Unified Ecosystem for Smartphones as Mobile Mini-Cloud Computing

Group Members

Sagar Palao | 48

Abhishek Shahasane | 62

Why?

- 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).

So what are we doing?

- Let the researchers load a job in the central system.
- The researcher needs to split the job into smaller tasks.
- When a mobile connects to the hotspot, the mobile is initiated a mini cloud for the server system.
- Now server sends a packaged job to the mobile cloud assuming it as its infrastructure for execution of the job.
- The mini cloud computes the job and send the result to the server.
- If the mini cloud job result is received by the server it updates its central map.
- When all the task are received in the central map, it updates the task to combine stage where the results are combined.
- This again is performed by the mini clouds.
- Finally when combine is completed, the final result is stored in the system.

How?

First Phase - Server:

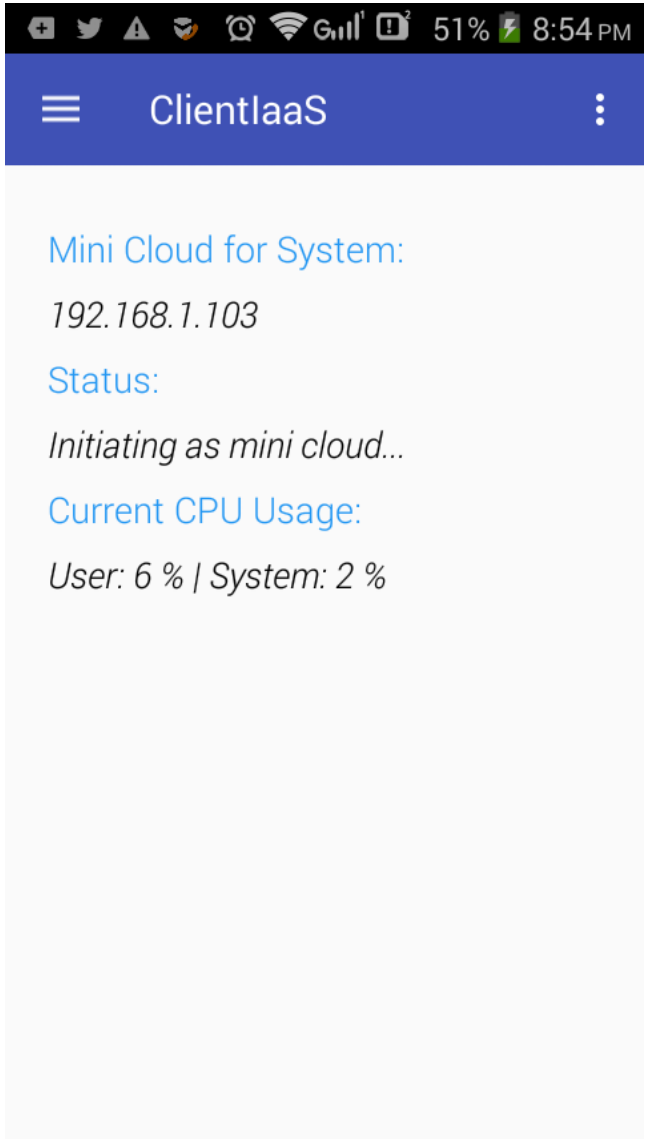
- The first phase is to build the Server end. The Server is designed in Java to be compatible across multiple platforms. The server is minimal and designed with polished interface using Sockets to enable communication with the mobile clients.
- When the client connects to the hotspot, the server sniffs 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

