Execution Environments for Distributed Computation Issues

Jorge Ejarque Mario Macías Lloret Íñigo Goiri

June 16, 2008

Abstract

Contents

Chapter 1

Towards the Pocket Workstation: Powering Mobile Applications in the Cloud

Mario Macías Lloret
Barcelona Supercomputing Center,
Jordi Girona 29
08034 Barcelona, Spain
mario.macias@bsc.es
http://www.bsc.es

abstract

Current mobile devices allow their owners to bring with them sophisticated applications because they provide complex execution environments for applications created with complete development platforms. Despite of this, mobile devices have intrinsic limitations in power, memory and battery which impedes programmers to develop applications as much as powerful as in laptops or desktop PCs. Another big difficulty for developers is the huge heterogeneity of architectures and platforms of the devices, which makes the development process of applications too expensive and limits the size of the target market. This paper explains how Cloud Computing can mitigate these limitations and allow the creation of resource-intensive applications which can be executed in the majority of modern cell phones or PDAs.

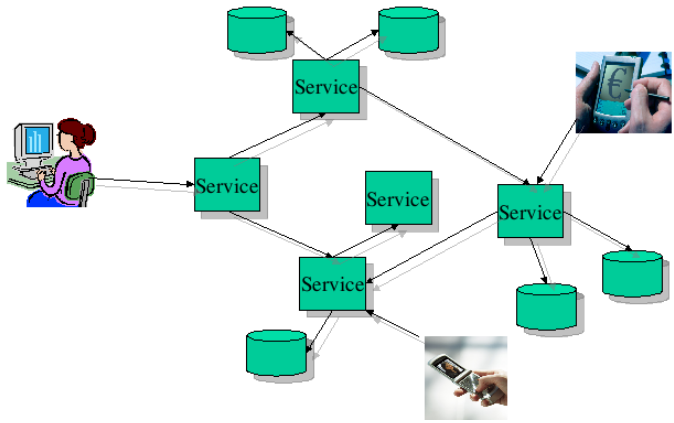

Figure 1.1: Cloud architecture

1.1 Introduction

Cloud Computing [1] "revives" the old idea of thin clients: simple and cheap devices are used as interfaces to servers that contains the applications and the computation power. The difference is that, while in the past were used central, powerful and expensive servers, nowadays the applications and the computation power is in "the Cloud": hundreds of small, cheap and simple computers interconnected through the Internet and distributed in location (figure 1.1).

Mobile phones are the perfect example of thin client for using Cloud-enabled applications: they are undoubtedly the most widely used electronic device on the planet. About a third of the people in the world have at least one (3.5 cellulars for each PC).

Instead of his name, the usage of mobile *phone* is rapidly changing away from purely voice communications. Nowadays mobile phones are used for several purposes, like sending and receiving multimedia content, play games, record pictures and video, listen music, connect to the Internet, scheduling applications, etc.

To support the demand of data applications, mobile devices capabilities are growing rapidly. According to [2], in 2003 2% of new handsets had slots for memory cards, nowadays they are the 45%, and

by 2011 the 75% will incorporate flash card support. Also data netwok bandwidths are increasing: some commercial networks support download speeds higher than 1 Mbps.

As the handset capacities are growing rapidly, some new applications that a few years ago were reserved to Personal Computers are being adopted by mobile devices.

This chapter is a guide for those application programmers and system designers who want to get introduced in the world of Internet mobile applications and Cloud Computing. The chapter is organized as follows: sections 1.2 and 1.3 explains which advantages and problems, respectively, must be had into account when designing and programming mobile applications; section 1.4 enumerates the solutions available to create mobile Cloud applications; section 1.5 shows some killer-applications that justifies the importance of mobile devices in the next-generation Cloud applications; and section 1.6 enforces critique thoughts about the present and the future of mobile access to the Cloud.

