

Building a Platform to Bridge Low End Mobile Phones and Cloud Computing Services

Fung Po Tso, Lin Cui, Lizhuo Zhang, and Weijia Jia

(City University of Hong Kong)

Abstract: Two waves of technology are dramatically changing daily life: cloud computing and mobile phones. New cloud computing services such as webmail and content rich data search have emerged. However, in order to use these services, a mobile phone must be able to run new applications and handle high network bandwidth. Worldwide, about 3.45 billion mobile phones are low end phones; they have low bandwidth and cannot run new applications. Because of this technology gap, most mobile users are unable to experience cloud computing services with their thumbs. In this paper, a novel platform, Thumb-in-Cloud, is proposed to bridge this gap. Thumb-in-Cloud consists of two subsystems: Thumb-Machine and Thumb-Gateways. Thumb-Machine is a virtual machine built into a low end phone to enable it to run new applications. Thumb-Gateways can tailor cloud computing services by reformatting and compressing the service to fit the phone's profile.

Keywords: mobile cloud computing; low end mobile phone; mobile OS; middleware

1 Introduction

Thumb-in-Cloud is a platform that delivers cloud computing services to low end mobile phones, and it has the potential to serve millions of people.

1.1 Motivations

A low end phone typically only has built-in applications and no means of installing or removing applications. Such devices usually have a 104 MHz ARM-7 processor and a lightweight Operating System (OS). In 2010, there were more than 5 billion mobile phones

in use in the world [1]. Of these, about 3.45 billion were low end phones. Although shipment of high end smartphones increased rapidly in recent years, low end mobile phones still played a dominant role in the market [2], particularly in Asia and Africa.

Cloud computing has become more widespread in recent years due to vigorous promotion by Google, Amazon, Yahoo and others. Cloud computing involves moving services, computation, or data off-site to a centralized internal or external facility or to a contractor. By making data available in the cloud, it can be more easily and ubiquitously accessed, often at much lower cost. The value of the data itself may also increase as opportunities for collaboration, integration, and analysis are enhanced on a shared common platform [3]. A mobile cloud user need only put fingertip commands into the Internet for

cloud services to "flow/push" onto the terminal screen.

However, because it is often not possible to install new applications on low end phones, many users cannot access powerful cloud computing services and storage. To bring these services to users, fundamental technology gaps should be filled:

(1) System gap: A low end mobile phone only has built-in software. This is burnt into the Read-Only Memory (ROM) during manufacturing to keep the price of the phone low. None of these built-in applications allow access to cloud computing services or even installation of new applications that would facilitate access. This means low end phones are extremely isolated. Also, their below average hardware is usually incapable of handling advanced services.

(2) Hardware gap: Cloud computing services are often in multimedia format. For low end phones, the display ability

The paper is partially supported by CityU Applied Research Grant (ARG) under Grant No. 9667033, Shenzhen Basic Research Grant under No. JC200903170456A, Shenzhen-HK Innovation Cycle Grant under No. ZYB200907080078A, RGC General Research Fund (GRF), HK SAR under Grant No. CityU 114609, CityU Applied R & D Centre (ARD (Ctr)) under Grant No. 9681001, and China NSF under Grant No. 61070222/F020802.

has been burned in and fixed. With very limited processing power, display processing cannot be handled by a low end phone.

(3) Energy gap: Low end phones have very limited battery life and do not meet the needs for the most cloud computing tasks, especially realtime data services.

(4) Communication gap: High bandwidth and a stable Internet connection are usually required for accessing cloud services. However, in low end mobile phones a stable and secure Internet connection cannot be guaranteed. Furthermore, mobile Internet access in some developing countries is still expensive for the general public.

1.2 Contributions

We have designed and implemented a novel and sophisticated Thumb-in-Cloud platform to overcome these challenges. Thumb-in-Cloud operates commercially in China to provide cloud services to low end mobile users. It includes two components: Thumb-Machine and Thumb-Gateways. Thumb-Machine is a virtual machine embedded in low end phones to enable them to execute and manage software applications. It is like a web browser that accepts user input, relays it to Thumb-Gateways, and displays results on the screen.

