Robust Mobile Cloud Services Through Offline Support

Maxwell Omwenga Momanyi
School of Computing and
Informatics Technology
Makerere University
Kampala, Uganda
Email: momwenga@cit.mak.ac.ug

Julianne Suzanne Sansa-Otim
School of Computing and
Informatics Technology
Makerere University
Kampala, Uganda
Email: sansa@cit.mak.ac.ug

Agnes F. Namulindwa Lumala
School of Computing and
Informatics Technology
Makerere University
Kampala, Uganda
Email: anamulindwa@cit.mak.ac.ug

***Abstract*— Cloud computing improves the efficiency of mobile applications. This is because data storage and processing are moved from the low resourced mobile devices to the cloud. Mobile cloud computing however, faces the challenge of intermittent cellular network connection since it is through the connection that services are reached. In this paper we argue that providing mobile cloud applications with temporary off-line file storage increases their robustness without burdening the low resourced mobile handset. We develop a module that offers off-line support which we plug into a mobile cloud calendar and we call it MobiCal.**

Keywords: ***Mobile Cloud Computing; Off-line Support; Robust Application***

1. Introduction

Cloud computing is a technology that uses the Internet and central remote servers to allow sharing of resources such as computing power, memory, storage, on-line applications and networking infrastructures [1], [2], [11]. Cloud computing allows consumers and businesses to use applications without local installation on site and to access their personal files at any computer with Internet access. Several companies such as Amazon, Google, Microsoft, HP, and IBM have developed their own cloud computing platforms. Figure 1 compares the cloud platforms of Amazon and Google.

Figure 1 shows that the cloud platforms for both Amazon and Google can be viewed from three layers namely: (1) the general purpose support services; (2) the cloud computing services; and 3) the client capabilities. The general purpose support services of both Amazon and Google have several components and APIs. The cloud computing services of Amazon run the platform of the users choice without much integration between the services. Google, on the other hand, uses widely available and open source languages and frameworks like Python and Django and through the Google Applications Engine provides some real glue that pulls together an integrated platform solution for its customers. As for the client capabilities, Amazon does not offer client-side tools and lets customers choose whatever they like while Google provides this flexibility but also provides many compelling client tooling options of its own. This type of computing

Figure 1: Amazon and Google Cloud Platforms [4]

paradigm provides many advantages for businesses, including service access flexibility, application efficiency, scalability, shorter start-up time for new services, higher utilization through virtualization, easier disaster recovery, and cost reduction in terms of lower maintenance, licensing and operation cost. Mobile devices are small, hand-held and generally low resourced. In recent times there has been advancement in processing power and storage capacity especially of smart mobile devices. Examples include (but are not limited to) iphone, samsung and HTC smart models. These however are not affordable in developing countries (which is the context of this study) which limits such countries to the low resourced

978-0-7695-4909-5/12 $26.00 © 2012 IEEE
DOI 10.1109/ACSEAC.2012.27

mobile devices like the highly demanded HUAWEI models.

These low resource devices, can only support light weight applications which are not even guaranteed to run efficiently. Since the cloud computing paradigm shifts both the data storage and processing away from the end devices to the cloud, it can improve the efficiency of mobile applications. Mobile G mail and Google Maps are two popular mobile cloud applications, however, both still use the data storage and processing capacities of mobile devices. In order to move mobile computing from the mobile device to the cloud, key issues such as energy efficiency, security, as well as connectivity and availability have to be considered.

This paper focuses on the connectivity and availability challenges. There is a very big challenge with the quality of service(QoS) of cellular networks in developing countries. More pronounced are the connectivity and availability attributes of this QoS. This creates a pressing need to address these attributes in the context of our study. Carnegie Melon University addresses availability through their CODA file system [6]. Firstly, CODA requires at least 256MB of RAM to run efficiently. Recall that on average, the mobile device models we are targeting usually support below 256MB of RAM. Secondly, CODA caches the entire files on to the local device permanently [8]. Recall also that the target devices for our study have limited storage capacity. Microsoft has the windows Azure platform which uses smaller but faster memory cache [12]. However, the mobile applications developed with Azure run on smart mobile devices. Therefore there is still a pressing need to address connectivity and availability in the context of our study.

