Available online at www.sciencedirect.com

ScienceDirect

Procedia Computer Science 73 (2015) 193 – 198

Procedia Computer Science

The International Conference on Advanced Wireless, Information, and Communication Technologies (AWICT 2015)

Communication within Cloudlet using the Raspberry

Nawel Kortas, Haythem azzabi, Anis Ben Arbia, Jamel Belhadj tahar

nawelkortas@gmail.com
azzabi.haythem@gmail.com
anis.benarbia@infcom.rnu.tn
belhadjtahar.jamel@gmail.com

Abstract

Communication devices, such as Laptops, computers smartphones and personal media players, have extensively increased in popularity thanks to the rich set of cloud services that they allow users to access.

This paper focuses on setting solution of network latency and security problem for devices communication by the use of Cloudlets. It proposes a conception of a local datacenter that allows users to connect to their data from any point and through any devices by the use of the Raspberry.

Peer-review under responsibility of organizing committee of the International Conference on Advanced Wireless, Information, and Communication Technologies (AWICT 2015)

Keywords: Cloudlets; Cloud computing; Raspberry PI; datacenter; Device communication.

1. Introduction

Cloud server process and maintain all the service data or users need to contact the cloud server whenever they want to consume data or contribute data that they have generated.

The goal of this paper is to present firstly a comparison between Cloud services and cloudlet[1]. Then it deposits the solution of Cloudlets mechanism for network latency problems. Thirdly, it presents the Raspberry[2] mechanisms and their diverse use. Finally, the conclusion and future work.

* Corresponding author nawel kortas. Tel: + 216 22796995
E-mail address: nawelkortas@gmail.com

Peer-review under responsibility of organizing committee of the International Conference on Advanced Wireless, Information, and Communication Technologies (AWICT 2015)
doi:10.1016/j.procs.2015.12.012

2. Cloud services and Cloudlet

2.1. Non-Geo-locality and Geo-locality Cloud-based Services

For usual non-geo-locality cloud services, caching has been broadly employed to reduce latency and server load. Caching has been commercially deployed or proposed to improve access to distributed network file and database systems, DNS, web content and search results[3]. Deployed and planned approaches for cloud service caching are based on a multi-tier architecture in which the data are cached both on the cloud server and on intermediate proxy devices. Several of these services, including web content, are heavily cached on the user's mobile device as well. For others (e.g., web search), however, only cloud-based (server, proxy) caching schemes have been proposed.

Table 1. Difference between non-geo-locality and geo-locality Cloud based Services.

	Non-geo-locality Services	Geo-locality Services
Data Type	Typically very large in size; data of global importance.	Relatively smaller; data of only local importance.
Data Generation	Distributed users as a function of their personal interests.	Location-dependent. Input/ sensed by users/devices within a specific geographic region.
Data Consumption	Typically location-independent. Based only on user's personal interests.	Location-dependent. Consumed by users within a specific geographic region (same region where data was initially generated).
Data Caching	Location-independent. Depends only on personal user interests.	Location-dependent. The type/subset of data (sensed and) cached varies depending on location.
Request Serving	From local cache when possible. Otherwise from cloud server over long-range communication.	From local cache when possible. Otherwise from neighboring mobile devices over short-range communication.

2.2. Why Cloudlet

One solution to defeat these resource limitations is mobile cloud computing[4]. By leveraging infrastructure such as Amazon's EC2 cloud or Rackspace[4], computationally expensive tasks know how to be offloaded to the cloud. However, these clouds are usually far from the mobile user, and the high WAN latency makes this approach insufficient for real-time applications. To cope with this high latency, Satyanarayanan[5] introduced the concept of cloudlets: trusted, resource rich computers in the near vicinity of the mobile user (e.g. near or collocated with the wireless access point). Mobile users can then rapidly instantiate custom virtual machines (VMs) on the cloudlet running the required software in a thin client fashion[6].

Even if cloudlets might determine the issue of latency, there are two important drawbacks of the VM based cloudlet approach that we have not mentioned. The first one remains dependent on service providers that actually deploy such cloudlet infrastructure in LAN networks. To improve this constraint, we suggest a more dynamic cloudlet concept, where all devices in the LAN network can assist in the cloudlet, as depicted in Fig. 1. Next to the cloudlet infrastructure provided by service providers in the mobile network or by a corporation as a corporate cloudlet, all devices in the home network can share their resources and form a home network cloudlet. On the train, different users can also share resources in an ad hoc cloudlet.

