International Journal on Recent and Innovation Trends in Computing and Communication ISSN: 2321-8169
Volume: 2 Issue: 12 3906 - 3910

Development of Android Mobile Application for Cloud Server

Miss. Bhagyashri D. Patil
Department of Computer Science and Engineering
H.V.P.M's College of Engg.& Technology, Amravati
Maharashtra (India)
Email: patilbhagyashripatil@gmail.com

Prof. P. L. Ramteke
Department of Computer Science and Engineering
H.V.P.M's College of Engg.& Technology, Amravati
Maharashtra (India)
Email: pl_ramteke@rediffmail.com

Dr. D. N. Chaudhari
Department of Computer Science & Engineering
Jawaharlal Darda Institute of Engineering & Technology, Yavatmal
Maharashtra (India)
Email: dnchaudhari2007@rediffmail.com

Abstract- The number of Smartphone users and mobile applications are growing rapidly. Though smart phones are expected to have PC-like functionality, hardware resources such as CPUs, memory and batteries are still limited. To solve this resource problem, many researchers have proposed architectures to use server resources in the cloud for mobile devices. The system proposed conceptual architecture of development of android cloud for efficient implementation of platform as a service, which enables multiple user Android applications on cloud server via network. Though Android is mainly designed for physical Smartphone, Android's too the features are useful to construct a server platform. Android is open-source product and run sonanx86 CPU. Android is an open-source mobile OS initiated by Google. The main reason to use Android as a server platform is that it is able to run not only for smart honesbutalsoforthex86platformincludingservers.Weshowthreetypesofmulti-tenant architecture for an Android server platform and discuss the direction to take to it reality.

Keyword: *Mobile Computing, Cloud Computing, Android Application, Multi-tenant.*

*****

1. INTRODUCTION

The number of Smartphone users and mobile application are growing rapidly. According to be cent report, 45 million people in the U.S. own Smartphone's and 234 million people subscribe to the mobile phone application stores [1]. There are several mobile Operating Systems (OSs), such as Symbian, iOS, Android, and Windows Mobile. Because thousands of application developers construct many kinds of applications for these platforms, users can easily enjoy the individual Smartphone lifestyle. Though Smartphone's areexpected to PC like functionality, hardware resources such as CPUs, memory, and batteries are still limited. Therefore, many application developers are forced to take into account these limitations. To solve this resource problem, some researches have proposed using server resources in the cloud for Smartphone's.Mobile devices are replacing laptops and traditional computers. These devices are used not just for communication but also for multimedia applications such as listening to music, watching videos, and playing games. Thus the storage space available on these devices limits how much multimedia files can be used on the device and the user is constantly removing files to make space to add new ones. Addition of extra storage space either by increasing internal storage by manufacturers or addition of SD cards only serves to temporarily all the problem until we run out of space again. There is a need to permanently solve this problem and integration to cloud based storage elegantly solves this problem. Cloud Based File System solves this problem by providing anytime/ anywhere access to the unlimited storage of a "cloud" to the Mobile Device users. Most of today's smartphone applications are geared towards an individual user and only use the resources of a single phone. There is an opportunity to harness the collective sensing, storage, and computational capabilities of multiple networked phones to create a distributed infrastructure that can support a wealth of new applications. These computational resources and data are largely under-utilized in today's mobile applications. Using these resources, applications could conveniently use the combined data and computational abilities of an entire network of smartphones to generate useful results for clients both outside and within the mobile network [1]. This interface and the underlying hardware would create a mobile-cloud upon which compute jobs could be performed. We define mobile-cloud computing to be an extension of cloud computing in which the foundational hardware consists at least partially of mobile devices.

