EUSRM
A Universe of Research
ISSN(ONLINE) : 2319-3069

Engineering Universe for Scientific Research and Management
(International Journal)
Impact Factor: 3.7

Vol. 6 Issue 10
October 2014

Implementation of Sas of Cloud for Mobile Computing

Arushi Singh Bhadouria[1], Pawan Patidar[2], Dr. M. K. Rawat[3]
M.Tech. Research Scholar, CSE, LNCT Indore, MP, India[1]
Asst. Professor, CSE, LNCT Indore, MP, India[2]
Professor & HOD, CSE, LNCT Indore, MP, India[3]
arushibhadouria15@gmail.com[1], pawanpatidar4u@yahoo.co.in[2], cshodlnct@gmail.com[3]

Abstract

Mobile devices are becoming the universal interface to online services and cloud computing applications. Since mobile phones have limited computing power and battery life, there is a potential to migrate computation intensive application components to external computing resources.

Our main aim in this paper to show my research work is to empower the resource constrained cell phone in a way such that by using cloud computing we can perform the computation on the cloud end and the mobile will get the final result. In this work we are going to open the MS word documents in the mobile phones that are not able to process some basic doc files which contain images, texts and the combination of different text styles or special data. The "Dynamic Outsourcing Mobile Computation to the Cloud "that was time consuming in response here we proposed new architecture that is faster than previous architecture. In this paper we will show the implementation details.

Keywords : cloud, offloading.

1. Introduction

In now a day's mobile devices are becoming the universal interface to online services and cloud computing applications. We can empower the processing power of any devices by using the cloud computing. In my proposed work, cloud computation will empower the mobile devices , so for they can process the document files because the basic mobile phones are not able to process and show the content of the document (. doc) files. The Cloud is an attractive platform for offloading due to its elastic resource provisioning and the ability to support large scale service deployment for the different computational work.

In this paper we can show the implementation detail by which we can save the processing time along with the battery life of the cell phone.

In this research work we are going to use the cloud computing for offloading of mobile computation that can increase the performance of the mobiles which don't have the processing and memory power like Desktop and Laptop computers. These research works also propose an architecture that will save the processing time along with the battery life of the cell phone. By using the cloud computation as well as we also changed the architecture that use the proxy and the code repository for the authentication purpose that is time consuming process.

2. Implementation

Implementations of the system based on the various methods classes and the library's of the technology that are help to achieved the-

2.1 User Defined Classes

In this section we provide the classes implemented by us, there are we include some important classes that are necessary for describe here.

Table 1: Description of user define classes

S.No.	***Class name***	***Description***
1	myTest	It is an J2me midlet and used to provide the user interface for mobile user and in same way provide the interface for selecting the file for open
2	UploadService	That is server side java web service class and help to upload files selected
3	DoHTMLService	This services are convert data from one format to

		other
4	DoConvert	This service is used to combine data which is accepted by the client
5	LoginService	This service is used to provide authentication for mobile client

2.2 System Classes and Library

Here we describe some classes that are provided by the java library and used with class for implementation.

Table 2: Description of System Classes and Library

S.No.	***Class name***	***Description***
1	java.io.File	An abstract representation of file and directory pathnames. User interfaces and operating systems use system-dependent pathname strings to name files and directories. This class presents an abstract, system-independent view of hierarchical pathnames.
2	java.util	Contains the collections framework, legacy collection classes, event model, date and time facilities, internationalization, and miscellaneous utility classes (a string tokenizer, a random-number generator, and a bit array).
3	javax.swing	Provides a set of "lightweight" (all-Java language) components that, to the maximum degree possible, work the same on all platforms.
4	java.awt.event	Provides interfaces and classes for dealing with different types of events fired by AWT components.
5	Java.sql.*	Provides the API for accessing and processing data stored in a data source (usually a relational database) using the Java programming language. This API includes a framework whereby different drivers can be installed dynamically to access different data sources. Although the JDBC API is mainly geared to passing SQL statements to a database, it provides for reading and writing data from any data source with a tabular format. The reader/writer facility, available through the javax.sql.RowSet group of interfaces, can be customized to use and update data from a spread sheet, flat file, or any other tabular data source
6	Javax.jws.webservice	Marks a Java class as implementing a Web Service, or a Java interface as defining a Web Service interface.
7	Javax.jws.webmethod	Customizes a method that is exposed as a Web Service operation. The associated method must be public and its parameters return value, and exceptions must follow the rules defined in JAX-RPC 1.1, section 5. The method is not required to throw java.rmi.RemoteException

