INTERNATIONAL JOURNAL OF PURE AND APPLIED RESEARCH IN ENGINEERING AND TECHNOLOGY

A PATH FOR HORIZING YOUR INNOVATIVE WORK

MOBILE DISPLAY SOLUTION FOR CLOUD

PROF. NIKHIL. S. BAND

Department of Computer Science & Engineering, IBSS COET, Amravati.

Accepted Date: 05/03/2015; Published Date: 01/05/2015

Abstract: With the advancement in wireless technology in last few decades and the encroachment in mobile computing in last few years has led to large amount of services being made available to mobile end user. With this tremendous increase in mobile cloud computing the mobile devices today thus needs to capture user inputs and render the display updates received from the distant server quickly. However the variations in wireless channel, short battery lifetime and interaction latency commence major challenges for the remote display of cloud applications on mobile devices. This paper, discusses an adequate solution to tackle the main issues associated with the remote display of cloud services on mobile devices.

Keywords: Mobile phone, Mobile Cloud Computing, web services, Cost effective Cloud.

Corresponding Author: MR. VAIBHAV K. LIKHITKAR

Access Online On:

www.ijpret.com

How to Cite This Article:

Nikhil S. Band, IJPRET, 2015; Volume 3 (9): 1561-1571

PAPER-QR CODE

INTRODUCTION

Cloud Computing is a buzz-word in today's information technology (IT) that nobody can escape. Cloud computing uses virtualization and the modern Web to dynamically provide resources of various kinds of services which are provisioned electronically. Moreover these services are available in a reliable and scalable way so that multiple consumers can use them either explicitly upon request or simply as and when required. An often-quoted definition of cloud computing was established by the National Institute of Standards and Technology (NIST) in the U.S. [1].

According to ABI Research [2], "By 2015, more than 240 million business customers will be leveraging cloud computing services through mobile devices, driving revenues of $5.2 billion". Mobile cloud computing (MCC) thus serves as a benefactor for mobile device users. Users are able to share resources and applications without a high level of capital expenditure on hardware and software resources. Further the 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. One another of the key aspects that has led to eminence of cloud computing is that it hides the complexity of IT technology from users and developers. No need to know details of how a service is generated – it is the service provider's job to provide a corresponding abstraction layer. End users will see a plethora of unique features enhancing their phones because of mobile cloud computing.

Figure 1 shows the four layered model architecture of a so called cloud and virtualization; The Hardware, IaaS (Infrastructure as a Service), PaaS (Platform as a Service), SaaS (Software as a Service) [2].

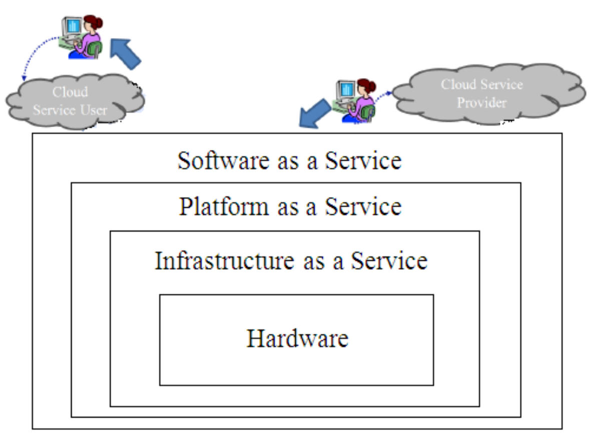

Figure 1: Cloud Layers

A. Hardware

The Hardware refers to the highly capable computing and networking equipment. It includes efficient processing engines, storage solutions, fast networks and larger memory.

B. Infrastructure as a Service

As there would be a large number of users to utilize the resources simultaneously, management is necessary to distribute the hardware among different users i.e.,OS and its virtualization. The same hardware will behave as different machines for different users. The users will be allocated with dedicated CPU, Memory, and Disk space virtually depending upon their liability.

