ISSN No. 0976-5697

Volume 4, No. 9, July-August 2013

International Journal of Advanced Research in Computer Science

REVIEW ARTICLE

Available Online at www.ijarcs.info

Implementation of GCM for Mobile Cloud Computing in Android Devices

Sundeep Kumar, Manoj Kumar*
Department of Computer Science
Maharaja Surajmal Institute C-4, Janakpuri New Delhi
sundeep_chaudhary@rediffmail.com, manoj.rke77@gmail.com

Abstract: This paper presents a brief overview of Mobile Cloud Computing which consists of a group of low cost servers. Cloud Computing is a buzzword these days, however there is a shift towards Mobile Cloud Computing as the uses of mobile devices have registered an exponential growth. Recently launched Google Cloud Messaging (GCM) is a free service that helps developers to send data from servers to their Android applications on Android Devices. We discuss the architecture of GCM and the process of registration of the device in detail. We also discuss the different permissions required by the application, limitations of mobile devices and the conclusion of this paper.

Keywords: Mobile Cloud Computing, Google Cloud Messaging, GCM architecture, Mobile Computing

I. INTRODUCTION

The concept of Mobile cloud computing (MCC) has been introduced after invent of cloud based computing. Mobile cloud computing integrates the advantages of mobile computing, mobile networks and cloud computing [1]. It is a combination of all the 3 things for the benefit of mobile customers, cloud providers and mobile network operators. The essence of mobile cloud computing is to provide valuable, precise and real time information to any clients at any time, at any place[2].

In lack of the consensual or standard definition provided by any authorized body, several developers and organizations describe it according to their own perspective.

However, the term cloud computing is defined by NIST as- "Cloud computing is a model for enabling ubiquitous, convenient, on-demand network access to a shared pool of configurable computing resources (e.g., networks, servers, storage, applications, and services) that can be rapidly provisioned and released with minimalmanagement effort or service provider interaction.

This cloud model is composed of five essential characteristics (on-demand self-service, broad network access, resource pooling, rapid elasticity&measured service), three services(Software as a Service (SaaS),Platform as a Service (PaaS), and Infrastructure as a Service (IaaS)), and four deployment models(Private cloud, Community cloud, Public cloud and Hybrid cloud)."

Today, devices like smart phones, tablets, notebooks, and other PDA's can access the cloud services.There is an exponential growth of the mobile users. According to [3],the number of MCC subscribers will be almost 1 billion by 2014 and driving nearly $ 5 billion in revenues by more than 240 million mobile business users by the end of 2015. Cloud Computing will have an increased impact on the enterprises and other organizations.

In comparison with Personal Computers (PC), mobile devices have less computing ability, memory space, keyboard screen, battery backup and connection bandwidth.MCC shifts the computation tasks onto large storage and multiprocessors through clouds.

In order to implement Cloud Services, Google has introduced Google Cloud Messaging (GCM) services which allow to send data from a server to Android powered devices and also to receive messages from devices on the same connection [5]. This free service sends a lightweight message informing the Android application of new data to be fetched from the server.

II. GCM

Recently launched Cloud to Device Messaging (C2DM) service by Google's Android had a big attraction.GCM is a push notification services that help developers to send data from server to the application to a particular Android device.

GCM allows a serverto send the messages asynchronously to mobile devices. As soon as there is a new update, the server sends a message to the device rather than application polling the server between several time intervals.Now, the application requests an update from the server, knowing it exists.

Earlier, we could achieve it by the service classes in Android. The drawback with the service class is that sometimes other processes may have high priorities and eventually the service class dies off. In GCM, there is no polling mechanism. The server pushes the data in form of push notification to the device and our application will start in the process. This service provides a simple, lightweight mechanism that servers can use to tell mobile applications to contact the server directly, to fetch updated application or user data. GCM handles all aspects of queuing of messages and manages a perfect delivery to the target application running on the target Android device.

Here are the primary characteristics of Android Google Cloud Messaging:

a. While using GCM service, it is not necessary that an application running all the time to receive messages. The device will itself wake up the application via Intent broadcast when the message arrives. The application is required to have broadcast receiver and all the required permissions.
b. This process simply delivers the message data straight to the application which manages its full control.
c. Android 2.2 or higher version must be installed on the device to use GCM.
d. It uses an existing connection for Google services.

e. It supports a message containing up to 4kb of payload data.