2. Motivation

Intermittent cellular network connection is a key challenge in Mobile cloud computing. Remember that it is through the cellular network connection that services are reached. Intermittent cellular network connection can be caused by signal disturbance, spurious disconnection by hand- off, voluntary disconnection to use power efficiently and unpredictable transmission errors. A cloud based application stops working when a mobile device loses its cellular network connectivity. A number of mobile cloud applications exist to address the connectivity and availability challenge. These include (but are not limited to):

- *Mobile Market Agent Services (MASS)* [5] which is a cyber-market.
- *Cool-Tether* [9] which builds WiFi hot spots on-the- fly.
- *Universal Mobile Service Cell* [7] (UMSC) which monitors movement of mobile nodes between cells in a network.
- *Google applications* (Google docs [10], G mail Off-line [3] and Google maps) which offer word processing services, communication services and geographic services.
- *Microsoft Outlook* which is a mail client.

MASS, Cool-Tether and UMSC above do not offer off-line functionality. Google applications and Microsoft Outlook are available off-line but they utilize permanent storage of the divide for the off-line service. Apart from Google maps, all other Google applications and Microsoft Outlook also offer mirror storage as cloud. The reader should remember that storage is a key challenge for mobile devices. Therefore it is important that when off-line services are available, temporary storage is used. We found no mobile cloud application in literature that offers off-line temporary storage.

We argue that temporary off-line file storage can increase robustness of mobile cloud-based applications in the face of intermittent connectivity, while relieving the mobile device of the burden of using its permanent storage. This temporary storage allows continued access to the application. In our research we develop, a module for the mobile device that supports cloud-based applications by temporarily saving files for off-line use. Unlike CODA [8], our module temporarily caches a piece of the system on the local device. Our module also supports minimal local processing since most processing is done in the cloud while on-line. We plug the module into a mobile cloud calendar as proof of concept.

3. The Application

We develop the off-line module using HTML5, PHP and Google Gears. HTML5 is a scripting language which offers local caching. PHP is a light weight server end scripting language. Google Gears is a browser plug-in that manages the off-line process by ensuring that specific files are available when a device is disconnected. This module is plugged into a calendar, which is a cloud application.

This database driven calender is a light weight open source application, which makes it suitable for mobile devices. We call this application MobiCal. While in the cloud it uses a database. We extract the appropriate data files for off-line purposes using an XML engine and a manifest file. The following steps (illustrated in Figure 2) describe how MobiCal works.

- Install the gears plugin.
- On first access to MobiCal, the user is required to SAVE files which will be used for off-line purposes on to the mobile device.
- While the user is accessing MobiCal, the mobile device may LOSE CONNECTIVITY. The disconnection could be caused by signal disturbance, spurious disconnection

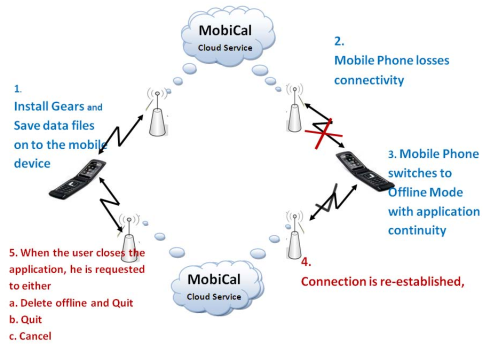

Figure 2: High Level Architecture of MobiCal

by hand-off, voluntary disconnection to use power efficiently or unpredictable transmission errors.

- Due to the disconnection, the mobile device automatically switches to OFF-LINE MODE, thus providing the user application continuity.
- While in off-line mode connectivity may be RE ESTABLISHED due to
  - Voluntary connection by user
  - Mobile device is within the network cover range
- On QUITTING, the application deletes all the off- line files from the mobile device which guarantees temporary off-line storage.