C. Platform as a Service

It points to the programming models, execution method and programming environment. The service should be capable of providing different execution environment [2].

D. Software as a Service

This is where the selected software will be availed to the user as a service. The users here would be accessing the software online and storing the data back in the cloud. The management of the services of different levels and the user accountability is implemented by an accounting server [3].

The essence of mobile cloud computing is to physically separate the user interface from the application logic. Only a viewer component is executed on the mobile device, whereas the actual applications run on distant servers in the cloud. Any remote display framework is composed of three components:

- Server side component that encodes and transmits the application graphics to the client.
- Viewer side component on the client
- Remote display protocol to transfer display updates and user events between the server and the client.

2. Challenges of Mobile Cloud Computing

Servers expose services that clients can interact with. These services are accessible to clients that would like to make use of these services. As a server exposes services, it exposes potential vulnerabilities that can be attacked. As mentioned earlier some of the common challenges of mobile cloud computing are wireless bandwidth availability, battery lifetime and interaction latency. A number of techniques have been proposed to overcome these challenges, some of which are discussed below.

While computing and communication capabilities of handheld devices have increased by orders-of-magnitude in the past two decades, battery energy density has only tripled in the same period of time [4]. Therefore, a variety of methodologies and techniques have been adopted to reduce the amount of energy drawn from the battery. These methodologies and techniques include low-power electronics, smart batteries [5] to report their state of charge (SoC) [6] to operating systems and applications, power-saving modes of operation of processors, wireless network interface cards, and displays, efficient algorithm design and implementations, and efficient graphical user interface (GUI) design. The concepts of proxies between servers and handheld devices and source level (i.e. server level) media stream control have also been introduced to make handheld devices energy efficient. The Transmission Control Protocol (TCP) is widely used in web-based applications to transport data between clients and servers in a reliable manner. Researchers have also worked on TCP-based energy saving in handheld devices. Specifically, there are two relevant cases in which TCP experiences idleness: full buffer and empty buffer [7]. The authors have proposed a technique to turn on/off the WNIC card of a handheld device based on the occupancy of TCP buffer.

As compared with wired access networks, bandwidth availability on modern broadband mobile and wireless technologies is limited and variable. Moreover due to user mobility and interference and fading effects the actual throughput further reduces. To overcome this limited BW availability techniques such as motion-based differentiated encoding [5], individual object encoding [6] and real-time adaptation of encoding parameters can be used.

In thin client computing, data spikes are caused by the need of transferring a large amount of screen update data in a very short time. These data spikes cause long interactive latencies when the available bandwidth is not enough, giving users unsmooth usage experiences. A hybrid cache-compression scheme, DSRS can be employed to reduce the data spikes. DSRS is a hybrid cache-compression scheme which reduces data spikes by caching their main contributors and uses the cached data as history to better compress the recurrent screen updates in possible data spikes. It costs only a little additional computation time and the cache size can be negotiated between the client and server. Further using remote idle memory as secondary store for networked systems is motivated by the observation that network speeds are getting faster more quickly than are disk speeds [10]. However, in handheld devices storage memory is small or absent due to weight, size, and power constraints. Therefore, it is useful to evaluate the efficacy of implementing a network-based virtual memory system for handheld devices.

A key characteristic of a network-based virtual memory system is the need for additional energy cost incurred due to page swapping over the wireless link between a handheld device and a fixed network. It is important to note that the effectiveness of any DPM (Dynamic Power Management) strategy is strongly dependent upon the workload characterization [11]: the higher the burstiness of the workload, the longer the idle periods, and consequently, the higher is the energy saving.

Further Long-distance redundancy reduction scheme (LDRS) can be employed, to reduce the data redundancy occurring at long distance. LDRS is based on the dictionary-based Lempel-Ziv algorithm (LZ) [12], which is widely used in thin client systems. LDRS extends the history buffer of LZ algorithm in a vertical and discrete way. This gives LDRS more flexibility and scalabilities over a flat extension.

