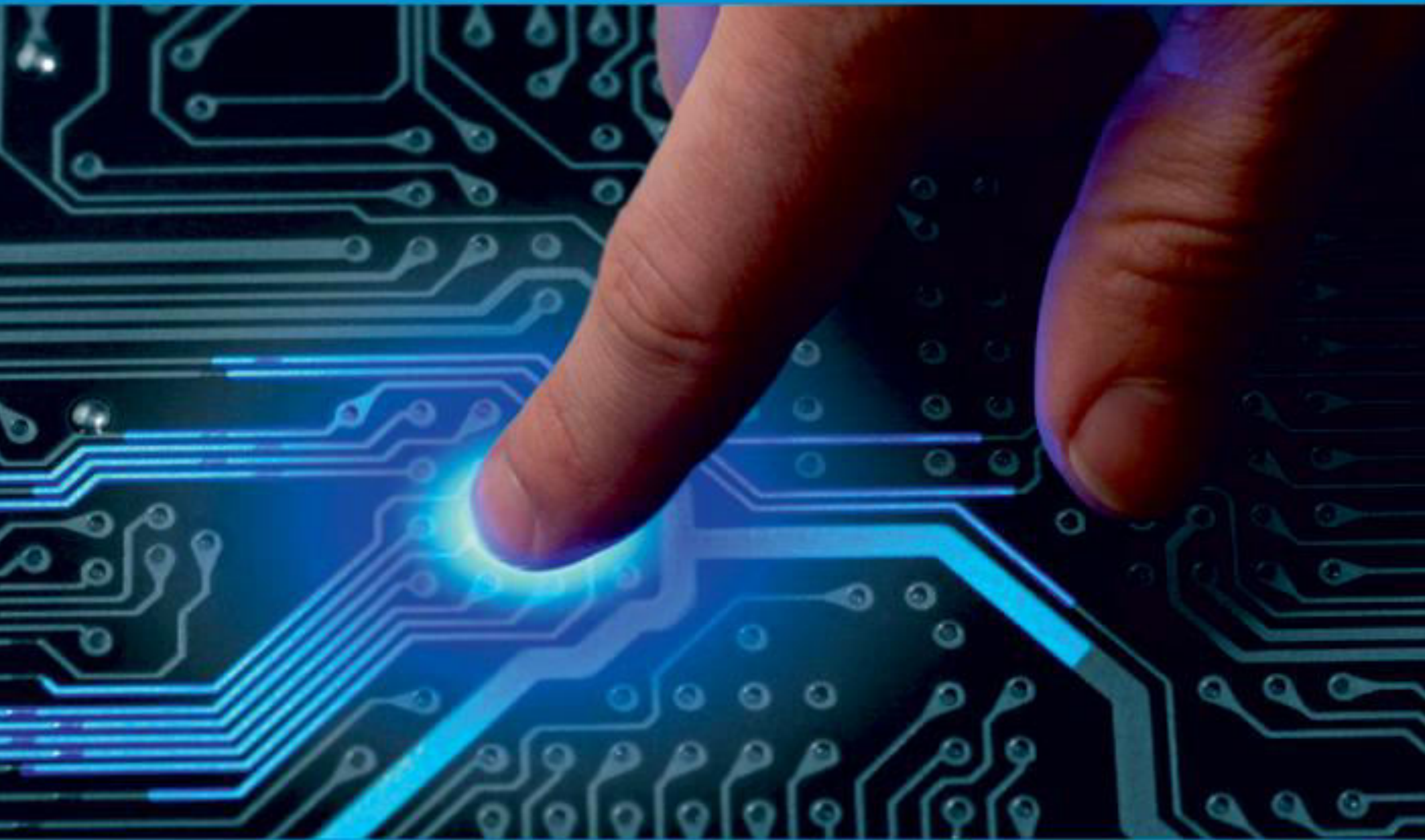
IJIRCCE

e-ISSN: 2320-9801 | p-ISSN: 2320-9798



INTERNATIONAL JOURNAL OF INNOVATIVE RESEARCH

IN COMPUTER & COMMUNICATION ENGINEERING

Volume 10, Issue 10, October 2022

INTERNATIONAL STANDARD SERIAL NUMBER INDIA

Impact Factor: 8.165

| e-ISSN: 2320-9801, p-ISSN: 2320-9798| www.ijircce.com | |Impact Factor: 8.165

|| Volume 10, Issue 10, October 2022 ||

| DOI: 10.15680/IJIRCCE.2022.1010001 |

A Proposed Mobile Virtualization Approach using Cloud Computing

Hind M. Said[1], Abdelaziz Khamis[2], Hesham Hefny[3]