III. GCM ARCHITECTURAL OVERVIEW

Key terms and concepts involved in GCM are divided into 2 categories:-

A. *Components:*

a) ***Mobile Device:*** The device that is running an Android application that uses GCM. This must be a 2.2 Android device that has Google Play Store installed, and must have at least one logged in Google account.Alternatively, an emulator running Android 2.2 with Google APIs may be used for testing purpose.

b) ***3rd Party Application Server:*** An application server that developers set up as part of implementing GCM in their application. The 3rd party application server sends data to an Android application on the device via the GCM server.

c) ***GCM Servers:*** The Google servers involved in taking messages from the 3rd party application server and sending them to the device.

B. *Credentials:*

a) ***Sender ID:*** A project ID is acquired from the API console to identify an Android application that is permitted to send messages to the device

b) ***Application ID:*** The Android application that is registering to receive messages.

c) ***Registration ID:*** An ID issued by the GCM servers to the Android application that allows it to receive messages.

d) ***Google User Account:*** For GCM to work the mobile device must include at least one Google account if the device is running lower than android 4.0.4.

e) ***Sender Auth Token:*** An API key that is saved on 3rd - party application server that gives the application server authorized access to Google services

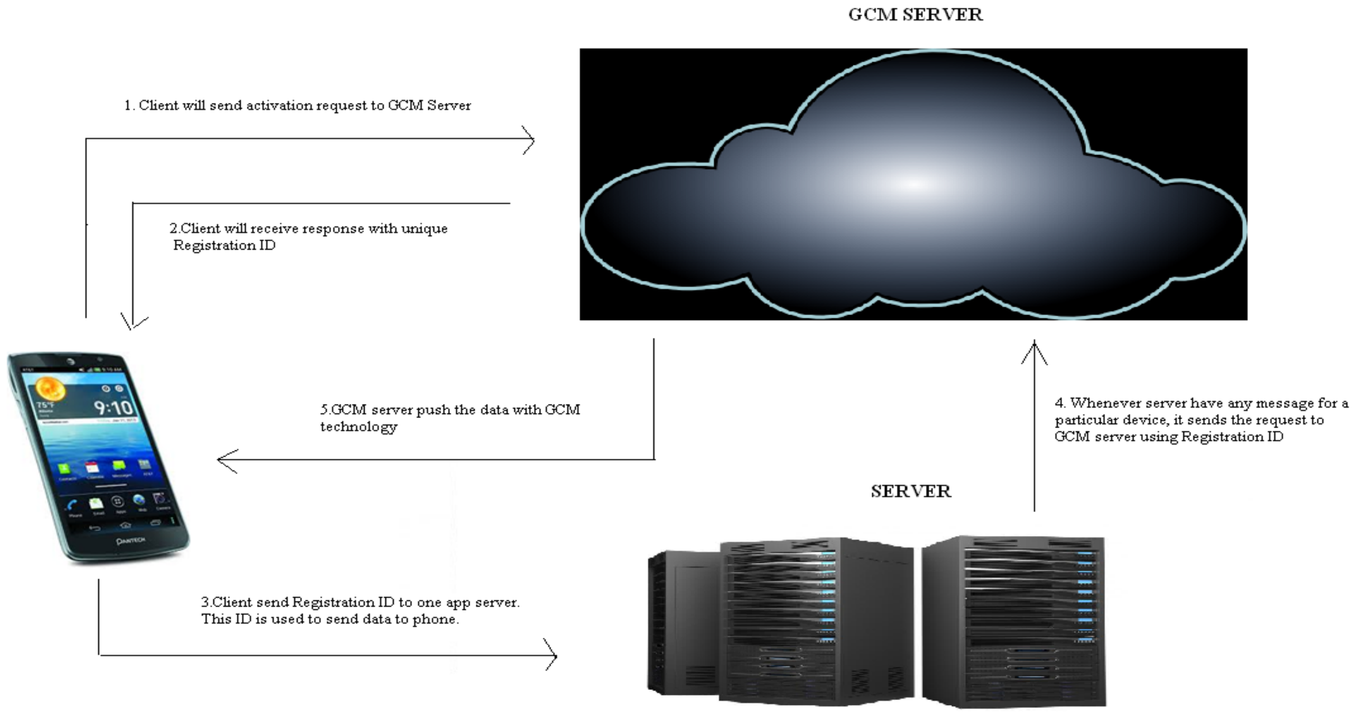
GCM SERVER
1. Client will send activation request to GCM Server
2.Client will receive response with unique Registration ID
5.GCM server push the data with GCM technology
4. Whenever server have any message for a particular device, it sends the request to GCM server using Registration ID
SERVER
3.Client send Registration ID to one app server. This ID is used to send data to phone.