The existence of static objects in an application's GUI, a static object cache technique to reduce the network traffic caused by redundant transferring of static objects' presentation data greatly helps. Our experiment results show that this approach reduces both the baseline VNC's network traffic and interactive latency up to 60% with only a little more CPU usage. While this application-specific approach is efficient, it requires offline profiling to collect static object information of an application. Hence it's only suited for the enterprise environment where applications need to be shared by many users [12].

One of the newest and still researched on technique is forming a cloudlet, in which rather than relying on a distant “cloud,” a mobile user instantiates a “cloudlet” on nearby infrastructure and uses it via a wireless LAN. Crisp interactive response for immersive applications that augment human cognition is then much easier to achieve because of the proximity of the cloudlet. Further rapid customization of cloudlet infrastructure, is achievable through dynamic VM synthesis [13]. This idea has opened the door to a new world of mobile computing in which seamless cognitive assistance of users occurs in diverse ways at any time and place.

3. Proposed Method

By storing their data into the cloud, the data owners can be relieved from the burden of data storage and maintenance so as to enjoy the on-demand high quality data storage service. But the fact that data owners and cloud server are not in the same trusted domain may put the outsourced data at risk. Thus it requires the data to be encrypted prior to outsourcing for data privacy and preventing unwanted accesses. However, data encryption makes effective data utilization a very challenging task given that there could be a large amount of outsourced data files. Further, this complexes data searching as the individual users might want to only retrieve certain specific data files they are interested in during a given session.

The main challenge is the requirement of a high efficiency. To make search really interactive, for each keystroke on the client browser, from the time the user presses the key to the time the results computed from the server are displayed on the browser, the delay should be as small as possible. Providing a high efficiency on a large amount of data is especially challenging because of two reasons [12].

- The query keywords may appear in different attributes with an arbitrary order, and the "on-the-fly join" nature of the problem can be computationally expensive.
- To support fuzzy search by finding records with keywords that match query keywords approximately.

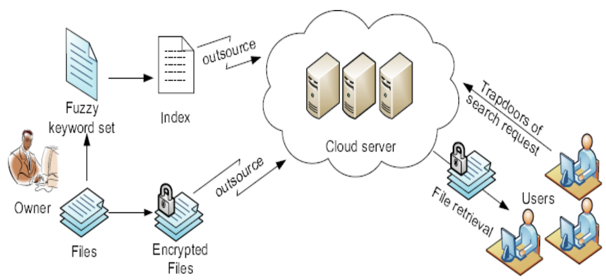

Figure 1: Architecture of Fuzzy Search.

In this paper, we focus on enabling effective yet privacy preserving fuzzy keyword search in Cloud Computing. Figure 1 above shows the general architecture of Fuzzy search environment. The data owner stores the data files in encrypted format along with the index file which later helps to simplify and fasten the searching of data files. Fuzzy keyword search greatly enhances system usability by returning the matching files when users' searching inputs exactly match the predefined keywords or the closest possible matching files based on keyword similarity semantics, when exact match fails. Further a wildcard-based technique is employed, for the construction of fuzzy keyword sets. This technique eliminates the need for enumerating all the fuzzy keywords and the resulted size of the fuzzy keyword sets is significantly reduced. Based on the constructed fuzzy keyword sets, we propose an efficient fuzzy keyword search scheme.

To illustrate the competence of the proposed method consider the situation as in figure 2 where there are two servers Home and College. The Home server has a table named tblHomeTasks. The College server has three tables named tblNormal, tblFuzzy, tblLowMarks, tblMediumMarks and tblHighMarks.

