

# Unified Ecosystem for Smartphones as Mobile Mini-Cloud Computing

## ***Abstract-***

In a research environment or a system environment where a large amount of computationally exhaustive task is required to be done, the company or organization demands a high expensive system or hosts a private cloud and pay for it. Imagine you have a set of peoples/employees in using your organization or research locality who uses your resource such as hotspots. Now if you could turn this into using the mobiles as a mini cloud for your system using their resources to get your task done. This enables an environment for bringing live mobile mini clouds and using them as IaaS (Infrastructure as a Service).

## **1. Introduction**

Let us first take a brief case of how are mobiles are used today. Mobiles today have a lot more computation power than expected. Even a low budget phone runs a Quad core processors with a 3 gigs of RAM or even more. Now the question how much of it do we really use? On a normal basis of usage of smart phone we hardly reach the least computation capability at all. In fact most of the times the phone seems to be idle.

The idea is to use this power which is unutilized. To do this the idea is to create a smart ecosystem which enables performing high level computations from this mobiles.

The ideology goes as:

1. Let the researchers load a job in the central system.
2. The researcher needs to split the job into smaller tasks.
3. When a mobile connects to the hotspot, the mobile is initiated a mini cloud for the server system.
4. Now server sends a packaged job to the mobile cloud assuming it as its infrastructure for execution of the job.
5. The mini cloud computes the job and send the result to the server.

6. If the mini cloud job result is received by the server it updates its central map.
7. When all the task are received in the central map, it updates the task to combine stage where the results are combined.
8. This again is performed by the mini clouds.
9. Finally when combine is completed, the final result is stored in the system.

Assume a scenario where such a system is installed in the railway station of India. Thousands of people connect to our railway hotspot. Using their computation power to get some exhaustive task done by the government can be a great resource.

## 2. Implementation

The implementation is done in 2 phases.

### *First Phase - Server:*

- The first phase is to build the Server end. The Server is designed in Java to be compatible across multiple platforms. The server is minimal and designed with polished interface using Sockets to enable communication with the mobile clients.
- When the client connects to the hotspot, the server sniffs it and sends a request to the client for connection with the server. The request is general handshake which happens at the socket level.
- When client is connected the server creates a thread dedicated to the client for managing with the client.
- Server has actually defined a matrix multiplication tasks of 20 matrices where each matrix is 1000\*1000 dimension long in floating type.
- Now comes the key part where the task is accomplished in levels:
  - Level 1: Multiply in pair of 2 matrix to get resultant 10 matrix
  - Level 2: Resultant 10 matrix are further multiplied in pairs to get 5 matrix
  - .
  - .
  - Till there is 1 result matrix.

- To send a job we have used Serialization in java which is a wrapper for the objects. Each task is wrapped and sent to the client mini cloud.
- The result is then again received by the server as a serialized object. Which stores the result in files.
- Each thread s destroyed as the task is completed successfully.
- Threads are synchronized in Java using the standard mutual exclusion packaging.

*Second Phase – Mobile Application as mini cloud:*

- The second phase was to create a mobile application.
- The application area currently covered is only for Android as of now.
- The application is responsible for first creating a connection with the server the instant when it connects to the hotspot. It supports the server three handshake for confirming the connection.
- Once it is connected it is ready to offer itself as Infrastructure as a Service to the server. In other words it has initiated itself as a mini cloud.
- It then is responsible to receiving the job which is packaged by Java serializer, deserialize it and complete the task in the method specified.
- It the packages the solution again with Java serializer and sends it to the Server.
- The computation is performed as a Service in Android, thus the front interface is of no use.
- However the interface is created which specifies
  - The server to which the client is acting as mini cloud.
  - The current state of mini cloud which can be:
    - Initializing
    - Computing the job
    - Completion of the job
  - The CPU usage by the system and the user.

### 3. Results

The system was tested with the multiple clients connected to the Server which was controlled in a laptop. The mobile applications were tested with set of test cases as shutting down when the task is on, running it in high load system etc.

The results seems to be pretty fair.

The computation task loaded was matrix multiplication tasks of 20 matrices where each matrix is 1000\*1000 dimension long in floating type.