2.3 Method Signature

This section contains the methods functions and other important members of the system with their description.

Table 3: Description of Method Signature

S.No.	***Method***	***Signature***
1	ValidateLogin()	That function work with client GUI and enable mobile user to authenticate
2	FileBrow()	This function help the user to create a file browser
3	showFile()	After conversion from cloud server this function help mobile user to view any file over local browser
4	traverseDirectory()	That is a mobile user function to traverse selected directory and its contains
5	showCurrDir()	This function help user to explore directories of mobile
6	readDocFile()	This function helps to read document or extract text and image from file
7	mergeFile()	This function used when the file is large and file is read in chunks after that chunks are recombine using this function

3. TESTING OF THE SYSTEM

Software testing is conducted to provide stakeholders with information about the quality of the product or service under test. Software testing can also provide an objective, independent view of the software to allow the business to appreciate and understand the risks of software implementation. Software testing can be stated as the process of validating and verifying that a software program / application / product –

1. Meets the requirements that guided its design and development;

2. Works as expected; and

3. Can be implemented with the same characteristics.

Software testing is depending on the testing method employed. Testing can be implemented at any time in the development process of the system. Testing takes place after the requirements have been defined and the coding process has been completed. As such, the methodology of the test is governed by the software development methodology adopted.

Different software development models will focus the test effort at different points in the development process. Newer development models, such as Agile, often employ test driven development and place an increased portion of the testing in the hands of the developer, before it reaches a formal team of testers. In a more traditional model, most of the test execution occurs after the requirements have been defined and the coding process has been completed.

3.1 Testing Adopted

To test our application we perform the unit testing in this method we test the functions of the prepared system. Unit testing refers to tests that verify the functionality of a specific section of code, usually at the function level. This is usually at the class level, and the minimal unit tests include the constructors and destructors. These types of tests are usually written by developers as they work on code (white-box style), to ensure that the specific function is working as expected.

For that reason we are apply UNIT TEST for our project. And we conclude the test results by screen by screen (by classes).

3.2 Test Cases

In this section we include the test case that reflects the correct working of all classes that implemented by us.

Table 4:Test cases execution plan

S.No.	*Class Name*	*Expected outcome*	*Actual outcome*	*Remark*
1	myTest	It is an J2me midlet and used to provide the user interface for mobile user and in same way provide the interface for selecting the file for open	Same as expected	Pass
2	UploadService	That is server side java web service class and help to upload files selected	Same as expected	Pass
3	DoHTML Service	This services are convert data from one format to other	Same as expected	Pass
4	DoConvert	This service is used to combine data which is accepted by the client	Same as expected	Pass
5	LoginService	This service is used to provide authentication for mobile client	Same as expected	Pass

4. Results & Discussion

This part of paper will describe results and the performance of our developed system on different parameters.

4.1 Conversion Accuracy

That is our main task to provide the better accurate conversion accuracy from the given set of data, that parameter is evaluated according to the user feedback basis, and provided ranks how much effectively and accurately the application provide the response for any data conversion.

Table 5: performance accuracy in %

Data size	*Accuracy (%)*
.5 MB	90
1 MB	70
2 MB	80
4 MB	95
6 MB	100
8 MB	70
10 MB	80

After analysis we found the system provide the proper response for conversion different data sited over the documents. but sometimes converted data may place in different other places by which the user response is decreases.