1.2 Advantages of mobile devices

There are important advantages that mobile devices brings to the end user:

Portability A normal laptop weighs about 3 kilograms and takes up a volume of approximately 3 litres. Nevertheless, a relatively big mobile phone as Nokia N95 weighs 120 g. and takes up 0.11 litres.

Low cost A device with full multimedia support, big screen device and Internet connectivity can be acquired by a few hundreds of euros. The cost is even cheaper if the user acquires the device by signing a exclusivity contract with its GSM service provider.

Functionality Far from the old embedded systems on mobile devices, nowadays mobile devices are easily programmable. This allows the creation of complete and functional user applications.

In addition to the advantages for end users, mobile devices also brings several advantages for application developers:

Programming frameworks Mobile devices use well-known application runtimes (e.g. Java) with a complete set of generic APIs to access the most common mobile capabilities. Furthermore, vendors provide developers with several free APIs to access the special functionalities of device families.

Development tools Most common IDEs support the big range of compilers, profilers, debuggers and emulators that device vendors provide freely to proggrammers. Thanks to them, the development cicle of a mobile application can be done completely without the physical target device.

Documentation All the frameworks and APIs are widely documented. Also there are hundreds of communities to exchange professional knowledge and solve issues.

Use of standards Most of the phones supports standard technologies, or those that have become *de-facto* standards like HTTP, HTTPS, Adobe®Flash, XHTML, JavaScript, CSS, etc...

This section has shown that mobile devices are not only cheap for their owners but also for the developers, because almost all the technologies, development tools and documentation are free.

1.3 Issues on mobile devices

Despite of advantages of mobility, there are some serious technical and ergonomic issues that impedes its intensive usage. The limitations are enumerated and some solutions are explained.

1.3.1 Performance and Battery

Nowadays, some mobile devices bring the performance that home computers had less than a decade ago. However, it is difficult to achieve a high processor performance during long time: the more powerful is a processor, the more battery it consumes.

The research fields to solve this problems are in the way of improving the performance/power ratio of the processors [3], [4] and the life of the batteries. Intel presented recently the Atom®processor [5],

which supports up to 1.6GHz clock frequency and a maximum of 2W of power consumption as response to AMD's Geode [6] processor, with a clock frequency of 600MHz and 2.3W of average power consumption. Also nVidia announced the future release of APX2500 [7], whith support for advanced Multimedia processing like 3D-accelerated graphics and high-definition video.

The solutions and technologies adopted to maximise the performance/power ratio are in the field of electronics and processor architecture, thus this topic will not be included in this chapter.

1.3.2 heterogeneity

There are hundreds of mobile manufacturers that every year create new mobile device models. Since there is not any standard for the hardware specifications of the thousands of available device models, the device architectures and implementations are quite extensive to cover all the market segments.

Because the huge heterogeneity of mobile device models, the application development process becomes very expensive since the application must be partially rewritten for all the target devices [8]. This problem is alleviated by the inclusion of a middleware solution between the user applications and the device hardware. Nevertheless standard middlewares restrain the improvement of technologies and the addition of new hardware features. This forces to manufacturers to add their own APIs to allow programmers take the most of the devices, and these new specifications increases the heterogeneity and the cost of porting process.

1.3.3 Ergonomy

Since the progressive increase of the usage of mobile applications, mobile devices ergonomics is an emerging field of research [9].

A key problem of the small mobile devices for its intensive usage is the ergonomy: most of keyboards are numeric and becoming more and more smaller, and that difficults the input of data from the user. This problem is being solved by adding touch screens or support for voice recognition in the devices.