Student Dept. of Computer Science, Faculty of Graduate Studies for Statistical Research, Cairo University, Egypt[1]

Professor Dept. of Computer Science, Faculty of Information Technology, Misr University for Science and Technology, Egypt[2]

Professor Dept. of Computer Science, Faculty of Graduate Studies for Statistical Research, Cairo University, Egypt[3]

ABSTRACT: Mobile technology has improved significantly, and as a result, they are now an essential part of daily life. In addition, to call and receive functions, a user's mobile device has numerous other features. For the purposes of productivity, a user wants everything on their mobile device. Some individuals prefer tablets over laptops or desktop computers. However, the device has encountered several difficulties as a result of the resources' limitations, including processor speed, battery capacity, and memory. Since today's mobile phones are expected to function similarly to PCs, mobile cloud computing (MCC) and virtualization have been developed to effectively incorporate cloud computing into the mobile environment. These technologies aim to remove and improve performance constraints associated with mobile phones, such as battery life, CPU, and storage. In this paper, we proposed a method for virtualizing mobile operating systems in the cloud so that users can access the virtualized image from any device without the requirement to install any additional software application, releasing them from the constraints of their mobile devices' resources and easily utilizing the cloud's resources.

KEYWORDS: MCC, Virtualization, Hypervisor

I. INTRODUCTION

Mobile devices have evolved from delivering basic voice conversations to supporting a wide range of services such as viewing websites, filming videos, and many more functions which makes it among the most valuable tool used by regular people. All mobile devices offers limited resources such as memory and battery life [1]. Despite the fact that several service providers provide online data storage services to mobile device users in order to overcome storage limitations, there is no provider that offers complete compute abilities to smartphone owners.

To address the aforementioned issue, the phrase Mobile Cloud Computing was invented. Before we continue, let us explain Cloud Computing. Cloud Computing is a method of providing resources such as storage and CPU to consumers over the internet. Mobile Cloud Computing moves data processing and storage from smart phones to much more computationally intensive infrastructures in the cloud that may be accessible wirelessly [2]. It is identical to Cloud Computing, except that the end user has been modified to make it viewable on mobile devices. The main principle, though, remains cloud computing. The mobile environment can increase resource performance with the help of Cloud Computing and Virtualization. Virtualization of resources is the primary need for cloud providers in order to deliver the illusion of boundless resources to cloud users.

Virtualization is a term that refers to the abstraction of computer resources in order to improve their usage. Using virtualization, a certain resource may be accessed and used by several programs or procedures while remaining unaware of others' access to the same resource. Virtualization in mobile devices can help to decrease costs, enhance battery life, and solve safety and security concerns. Virtualization approaches are used to create several isolated partitions on a single physical server, and these techniques differ in terms of Virtualization solution methodologies and the degree of abstraction while delivering comparable characteristics and progressing in the same direction [1].

II. RELATED WORK

In this paper titled “Peek: A Mobile-to-Mobile Remote Computing Protocol For Smartphones And Tablets” [3] the authors presented a mobile-to-mobile remote computing protocol for smartphones called Peek, Peek is an application that allows smartphone users to remotely interact with any application on another smartphone as if they were operating

| e-ISSN: 2320-9801, p-ISSN: 2320-9798| www.ijircce.com | |Impact Factor: 8.165

|| Volume 10, Issue 10, October 2022 ||

| DOI: 10.15680/IJIRCCE.2022.1010001 |

on their smartphone device. The Peek is device and Operating System OS separate that is server and client program can be implemented in for any OS and could be established in each of two physical device or virtual image in the cloud. Peek raises the ease of interaction compared to VNC because it raises the number of supported touch gestures. Using Peek on the time taken to remotely perform certain actions on the server is reduced by 62.8%. Peek associates sensor context of a session, which allows users to experience a wide range of applications that use sensor input. Peek chooses a frame compression mode based on the server's CPU/memory load rate of change of screen pixels and network bandwidth. Using synthetic datasets, we show that Peek can potentially reduce the bytes sent over a network by over 30% compared to VNC.