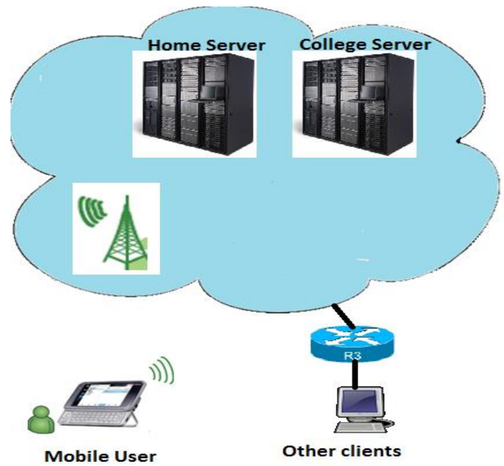

Figure 2: Model for proposed mobile display for cloud

The above scenario is implemented using Android and ASPcode. The Home server is used to store the information about the home tasks. Figure 3(a) shows the Login screen. After login the user can fill in the task info and click on insert to save the task info in *tblHomeTasks* table on the Home Server on cloud as in Figure 3(b). Now on the Android mobile the user launches the application named *CloudDisplay* as in Figure 3(c), the user logs in and can see the tasks created by him. Every time the user logs in the user is assigned a unique *token number* for the session thus making the access the more secure.

