Virtual Cloud for Mobile Phones

ABSTRACT
Mobile phones are becoming the important part of human being’s life. Now a days mobile phones becomes a need of a user. Using it, a user not only receives and makes calls, but also performs information tasks. Every user wants to execute any application on top of a mobile device, regardless of its capabilities. But mobile phones lacks in resources compared to a conventional information processing device such as a workstation or a laptop. To alleviate this, a mobile phones should get resource from an external sources. One of such source is cloud computing. Nevertheless an access to this platform is not always guaranteed to be available/or is too expensive to access them. We try to propose that due to pervasiveness of mobile phones and enhancement in their capabilities this is feasible.

General Terms
Networking

Keywords
Cloud Computing, Mobile phones, Ad Hoc.

1. INTRODUCTION
According to the latest report, worldwide mobile phone subscription grew from 12.4 million to over 6 billion, penetrating about 87% the global population and reaching the bottom of the economic pyramid [1]. Now a day’s mobile phone become the need of a user instead of luxury. Because of advances in mobile phones users starts to use the mobile phones as a personal information processing tool, so that user can execute any application on the mobile. However mobiles still lacks in resources compared to conventional information processing devices such as workstation or a laptops.

This limitation can be overcome using mobile cloud computing [2]. The cloud computing technique for mobile phones aims for storage and processing of data on mobile devices, thereby reducing the limitations imposed on mobile devices. According to ABI research “By 2015, more than 240 million business customers will be leveraging cloud computing services through mobile devices, driving revenues of $5.2 billion”. While it must be noted that there were only 42.8 million Mobile Cloud Computing subscribers in 2008.

The end mobile device user will eventually be the benefactor of the mobile cloud computing. Company users can share resources and application without a high level of capital expenditure on hardware and software resources. Nature of the cloud computing is also advantageous for users since they do not need to have very technical hardware to run applications as these computing operations are run within the cloud. This reduces the price of mobile computing to the end user. They could see a huge number of new features enhancing their phones due to Mobile Cloud Computing.

The Mobile Cloud Computing also provide benefits to the Developers. The largest benefit of cloud computing for developers is access to a broader audience of a wide range of mobile subscribers. Since cloud computing applications go through a browser, the end users mobile operating system does not have any impact on the application [3].

In this paper we are trying to present the concept to create framework for virtual cloud computing. This framework detects the nearby nodes that are in the stable mode. If such a node is found the service or target provider for the mobile application is changed with a virtual provider among the users.

2. RELATED WORK
Integration between cloud computing and mobile phones is presented in various previous works. X. Luo introduces the idea of using cloud computing to enhance the capability of mobile devices [4]. The main approach to this work is to show the feasibility of such implementation, introducing the new scheme for the tasks.

Chun and Maniatis examine the use of cloud computing to execute mobile applications in behalf of the device [5]. They propose the creation of clone VMs to run different applications/services in the same way as they run on mobile devices. Their work is strongly tied to distributed file systems, and assumes connectivity to the cloud. Marinelli [6] introduce Hyrax, a mobile cloud computing client which allows mobile devices to use cloud computing platforms. Based on Hadoop, the core of this work is to locate a client into a mobile device to enable the integration. The author introduces the concept of using mobile devices as resource providers, but experimentation is not included.

3. MOBILE CLOUD COMPUTING
3.1 What is Mobile Cloud Computing
A new term “Mobile Cloud Computing” has been devised for a combination of cloud computing and mobile applications. In mobile cloud computing, data storage and data processing occurs outside the mobile device and results are displayed through screen or speakers. GPRS, Gmail, and Google Maps
are already being used are pioneer examples of mobile cloud computing. In a few years’ time we can expect a major shift from traditional mobile application technology to mobile cloud computing.

Mobile cloud computing can give mobile device users a number of advantages. Company users are able to share resources and applications without a high level of capital expenditure on hardware and software resources. Due to the nature of cloud applications, users do not need to have highly technical hardware to use applications as complex computing operations are run within the cloud. This lessens the cost of mobile computing to the client. End users will see a plethora of unique features enhancing their phones because of mobile cloud computing. A few examples of such applications can be seen to emerge such as applications that give users the ability to watch home security systems and other services which allow users to create location based social networks

3.2 How Mobile Cloud Computing Works?
The Architecture for Mobile Applications in Cloud Environment i.e. mobile cloud platform is called open mobster. This architecture is as given in the Figure 1, and Figure 2 shows the general client cloud stack.