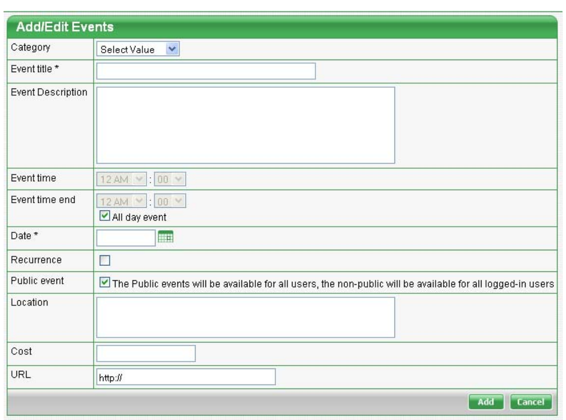

Figure 3: Add Events Form

MobiCal requires a first time user to create an account whose credentials the user will log in to access it whenever they wish to. Once logged in the user may add or edit the events (see Figure 3) they wish to schedule. In addition the user (while in off-line mode), may view a whole months calendar (see Figure 7), highlighting summaries of all scheduled events for that month. While in the month view, the user may see the details of any specific event by selecting it (shown in Figure 4). MobiCal also provides a search feature, which allows the user to search by event keyword, category or date. Figure 5 shows MobiCal as it captures files which would be used in the event that the mobile device is disconnected. MobiCal supports garbage collection (in on-line or off-line mode) to ensure temporary storage of any files