1.1 Mobile Computing

Mobile computing is the performance of computing tasks whiles the user in on the move, or visiting place other than their usual environment. In case of mobile computing a user who is away from his "home" environment can still get access to different resources that are too computing or data intensive to reside on the mobile terminal. Mobile distributed systems are based on wireless networks that are known to suffer from low bandwidth, low reliability, and unexpected disconnection [2]. Over the past few years, the number of Smartphone users has rapidly increased. As

Smartphone interfaces are now convenient and user friendly, users can create various types of content. Although ad hoc networks can easily be constructed with smartphones as they are equipped with various network interfaces, such as Bluetooth and Wi-Fi, the connectivity between smartphones is expected to be intermittent due to the movement patterns of carriers and the signal propagation phenomena [3]. Recently, the development in wireless devices has made it possible to connect numerous devices by constructing networks only amongst themselves [4]. With today's technology, many applications rely on the existence of small devices that can exchange information and form communication networks [5].Smart Phones are not just for contact to people but it plays vital role in the life. Now smart phones having PC-like functionality, Wi-Fi communication, camera, videos and all important data are present in it. Yet hardware resources such as CPU's, memory and batteries are still limited [6].

1.2 Cloud Computing

Cloud computing is a style of computing in which dynamically scalable resources are provided as a virtualized service Knorr and Grumman [1]. It allows service providers and other users to adjust their computing capacity depending on how much is needed at a given time or for a given task. According to Myerson, cloud computing requires three components: thin clients, grid computing, and utility computing. Thin clients are applications that make use of the virtualized computing interface. Users are commonly exposed to cloud computing systems through web interfaces to use services such as web-based email, search engines, and online stores. Grid computing harnesses the resources of a network of machines so that they can be used as one infrastructure. Utility computing provides computing resources on demand, where users "pay as they use". This is exemplified by Amazon EC2, which allows users to allocate virtual servers on demand, paying an hourly fee for each allocated server. In mobile-cloud computing, the same type of virtualized interface is provided to users, but the system is ultimately supported by mobile devices or a combination of mobile and static devices. The possibility of heterogeneous clusters of servers and mobile devices in which the capabilities of each are used in conjunction is not excluded [7].

1.3 Android

Android Open Handset Alliance is an open source mobile operating system developed by Google and the Open Handset Alliance. It is built on top of the Linux kernel and provides an SDK for application development in Java. Android uses the Dalvik Virtual Machine to execute applications. Dalvik is optimized to run on devices with constrained CPU, memory, and power resources. It implements a subset of Java 2 Platform Standard Edition (J2SE) using libraries from the Apache Harmony Apache Java implementation, giving it an advantage over other mobile platforms that only support Java 2 Platform Micro Edition (J2ME), which is limited by comparison. Java class files must be compiled to Dalvikbytecode (.dex format) and packaged in a .apk file in order to be used on Android. Android provides an interface to system devices and services through a set of Java packages, including android OS, android Hardware, Android Location and android media. This makes it easy to access and operate on multimedia data, sensor values, system resource usage data, and location information. Unlike some mobile operating systems, Android applications can use the file system directly, making it possible to manage files as on a traditional Unix system. Android also provides a shell interface, but it lacks many of the abilities of a typical Linux shell. Some of the missing utilities can be added by installing BusyBox Denys Lysenko [8].

1.4 Multi Tenancy

A tenant is a group of users sharing the same view on an application they are use. This view includes the data they access, the configuration, the user management, particular functionality and related non-functional properties. Usually the groups are members of different legal entities. This comes with restrictions (e.g. data security and privacy). *Multi-tenancy* is an approach to share an application instance between multiple tenants by providing every tenant a dedicated "share" of the instance, which is isolated from other shares with regard to performance and data privacy. A commonly used analogy for explanation is a living complex where different parties share some of their resources like heating to reduce costs, but also love to enjoy their privacy and therefore demand a certain level of isolation (in particular when it comes to noise). Besides multi-tenancy there is also the notion of *tenant space* [9].