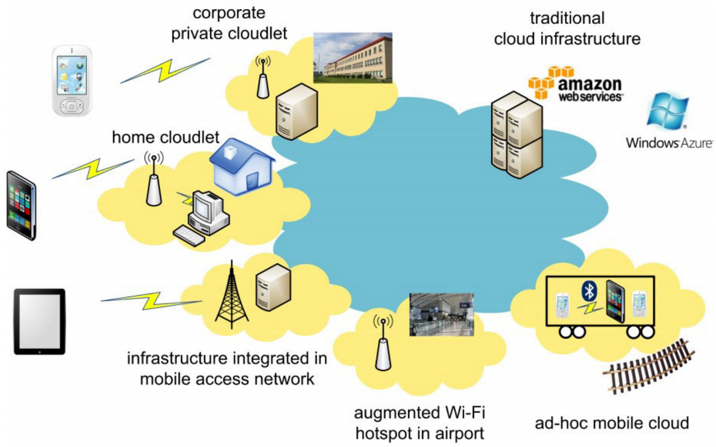

Fig. 1. The cloudlet infrastructure provided by service providers in the mobile network.

2.3. Cloudlets solution mechanism for network latency problem

Clouds are usually far from the mobile user, and the high WAN latency makes it deficient for real-time applications. To cope with this high latency, Satyanarayanan[5] introduced the concept of cloudlets: trusted, resource rich computers in the near vicinity of the mobile user (e.g. near or collocated with the wireless access point). Mobile users are able to quickly instantiate tradition virtual machines on the cloudlet operation the required software in a thin client fashion[9].

As distinct in Caceres' paper[7], cloudlets are decentralized and generally dispersed Internet infrastructure whose compute cycles and storage resources can be leveraged by closely mobile computers. A cloudlet might be a collection of multicore computers, with gigabit internal connectivity and a high bandwidth wireless LAN. A cloudlet can also be a very dominant multi-core server with Internet connectivity depending on the application scenario. Cloudlets shortly had been planned to assist mobile users, directly connected to them in terms of storage and processing[7]. Actually, to support mobile devices linked with them in storage and processing, cloudlets can be used to cache and transfer content to mobile nodes using affordable wireless technologies such as Wi-Fi and Wi-Fi repeaters[8], and/or Flashing[9]. Cloudlet also simplifies the challenge of meeting bandwidth demand of multiple users, such as HD video and high-resolution images.

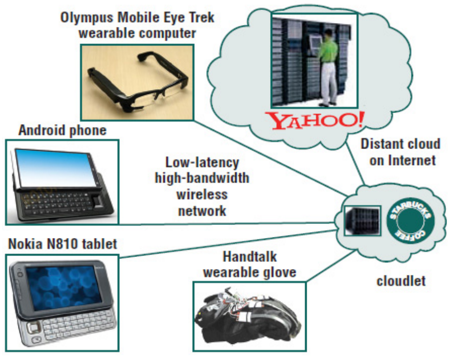

Fig. 2. The concept of cloudlet.

2.4. Comparison between Cloud and Cloudlets

With cloud computing there are a few challenges. Main challenge of Network Latency we can progress using cloudlet. Here in next Table shows the differences between Cloudlet and Cloud.

Table 2. Comparison of Cloudlet vs. Cloud.

	Cloudlet	Cloud
State	Only soft state	Hard and soft state
Management	Self-managed little to not professional attention	Professionally administered, 24*7 operator
Environment	“Datacenter in a box” at business premises	Machine room with power conditioning and coding
Network latency	Need of Low LAN latency	Need of High Internet latency
Sharing	Few users at a time	100-1000 s of users at a time
Distance	Near to the mobile users	Far to the mobile users
Ownership	Decentralized ownership by local business	Centralized ownership by Amazon, Yahoo!, etc.

3. The Raspberry PI

The Raspberry Pi[2] is a credit card sized computer with versions costing between £20 and £35. It runs a complete version of the Linux System. Its records are held on an SD card typically holding between 2 and 32 Gigabytes of data. While connected to a power provide, a USB keyboard and mouse, and attached to a TV via an HDM, it behaves like a regular laptop. Programs for it can be written in diverse languages such as Python, C and Java.