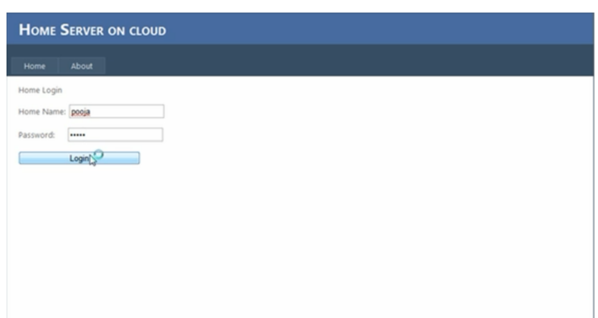

Figure 3(a): Login screen

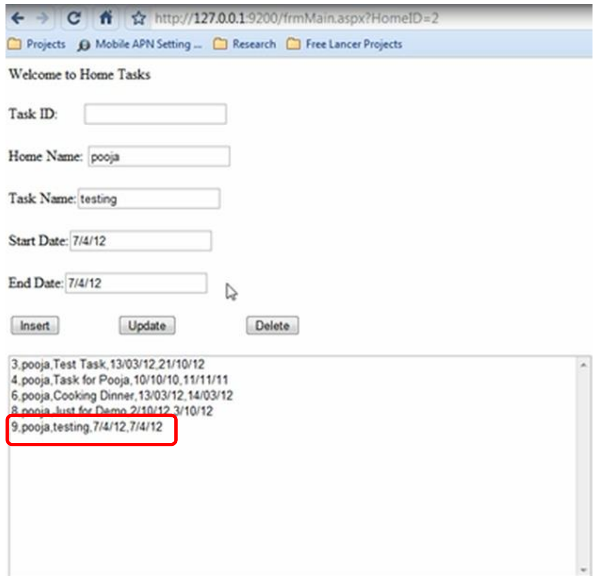

Figure 3(b): Inserting task info on Home Server on cloud

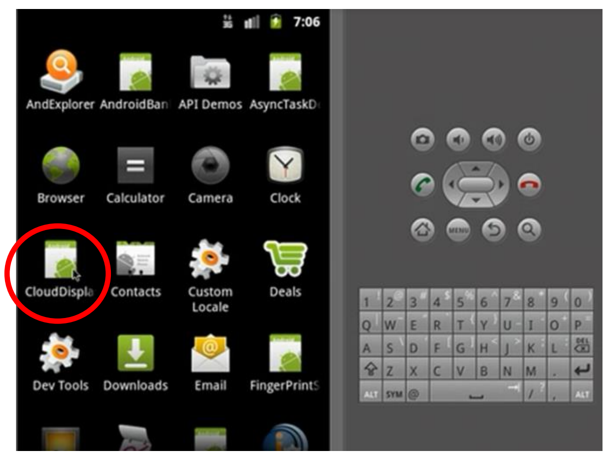

Figure 3(c): Application on Android Mobile

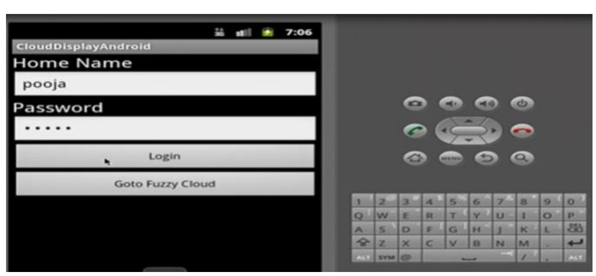

Figure 3(d): Login to Home Server in cloud

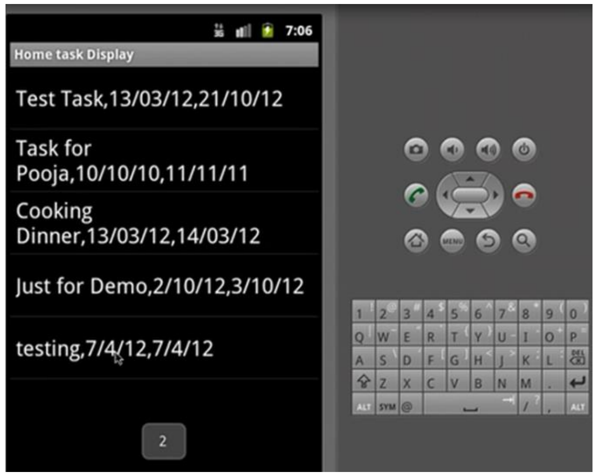

Figure 3(e): Tasks visible to mobile user searched from Home server

To highlight the importance and ease of fuzzy searching in cloud computing consider the College Server. This server stores the marks of the students. The marks are stored in *tblNormal*table whereas*tblFuzzy*serves as an index to sort the data into *tblLowMarks, tblMediumMarks and tblHighMarks*, as per the range of marks defined, this facilitates easy and fast searching of data.

Figure 4(a) and (b) shows the data being inserted into the tables in college server. As in Figure 4(b) for student ID =10 and marks 45/50; the data is inserted in *tblNormal and tblHighMarks*. Now the data is searched using the Android application. Figure 4(c) shows the fuzzy search for students with marks within the range of 89 to 91. Now as the marks are in the higher range the search is carried out in *tblHighMarks*only this increases searching speed. Another thing to note here is that the search gives list of students with marks within 89 to 91% out of 50 as well. The same searching takes 150 ms whereas with fuzzy searching requires 35 ms.