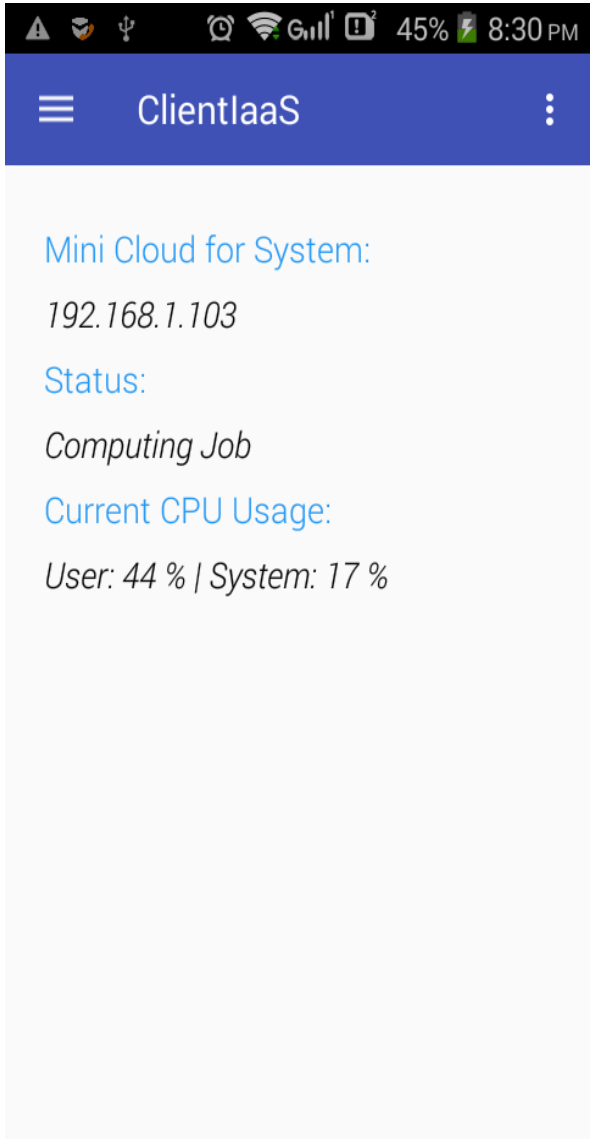
  …
  - Till there is 1 result matrix.
- To send a job we have used Serialization in java which is a wrapper for the objects. Each task is wrapped and sent to the client mini cloud.
- The result is then again received by the server as a serialized object. Which stores the result in files.
- Each thread s destroyed as the task is completed successfully.
- Threads are synchronized in Java using the standard mutual exclusion packaging.

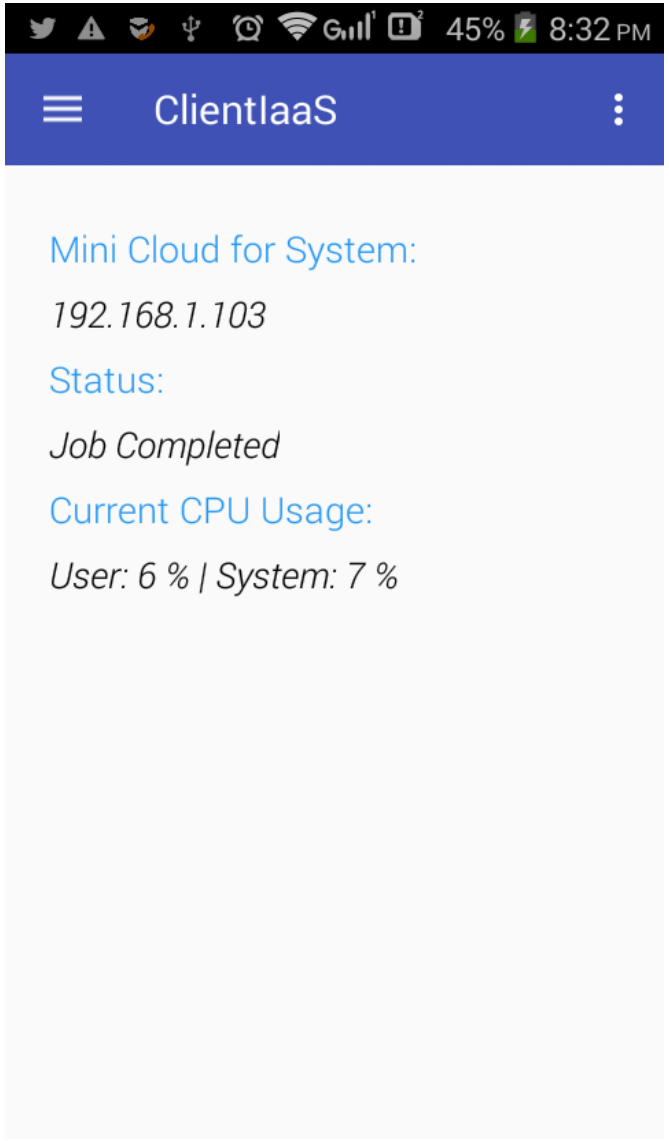
Second Phase – Mobile Application as mini cloud:

- 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, desterilize 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.