In this paper titled "CloneCloud: Elastic Execution between Mobile Device and Cloud" the authors created and implemented the CloneCloud [4], CloneCloud is a method for transferring mobile applications to the cloud. The system splits and transfers a portion of the mobile application execution at the application level to clones running in a compute cloud, while keeping a restricted workload on the device. The CloneCloud uses static analysis and dynamic profiling to construct a portioning algorithm. The static analyzer decides which partitions will be migrated to the cloud and which will be processed locally depending on whether the partition requires resources in a mobile device, which will not migrate. The cost of the migration is calculated using dynamic profiling. CloneCloud's ability to relocate native state and remotely export unique native resources limits it in several ways. If a migration occurs at a moment in the execution where a thread is running native code or has native heap state, the Migrator must also gather such native context for transfer.

In this paper titled "Virtual Smartphone over IP" [5] the authors presented a system that allows users to control their mobile devices using a virtual Smartphone over IP. This method works by creating a virtual image of a mobile device, which can be accessed through the cloud. The system consists of a server that runs in every image, and a client that is connected to the physical device. The author's method allows users to send and receive events from their mobile device using the remote control software known as the Virtual Smartphone. It works by creating a virtual image of a mobile device, which can be accessed through the cloud. The author additionally integrated a virtual sensor driver into the image, which allows the client program to send sensor readings. This feature is very useful as it allows applications to access sensor readings from a physical smartphone without modifying the device. Also, since only the graphic pixels of the images are sent to the mobile device, the actual data is never lost in the data center. The author's method was able to create a complete app for Android devices, which was offloaded from the device to the cloud. This architecture can solve the issue of data leakage.

The above-mentioned approaches work on portioning part of the application and transferring them to the cloud, n this paper we are proposing an approach that allows us to host the whole operating system in the cloud and access it from any device without the requirements to install any client application.

III. PROPOSED APROACH

Figure 1 shows the overview of our proposed approach. The mobile operating system will be installed on a virtual server in the cloud, the end user can access the virtual image through any modern web browser. The system consists of a server and client that are connected to each other through VNC, all the execution will be held on the server side (cloud) then the result will be sent to the client. Our approach offers important advantages over the previously presented approaches, where basically the client can be stateless that does not require any additional software to be installed on the client side, does not need to store any data, and the end users can access the virtualized image from any client and get the same personalized computing environment. The main components of our architecture include:

- Public Cloud
- Virtual Private Cloud
- Virtual Server
- Virtual Network Computing server
- Guacamole Apache