2. LITERATURE REVIEW

Android is a relatively new platform. It is produced by Google, Inc., and its first release was presented in 2007 [10]. Android is installed on many different mobile devices and its users can download Android apps and other content through Google Play service, which replaced the old Android Market [11]. This thesis discusses technologies incorporated in Android application development and how they apply to the research problem. As the official Android website describes this platform, "Android is a software stack for mobile devices that includes an operating system, middleware and key applications" ("What is Android," 2012). Android provides the "core set of applications including an email client, SMS program, calendar, maps, browser, contacts, and others" ("What is Android," 2012), while additional applications can be downloaded through Google Play service (Bishop, 2012) [11] [12].

2.1 Android Fundamentals

Many authors described Android application development fundamentals, which include setting up Android development environment on the machine, AndroidManifest.xml file, Activities, Intents, and XML layouts. Jackson (2011) outlines "three major components of

an Android development environment: Java, Eclipse, Android" and provides instructions on how to download and install necessary files to establish this environment [12]. Felker (2011) does not explicitly state the components but rather points out that Java JDK, Android SDK, Eclipse IDE, and Android ADT need to be installed and configured on a machine [13]. The steps provided by these two authors are standard. They appear in many books written on Android development and are also presented on official Android website ("Installing the SDK"). Ableson, King, and Sen (2011) present "four primary components of Android applications": Activity, Service, Broadcast Receiver, and Content Provider. It is noted that "a particular Android application might not contain all of these elements, but will have at least one of these elements" [11]. Since Activity "displays a UI (user interface) and responds to system and user initiated events", it is used very frequently for Android applications. These Activities are declared in AndroidManifest.xml file, which provides "the foundation for any Android application". Activities present their views through XML layouts and "communicate" with each other through Intents. Clear understanding of these concepts and Java programming language is a prerequisite to start implementing the development techniques used in Android applications.

2.2 Concept of Multi tenancy

Multi-tenancy is a new field of research and started to get in focus with the arising of enterprise SaaS applications. Koziolek describes an architectural style of MTAs called SPOSAD based on the well-known multi-tier web application model with its architectural constraints and tradeoffs [29]. Mietzner describes a multi-tenant architecture based on the Service Component Architecture (SCA) [16]. We instead focus on the most important features for the tenant and its implications to some architectural concerns [8]. Furthermore, our assumptions and restrictions regarding the architectural style are not as restrictive as presented by Koziolek [29]. A list of some key characteristics, a conceptual architecture of MTAs and the resulting challenges is listed in (Bezemer and Zaidman, 2010) [20]. Nevertheless, the challenges are rather discussed on a technical level and the authors did not discuss how one challenge influences conceptual architectural decisions. There are publications discussing how to separate the tenant's data and how to create tenant specific data models. Wang also did some performance related research [17] [4]. Ever the less, there is no discussion about mutual influences of non-database and database related concepts. Besides that, a number of publications associated with performance and resource optimization [7].

2.2.1 Different Layers of multi-tenancy

When a *single code base* is shared between different customers/tenants it comes up with some requirements for the code. If one single code base is used the application has to be widely configurable to be adapted for customer specific needs. Sharing the code base yields reuse and is omnipresent. Nevertheless, a single code base is not sufficient enough to reduce the operational costs. Developing widely configurable software instead of customer specific branches is a question related to product line engineering and not specific for MTAs.

Sharing a data center is the lowest level of resource sharing one could imagine. Reusing the facilities environment like air conditioner or network infrastructure is the simplest way of decreasing costs. Application Service Providers already have adopted these concepts for years. However, sharing the data center only has a very limited cost saving potential, e.g. workload fluctuations of different customers can't be considered for resource optimization. *Virtualization* provides an easy way for sharing a single server. Running a separate instance of the application within one VM for each customer is a first step towards efficient operation and probably today's most widely adopted sharing approach. In opposite to a shared data center, virtualization allows leveraging workload fluctuations by overcommitting the servers, while allowing a good isolation. However, the overhead of this solution per customer is still quite high. Virtualization is a well-established field of research with challenges and goals on a hardware related level and should not be referred to multi-tenancy which is a concept on the applications level.