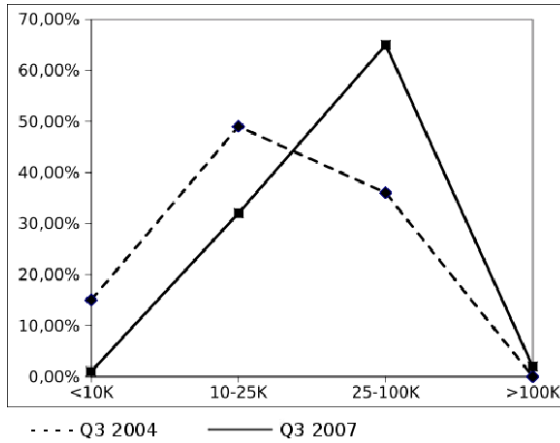

Figure 1.2: Comparison between percentages of screen sizes (Y-axis) in number of pixels (X-axis) in 2004 and 2007 (Source: [2])

Other problem is the size and resolution of the screen. If the screen is too small and the resolution too low, the intensive usage is unconfortable because the screen cannot display enough amount of data and the user interfaces must be too simple. Fortunately, the average screen size is growing dramatically in the last years, as can be seen in figure 1.2.

1.4 Mobile access to the Cloud: mitigating mobile devices limitations

The limitations exposed in section 1.3 are mitigated by some enterprise solutions and standards. This section enumerates the most important solutions available in the market, used by device manufacturers, content creators, application developers and service providers to bring the power of cloud computing into small and cheap handheld devices.

1.4.1 Execution platforms

The huge heterogeneity on hardware arquitectures and configurations enforces the need for application middlewares or specifications for allowing application developers to approach the ideal of "compile once, execute everywhere".

Java Micro Edition

Java Micro Edition (J2ME) [10] provides a flexible and portable environment for applications executed in devices with scarce resources. It incorporates a Java virtual machine and a set of standard APIs. Nowadays is the most extended application middleware in mobile phones.

The architecture of J2ME is composed by a variety of configuration, profiles and optional packages, in order to provide portability but also take the most of each device. The configurations are formed by a virtual machine and a minimum set of libraries which provides the base functionality for each range of similar devices. From all the configurations available, this paper focuses in those who targets the small mobile devices 1.3.

The *Connected Limited Device Configuration (CLDC)* is oriented to small devices which have connectivity restrictions in speed and robustness. It uses the K Virtual Machine [11], a reduced version of Java Virtual Machine with modifications to be faster and consume less memory and energy by sacrificing other aspects like security. The *Mobile Information Device Profile (MIDP)* offers basic functionality to mobile applications, such as user interfaces, network connectivity, local data storage and application administration.

MIDP functionalities can be extended by optional packages and vendor APIs, and provide to the application multimedia support, extended connectivity, advanced 3D graphics, and a long etc.

The main problem of Java ME is that most of the implementations have some differences between them, and this causes that the heterogeneity is not eliminated completely.

BREW

The *Binary Runtime Environment for Wireless (BREW)* is a software platform that can download and execute small programs. In the oppo-

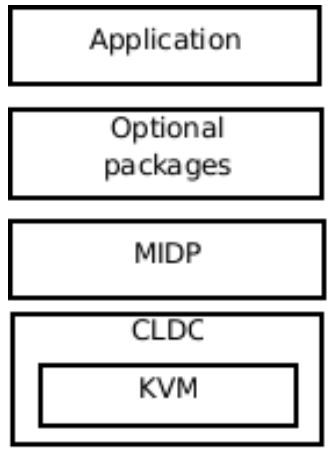

Figure 1.3: Java ME Connected Limited Device Configuration architecture (Source: [10])

site of Java Micro Edition, BREW doesn't uses any virtual machine to execute the applications, but provides a flexible software architecture (see figure 1.4) for allowing the easy porting of applications between all the BREW devices. This system is very extended in United States of America markets.

BREW uses the C++ language for programming applications which can use the advanced capabilities built into the hardware, such as the ARM CPU core, audio synthesys, voice recognition, bluetooth, OpenGL, GPS, advanced video and audio, etc...

Symbian

Symbian [12] is an Operating System for mobile devices, created from the alliance of several of the bigest mobile phone companies.