Some Snaps

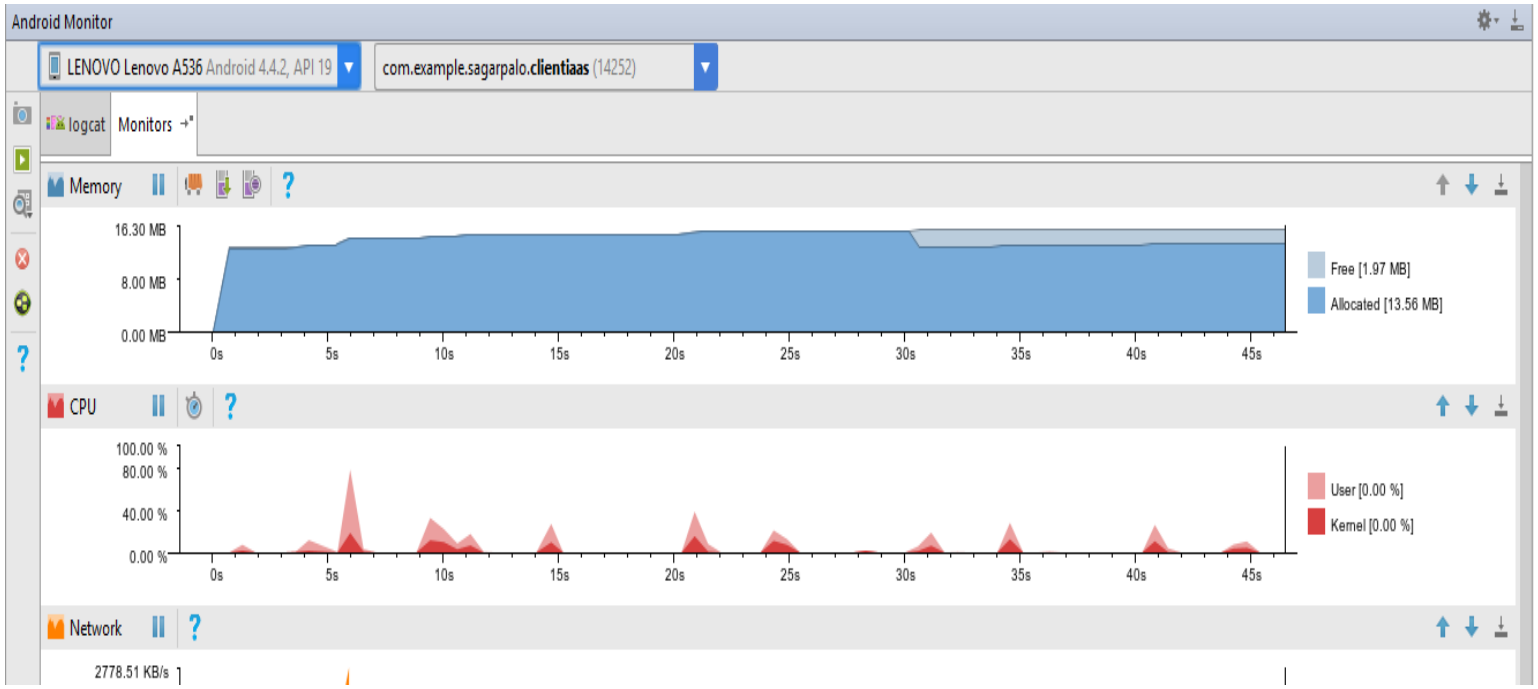

51% 8:54 PM
ClientIaaS
Mini Cloud for System:
192.168.1.103
Status:
Initiating as mini cloud...
Current CPU Usage:
User: 6 % | System: 2 %


45% 8:30 PM
ClientIaaS
Mini Cloud for System:
192.168.1.103
Status:
Computing Job
Current CPU Usage:
User: 44 % | System: 17 %


45% 8:32 PM
ClientIaaS
Mini Cloud for System:
192.168.1.103
Status:
Job Completed
Current CPU Usage:
User: 6 % | System: 7 %

Some more…

```
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
```



Finally it ended…