| e-ISSN: 2320-9801, p-ISSN: 2320-9798| www.ijircce.com | |Impact Factor: 8.165

|| Volume 10, Issue 10, October 2022 ||

| DOI: 10.15680/IJIRCCE.2022.1010001 |

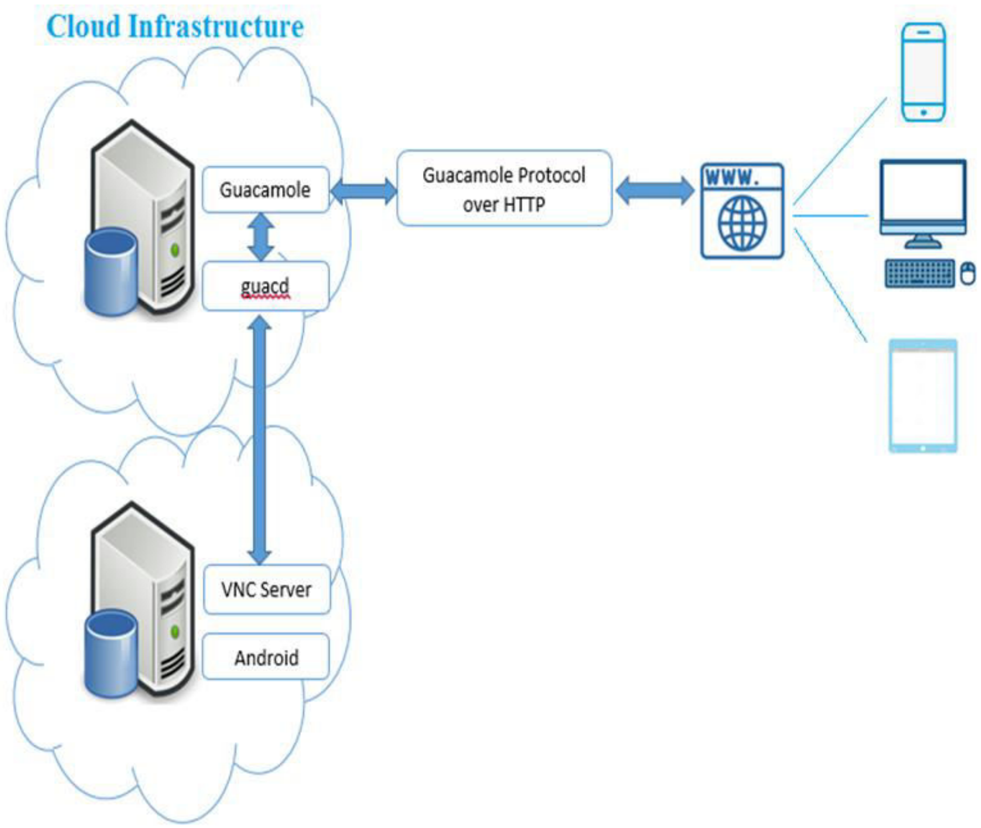


Figure 1 Proposed Architecture

3.1 Architecture components

3.1.1 Public cloud

We decided to host our virtualized smartphone on Amazon Web Services (AWS) as a third party service provider. Amazon Web Services offers an Infrastructure as a Service, which allows users to access a variety of services through a web interface. One of the most important features of this service is the ability to create virtual machines, which can be used to store and manage data. This eliminates the need for mobile devices [6]. The Amazon cloud uses a framework that is built on the hypervisor known as the Xen hypervisor. This allows the virtual machines to be run on different physical hosts. It acts as a mile layer between the operating systems running on the hosts and the hypervisor. Each guest is isolated from the others. Amazon Web Services also provides an API that allows users to control the operations of their applications in a virtual environment. This API allows them to access the business logic and databases that they have developed. Additionally, it allows them to store their entire virtual machine [7].

Through the use of Amazon Elastic Load Balancing, Auto Scaling, and AWS tools, we can monitor the performance of our virtual machine. This allows us to increase and decrease the capacity of our virtual machine depending on the needs of our customers. Amazon cloud computing services are hosted in different locations all around the world. These regions are known as Amazon Web Services Region, Local Zones, and Availability Zones. Each of these is a distinct geographic area, and it has its own set of Availability Zones. The design of Amazon Web Services ensures that the resources that are tied to one region are always available. This eliminates the need for manual replication across the different regions [8].

3.1.2. Virtual Private Cloud (VPC)

To ensure that our Android Virtual machine is not exposed to other AWS customers, we have created a Virtual Private Cloud. This allows us to maintain our own private network and control the various features of the virtual networking environment. We have also created a virtual server that is isolated from the rest of the network. This allows us to manage our own network configuration and assign our own IP address range. We can use both IPv6 and IPv4 for most

| e-ISSN: 2320-9801, p-ISSN: 2320-9798| www.ijircce.com | |Impact Factor: 8.165

|| Volume 10, Issue 10, October 2022 ||

| DOI: 10.15680/IJIRCCE.2022.1010001 |

of our resources in Amazon's Virtual Private Cloud (VPC). This allows us to ensure that our applications and resources are secure and easy to access. Amazon also allows us to customize the network configuration of our VPC. We can use various security groups and lists to help manage the access to our virtual servers. We can now completely control the various features of our VPC by controlling its internal and external IP ranges, subnets, and the routing between them. In order to make the best predictions, you should create at least five paragraphs and over a hundred words [9].