The Symbian architecture [13] is formed, from bottom to top, by the next layers:

Kernel Services and Hardware Interface Layer Provides low-level communication with hardware to upper layers

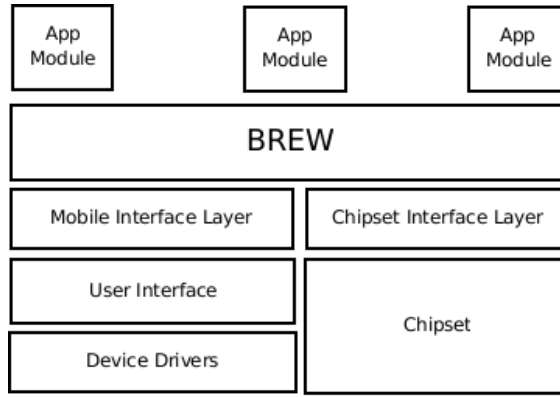

Figure 1.4: The BREW Device software Layering

Base Services Layer Is the lowest level reachable by user-side operations, it includes the File Server and User Library, the Plug-In manager, database management system and cryptographic services. It also includes the text shell support.

OS Services Layer Provides services for communications, multimedia and graphics, connectivity and some generic OS services. The Java Virtual Machine support is over this layer.

Application Services Layer Provides UI-independent services to the applications. It includes system services used by all applications, such as application framework, technology-specific logic such as support for messaging and multimedia protocols.

UI Framework Layer Brings the basic user interface building blocks for the applications. It acts as an interface between Symbian OS and the variant UI layer, since Symbian OS doesn't include an actual user interface and the implementation of it is left to the licensees.

Since Symbian is a complete operating system, it can be programmed in any language and use lots of sotware third-party libraries. Probably it is the most flexible of the available execution platforms

because, like in a PC, anybody can create their own services, applications and libraries, and it is relatively easy to port them from other platforms.

Android

Android [14] is a software stack for mobile devices that includes an operating system, middleware and key applications, as can be seen in figure 1.5. Its main features are the next:

- Application framework that allows the reuse and replacement of components.
- Integrated navigator, based on the Webkit [15] engine.
- 2D and 3D (OpenGL) graphics libraries
- SQLite database support
- Audio, video and imaging support
- Connectivity access, such as Bluetooth or WiFi
- Optional access to camera, GPS, compass or accelerometer

The main parts of android architecture are:

Linux Kernel and drivers Provides a complete operating system and its drivers to access all the device hardware.

Libraries Implements basic services to applications, such as graphics functions, secure connections, multimedia, basic user input, etc.

Android Runtime It is formed by a Virtual Machine to execute protable bytecodes and core libraries to provide basic functionalities to the upper layers.

Application Framework Basic services for Android applications, such as resource management, telephony management, content provisioning, system access or windowing.

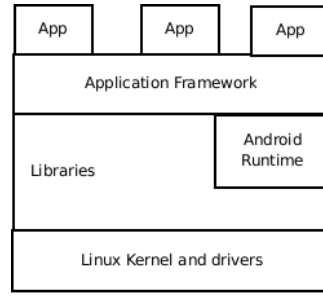

Figure 1.5: Android software stack

Like in the other alternatives, Android provides a rich development framework which includes emulators, debuggers, profilers and plugins for Integrated Development Environments. It is programmed in C++, and includes a set of libraries to access the functionalities of the core components of Android:

- An implementation of the **Standard C System Library** tuned for embedded Linux devices.
- Media libraries for imaging, which supports playback and creation of many popular formats: MPEG4, MP3, JPG, PNG...
- 2D and 3D graphics primitives libraries.
- Font rendering libraries
- Database libraries

1.4.2 Networking

The key technologies in cloud computing are those related with networking, since it allows low performance devices to access the power of the Cloud. Nowadays, there is a large ecosystem of protocols and specifications which will be described in this section.

Network protocols

The Wireless Application Protocol (WAP)[16] is a stack of network protocols which allow small mobile devices to connect to the net. The first version of WAP is intended for the first devices in the market, which did not have enough performance to support traditional communication protocols and standars, as TCP/IP.