Figure 1: GCM Architecture

IV. IMPLEMENTATION PROCESS

To start the integration process one should go to the API Console, get a server key, and put it in application. After this initial setup, anyone can enjoy the features of GCM, like message multicasting (send the same message to multiple devices), messaging collapsing (collapse message with the same collapse key, delivering only the last one), multiple servers for each application plus many more. An interesting development is the message life time field which can vary from 0 to 4 weeks. This feature allows to send messages that can be saved in the cloud for as long as it is predefined by the user if the device is offline and get delivered as soon as the device is available. Setting a message lifetime of 0 will send the message if the device is available at the exact moment, or simply discard it. Apart from that, it canalso select to delay the messages if the device is idle, which will treat the device as offline if the device is online but not active. Combining these approaches, it is possible to maximize the personalized experience.

After usual steps of creating a project the manifest uses-permissions will be as follows

a. GCM connects to Google Service:
android.permission.INTERNET
android.permission.ACCESS_NETWORK_STATE
android.permission.READ_PHONE_STATE

b. GCM requires a Google Account
android.permission.GET_ACCOUNTS

c. Awake the processor from sleeping when message is received
android.permission.WAKE_LOCK

d. Create a custom permission so that only this app can receive its messages-
PACKAGE. permission.C2D_MESSAGE
Where PACKAGE is the applications package name

e. Broadcast Receiver will receive the intents coming from GCM servers and handles them

f. The permission com.google.android.c2dm.permission.SEND is necessary so that only the GCM servers can send data messages to this app.

g. The action com.google.android.c2dm.intent.RECEIVE will be required to receive the actual message

h. The action com.google.android.c2dm.intent.REGISTRATION will be required to receive the registration id

V. ERRORS IN THE PROCESS

While implementing GCM, some errors may be there in the process.

A. Service_Not_Available:

The device can't read the response, or there was a 500/503 from the servers that can be retrived later. The Android Application should use exponential back off and retry.

B. Account_Missing:

There is no Google account on the phone. The android application should ask the user to open the account manager and add a Google account. Fix on the device side.

C. Authenitication_Failed:

Bad Google account password. The android application should ask the user to enter his/her google account password and let the user retry manually later. Fix on the device side.

D. Invalid_Sender:

The sender account is not recognized. This must be fixed on the Android application side. The developer must fix the application to provide the right sender extra in the com.google.android.c2dm.intent.REGISTER.intent

E. Phone_Registration_Error:

Incorrect phone registration with Google. This phone doesn't currently support.

F. Gcm.Invalid_Parameters:

The request sent by the phone does not contain the expected parameters. This phone doesn't support GCM.

VI. LIMITATIONS OF MOBILE DEVICES

The computing power, memory, battery power and screen size of the mobile device are small in comparison to the PCs. Connectivity issues and Connection bandwidth also pose challenges for mobile devices as the device is on the move depending upon the requirement of the user. Security and privacy are always a threat to the wireless devices.

VII. CONCLUSION

Mobile Cloud Computing is getting popularity but it is still at the early stage of the development and will require a lot of research for the maturity of the field.As far as the hardware is concerned, it will require better CPU, memory and resolution of the screen. Improvement in the battery backup had been a challenge for the hardware developers. Using GCM, the application no longer needs to poll the server for the updates, it saves both battery and data usage. The application gets also updated in almost real time, since the message processing and sending is done in as little as 4.7ms. Robust authentication algorithm and privacy needs to be ensure at the user end and cloud end respectively.

VIII. REFERENCES

[1]. Longgzhao Zhong, Beizhan Wang and Haifang Wei," Cloud Computing Applied in the Mobile Internet", 2012, IEEE, 7th International Conference on Computer Science and Education (ICCSE)

[2]. Weiguang SONG, Xiaolong SU, "Review of Mobile Cloud Computing", 2011, IEEE, 3rd International Conference on Communication Software and Networks (ICCSN).

[3]. Priya A. Kotwal, Adwitiy R. Singh, "Evolution and Effects of Mobile Cloud Computing, Middleware Services on Cloud, Future Prospects: A peek into the Mobile Cloud Operating Systems"2012, IEEE, International Conference on Computational Intelligence and Computing Research (ICCIC).

[4]. Han Q and Abdullah Gani, "Research on Mobile Cloud Computing: Review, Trend and Perspectives", 2012, IEEE, Second International Conference on Digital Information and Communication Technology and its Application (DICTAP).

[5]. http://developer.android.com/google/gcm/index.html