2.2 Cloud computing service models

"Cloud computing isn't so much a technology as it is the combination of many preexisting technologies. These technologies have matured at different rates and in different contexts, and were not designed as a coherent whole; however, they have come together to create a technical ecosystem for cloud computing", [10]. The cloud computing concept is also divided into three different service models, also referred to as infrastructure models or services delivery models. The service models are generally categorized as Software as a Service (SaaS), Platform as a Service (PaaS) and cloud Infrastructure as a Service (IaaS).

2.2.1 Software as a Service (SaaS)

Software as a Service or SaaS refers to: "The capability provided to the consumer is to use the provider's applications running on a cloud infrastructure". The traditional method of purchasing software requires the customer to locally install an application on their computer and use licenses to authorize the usage. With SaaS the customer pays for the software on a subscription level and does not need to install any software on their computers. The software, application, is instead accessed via the Internet, through a web browser. An example of this is Google Docs which is a word processing application offered online. The user can access the application through a web browser, create documents and use all the features of the application.

2.2 2. Platform as a Service (PaaS)

"In a platform-as-a-service (PaaS) model, the vendor offers a development environment to application developers, who develop applications and offer those services through the

ISSN: 2321-8169
3906 - 3910

provider's platform", [10]. In comparison to SaaS where the application already exists, and is usually owned by the cloud provider, PaaS offers the possibility to create and modify applications. It is an outgrowth of the SaaS application delivery model [21]. To aid the developer, different tools are provided like programming languages and Application Programming Interfaces (API). In comparison to cloud Infrastructure as a Service, IaaS, the user does not control the virtualization instance or network configuration of the cloud server. An example of PaaS is Google App Engine that offers the possibility to create Java, Python and Go applications on servers hosted by Google.

2.2.3 Cloud Infrastructure as a Service (IaaS)

"Infrastructure as a Service (IaaS) is the delivery of computer hardware (servers, networking technology, storage, and data center space) as a service. It may also include the delivery of operating systems and virtualization technology to manage the resources. The actual network infrastructure of the cloud servers does not lay in the hands of the user, but rather network options like firewalls, storage, and operating systems etcetera [27]. An example of IaaS is Amazon EC2, where virtual servers can be set up and configured over a web based interface within minutes [22] [34]. The customer can choose operating system, database and application development environment which gives the customer greater control over the hardware in comparison to PaaS. The customer has the possibility to configure the servers based on their needs, which generally includes more maintenance in comparison to PaaS but also more options. These three service models constitutes the general model of cloud computing. It is a very broad concept and there are many different definitions, and new ones are coined frequently. SaaS, PaaS and IaaS are the most encountered in cloud computing literature and are basically divided by hardware abstraction level.

3. ANALYSIS OF PROBLEM

3.1Mobile Application Platform on Cloud Server

As a numbers of service providers such as Drop box and Zumo drive provide online storage services, the architecture for remotely using mobile application on server has many benefits for users. This approach, called Mobile Application Platform on Cloud Server, intends to handle not only user data but also user applications in a cloud server. This approach changes the application lifecycle as follows. "Write once, run everywhere. Install once, use everywhere." Figure 3.1 illustrates an overview of the concept. By executing a mobile application in the cloud server, users and developers free from device limitation such as CPU power, memory, and battery, and from device software environment such as OS or version. Moreover, once a user installs an application on the cloud server, she/he can use the application anywhere, an any device.