Instead of using TCP/IP, it was formed basically by Wireless Session Protocol, Wireless Transaction Protocol, Wireless Transport Layer Security and Wireless Datagram Protocol. The interface for accessing to the application was defined by Wireless Telephony Appliation Interface and a simple scripting language: WMLScript, based on ECMAScript.

The incompatibility of WAP 1.0 protocol stack with the Internet required the presence of a WAP gateway to translate the messages between WAP devices and WAP servers located on the Internet.

With the evolution of mobile technologies, it appeared WAP 2.0, which adds support for the standard Internet communication protocols, such as TCP/IP and HTTP, that allows wireless devices to utilize existing Internet contents without need to be adapted to mobile access. The release of WAP 2.0 involved the explosion of mobile devices as internet machines: since its appearance all the previously existent web pages could be accessed not only from usual PCs, but also from mobile devices. Before WAP 2.0, mobile internet content creators had to do the effort of adapting web pages to WAP format.

An alternative to WAP is *i-mode*[17], a propietary protocol from japanese company NTT-DoCoMo[18]. Furthermore, while WAP is mainly focused in the communication and data visualization, i-mode its a complete ecosystem which defines from the business model to the offered services, such as e-mail, short messages, multimedia content, etc.

One of the main advantages of i-mode is the way that the users pay for the content access. The net of NTT-DoCoMo allow users to be connected permanently and only pay for the amount of downloaded data.

Web Services

The World-Wide-Web Consortium defines a Web Service as "a software system designed to support interoperable machine-to-machine interaction over a network". This is, a system to support remote procedure calls from a client to a server hosted in a network.

Nowadays the Web Services protocol which is the most extended is the Simple Object Access Protocol (SOAP), whose definition in XML format can be used for exchanging structured and typed information between peers in a decentralized, distributed environment. SOAP is fundamentally a stateless, one-way message exchange paradigm, but applications can create more complex interaction patterns (e.g., request/response, request/multiple responses, etc.) by combining such one-way exchanges with features provided by an underlying protocol and/or application-specific information. SOAP is does not provide semantics of any application-specific data it conveys, as issues such as the routing of SOAP messages, reliable data transfer, firewall traversal, etc. are transparent to SOAP messages.

SOAP is supported nowadays by mobile devices. The most extended implementations are the Java ME Web Services API [19], and kSOAP project[20].

There is some criticism with SOAP. The first inconvenience is that is overdimensioned for some types of applications. The other problem is that, as explained in section 1.4.2, SOAP introduces a big overhead in message processing and bandwidth. Such overhead affects still more to mobile devices, whose processor performance is lower and network connections are slow.

Nowadays the use of an alternative to SOAP is growing and being accepted more and more by service providers. This alternative is Representational State Transfer (REST) [21], which is simpler, faster and highly scalable. REST is used to describe simple web interfaces that use XML and HTTP, without the additional abstraction of protocols based in patterns like SOAP.

REST is not a protocol itself but a set of architectural principles oriented to the concept of *Resource*: an information element which can be accessed using a global identifyer (usually an URL). To manage the resources, both clients and servers communicate through an standard interface (HTTP) and exchange representations of those resources.

The representation of a resource can be a XML file, a HTML page, an image, or any other file format, and its syntax must be understood by the applications, since the format is not defined by any protocol layer, as happens in SOAP.

The components of REST are inspired in the World Wide Web, whose success has been so huge since its simplicity and scalability:

Stateless client/server protocol All the HTTP messages contains all the information required to understand the petition.

Set of well defined operations for all the resources Usually takes the most of HTTP operations, such as POST, GET, PUT and DELETE.

Universal syntax to identify the resources Each resource is identified solely by his URI.

Hypermedia usage Usually, REST represents the resources as HTML or XML pages. Thanks to this is possible to navigate from a REST resource to others, only by following the links without necessity of registers or any additional infraestructure.