Thumb-Gateways are aware of the network environment and phone profile so that cloud computing services can be tailored to suit each low end phone.

To the best of our knowledge, Thumb-in-Cloud is the first system to bring cloud services to low end phones by loosening their system software and hardware. This is a step in a different direction for mobile computing because research communities have tended to focus on high end rather than low end mobile devices.

2 Related Works

A cloud service has three distinct characteristics that differentiate it from traditional hosting: It is sold on demand, typically by the minute or the hour; it is elastic, so that a user can have as

much or as little of a service as they want; and the service is fully managed by the provider [4]. Significant innovations in visualization and distributed computing, as well as improved access to high-speed Internet and a weak economy, have accelerated interest in cloud computing [5]. Since the service provider hosts both the application and the data, the end user can use the service from anywhere [6].

There are several well-known contributors to cloud computing [7]. Amazon Web Services is the largest public cloud provider. A major issue in cloud computing, especially with public clouds, is protection of user data. One concern is that providers themselves may have access to unencrypted customer data—whether it is on disk, in memory, or transmitted over the network [8]. To limit this exposure, many sources have recommended never giving providers access to unencrypted data or keys.

As data moves into the cloud, storage companies are taking advantage of virtualization and adding more memory to the data center. Storage virtualization can improve existing storage hardware and make provisioning easier, while adding memory to the data center can make accessing information faster [9]. Caching on the edge of the Internet is becoming a popular technique to improve the scalability and efficiency of clouds. Caches belonging to a cloud cooperate in order to serve misses and to maintain freshness of the cached document copies [10].

Today, there are already some good examples of mobile cloud computing applications, including mobile Gmail, Google Maps, and some navigation apps. However, the majority of applications today still perform most of the processing and store most of the data on the mobile device itself rather than in the cloud. A report from ABI research predicts the number of mobile cloud computing subscribers will total nearly one billion by 2014. There are two reasons why ABI believes cloud computing will become a force in the mobile world. First, the number of users

that cloud technology has the power to reach is far more than the number of smartphone users alone. Second, all that is needed is access to the Internet for access to mobile applications.

3 Thumb-in-Cloud

Thumb-in-Cloud resides at the edge of a generic cloud to provide lightweight cloud services to isolated low end mobile phones or a bridge to popular cloud services such as an online photo album. Thumb-in-Cloud provides text, image, sound and gaming services. In general, lightweight cloud services provided by Thumb-in-Cloud are satisfactory for the needs of low end mobile phone users. If they are not, Thumb-in-Cloud can relay service requests to a generic cloud service provider such as Google for further action.

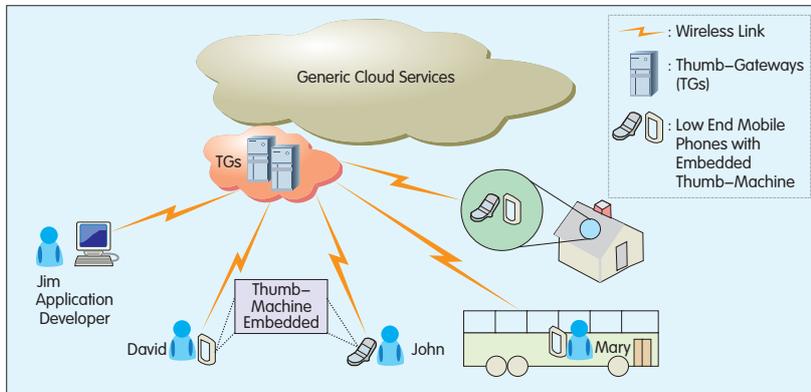
3.1 Overview

Fig. 1 is an overview of the Thumb-in-Cloud system. It comprises low end mobile phones and a group of supporting gateways. Implementing Thumb-in-Cloud is very challenging because low end phones are weak and every bit of them needs to be exploited to maximize processing capability. On the other hand, Thumb-Gateways need to carefully process content before pushing it to phones so that the content can be accepted.