4.2 Memory Consumed

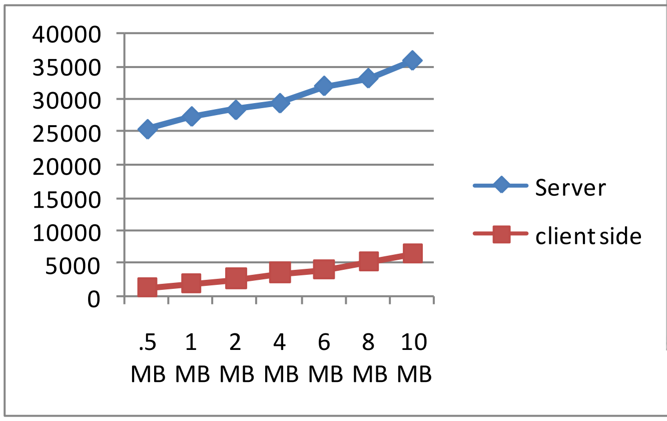
The amount of memory resource consumption during smooth execution of the program is known as memory consumption.J2meapplication is basically designed for simple calling and small utility, thus the required hardware to execute the larger task is too poor. The memory requirement is directly dependent over the size of data which is required to consume, this parameter is evaluated from both ends mobile device and as well as server end.

Table 6: Memory Consumptions at server and client end

File size	*Server side (KB)*	*J2me Client side (KB)*
.5 MB	23464	1283
1 MB	27392	2782
2 MB	25384	3282

4 MB	26392	3377
6 MB	27329	3988
8 MB	23721	5238
10 MB	24838	6329

After results evaluation of results we found that the size of data is not effect on the client machine it is directly affected by the server side execution. Which is clearly seen in the below given graph.



Graph 1 : shows memory consumption of both ends

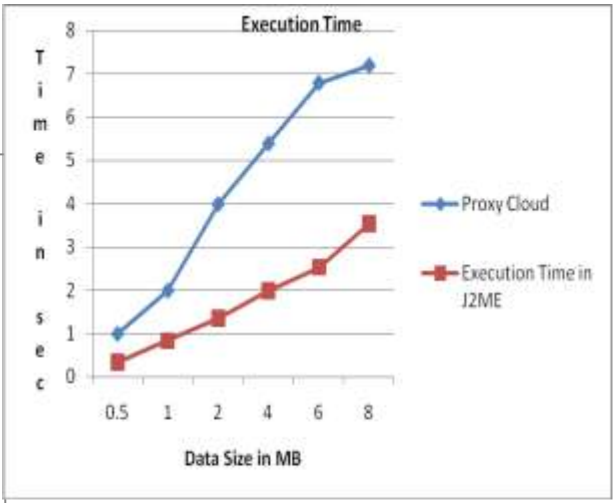
4.3 Job Execution Time

That another effective parameter which provide the information how much time consumed to convert data from doc and Doc format to HTML. That is measured at the server end and that results are the best results between the number of experiments derived by us.

Table 7: Shows the job execution time

Data size	***Execution time in j2me***	***Execution Time using proxy***
.5 MB	.3278	1
1 MB	.8248	2