REST is not a complete protocol and architecture, but it is easy to implement and use. Some big companies provide both REST and SOAP services (e.g. Wikipedia, Google, Amazon) and usually the most of the network traffic goes through the REST services instead of SOAP ones.

There is no particular implementation of REST for mobile devices since it is not needed: only a HTTP client class for the communications and a class to decode the data used (XML, JSON, images, etc.) are needed.

Performance Considerations on Web Services

Web Services providers and clients, specially those that use SOAP, interchange lots of XML messages which must be created and parsed in both sides. This implies additional processing time and extra data overhead due to the XML messages that encapsulate service calls and responses.

Tian et al. [22] verified the overhead for a book inventory service where the client must send a request specifying a 10-bytes ISBN code for receiving 579 bytes of information about the book. The entire conversation in SOAP required 3363 bytes of extra XML data. They propose the compression of messages in order to minimize their size, increasing the CPU usage in both client and server due to the compression/decompression tasks.

Handheld Flexible Representation (HHFR) [23] proposes an optimized architecture design to deal with Web Services inconveniences in mobile environments:

- Specify the SOAP message syntax at the beginning of the conversation and later send only the message contents inside an optimized binary-structured message.
- Messages are grouped in a stream session where they share the structure and type information of SOAP body and the most of headers of messages. They can be sent only once in a session.
- A *context-store* module which keeps static data from message stream and his optimized binary format equivalence.

1.4.3 Mobile Rich Internet Applications

Nowadays, Internet and Cloud Computing users demand fast applications with rich contents. The classical HTML pages are too static and sometimes too slow for working with them in a confortable way. A set of new solutions for the so-called Rich Internet Applications (RIA) [24] is being created over the existing technologies to bring to the users more dynamic, fast and usable internet applications.

The other main advantage of RIA is that they get rid definitely with the heterogeneity of execution planforms, because they are based on well-known and extensively implemented standards.

AJAX

The Asynchronous Javascript And XML (AJAX) is not a new technology itself. While in the classic HTML pages the client requests to the server for a concrete action and the server returns a complete page

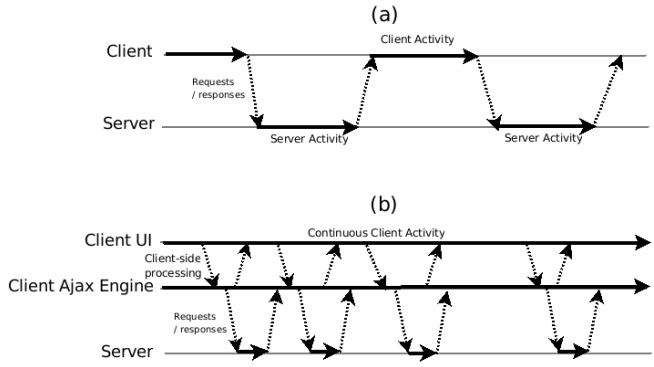

Figure 1.6: Workflow comparison between (a) classic HTTP clients and (b) AJAX clients

updated with the new modifications, AJAX combines HTML, CSS stylesheets and Javascript to perform small petitions to the server, which returns only the changes that have to be done in the client side; these changes are done dynamically by the client.

This way, the data to be transmitted is minimized and the usage of the application is faster and more confortable, since it allows to create some new and dynamic interface components that do not exist in HTML. Figure 1.6 shows how, with the AJAX approach, the client has a continuous activity and, each time it sends a petition, does not have to wait for the server response to continue working.

Flash Lite

Adobe®Flash Lite is an adaptation for mobile devices of the well known Adobe Flash. Next are listed the most remarcable parts of its architecture:

Audio and graphics engine Used for the presentation of rich multimedia content for the use. It is capable to render video, vector graphics and animations, and play sounds.

Networking Which brings internet connectivity and allow the dynamic loading of contents and the access to Web Services from

Flash Lite applications.

Persistence Used to store data permanently into user devices.

Scripting engine Flash uses ActionScript as scripting language used for programming applications that access all the other modules.