In the following, we discuss four typical applications to show how Thumb-in-Cloud plays a critical role for cloud computing services on low end phones. Assume that Jim, David, John, and Mary have low end mobile phones. Jim is also an application developer (Fig. 1).

(1) Scenario 1: Jim is very satisfied with Thumb-in-Cloud. He has just developed a mini game and wants to release it to the public for free. He can do this by signing up to an application server in Thumb-Gateways and log his mini game to the application resource index.

(2) Scenario 2: David wants to receive digital entertainment but his low end phone does not allow it. With Thumb-Gateways and built-in



▲ Figure 1. Architecture of Thumb-in-Cloud.

Thumb-Machine middleware he can download new games from Thumb-Gateways and install them on his handset with Thumb-Machine.

(3) Scenario 3: Mary has subscribed for weather information and daily news, and every morning content is pushed to her phone. She can check weather conditions before leaving for work. She can also read the daily news on the bus.

(4) Scenario 4: John likes taking pictures so he subscribed to a photo album service with a well-known cloud service provider. He can do this with his low end phone via Thumb-Gateways. He first downloads an application to connect to Thumb-Gateways, and then Thumb-Gateways relay his album access requests to the service provider.

3.2 Architecture

Fig. 2 shows Thumb-in-Cloud system components. A low end mobile phone with Thumb-Machine communicates with Thumb-Gateways via the General Packet Radio Service (GPRS) network. Thumb-Gateways have three operational groups for delivering content, for computation, or for security support.

We lodged a Thumb-Machine in target low end mobile phones to allow them to accept new applications. Thumb-Machine includes a component that manages installed applications and downloads new applications from cloud. It is also a virtual machine that can efficiently translate application scripts to machine code and execute them.

Thumb-in-Cloud cloud computing services are provided by Thumb-Gateways, and Thumb-Gateways are formed by a group of specific servers—an MP-aware functional group, a resource group, or a developer group—depending on their service nature.

3.3 Thumb-Machine

A high end phone can become programmable by downloading or embedding new application software whereas a low end phone has built-in applications that cannot be updated with new software installations. One of the objectives of Thumb-in-Cloud is to enable low end phones to be programmable using low bandwidth wireless channels through support of Thumb-Gateways system. By implementing a container-software called Thumb-Machine, low end

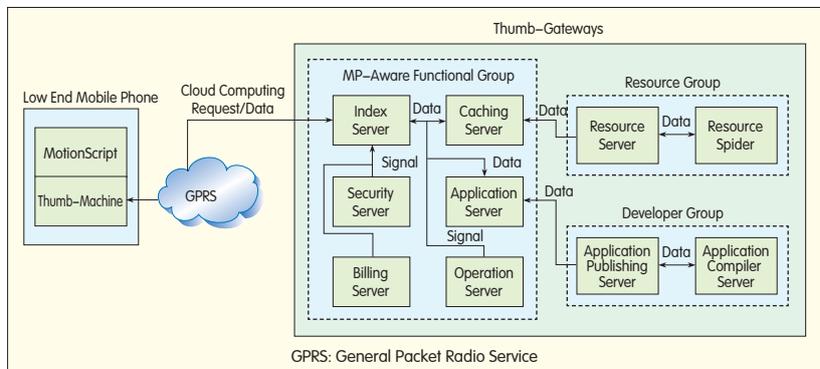
phones can load new applications dynamically without the need for a user to intervene or even be aware of the process. The implementation is particularly focused on low end phones such as MTK phones with Nucleus OS that usually cannot satisfactorily access the cloud.

A typical low end phone from MTK costs US\$30–50, which is reasonably affordable to the public. MTK phones often come with a 104 MHz ARM 7 processor and a lightweight OS.