### 4. Snapshots

Mobile App (ClientIaaS)

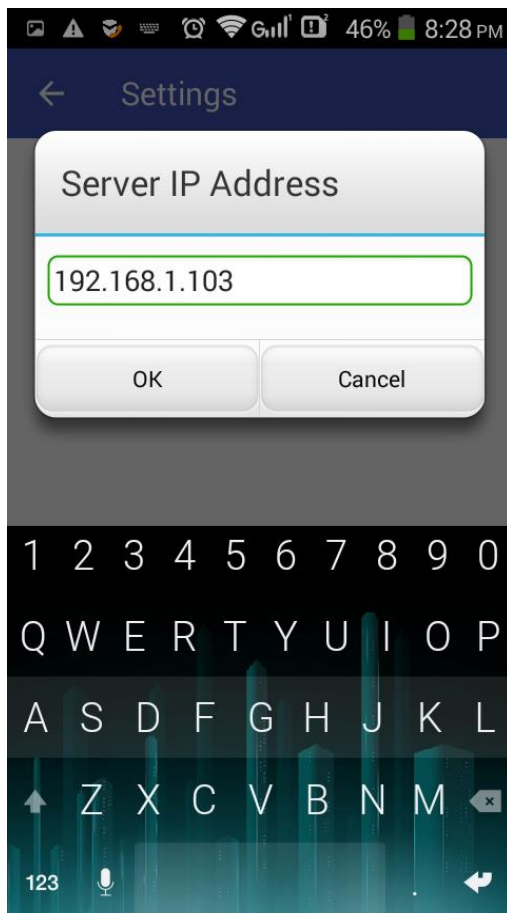


Fig: Setting the server in application.