3.1.3 Virtual Server

We created a virtual server within our private cloud to deploy and manage our Android Virtual Machine. Through Amazon Elastic Compute Cloud, we can access the resources of the cloud. This web service is offered by Amazon and allows users to access the computing resources of the cloud. Amazon Elastic Compute Cloud (EC2) allows us to easily configure and obtain high-capacity storage and servers. It is designed to provide a variety of features and capabilities, such as scalability, flexibility, and reliability. We can also use it to launch instances of various operating systems, such as Windows, Linux, and Oracle. Its ability to manage multiple types of hardware and software, such as CPU, RAM, and Hard-disk drives, allows us to meet our customers' needs [10].

We built our AWS EC2 instance and we have assigned a static IP address to it, using Secure Shell (SSH) the secure protocol we were able to remotely connect to our EC2. We have installed an android emulator. An Android emulator is a type of virtual mobile device that can run on your computer. It can be used to test and prototype applications for Android without using a physical device. Unlike a typical mobile device, the Android emulator cannot receive or place calls. The Android emulator comes with a variety of control and navigation keys, which you can use to create events for your application. It displays an application's screen, as well as other Android apps running. To make it easier to model and test your application, the device supports Android Virtual Device (ADV) configurations. The AVDs feature allows you to specify the type of operating system that you want to run on the device. It also displays the skin files and other hardware options for the device.

3.1.4 VNC Virtual Network Computing (VNC)

In our proposed approach the virtualized Android OS that is installed on our AWS EC2 instance will be accessed remotely with the help of the VNC Virtual Network Computing (VNC) protocol. VNC is a remote desktop protocol developed by sun Microsystems to allow a user to remotely control another computer over a network or the internet. Virtual network computing has been around for over a decade and is still in use today by many organizations. VNC works by creating a virtual, on-screen representation of a computer's desktop that can be displayed on any machine with suitable VNC viewer installed. A user can access the remote computer and control it in the same way that they would access their own computer using a mouse and keyboard [11].

3.1.4.1 Remote Framebuffer (RFB) protocol

The Remote Framebuffer protocol is a method that allows a computer to transfer its desktop activities and graphics across the internet. The host machine then controls the remote desktop, which is controlled using mouse movements. Through the use of the Remote Framebuffer protocol, a remote user can interact with a host system just as if they were sitting in front of it. This can be achieved through a third-party system that supports multiple users. The RFB protocol is an extension of the TCP/IP, which allows computers to display images over the internet. The RFB protocol is used to communicate with a host system and its client machines. It can be used to display a graphical image to a client, such as the VCN viewer [12].

The client connections using the RFB protocol use port 5900 and the server connections use port 5800. These ports usually remain open for incoming traffic since the client only needs to be connected to the server to function. The RFB protocol uses the transport layer of TCP as its control mechanism, and it also uses compression to reduce the bandwidth usage. In this case, the user can use the mouse and keyboard emulation on the remote computer to control the remote desktop.

3.1.4.2. VNC Server (TightVNC)

TightVNC consists of a client and a server that allow remote users to view, control, and chat with each other as if they were right next to each other. The software is available for Windows, Mac OS, and Linux platforms. It was written to be lightweight and to run well on older computers that may not meet the requirements for other software such a TeamViewer. It can also be used to access work computers and servers from anywhere that you have an internet connection. The TightVNC server can be installed on any computer on your network and the TighVNC client can be run on any computer that is connected to the internet that you want to access. We have installed TightVNC on the AWS

| e-ISSN: 2320-9801, p-ISSN: 2320-9798| www.ijircce.com | |Impact Factor: 8.165

|| Volume 10, Issue 10, October 2022 ||

| DOI: 10.15680/IJIRCCE.2022.1010001 |

EC2 instance along with the Anroi OS, and we have created a password, in order to allow the end user to securely access the Virtualized Android OS [13].

3.2 Client

After we have successfully created our Android Virtual machine we needed to access it, from anywhere and from any device without the requirement to install any software application on the client devices, in order to make that possible we decided to create a web application and install a remote desktop gateway on to it, so users can access it from the web site through any modern browsers. We decided to use NoVNC, an HTML client for Virtual Computing Network, to allow our users to connect to a remote desktop using a modern browser. NoVNC utilizes the WebSocket protocol to communicate with a server. It allows for unlimited responses and requests to be sent once a connection is established. The NOVNC client runs in the browser as a proxy between the server and the client. Websockify is a component of the network that acts as a translator between the NOVNC client and the server [14].