High end phones achieve cloud computing access via a full operating system (iPhone OS, Android, Symbian, and Windows Mobile) and powerful hardware (1 GHz CPU, 512 MB RAM). Mid range phones usually have Java VM. Low end phones, which have the weakest hardware and lightest OS, often use binary loader or WAP to enhance their internet access capability. Although these solutions work well in their own classes, some are impossible to build into low end devices. Taking hardware limitations into account, we examined possible solutions to see if they fulfill user requirements. Comparisons of these possible solutions are listed in Table 1:

(1) Binary Loader: Although its object code is small and execution efficiency high, developing binary code is difficult for developers. In addition, binary code is highly dependent on platform because the binary level instruction set varies from platform to platform.

(2) WAP Browser: Wireless Application Protocol (WAP) is an open international standard for

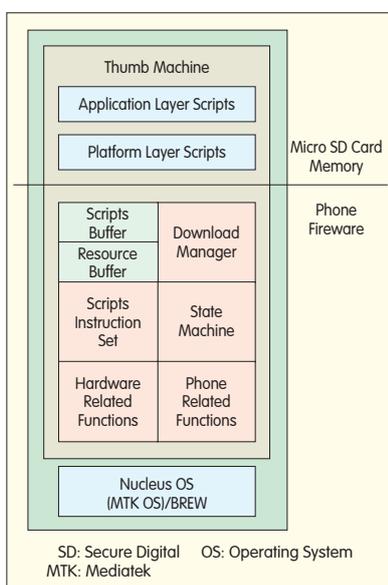


▲ Figure 2. Thumb-in-Cloud architecture and signals.

▼ Table 1. Comparison of binary loader, WAP browser, Java VM and Thumb-Machine

	Execution Efficiency	Application Portability	Development Difficulty	Application Object Size	Hardware Accessibility	User and Phone Interaction
Binary Loader	High	Hard	Difficult	Small	Easy	Good
WAP Browser	Low	Easy	Easy	Big	Difficult	Bad
Java VM	High	Easy	Easy	Big	Difficult	Fair
Thumb-Machine	High	Easy	Easy	Medium	Easy	Good

VM: Virtual Machine WAP: Wireless Application Protocol



▲ Figure 3. Architecture of Thumb-Machine.

application-layer network communications. A WAP browser provides mobile users with web-like access to Wireless Markup Language (WML) websites. Text-based web applications always come up short in execution efficiency, hardware accessibility, and object size.

(3) Java VM: Java's philosophy is to let application developers "write once, run anywhere." Using a Java Virtual Machine (VM), Java's bytecode can be run without change. But for low end mobile phones, Java VM is too big for the OS, and porting Java VM to the MTK platform demands even more effort than developing new middleware.

These solutions have different strengths in execution efficacy (binary loader), development simplicity (WAP), or code portability (Java); but their weakness is that they cost too much to

transplant into low end mobile phones. Binary loader lacks code portability; WAP is big and execution efficiency is low; and Java VM is too big and complicated to exist in MTK low end devices. Therefore, no existing alternatives are fit for low end mobile phones. Device weaknesses are detrimental to user experience. In light of this, a lightweight, scalable, efficient, and portable substitution is very beneficial to these devices.

The architecture of Thumb-Machine is illustrated in Fig. 3. Together with the OS, Thumb-Machine should be burnt into the ROM as phone firmware. Thumb-Machine consists of seven components which coordinate to execute application scripts. There are two types of buffers residing in the Thumb-Machine: script buffer and resource buffer. When executing an application script, Thumb-Machine loads a target script (application) into script buffer. During execution, the script may ask for additional resources; for example, a picture to display or a sound to play. The resource files are loaded into the resource buffer