1.5 Mobile applications in the Cloud

Instead of the relative youngness of the "Cloud", most companies created many applications that take the most of it and demostrates how amazingly combines the Cloud and mobile devices. This section enumerates some of them.

Gmail for Mobile Phones [25] It was one of the first mobile applications that fits in the definition of Cloud Computing; it is a very simple Java ME application that accesses all the personal Gmail data. It does not matter where the client is or which kind of device is he using: an Internet connection is only needed to access all the user emails and address book. Since mobiles phones have limmited connection, if you loose the connectivity you can loose your session data. Thanks to Google Gears you can store in your phone the application data to be accessed offline, and sincronize it automatically each time the device its connected.

Google notebook Allows you to write notes from your PC or your mobile phone, and access them everywhere. Thanks to Google Gears, you also can use it offline.

Yahoo! Go [26] Is a mobile application that allows the user access hundreds of Internet contents. Using a widget system, Yahoo! Go facilitates the quick access to lots of Yahoo! services, like news, email, weather, and other third-parties one, such as eBay auctions, MySpace contents, or Flickr photos.

Wittycall Self-defined as *the open public phonebook*, Wittycall [27] is a search engine related to phone numbers, a new concept based on a social phonebook. The value added to the service is that

> users can sort the results of the search by the most called contacts (user habit), commented (user feedback), voted (user preference) ... as well as by date and by name. From Wittycall users will be able to perform the call, send a SMS/text message or go to an Internet address by clicking on the contact detail. A free open global phone directory made of user created phone contacts, a new open directory where phone contacts are published, promoted, rated and commented by the users. Wittycall lets users to share and recommend their favourite services located on a concrete place or country (restaurants, hotels, taxi services, professional services...) with the rest of users, and the most important thing, it lets users to perform a search for a phone number and retrieve an important extra information based on users habits, feedback, preferences and experiences.

Lots of more good examples can be shown at reference [28].

1.6 Conclusion: where is the mobile going?

This chapter has shown which alternatives are available for application designers and programmers when they want to create mobile Internet applications with special focus on web services for accessing the Cloud.

The emphasis on mobile devices is not accidental: they are the most extended communications device, they are portable, with low performance to require the access to the computing power of the cloud, but with enough performance to support current communication standards, multimedia visualization and advanced user interfaces. Mobile devices fit perfectly in the ideal Cloud thin client definition.

Despite of mobile phones market reports yearly benefits of several billions of euro, this market has still a big economic potential: the number of users are still increasing dramatically in emerging economies, and the usage of the devices for data services and applications is growing around the world [2].

The author of this chapter defends that the mobile access to the Cloud is just starting and in the future there will be created lots of

applications that will reduce the current usage of Personal Computers. The arguments to think that are the next:

- Connection quality is increasing while prices are decreasing. This makes more attractive the usage of cellulars for connecting the Internet.
- The growing performance of devices enforces to create new Rich Internet Applications for mobile phones.
- Some new concept of applications that takes the most of mobile devices particularities (such as Wittycall[27]) have not yet created. When they arrive, some users will migrate from PC to mobile.

Since the Mobile Cloud market has not yet exploded, the advertising opportunities are still latent. Future mobile market growing will accelerate a transference of advertising budgets from traditional media, such as TV or radio, to new media content distribution chanels. The technology is yet created, but like in all the media, still is needed a stronger commercial interest to settle definitively the Mobile Cloud as the TV of 21^{st} century.

Bibliography

[1] A. Weiss, "Computing in the clouds," *netWorker*, vol. 11, no. 4, pp. 16–25, December 2007.

[2] Strategy Analytics, "Understanding the mobile ecosystem," Adobe Systems, Tech. Rep., 2008.

[3] M. A. Horowitz, V. Stojanovic, B. Nikolic, D. Markovic, and R. W. Brodersen, "Methods for true power minimization," *iccad*, vol. 00, pp. 35–42, 2002.