![Image](image_url)

**Fig 1: The Open Mobster Architecture**

3.2.1 Essential services needed by the mobile cloud client:

**Sync:** This service synchronizes all state changes made to the mobile or its applications back with the Cloud Server.

**Push:** It manages any state updates being sent as a notifications from the cloud server. This improves the user’s experience as it does not require the user to pro-actively check for new information.

![Image](image_url)

**Fig 2: Client Cloud Stack**

OfflineApp: It is a service which carries the management capabilities to create smart coordination between low-level services like Sync and Push. It frees the programmer from the burden of writing code to actually perform synchronization as it is this service which decides synchronization management and mechanism which is best for the current state. The moment the data channel for any mobile application is established, all synchronizations and push notifications are automatically handled by OfflineApp service.

Network: It manages the communication channel needed to receive Push notifications from the server. It carries the ability to establish proper connections automatically. It is a very low-level service and it shields any low-level connection establishment, security protocol details by providing a high level interfacing framework.

Database: It manages the local data storage for the mobile applications. Depending on the platform it uses the corresponding storage facilities. It must support storage among the various mobile applications and must ensure thread safe concurrent access. Just like Network service it is also a low-level service.

InterApp Bus: This service provides low-level coordination/communication between the suites of applications installed on the device.

3.2.2 Essential services needed by Mobile Cloud Server

These are the essential services that must be provide to the mobile apps by the server.

**Sync:** Server Sync service synchronizes device side App state changes with the backend services where the data actually originates. It also must provide a plugin framework to mobilize the backend data.

**Push:** Server Push service monitors data channels (from backend) for updates. The moment updates are detected, corresponding notifications are sent back to the device. If the device is out of coverage or disconnected for some reason, it waits in a queue, and delivers the push the moment the device connects back to the network.

**Secure Socket-Based Data Service:** Depending on the security requirements of the Apps this server side service must provide plain socket server or a SSL-based socket server or both.

**Security:** Security component provides authentication and authorization services to make sure mobile devices connecting to the Cloud Server are in fact allowed to access the system. Every device must be first securely provisioned with the system before it can be used. After the device is registered, it is challenged for proper credentials when the device itself needs to be activated. Once the device is activated, all Cloud requests are properly authenticated/authorized going.

**Management Console:** Every instance of a Cloud Server must have a Command Line application such as the Management Console as it provides user and device provisioning functionalities. In the future, this same component will have more device management features like remote data wipe, remote locking, remote tracking, etc.
4. MOBILE DEVICE AS A CLOUD PROVIDER

4.1 Scenario and Motivation

If we consider the case of offloading to devices with similar characteristics, in which the performance will be similar to the source node, the overall performance of the task will be worse than running it on a single device due to the migration overhead. Therefore, we need to explore what makes the offloading to similar devices beneficial. On an economical basis, accessing cloud computing providers is associated with two costs: the cost of networking plus the cost of using the provider’s resources. The latter is not high nowadays – it can be as cheap as 5 USD per month considering the access of a small on-demand server 2 hours per day but is expected to increase to reach higher levels of uptime and better support [7]. On the other hand, wireless data fee is still very high. As an example, in South Korea the subscription plans for the iPhone3 GS(32GB) are near 70 USD per month, and if the user wants to download 1 GB of data he has to pay more than 200 USD. Besides, using 3G connectivity consumes more battery and is slower than network interactions with nearby devices using other interfaces such as Wi-Fi [8]. On a technical side, there are several benefits to consider: First, we still preserve conventional offloading benefits, such as allowing applications that cannot otherwise be executed on mobile devices due to a lack of resources. For example, if memory is not enough then creating instances of those objects on any remote device will allow the application to be executed. Second, performance can be enhanced if the execution sequence of an application can be reordered for increasing the level of parallelism. This can be achieved by maximizing the number of operations that can be executed while waiting for a result from other nodes [9] if the communication overhead does not affect the overall performance. However these two aspects lack incentives for users to share their resources. A way to overcome this is by finding users pursuing the same task, and splitting the elements of the task among them.