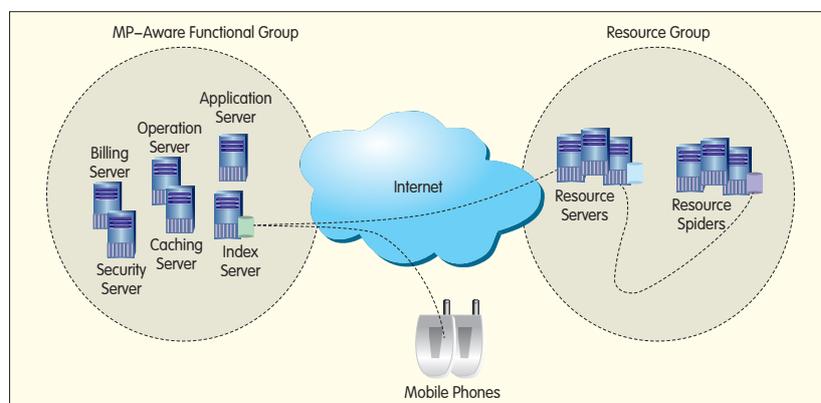
beforehand in order to speed up execution efficiency. Thumb-Machine also has a download manager for downloading a resource that is not locally available on the fly. The idea behind downloading on the fly is downloading only what you need; thus GPRS traffic is reduced. Download Manager also monitors the whole download procedure to avoid potential errors triggered by improper abandonment of a download.

3.4 Thumb-Gateways

A Thumb-Gateway System is composed of several groups of servers (Fig. 4) aimed at bridging low end mobile phones with dynamic application services and content rich cloud computing.

Thumb-Gateways can be divided into three groups based on functionality: a mobile phone-aware functional group, a resource group, and a developer group. The operational group provides operation and computation support to the whole system; the resource group distributes and collects resources from the cloud that the user is interested in or has subscribed to; and the developer group enables developers to rapidly create and publish applications for mobile phones.

The mobile phone-aware functional group supports fundamental operations such as resource indexing, application provision, and data mining of content usage, history, and subscription. This group involves an index server, an



▲ Figure 4. The Architecture of Thumb-Gateways system.

application server, an operation server, and sometimes a billing server. The index server is the entry point to Thumb-Gateways since every request arrives at the index server first. The index server interprets user requests and responds to them by dispatching the requests to actual destinations. The request can be classified as an application request or resource request. Index server redirects application requests to an application server; and upon receiving a request, the application server pushes the application(s) to the user's mobile phone. If users request content, the index server queries the caching server first or passes the request(s) to the resource server if the caching server reports a negative result. The billing server charges users according to their subscriptions.

In this group, an operation server works in the background to analyze content consumption and subscription history so that user interest in specific content can be determined. The operation server also analyzes UI specifications and display format of mobile phones. The system is then able to adjust content so that it is provided in resource server in a way that best fits users' interests, mobile phone UI, and display. At the same time a fine tuned billing strategy ensures users are charged fairly and accurately.

In the index server, all packets returning to clients are compressed using zlib [11]—which has a compression ratio of around 2:1 to 5:1.

The resource group is composed of resource servers and resource spiders. The resource server mainly provides storage space for text, pictures, and multimedia files. Resource servers are distributed at the edge of the cloud, serving every corner of the Internet provided they are registered in the index server. As well as resource provision, resource servers also have a unique feature to alleviate computation burden of low end mobile phones. This feature is borrowed from cloud computing, which promotes thin clients by shifting intensive computation tasks to servers. In real implementation, every mobile phone service request carries a

device fingerprint. Resource servers decode this fingerprint to extract the information such as screen resolution about a mobile phone and then pre-compute the proper organization of the content before it is pushed to the phone. As a result, a mobile phone need only display content on the screen without rearranging the content to be displayed.

In contrast, resource spiders are used for collecting content based on user subscription and analyzed results given by the operation server. Resource spiders retrieve desired information from the cloud regularly according to the information characteristics. For instance, resource spiders collect news provided free from news portals every morning at 5am. So users can read the news on the way to the office.

4 Conclusions

In this paper, we have presented an operational commercial system, Thumb-in-Cloud, for enabling cloud computing services access to low end mobile phones. In only one month, the system has been accessed by a wide range of low end mobile phone brands.