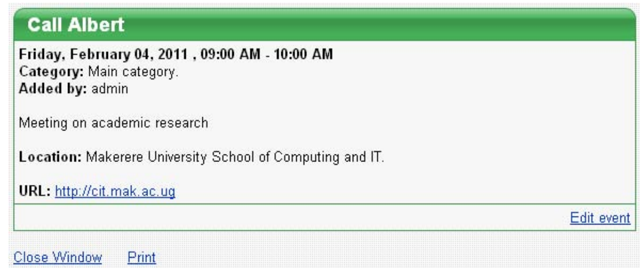

Figure 4: View an Event

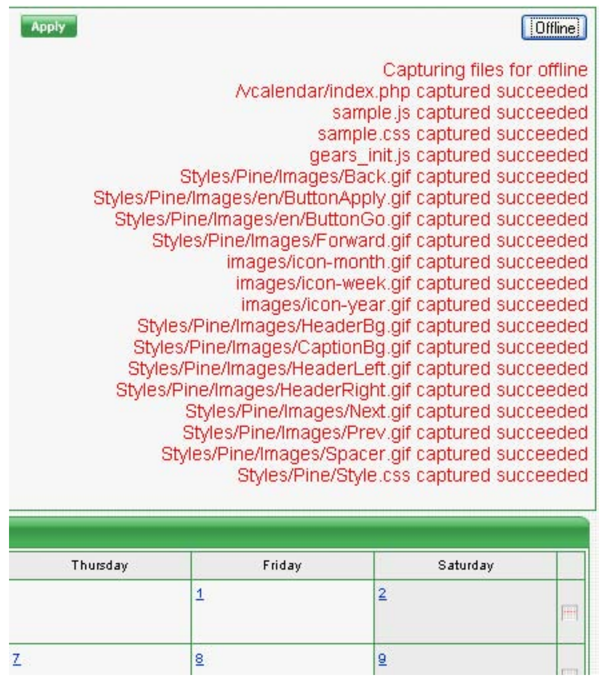

Figure 5: Files being downloaded for offline

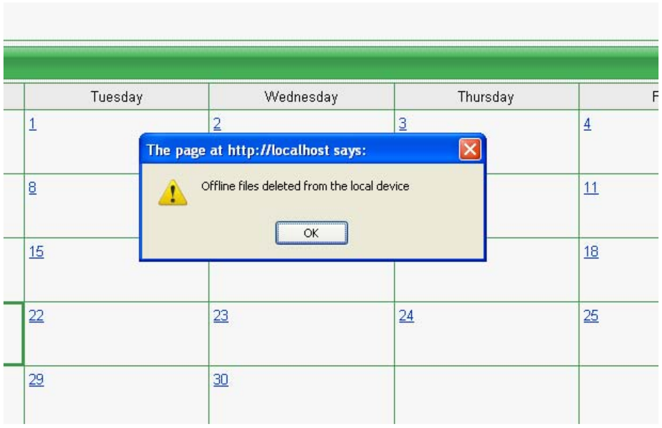

Figure 6: Garbage Collection

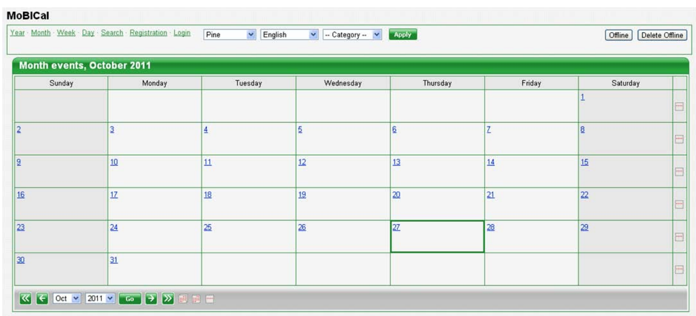

Figure 7: A whole Month's view

which may be downloaded on to the device. Figure 6 shows a successful deletion of files from the mobile device during garbage collection.

4. Conclusion

Mobile cloud applications stop working when the mobile device loses network connectivity. In this paper we develop a module that provides temporary file storage for off-line purpose (by ensuring deletion of files from the mobile device when they have served their purpose), thus increasing the robustness of the mobile cloud applications without burdening the low resourced mobile handsets. We plug this module into a mobile cloud calendar and call it MobiCal.

References

[1] M. Armbrust, A. Fox, R. Griffith, A. D. Joseph, R.H. Katz, A. Konwinski, G. Lee, D. A. Patterson, A. Rabkin, I. Stoica, and M. Zaharia. Above the clouds: A berkeley view of cloud computing. Technical Report UCB/EECS-2009- 28, EECS Department, University of California, Berkeley, Feb 2009.

[2] R. Aversa, B. Di Martino, M. Rak, and S. Venticinque. Cloud agency: A mobile agent based cloud system. In CISIS, pp. 132137, 2010.

[3] D. R. Herrick. Google this!: using google apps for collaboration and productivity. In Proceedings of the 37th annual ACM SIGUCCS fall conference, SIGUCCS 09, pp. 5564, New York, NY, USA, 2009. ACM.

[4] D. Hinchcliffe. Comparing amazons and googles platform-as-a- service (paas) offerings, 2008. ZDNet Newsletters.

[5] P. Jung Kim and Y. Ju Noh. Mobile agent system architecture for supporting mobile market application service in mobile computing environment. Geometric Modeling and Graphics, International Conference on, 0:149, 2003.

[6] J. J. Kistler and M. Satyanarayanan. Disconnected operation in the coda file system. ACM Trans. Comput. Syst.,1992, Vol.10, No.1,pp. 325.

[7] Q. Liu, X. Jian, J. Hu, H. Zhao, and S. Zhang. An optimized solution for mobile environment using mobile cloud computing. In Proceedings of the 5th International Conference on Wireless communications, networking and mobile computing, WiCOM09, pp. 54835487, Piscataway, NJ, USA, 2009. IEEE Press.

[8] M. Satyanarayanan, James J. Kistler, P. Kumar, M. E. Okasaki, Ellen. H. Siegel, and D. C. Steere. Coda: A highly available file system for a distributed workstation environment. IEEE Trans. Comput.,1990, Vol. 39, No. 4, pp. 447459.

[9] A. Sharma, V. Navda, R. Ramjee, V. N. Padmanabhan, and E. M. Belding. Cool-tether: energy efficient on-the-fly wifi hot-spots using mobile phones. In Proceedings of the 5th international conference on Emerging networking experiments and technologies, CoNEXT 09, pages 109120, New York, NY, USA, 2009. ACM.

[10] J. Viega. Cloud computing and the common man. Computer, 2009, Vol. 42, No.8, pp.106 108.

[11] J. Yang, S. Park, H. Seong, H. Byun, and Y.K.Lim. A fast image retrieval system using index lookup table on mobile device. In ICPR, pages 14, 2008.

[12] H. Li. 2009. Introducing Windows Azure. Apress, Berkely, CA, USA.