The processor at the mass of the Raspberry Pi system is a BroadcomBCM2835 system-on-chip (SoC)[10] multimedia processor. This means that the enormous majority of the system’s components, counting its central and graphics processing units beside with the audio and communications hardware, are built onto that single component hidden beneath the 256 MB memory chip at the center of the board (Fig. 3). It is not just this SoC design that builds the BCM2835 different from the processor found in your desktop or laptop, however. It also uses a different instruction set architecture (ISA), known as ARM.

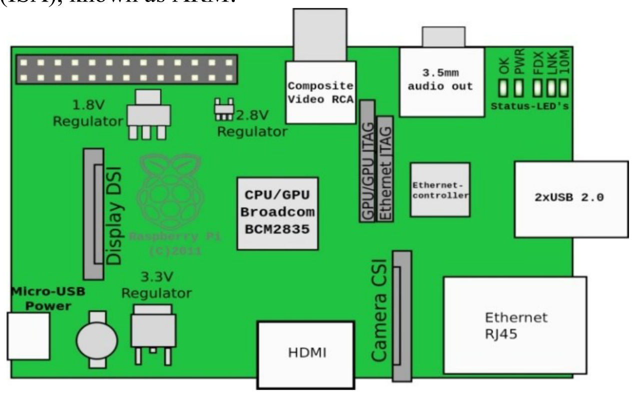

Fig. 3. The Raspberry PI.

3.1. Component of the Raspberry PI

Table 3. The Raspberry component

Feature	Specification
CPU	700MHz ARM1176-JZFS
GPU	Broadcom VideoCore IV
Memory	256MB LPDDR2-800
Video	HDMI, composite
Audio	HDMI, stereo analog
USB	2 x USB2.0 (model B)
Storage	SD card
Networking	10/100 Ethernet
Power	5V micro USB

3.2. Development of different models of Raspberry

In 2006, the premature concepts of the Raspberry Pi were based on the Atmel ATmega644 microcontroller[11]. Its schematics and PCB layout are widely available. Groundwork trustee Eben Upton assembled a group of teachers and computer enthusiasts to develop a computer to inspire children. The computer is inspired by Acorn's BBC Micro of 1981. Model A, Model B and Model B+[11] are references to the original models of the British educational BBC Micro computer, developed by Acorn Computers. The first ARM prototype version of the computer was mounted in a package the same size as a USB memory stick. It had a USB port on one end and an HDMI port on the other.

The Foundation's goal was to offer two versions, priced at US$25 and US$35. They started accepting orders for the higher priced model B on 29 February 2012, the lower cost model A on 4 February 2013, and the even lower cost (US$20) A+ on 10 November 2014[12].

3.3. Why we chose the Raspberry?

In recent years, there has indeed emerged many demo boards and modular development to less than $ 100, from different sources. Yet, beware; even if the price levels are all similar, user experience, functionality and flexibility are different at the highest point. These cards are now famous names, such as Raspberry Pi, Arduino, Beagle Board or Tower. Each has advantages and disadvantages to try to meet the challenging demands of today's designers. For if the purchase price is low, regardless of the card, this fact does not absolve spend some time in advance to ensure that the initial requirements will be gotten.

The Raspberry Pi platform is undoubtedly the best known of these platforms. For only $ 25, it has a card, an integrated development environment (IDE) Open Source GNU and a community ready to help. This good platform can be sufficient if you create an automated refrigerator door alarm to his house. Furthermore, Raspberry Pi has many versions to suit the specific set of devices of various models used for microcontrollers and the family house use. The table below presents the differences between the Raspberry and other carts.