There are some technological hurdles in software, hardware, storage, and energy for some inexpensive mobile phones that prevent them accessing cloud computing. We have overcome these hurdles by introducing Thumb-Machine so that new applications (software) can be added onto the phone. Thumb-Gateways support cloud computing services or bridge them to general cloud computing services and Thumb-Gateways. We have elaborated on the design and implementation Thumb-Machine and highlighted some challenges encountered during research and development. Evaluation of Thumb-in-Cloud reveals that it fits very well into low end mobile phones and provides a more than satisfactory level of cloud computing service.

References

- [1] BBC. [online]. Available: <http://www.bbc.co.uk/news/10569081>
- [2] MobiThinking. [online]. Available: <http://mobithinking.com/mobile-marketing-tools/latest-mobile-stats>

- [3] M. Creeger, "Cloud computing: An overview," *ACM Queue-Distributed Computing*, vol. 7, no. 5, p. 2, Jun. 2009.
- [4] M. A. Vouk, "Cloud computing Issues, Research, and Implementations," *J. Comput. and Inf. Tech.*, vol. 4, pp. 235–246, 2008.
- [5] B. Hayes, "Cloud computing," *Commun. ACM*, vol. 51, no. 7, pp. 9–11, 2008.
- [6] R. Buyya, C. S. Yeo, S. Venugopal, J. Broberg, and I. Brandic, "Cloud computing and emerging it platforms: Vision, hype, and reality for delivering computing as the 5th utility," *Future Generation Computer Systems*, vol. 25, no. 6, pp. 599–616, 2008.
- [7] Infoworld. [online]. Available: <http://www.infoworld.com/d/the-industry-standard>
- [8] A. Fox, R. Griffith, A. D. Joseph, R. Katz, A. Konwinski, G. Lee, D. Patterson, A. Rabkin, I. Stoica, and M. Z. M. Armbrust, "Above the clouds: A Berkeley view of cloud computing," Dept. Electrical Eng. and Comput. Sciences, University of California, Berkeley, Rep. UCB/ECS 2009 28, Feb. 2009.
- [9] S. Higginbotham, "Who will cache in on cloud storage?," [online]. Available: <http://gigaom.com/2008/04/26/who-will-cache-in-on-cloud-storage/>
- [10] L. Liu and A. I. L. Ramaswamy, "Cache clouds: Cooperative caching of dynamic documents in edge networks," In *Proc. 25th Intl. Conf. Distributed Comput. Systems (ICDCS 2005)*, Columbus, OH, pp. 229–238.
- [11] Zlib. [online]. Available: <http://www.zlib.net>

Biographies

Fung Po Tso (posco.cs@cityu.edu.hk) received B.Eng. and M.Phil. degrees from City University of Hong Kong. He is currently working towards a Ph.D. degree. His research interests include the design and applications of mobile communication systems, and system performance measurement and optimization.

Lin Cui (lincui2@student.cityu.edu.hk) is a Ph.D. candidate in the Department of Computer Science, City University of Hong Kong. He received his bachelor's degree from Shandong University in 2007, and a master's degree from Harbin Institute of Technology in 2009. His research interests include mobile computing systems and wireless ad hoc/sensor networks.

Lizhuo Zhang (lizhuocs@cityu.edu.hk) received his Ph.D. degree from the Central South University, China. He is currently a senior research associate in City University of Hong Kong. His research interests include multimedia communication, wireless networks, and mobile systems.

Weijia Jia (wei.jia@cityu.edu.hk) is a professor in the Department of Computer Science, City University of Hong Kong, and director of Future Networking Center, Shenzhen Research Institute of City University of Hong Kong. He received B.Sc. and M.Sc. degrees from Center South University, China, and a Master of Applied Science and Ph.D. from Polytechnic Faculty of Mons, Belgium. He was a research fellow at the German National Research Center for Information Science (GMD) in Bonn (St. Augustine) from 1993 to 1995. His research interests include next generation wireless communication, protocols and heterogeneous networks, distributed systems, and multicast and anycast QoS routing protocols. He has published in various IEEE journals, has contributed chapters to books, and has refereed international conference proceedings.