenticate the data to encrypted. In addition simultaneous to execute the cloud to reduce the consumption of battery, Memory usage and execution time.

REFERENCES