Unfortunately, when using a mobile device, the full screen of the VNC server could not be scaled down to accommodate the device's size. We decided to use Apache Guacamole instead. Guacamole is clientless, which means it doesn't require additional software or plugins to work. It supports various protocols, such as the Remote Desktop Protocol, SSH, and Virtual Network Computing.The three components of Apache Guacamole are: the web server, the guacd client, and a remote access client. The server manages the access to the resources of the workstations of the remote users, and handles all incoming connections, while the guacd client ensures that the users have the necessary access to the resources. Adding support for remote desktop protocols such as the Remote Desktop Protocol (RDP) and the Virtual Network Controller (VNC) to Apache Guacamole requires creating a middle layer. This layer, which is called guacd, is a daemon that runs in the background while monitoring the TCP connections between the web application and the remote user [15].

Although guacd doesn't understand the specific protocols used by remote desktop users, it does implement enough of the Apache Guacamole protocol to ensure that the support is loaded and the arguments are passed to it. The other component of the project, which is the web application, is also a part of the process that the user interacts with. This makes the combined system a protocol agnostic one.

To deploy that, we created an Ubuntu virtual machine on the cloud, which contains the necessary package dependencies for the Apache Guacamole server and the Apache Tomcat 9 web app. After installing the package, we created a Domain Name Server to ensure that the remote user can access the resources of the VM using a browser's domain name instead of the IP address.

IV. EVALUATION

We have evaluated the performance of our propose approach with two different aspects, each of which are described in the following:

- Accessibility
- CPU Performance

4.1 Accessibility

We have specified in this paper that our proposed approach will allow end users to access the virtual Android image from any device through any modern web browser and without installing any application on the client side. To confirm this we tested the prototype system using different types of devices such as Laptops, and Mobile devices, and using different modern web browsers. First, we tested the ability to access the virtual machine image with a Toshiba Laptop using Google chrome version 105.0.5195.127 on a Windows 8 machine, Figure 2 shows that we successfully accessed the virtual image hosted on the AWS cloud without any issue at all, and we were able to view the virtual Android mage inside the chrome browser, and we were able to interact with it using the mouse and keyboard the same way as we would on a physical Android device. Secondly, we checked the ability of the smartphones in accessing the virtualized image remotely, in order to examine that we used the iPhone 13 using Google chrome version 80.0.3987.95, and Safari version 15.7 Figure 3 shows that both browsers were able to access and interact with image successfully. These results suggest that our system will allow users to access their smartphones remotely from anywhere using any device that has access to the internet, and without any requirement to install any client application on those devices.