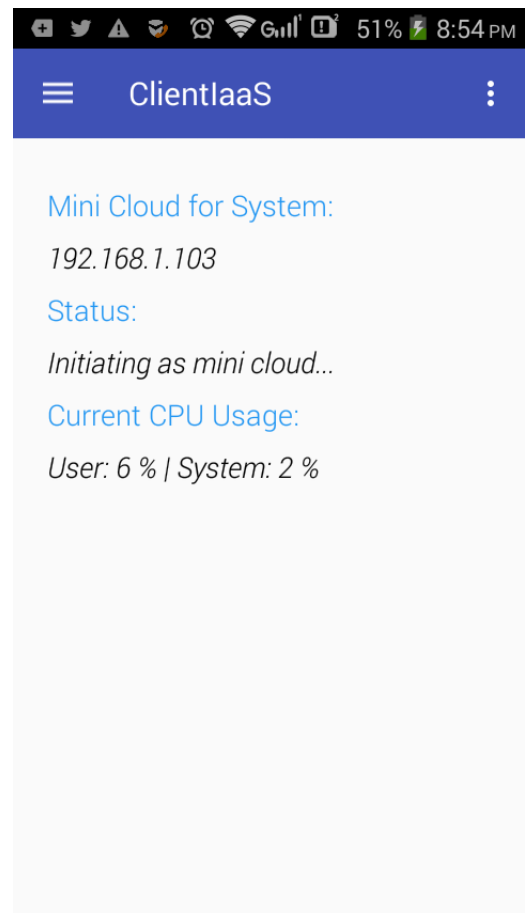


Fig: Initializing mini cloud

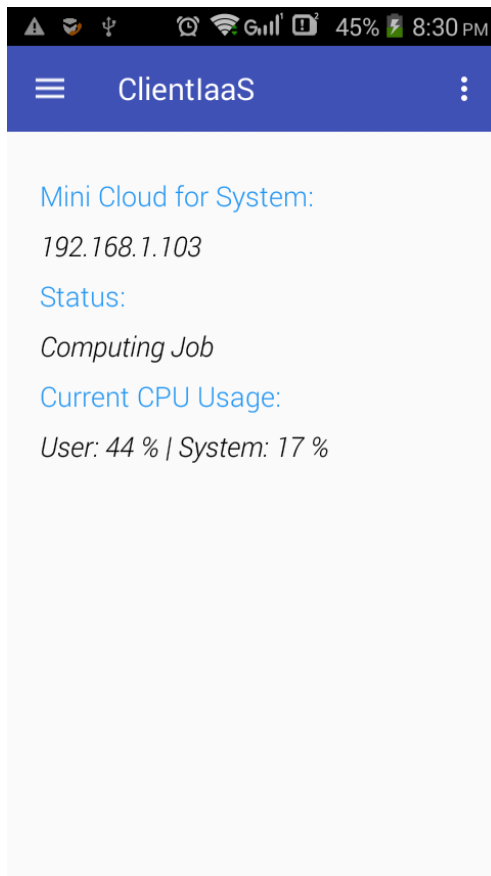


Fig: Computing Operation

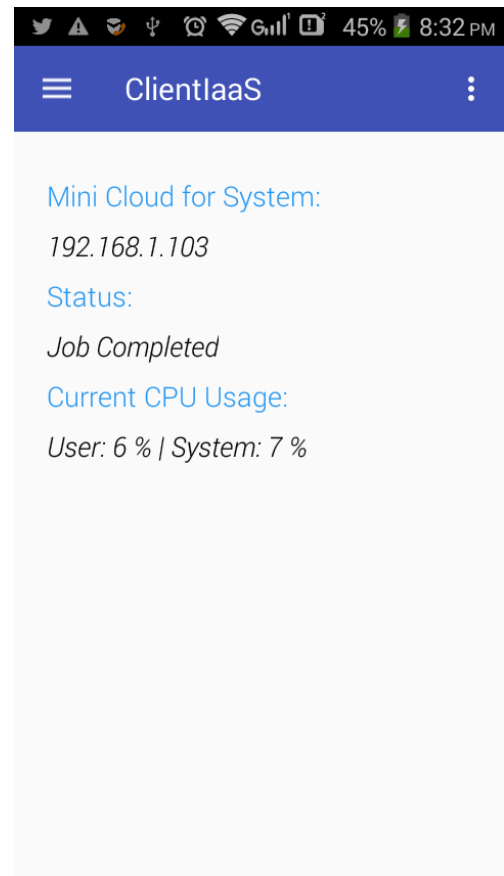


Fig: Task Completed

```
Current Level: 1 | nTask: 10 | Mats: 20
1 0 0 0 0 0 0 0 0 0 Message sent to the client is:
Client Job ID: 6001

/192.168.1.100 6001
Current Level 1
false

Message sent to the client is sending job

Task no.: 1 nTask: 10
Current Level: 1 | nTask: 10 | Mats: 20
1 1 0 0 0 0 0 0 0 0 Message sent to the client is:
Client Job ID: 6002

/192.168.1.100 6002
Current Level 1
false

Message sent to the client is sending job
```