2 MB	1.347	4
4 MB	1.372	5.4
6 MB	2.524	6.8
8 MB	3.53	7.2



Graph 2: shows comparative execution time

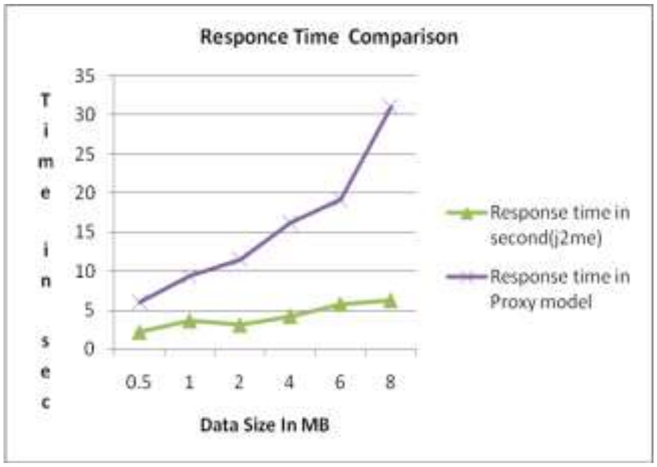
4.4 Response Time

The response time of the web domain is known as the time required providing the proper response after the job execution and the results obtained over the machine. in our case this time is measured as when we select the document to convert and in how much time we got a HTML response in our web browser from the remote machine. The results are produced here is a best results based on the number of experiments driven on the different size of data and compute their response time from the local mobile simulator.

Table 8: Show the Response time

Data size	***Response time in second(j2me)***	***Response time in Proxy model***
.5 MB	2.23	6
1 MB	3.72	9.35

2 MB	3.126	11.56
4 MB	4.249	16.2
6 MB	5.84	19.22
8 MB	6.32	31.12



Graph 3: shows the response time for different size of files

References

[1] Dynamic Outsourcing Mobile Computation to the Cloud ,Chonglei Mei, James Shimek, Chenyu Wang, Abhishek Chandra, and Jon Weiss man March 14, 2011

[2] CloudComputingDistributedInternetComputingforITandScientificResearch, Published by the IEEE Computer Society 1089 -7801/ 09 / $ 26 . 00 © 2009 IEEE

[3] Realization of Open Cloud Computing Federation Based on Mobile Agent, 978-1-4244-4738-1/09/$25.00 ©2009 IEEE

[4] Market-Oriented Cloud Computing: Vision, Hype, and Reality for Delivering IT Services as Computing Utilities, Grid Computing and Distributed Systems (GRIDS) Laboratory Department of Computer Science and Software Engineering The University of Melbourne, Australia

[5] The Case for VM-based Cloudlets in Mobile Computing, Mahadev Satyanarayanan, Paramvir Bahl, Ramon Caceres•, Nigel Davies Carnegie Mellon University, Microsoft Research,• AT&T Research, Lancaster University

[6] CLOUD COMPUTINGFOR MOBILE USERS:CAN OFFLOADINGCOMPUTATIONSAVE ENERGY? Authorized licensed use limited to: Purdue University. Downloaded on April 10,2010 at 16:10:45 UTC from IEEE Xplore
.

[7] A Virtual Cloud Computing Provider for Mobile Devices, ACM Workshop on Mobile Cloud Computing & Services: Social Networks and Beyond. MCS'10, June 15, 2010, San Francisco, California, USA. Copyright 2010 ACM 978-1-4503-0155-8

[8] Mobile Cloud Computing, IEEE COMSOC MMTC E-Letter, Dijiang Huang Arizona State University, Arizona, USA dijiang@asu.edu

[9] Securing Elastic Applications on Mobile Devices for Cloud Computing, CCSW'09, November 13, 2009, Chicago, Illinois, USA. Copyright 2009 ACM 978-1-60558-784-4/09/11 ...$5.00.

[10] Cloud Based Content Fetching: Using Cloud Infrastructure to Obfuscate Phishing Scam Analysis, Edward Ferguson, Joseph Weber, and Ragib Hasan Department of Computer and Information Sciences University of Alabama at Birmingham

[11] Cloud Computing Through Mobile-Learning, Cloud Computing Through Mobile-Learning, N.MallikharjunaRao1C.Sasidhar2 V. Satyendra Kumar3, Assistant Professor Annamacharya P.G College of Annamacharya Computer Studies, and Sciences, Rajampet, AP, India Rajampet, AP, India Rajampet, AP, India