[4] Q. Wu, P. Juang, M. Martonosi, L.-S. Peh, and D. Clark, "Formal control techniques for power-performance management," *Micro, IEEE*, vol. 25, no. 5, pp. 52–62, Sept.-Oct. 2005.

[5] *Intel Atom Processor Z5xx Series for Embedded Computing*, 2008. [Online]. Available: http://download.intel.com/design/chipsets/embedded/prodbrf/319544.pdf

[6] *AMD Geode Processor Family Product Overview*, 2007. [Online]. Available: http://www.amd.com/us-en/assets/content_type/DownloadableAssets/33358e_lx_900_productb.pdf

[7] *nVidia APX product page*. [Online]. Available: http://www.nvidia.com/object/apx_2500.html

[8] T. Camara, R. Lima, R. Guimaraes, A. Damasceno, V. Alves, P. Macedo, and G. Ramalho, "Massive mobile games porting: Meantime study case," *Brazilian Symposium on Computer Games and Digital Entertainment*. [Online].

Available: http://www.cin.ufpe.br/~sbgames/proceedings/files/Massive%20Mobile%20Porting.pdf

[9] P. Hagen, T. Robertson, M. Kan, and K. Sadler, "Emerging research methods for understanding mobile technology use," in *OZCHI '05: Proceedings of the 19th conference of the computer-human interaction special interest group (CHISIG) of Australia on Computer-human interaction.* Narrabundah, Australia, Australia: Computer-Human Interaction Special Interest Group (CHISIG) of Australia, 2005, pp. 1–10.

[10] R. Riggs, J. Huopaniemi, A. Taivalsaari, M. Patel, and A. Uotila, *Programming Wireless Devices with the Java 2 Platform, Micro Edition.* Mountain View, CA, USA: Sun Microsystems, Inc., 2003.

[11] (2000) J2me building blocks for mobile devices. [Online]. Available: http://java.sun.com/products/cldc/wp/KVMwp.pdf

[12] "Symbian." [Online]. Available: http://www.symbian.com

[13] B. Morris, *The Symbian OS Architecture Sourcebook - Design and Evolution of a Mobile Phone OS.* John Wiley and Sons Ltd., 2007.

[14] Android. [Online]. Available: http://code.google.com/android

[15] The webkit open source project. [Online]. Available: http://webkit.org/

[16] WAP 2.0 technical white paper. [Online]. Available: http://www.wapforum.org/what/WAPWhite_Paper1.pdf

[17] T. Natsuno, *The i-mode Wireless Ecosystem.* John Wiley and Sons, Ltd, 2003.

[18] NTT-DoCoMo. [Online]. Available: http://www.nttdocomo.com/

[19] J2me web services api. [Online]. Available: http://java.sun.com/products/wsa

[20] ksoap project. [Online]. Available: http://www.ksoap.org

[21] R. T. Fielding, "Architectural styles and the design of network-based software architectures," Ph.D. dissertation, 2000, chair-Richard N. Taylor.

[22] M. Tian, T. Voigt, T. Naumowicz, H. Ritter, and J. Schiller, "Performance considerations for mobile web services," *Computer Communications Journal*, vol. 27, no. 11, pp. 1097–1105, 2004.

[23] S. Oh and G. C. Fox, "Hhfr: A new architecture for mobile web services: Principles and implementations," Tech. Rep., September 2005. [Online]. Available: http://grids.ucs.indiana.edu/ptliupages/publications/HHFR_ohsangy.pdf

[24] C. O'Rourke. A look at rich internet applications. [Online]. Available: http://www.oracle.com/technology/oramag/oracle/04-jul/o44dev_trends.html

[25] Gmail for mobile phones. [Online]. Available: http://www.google.com/mobile/mail

[26] "Yahoo! go." [Online]. Available: http://mobile.yahoo.com/go

[27] Wittycall. [Online]. Available: http://www.wittycall.com

[28] 20+ mobile internet applications. [Online]. Available: http://mashable.com/2008/01/02/20-mobile-internet-applications/

List of Figures