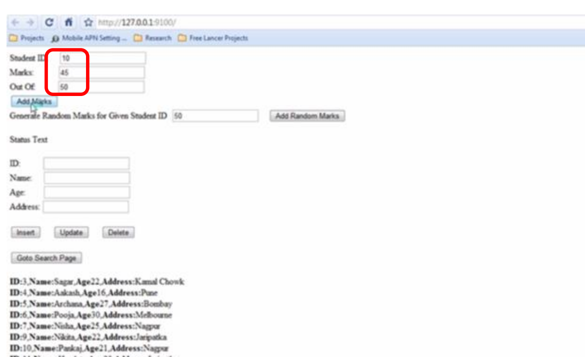

Figure 4(a): Inserting data into College Server

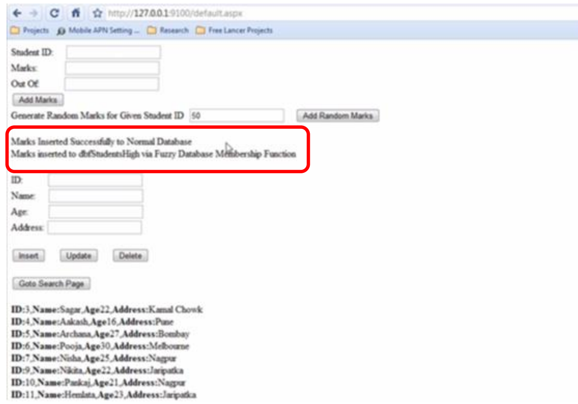

Figure 4(b): Data inserted into College Server

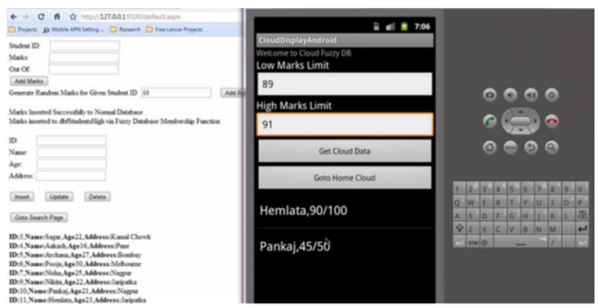

Figure 4(c): Fuzzy searching

4. CONCLUSION

In this paper, an adequate solution to tackle the main issues associated with the remote display of cloud services on mobile devices is presented. In order to reduce the searching time a fuzzy based searching technique is employed. The proposed method of fuzzy searching is not only fast but is more secure due to token assignment technique. The proposed method also indirectly addresses the problems of low bandwidth and limited battery life. The results show that the proposed method reduces the searching time by at least 25-30%.

REFERENCES

1. Christian Baun, Marcel Kunze, Stefan Tai and JensNimis, "Cloud Computing - Web Based Dynamic IT Servicres", Springer-Verlag Berlin Heidelberg 2011.

2. ABI Research. http://www.abiresearch.com/. 2010.

3. Martin Litoiu, Murray Woodside, Johnny Wong, Joanna Ng, Gabriel Iszlai, "A Buisness Driven Cloud Optimization Architecture", Proceedings of ACM in SAC'10, pp.380 – 385

4. R. Casas and O. Casas, "Battery Sensing for Energy-Aware System Design," IEEE Computer, Nov. 2005, pp. 48-54.

5. P. S. Marshall, "Brains and Brawns – The Power of Smart Batteries," Sensors Magazine, April 2002.

6. S. Piller, M. Perrin, and A. Jossne, "Methods for State-of-Charge Determination and Their Applications," Journal of Power Sources, June 2001, pp. 113-120.

7. D. Bertozzi, A. Raghunathan, L. Benini and S. Ravi, "Transport Protocol Optimization for Energy Efficient Wireless Embedded Systems," Proc. of Design Automation and Test Conference, 2003, pp. 706-711.

8. K. Pentikousis, "In Search of Energy-Efficient Mobile Networking," IEEE COMMUNICATIONS MAGAZINE, vol. 48, no. 1, pp. 95–103, JAN 2010.

9. K. Kumar and Y.-H. Lu, "CLOUD COMPUTING FOR MOBILE USERS: CAN OFFLOADING COMPUTATION SAVE ENERGY?" COMPUTER, vol. 43, no. 4, pp. 51–56, APR 10 2010.

10. T. Newhall, S. Finney, K. Ganchev and M. Spiegel, "Nswap: A Network Swapping Module for Linux Clusters," Proc. Euro-Par, 2003.

11. A. Acquaviva, E. Lattanzi and A. Bogliolo, "Power-Aware Network Swapping for Wireless Palmtop PCs," IEEE Trans. Mob. Computing, Vol. 5, 2006, pp. 571-582.

12. M. Bellare, A. Boldyreva, and A. O'Neill, "Deterministic and efficiently searchable encryption," in Proceedings of Crypto 2007, volume 4622 of LNCS.Springer-Verlag, 2007.

13. Sun Yang, "Optimization methods for the performance enhancement of thin client computing over bandwidth-limited and slow links", National University of Singapore, 2007.

14. M. Satyanarayanan, P. Bahl, R. Caceres, and N. Davies, "The Case for VM-Based Cloudlets in Mobile Computing," IEEE PERVASIVE COMPUTING, vol. 8, no. 4, pp. 14–23, OCT-DEC 2009.