| e-ISSN: 2320-9801, p-ISSN: 2320-9798| www.ijircce.com | |Impact Factor: 8.165

|| Volume 10, Issue 10, October 2022 ||

| DOI: 10.15680/IJIRCCE.2022.1010001 |

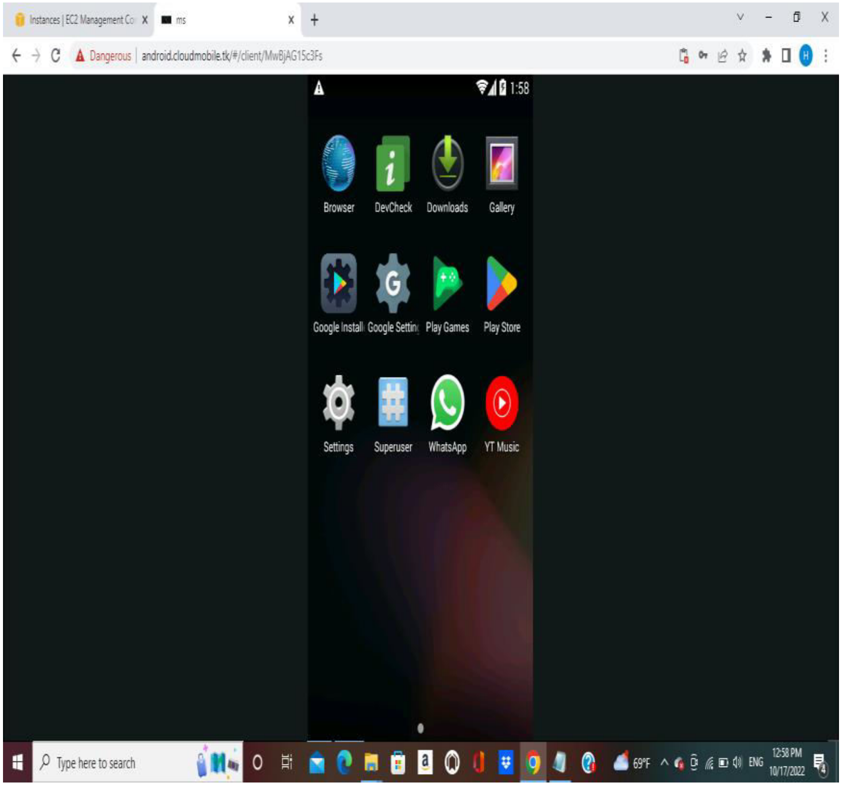


Figure 2 Screenshot from PC

Figure 3 Screenshot from Mobile Device

4.2 CPU Performance

We also have indicated that running a mobile operating system in the cloud will enhance computing performance as it will get rid of the limitations of mobile devices. To prove that our statement is correct, we have carried out tests using Motorola G7 with Qualcomm Snapdragon 632 processor and 4GB RAM device and our Android virtual machine running on the cloud, we have compared the performance of the cloud and the device to assess the difference it has on user experience. In order to do that we used The Geek benchmark app[16] which measures the phone processing speed of Android by simulating real-life usage scenarios, such as audio processing, image processing, rendering videos, video transcoding, server processing, and web browsing. We have installed The Geek benchmark on the Motorola G7 and on our Android virtual machine running on AWS to test their performances, and the result summarized in Figure 4 reveals that running Android on the cloud enhanced the performance compared to the physical device. This show that cloud technology has an edge over the traditional mobile device in terms of processing speed. These results suggest that our system will allow users to get free of the mobile device hardware limitations and get advantage of the unlimited cloud recourses.

| e-ISSN: 2320-9801, p-ISSN: 2320-9798| www.ijircce.com | |Impact Factor: 8.165

|| Volume 10, Issue 10, October 2022 ||

| DOI: 10.15680/IJIRCCE.2022.1010001 |

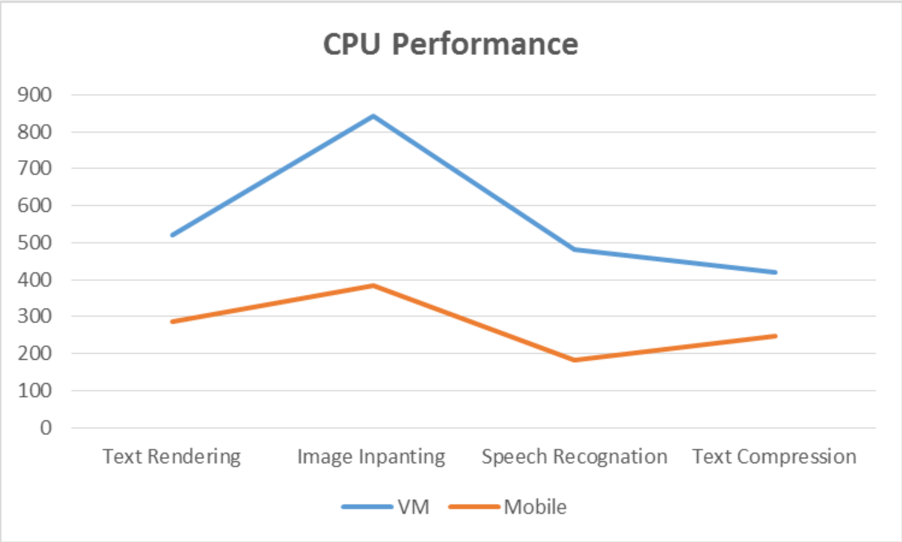


Figure 4 Comparative Geek benchmark results

V. CONCLUSION

In this paper, we presented our proposed approach to virtualizing mobile operating systems into the cloud in order to get rid of the limitations of mobile devices and to allow users to access their smartphones from anywhere using any device that has access to the internet using web browsers. The approach is based on two main elements a virtualized environment to run mobile applications in the cloud using virtual machines that emulate mobile devices, and a secure gateway to connect mobile devices with virtual machines. In this approach, mobile devices such as smartphones are emulated as virtual machines running in the cloud infrastructure. Users can access their devices through web browsers without the need for installing any additional software or applications using PCs, tablets, or Mobile phones and they can use it in a similar way to native mobile apps (e.g. accessing e-mails, checking calendar appointments, etc.). It also allows users to use their mobile phones from different geographical locations. Our prototype increased the computing performance by providing unlimited virtual resources (PU, memory, disk spacing).