Table 4. Difference between the Raspberry and other cards

CARD	CPU	ARCHITECTURE / ENVIRONMENT	RTOS ECONOMIC SUPPORTED
Raspberry PI	Broadcom 700 MHz Broadcom 900 MHz	Cortex-A7 / IDE	Linux (Raspbian, Debian GNU/Linux, Fedora, et Arch Linux ARM), RISC OS, FreeBSD, NetBSD, Plan 9
Arduino/Atmel	ATmega32/32x, ATmega2560	ATmega, ARM Cortex-M3 / Arduino IDE	FreeRTOS, ChibiOS/RT
BeagleBoard & BeagleBone/ TI	OMAP3/AM335x	ARM Cortex-A8 / IAR Embedded Workbenchnull	Linux, Android, Ubuntu
Cerebot /Microchip	dsPIC33f, PIC32MX	dsPIC, PIC32 (MIPS) / MPLAB IDE	FreeRTOS
ChipKit/Microchip	PIC32MX	PIC32 (MIPS) / MPLAB	FreeRTOS
Discovery/ STMicroelectronics	STM32	ARM Cortex / IAR Embedded	FreeRTOS, ChibiOS/RT
Freedom /Freescale	Kinetis K20 & KLxx	ARM Cortex-M0+/M4 / CodeWarrior IDE, IAR	MQX, MQX Lite
i.MX233-OLinuXino/ Freescale	i.MX23	ARM9	Linux, Android
LaunchPad/TI	MSP430, Piccolo, Tiva	MSP430, TMS320F282x, ARM Cortex-M4	TI RTOS, FreeRTOS
LPCXpresso /NXP	LPC1xxx	ARM Cortex-M0/M3 / Code Red	Linux
Pioneer/CypresSemi	PSoC 4	ARM Cortex-M0	

conductor		/ PSoC Creator IDE	
CY8CKIT-050/Cypress Semiconductor	PSoC5 LP	ARM Cortex-M3/ PSoC Creator IDE	FreeRTOS
Tower System /Freescale	MPC5125, Kinetis, HCS08/S08, S12, MCF51xx	e300, ARM Cortex M0+/M4, MC9S08/12, Coldfire V1 / CodeWarrior IDE, IAR Embedded Workbench	MQX, FreeRTOS
Wandboard/ Freescale	i.MX6	ARM Cortex-A9	Linux, Android, Ubuntu
Xplained, Xplained Pro / Atmel	ATmega128/256, AT32UC, ATSAM4	ATmega, XMEGA, AVR32, ARM Cortex-M0/M3/M4 / Studio 6 IDE	FreeRTOS
YRDKRL78G14/ Renesas	RL78/G14	RL78 / IAR Embedded	FreeRTOS, Android

4. Conclusion and future work

Using cloudlet, users seamlessly exploit near computers to obtain the resource profit of cloud computing without deserve WAN delays and jitter. Therefore, we need to present more focus and do research on the conception of cloudlet as a middleware for device communication within Cloud Computing.

References

1. Emmanouil Koukoumidis, Dimitrios Lymberopoulos, Karin Strauss, 'Pocket Cloudlets', March 5,(2011).
2. Martin Richards, Young Persons Guide to BCPL Programming on the Raspberry Pi, April 2 (2015).
3. Evangelos Markatos. On Caching Search Engine Query Results. Computer Communications,(2006).
4. I. Giurgiu, O. Riva, D. Juric, I. Krivulev, and G. Alonso. Calling the cloud: enabling mobile phones as interfaces to cloud applications. In Proceedings of the ACM/IFIP/USENIX 10th international conference on Middleware, Middleware'09, pages 83–102, (2009).
5. M. Satyanarayanan. Mobile computing: the next decade. In Proceedings of the 1st ACM Workshop on Mobile Cloud Computing & Services: Social Networks and Beyond, MCS '10, pages 5:1–5:6, (2010).
6. M. Satyanarayanan, P. Bahl, R. Caceres, and N. Davies. The case for vm-based cloudlets in mobile computing. Pervasive Computing, IEEE, 8(4):14 –23, (2009).
7. M. Satyanarayanan, P. Bahl, R. Caceres, and N. Davies. The case for vm-based cloudlets in mobile computing. Pervasive Computing, IEEE, 8(4):14 –23, 2009. M. Satyanarayanan, P. Bahl, R. Caceres, and N. Davies. The case for vm-based cloudlets in mobile computing. Pervasive Computing, IEEE, 8(4):14 –23, (2009).
8. CloudTrax. CloudTrax (A cloud-based network controller) network planning guide. CloudTrax, (2012).
9. M. S. Corson, R. Laroia, J. Li, V. Park, T. Richardson, and G. Tsirtsis. Toward proximity-aware internetworking. Wireless Commun., 17(6):26– 33, December, (2010).
10. A john Wiley and Sons, Ltd, Publication, 'Raspberry Pi User Guide Eben Upton and Gareth Halfacree',(2012).
11. Wong, George (24 October 2011). "Build your own prototype Raspberry Pi minicomputer". Ubergizmo, November 2, (2011).
12. Introducing Raspberry Pi Model A+". Raspberry Pi Foundation, November 10, (2014).