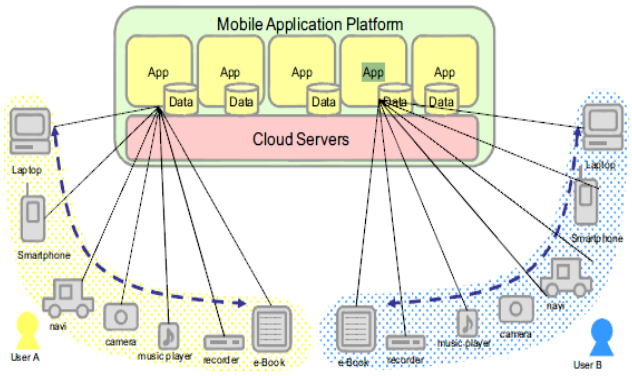

Figure 3.1 Mobile Computing Platforms

4. SYSTEM ARCHITECTURE

Cloud based file system Application is designed to meet the existing need for expanding storage of Mobile Devices along with easy to use Task oriented User Interface. It allows users to keep all the important files in a folder on own Cloud Server and user can access this folder anytime, anywhere from Mobile Device which generally have better security. So, now user is no longer bounded to Mobile Device's limited storage, because users have access to unlimited storage space of Cloud. Also, user doesn't need to struggle with Remote Access Applications which are very hard to use because of their poor user interface. An overview of basic Mobile Cloud Computing was presented in the previous section. A general architecture in more detailed representation will be presented in this section Fig 4.3 presents a typical Mobile Cloud Computing architecture.

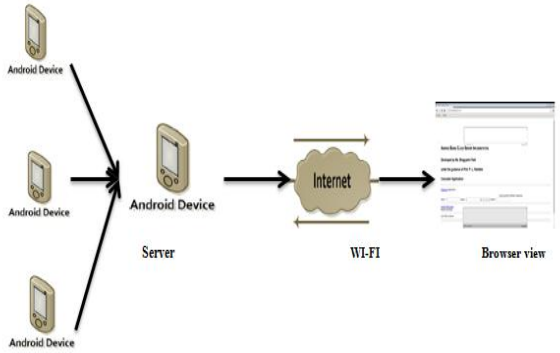

Figure 4.1Mobile Computing Architecture

The mobile devices are connected to the mobile networks through base stations that establish and control the connections (air interface) and functional interfaces between the networks and mobile devices. Mobile users' request and information are transmitted to the central processors that are connected to the servers providing mobile network services. The subscribers' requests are then delivered to a cloud through the Internet. Cloud controllers present in the Cloud, process the requests to provide the mobile users with the corresponding cloud services. These services are developed based on the concepts of utility computing, virtualization and service-oriented architecture. The details of cloud computing will be different in different contexts. The major function of a cloud computing system is storing data on the

cloud and using technology on the client to access that data. Some authors mentioned that Cloud Computing is not entirely a new concept.

5. Client Server Programming

A *socket* is one endpoint of a two-way communication link between two programs running on the network. A socket is bound to a port number so that the TCP layer can identify the application that data is destined to be sent. Normally, a server runs on a specific computerand has a socket that is bound to a specific port number. The server just waits, listening to the socket for a client to make a connection request [14].The client running on a computer or mobile device must know the host name of the computer on which the server is running and the port number on which the server is listening. The mobile device or computer client attempts to converge with the server computer on the server's computer and port to establish a connection request. The mobile device client also needs to introduce itself to the server computer, so it binds to a local client port number that it will use while this connection which is normally determined by the system. The server computer acknowledges the connection, if everything goes alright. After acknowledgment, the server computer occupy a new socket bound to the same local port of server computer and also has its remote endpoint set to the address and port of the mobile device client. Now server computer needs another new socket so that server computer can carry on listening to the primary socket for connection requests while serving to the requests of the connected mobile device client. If the connection is acknowledged on mobile device client side, a socket is successfully created and the mobile device client can use the socket to talk to the server. The mobile device client and server computer can now talk to each other by writing to or reading from their Ssockets. An endpoint is a combination of an IP address and a port number. Every TCP connection can be uniquely identified by its two endpoints. That way you can have multiple connections between your host and the server [14].