4.2 Design Considerations

To exploit a collection of nearby mobile devices as a virtual cloud computing provider, we believe that it should have the features as follows:-

- Resource monitoring and management to recognize when a task cannot be locally executed on a mobile device.
- Seamless integration with the existing cloud APIs. If provided by a cloud host, the goal is to mimic the same API on top of the ad hoc mobile P2P cloud.
- A partition and offloading scheme suitable for mobile devices. If an application is not built for cloud, or the job defined is too heavy, job splitting is required.
- Activity detection to find users of the same or similar goals. This detection should focus on detecting the task per se and to determine if users will remain together to minimize potential disconnections.
- Spontaneous interaction network support. In order to create a virtual cloud provider, the discovery and selection of mobile devices is needed.
- A memory cache scheme to save intermediate results.
- Lightweight and resource friendly architecture. We must not introduce excessive overhead that consumes resources on a local device in a faster pace than local execution.

4.3 Architecture

The process for the creation and usage of a virtual cloud provider is simple. If a user is at a stable place and wants to execute a task which need more resources than available in the device, the system search for nodes in nearby area? If that

Figure 3. General Architecture for Ad Hoc Mobile Cloud
available, the system intercepts the application loading and modifies the application in order to use the virtual cloud. To support this process, the proposed architecture shown in Figure 3.

It consists of five major features viz. Application manager; Resource manager; Context manager; P2P component and; offloading manager.

**The Application Manager** is in charge of launching and intercepting an application at loading time and modifying an application to add features required for offloading – proxy creation, RPC support - according to the current context. Since the idea is to replace calls to infrastructure-based clouds, the interception and modification should focus on modifying the reference to that provider with a reference to the virtual provider. This process is performed when an application is executed the first time. Once an application is modified, its modified copy is used to avoid further delays.

**The Resource Manager** is in charge of application profiling and resource monitoring on a local device. For each application, a profile is defined in terms of the number of remote devices needed to create a virtual cloud, and sensibility to privacy and amount of resources needed for the migration to happen (in average). This profile is checked by the application manager whenever an application is executed in order to determine whether an instance of the virtual provider should be created or not.

**The Context Manager** wields and synchronizes contextual information from context widgets and makes it available in some way for other processes. It is composed by three subcomponents: context widgets that communicate with the sources of information; a context manager itself that handles the information and extracts new contexts from them; and a social manager that is used to store the knowledge regarding relationship between users. Two basic contexts are of utter importance: the location and number of nearby devices. The former is used for the mobility traces. The latter for enabling of a cloud from the application manager, and it is given by the P2P component. This component is aware of the status of the devices in the surroundings: it sends events to the context manager in case a new device enters the space, or if an existing device leaves the space.

It utilizes an ad hoc discovery mechanism, and then groups the nodes using a P2P scheme, allowing for better scalability and distribution of contents. Once that information is captured, a context aggregator located in the context manager generates high level contexts from the basic contexts. They represent the consolidated information related to the user. We only define one high level context for this framework, which is whether the user is in a stable location or not. More details are in the next section.

**The Offloading manager** component is in charge of sending and managing jobs from the node to other remote devices, plus receiving and processing jobs sent from them. It communicates with the P2P component once a job is issued to the respective node, and waits for the results to be delivered back to the application. This component is the one in charge of detecting failures in the execution and to re-emit them. It also is in charge of creating protected spaces for the execution of the tasks coming from remote nodes. This protected spaces (represented here as a VM), are utilized to block the access to sensitive data located on the devices.

5. CONCLUSION

This paper present the motivation and preliminary design for a framework to create Ad Hoc cloud computing. This framework takes advantage of the pervasiveness of mobile devices, creating a cloud among the devices in the vicinity, so that they can execute jobs between the devices. The work presented here is preliminary, and creates the foundation for future work.

6. ACKNOWLEDGMENTS

Few select and pretty words will not be enough to acknowledge those intellectuals, who are involved in while preparing this paper. First of all we would like to express our deep gratitude to Prof. P. N. Kalawadekar whose guidance help us while creating this paper. We are indebted to his valuable guidance and help. We are also grateful to our friends who are directly or indirectly helped us in creation of this paper.

7. REFERENCES