Fig: Server State

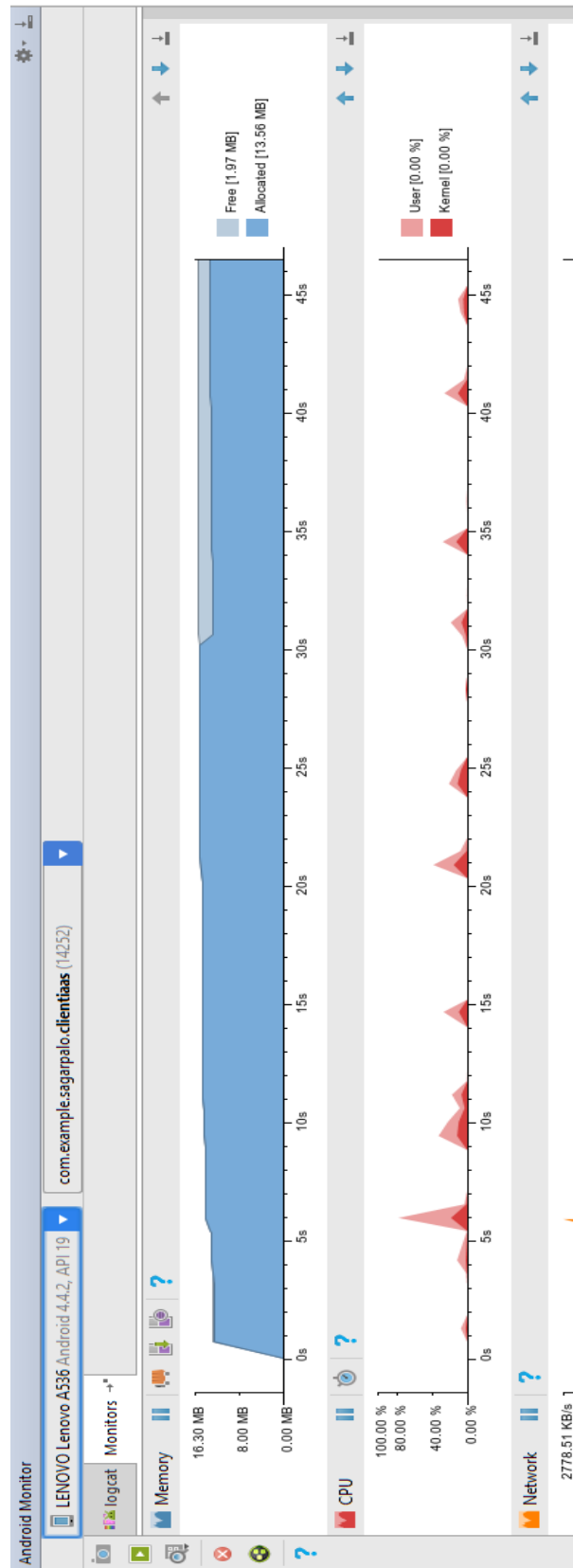


Fig: Client App Monitored

Name	Date modified	Type	Size
Intermediate	2017-03-29 3:56 PM	File folder	
Mat0	2017-03-26 6:53 PM	Text Document	178 KB
Mat1	2017-03-26 6:53 PM	Text Document	178 KB
Mat2	2017-03-26 6:53 PM	Text Document	178 KB
Mat3	2017-03-26 6:53 PM	Text Document	178 KB
Mat4	2017-03-26 6:53 PM	Text Document	178 KB
Mat5	2017-03-26 6:53 PM	Text Document	178 KB
Mat6	2017-03-26 6:53 PM	Text Document	178 KB
Mat7	2017-03-26 6:53 PM	Text Document	178 KB
Mat8	2017-03-26 6:53 PM	Text Document	178 KB
Mat9	2017-03-26 6:53 PM	Text Document	178 KB
Mat10	2017-03-26 6:53 PM	Text Document	178 KB
Mat11	2017-03-26 6:53 PM	Text Document	178 KB
Mat12	2017-03-26 6:53 PM	Text Document	178 KB
Mat13	2017-03-26 6:53 PM	Text Document	178 KB
Mat14	2017-03-26 6:53 PM	Text Document	178 KB
Mat15	2017-03-26 6:53 PM	Text Document	178 KB
Mat16	2017-03-26 6:53 PM	Text Document	178 KB
Mat17	2017-03-26 6:53 PM	Text Document	178 KB
Mat18	2017-03-26 6:53 PM	Text Document	178 KB
Mat19	2017-03-26 6:53 PM	Text Document	178 KB
Result	2017-04-19 8:36 PM	Text Document	207 KB

Fig: Matrix intermediate results computed in the server – task done by client

F:\Jobs\Result.txt - Notepad++

File Edit Search View Encoding Language Settings Macro Run Plugins Window ?