6. CONCLUSION

In this paper proposed Development of Android cloud for efficient implementation of platform as a service, which enables the use of sharing server-side Android OS among multiple users. The proposed Cloud Based File System App elegantly addressed the problem of limited storage of Mobile Devices, by providing unlimited storage of cloud to the mobile device users. Cloud Based File System app consists of Cloud Server app which runs on cloud and Mobile Device Client app which runs on Mobile Device. We tested the functionality of this app and found that it allows the Mobile Device user to connect with server from Mobile Device without any problem. Observations suggest that the Cloud Based File System app provides Mobile Device users with anytime/anywhere access to the unlimited storage of cloud while retaining functionality.

7. ACKNOWLEDGEMENT

This paper is the result of one year of hard work where many people have supported. It is a pleasant aspect that I have now the opportunity to express our gratitude to all of them. We have great pleasure to express deepest sense of gratitude and sincere thanks to all those who directly or indirectly supported us.

8. REFERENCES

[1] J. J. BARTON, S. ZHAI, AND S. B. COUSINS, eds., "*Proceedings of the Seventh IEEE, Workshop on Mobile Computing Systems & Applications*, Washington", D.C., 2006, IEEE Computer Society.

[2] H. KIM, N. AGRAWAL, AND C. UNGUREANU, eds., *Proceedings of the 10th USENIX conference on File and Storage Technologies*, Berkeley, 2012, USENIX Association.

[3] ElmurodTalipov, Yohan Chon, and Hojung Cha, *"Content Sharing over Smartphone-Based Delay-Tolerant Networks"*, IEEE TRANSACTIONS ON MOBILE COMPUTING, VOL. 12, NO. 3, MARCH 2013

[4] Kawamoto et al.*"MA-LTRT: Novel Method to Improve Network Connectivity and Power Consumption"*, IEEE TRANSACTIONS ON EMERGING TOPICS IN COMPUTING, VOLUME 1, NO. 2, DECEMBER 2013.

[5] Mietzner, R., Leymann, F., and Papazoglou, M. (2008). "*Defining composite configurable saas application packages using sca, variability descriptors and multitenancy patterns"*. In Internet and Web Applications and Services, 2008.ICIW '08. Third International Conference on, pages 156 –161.

[6] Miettinen, A.P. &Nurminen J.K. (2010). Energy efficiency of mobile clients in cloud computing. HotCloud'10 Proceedings of the 2nd USENIX conference on Hot topics in cloud computing. USENIX Association Berkeley, CA, USA

[7] M.satyanarayanan, V. Bahl ,R.Caceres, and N. Devies,*" The case for VM-based Cloudlets in Mobile Computing"*. IEEE persive Computing, Carnegie Mellon University, Microsoft Research, AT&T Research, Lancaster University , 2009.

[8] Naresh Kumar N (M.Tech), 2 Prof. Mohan K *," GCM Service Driven Communication With An Android Application In Cloud Computing "*, ISSN: 2278-0181, May 2013.

[9] Fehling, C., Leymann, F., and Mietzner, R. (2010). Aframework for optimized distribution of tenants in 2010 IEEE 3rd International Conference on, pages 252 –259.

[10] Meier, R. (2010). *Professional Android 2 Application Development*. Indianapolis, IN. Wiley Publishing, Inc.

[11] Koziolek, H. (2011). "*The sposad architectural style for multi-tenant software applications"*. In Proc. 9th Working IEEE/IFIP Conf. on Software Architecture (WICSA'11), Workshop on Architecting Cloud Computing Applications and Systems, pages 320–327.IEEE.

[12] Bishop, T. (2011, March 6). "Google Play replaces Android Market, cosolidates Google's media marketplaces". Retrieved from http://www.geekwire.com/2012/google-playreplaces-android-market-consolidates-googles-media-marketplaces.

[13] Felker, D. (2011). *Android Application Development For Dummies*. Hoboken, NJ. Wiley Publishing, Inc.

[14] Calvin, P. and Friedl, S. (2009). Lessons leraned: "Building multitenant applications with the windows azure platform".