CalculatorClient.java	CalculatorServer.java	Calculator.txt	Mat5.txt	Mat0.txt	Mat5.txt	Mat0.txt	Result.txt	MPLjava.java	MPLjava2.java
1	9.807575600342247E91	1.048430299650293E92	1.1065769254964896E92	9.55079873896604E91	9.105155598155416E91	1.0130100782017323E92	9.756283410857199E91	9.141076012631714E91	
2	9.116954253911653E91	9.746027014719828E91	1.0286548956452223E92	8.878258009174056E91	8.46399658375907E91	9.416765896810621E91	9.069272022330931E91	8.497387881314546E91	
3	9.188472343040902E91	9.822480034511759E91	1.0367242012130815E92	8.947903739218326E91	8.530392544113628E91	9.490636058827323E91	9.140416278110707E91	8.564045747285398	
4	9.9513590751422774E91	9.569005346019714E91	1.0099709778515086E92	8.716998148100852E91	8.310261339174898E91	9.24572445735651E91	8.904542428773498E91	8.343046212427389E91	
5	1.059252914550222E92	1.1323420399468367E92	1.195142351245257E92	1.031520203006491E92	9.93890734946121E91	1.0940870383942381E92	1.053713391276556E92	9.872689719262156	
6	9.647500639712336E91	1.0313182448670729E92	1.08861584703049E92	9.3949146696812E91	8.956654207867873E91	9.964761166987926E91	9.597046174649913E91	8.991879413550018E91	1
7	1.013726489072686E92	1.088674115478677E92	1.1497753409505644E92	9.87185620581208E91	9.411220262477048E91	1.047063258925626E92	1.0084248973683661E92	9.4483592668667E91	
8	9.331801044797043E91	9.975698782450424E91	1.0528988174448421E92	9.05748006466533E91	8.66345603776499E91	9.63867864704887E91	9.282995784103693E91	9.697634123416423E91	
9	8.272395145439001E91	8.849768438842397E91	9.334251688124816E91	8.056335504020778E91	7.68042616136877E91	8.54498085359028E91	8.22964659962163E91	7.710726586327576	
10	8.8319915692628E91	9.44140117761102E91	9.96502876402741E91	8.600755709414952E91	8.19944239349642E91	9.12243116213669E91	8.785798870696388E91	8.231790678267223E91	9.169
11	9.413182570030323E91	1.0062695840619551E92	1.062078003417216E92	9.166731003727691E91	8.739009014493459E91	9.722736700206781E91	9.363951918672662E91	8.773485126543293	
12	1.008410195256573E92	1.077991034352947E92	1.137777079757578E92	9.82008554471823E91	9.36187709518567E91	1.0415721682457861E92	1.003136430058885E92	9.398810173708637E91	
13	1.0942169715169544E92	1.169718700071862E92	1.2345919035182096E92	1.0655688933784697E92	1.0158490031216797E92	1.1302010157415553E92	1.0884946870609605E92	1.019856543045994	
14	8.892685515532379E91	9.50628316106296E91	1.0033850906104476E92	8.659860675007263E91	8.255789982555182E91	9.185121204002517E91	8.846175562667038E91	8.288359988003226E91	
15	8.65633142751768E91	9.253620037804245E91	9.7668334015544E91	8.429694388279698E91	8.036363585002791E91	8.940993750811665E91	8.611056924892422E91	8.068068040715958E91	8
16	9.15458313488618E91	9.786252384509765E91	1.0329005199684803E92	8.914901757883341E91	8.498930482578095E91	9.455632280610667E91	9.106704212778873E91	8.532459580449296E91	
17	9.285785775649594E91	9.926508342131431E91	1.0477039663765172E92	9.042669485054981E91	8.62073638428218E91	9.59114999816119E91	9.237221042116767E91	8.65474595791702E91	9.640
18	9.549851661477157E91	1.02084372823697E92	1.0774604203653425E92	9.299495793608089E91	8.865878808795046E91	9.86355458664765E91	9.499573329544344E91	8.90055418826281E91	9
19	1.005498533988978E92	1.0748731189014848E92	1.1344862466968205E92	9.7916282465862E92	9.33479421698934E91	1.038559584796911E92	1.0002350172044803E92	9.871625699585294	
20	9.058581489986835E91	9.693626391118053E91	1.0220687623370651E92	8.821413465686938E91	8.40980447962388E91	9.356473271903553E91	9.011204379615614E91	8.44298201115513E91	
21	9.926830167359763E91	1.050488617777887E92	1.1087489366388871E92	9.56954921723706E91	9.123031152207629E91	1.0149988605605637E92	9.775437332013013E91	9.159022078624346E91	
22	9.05793858157376E91	9.68293911523868E91	1.021996224220991E92	8.820787384041515E91	8.409207627079573E91	9.35580922383968E91	9.01054836319067E91	8.442382797716247	
23	9.52348396633683E91	1.018060825093532E92	1.0745231791282919E92	9.274144672415488E91	8.84141056438577E91	9.83665669807057E91	9.473676697392231E91	8.876290609706027E91	
24	9.691715443779932E91	1.0360448193468323E92	1.0935045548268434E92	9.439771826431462E91	9.997593851240759E91	1.0010430142150525E92	9.641028868775436E91	9.03308947549793E	
25	1.042729838691858E92	1.1146788236876804E92	1.1764995505712763E92	1.0154297048577942E92	9.62043981075752E91	1.0770205409595931E92	1.0372766834527266E92	9.7186837053097791	1
26	9.50917478466213E91	1.01653176303024E92	1.0772908697042048E92	9.26021012135978E91	8.82812626237544E91	9.821885968496714E91	9.459442370860486E91	8.862953906470406E91	9
27	9.01112469295431E91	9.63289438188788E91	1.0167142062135274E92	8.775198579126608E91	8.365746054684667E91	9.307455144343466E91	9.86399512384551E91	8.3987497293829E91	9
28	1.169307680627807E91	9.80199296148432E91	1.034561873905418E92	8.929240811823837E91	8.51260455688443E91	9.470841096677652E91	9.121381791611551E91	8.546183471618443	
29	9.303910428631284E91	9.9452424107243E91	1.0496812532644375E92	9.059735388260683E91	8.697005920825542E91	9.609251013195068E91	9.254654091704086E91	8.671079671301452E91	
30	1.021693978596003E92	1.0921914355610397E92	1.1527650261913688E92	9.94944568107489E91	9.485201102294529E91	1.055292840010261E92	1.0163807802371989E92	9.52262044902268	
31	8.905714155728637E91	9.5202108106891594E91	1.0048209129259299E92	8.672548222577402E91	8.2678551095194E91	9.1385783949242E91	8.85913610064943E91	8.300503228441533E91	
32	8.605697774089627E91	9.200850166314574E91	9.711136946530697E91	8.36162309600489E91	7.990535358663544E91	8.890006593818471E91	8.561951301009052E91	8.02205904027842E91	
33	9.52381214446869E91	1.018049842727565E92	1.0745115925385454E92	9.274044423734171E91	8.811315227015708E91	9.835559628478992E91	9.47357454191202E91	8.87614893636633	
34	9.492123875590792E91	1.014708492727036E92	1.0709848621469251E92	9.243605087876988E91	8.84225640183045E91	8.804274278168367E91	9.442408588870028E91	8.847616742329332	
35	9.9007128787959E91	1.0583866984857748E92	1.1170854998830767E92	9.6414976632692E91	9.191622380518192E91	1.0224301219153833E92	9.84893849950496E91	9.227883872008852E91	1
36	9.069522118156281E91	9.69532194055146E91	1.023303181106926E92	8.830267657020333E91	8.419961539709625E91	9.36777972484932E91	9.022087819852281E91	8.453179136556576	
37	9.108071912325438E91	9.73653176889661E91	1.027652711033051E92	8.869608209217703E91	8.4557035987857E91	9.407591367064719E91	9.06043610977473E91	8.489109165507545E91	9

Normal text file length: 211,817 lines: 101 Ln: 14 Col: 2,319 Sel: 0|0 Unix (LF) UTF-8 INS

Fig: Matrix final result computed in the server – task done by client

## 5. Conclusion

The system is a great starting point for a great research project. It has currently laid a setup system for working with large computation task and getting work done from mobile computer initiated as mini clouds for the system. The project has a greater scope in implementing many other distributed computations by dynamically creating destroying clouds from the resources which we underutilize out smart phones.

## Reference

- [1] *Mobile Cloud Computing*, Victor C. M. Leung, Yonggang Wen, Min Chen, Chunming Rong  
IEEE Wireless Network Journal
- [2] Web: Mobile Cloud Computing, Online: <https://www.ibm.com/cloud-computing/learn-more/what-is-mobile-cloud-computing/>