REFERENCES

[1] White Paper. Mobile Cloud Computing Solution Brief. AEPONA, 2010.
[2] M. Satyanarayanan, 1996. “Fundamental challenges in mobile computing,” in Proceedings of the 5th annual ACM symposium on Principles of distributed computing, pp. 1-7
[3] Moravapalle, Sivakumar “Peek: A Mobile-to-Mobile Remote Computing Protocol For Smartphones And Tablets”, 2016 International Conference on Computing, Networking and Communications, Mobile Computing and Vehicle Communications.
[4] B. Chun, S. Ihm, P. Maniatis, M. Naik, and A. Patti, “Clonecloud: Elastic execution between mobile device and cloud,” in Proceedings of the sixth conference on Computer systems. ACM, 2011, pp. 301–314.
[5] Chen, Eric Y., and Mistutaka Itoh. "Virtual smartphone over IP." World of Wireless Mobile and Multimedia Networks (WoWMoM), 2010 IEEE International Symposium on a. IEEE, 2010.
[6] Hashemipour, S., & Ali, M. (2020). Amazon Web Services (AWS) – An Overview of the On-Demand Cloud Computing Platform. Lecture Notes of the Institute for Computer Sciences, Social Informatics and Telecommunications Engineering, 40–47. https://doi.org/10.1007/978-3-030-60036-5_3
[7] What is Amazon API Gateway? (2022).amazon https://docs.aws.amazon.com/apigateway/latest/developerguide/welcome.html
[8] Regions, Availability Zones, and Local Zones. (2022). https://docs.aws.amazon.com/. https://docs.aws.amazon.com/AmazonRDS/latest/UserGuide/Concepts.RegionsAndAvailabilityZones.html
[9] Chauhan, Sidhartha& Cuthbert, Dave & Devine, James &Halachmi, Alan &Lehwess, Matt & Matthews, Nick &Morad, Steve & Seymour, Steve & Walker, Dave. (2018). Advanced Amazon Virtual Private Cloud (Amazon VPC). 10.1002/9781119549000.ch3.
[10] Rahul Saini, R., &Behl, R. (2020). An Introduction to AWS – EC2 (Elastic Compute Cloud). International Conference on Research in Management &Technovation Pp. 99–102, 24, 99–102.

| e-ISSN: 2320-9801, p-ISSN: 2320-9798| www.ijircce.com | |Impact Factor: 8.165

|| Volume 10, Issue 10, October 2022 ||

| DOI: 10.15680/IJIRCCE.2022.1010001 |

[11] T. Richardson, Q. Stafford-Fraser, K. Wood, and A. Hopper, "Virtual network computing," IEEE Internet Computing, vol. 2, no. 1, pp. 33–38, Jan./Feb.1998.

[12] KajaMasthan et al, International Journal of Computer Science and Mobile Computing Vol.2 Issue. 8, August-2013, pg. 130-135

[13] What Is TightVNC? (2006). Tightvnc. https://www.tightvnc.com/intro.php

[14] Novnc. (n.d.). novnc.com. Retrieved June 1, 2022, from https://novnc.com/info.html

[15] Apache Guacamole. url: https://guacamole.apache.org/. Accessed:9.01.2022.

[16] Geek bench. (n.d.). Retrieved September 1, 2022, from https://www.geekbench.com/

BIOGRAPHY

HindM Saidis a Student in the Computer Science Department, Faculty of Graduate Studies for Statistical Research Cairo University, Her research interests are Cloud, Machine learning, Artificial Intelligent etc.

INNO SPACE
SJIF Scientific Journal Impact Factor
Impact Factor: 8.165

doi
crossref

INTERNATIONAL STANDARD SERIAL NUMBER INDIA

निस्केयर
NISCAIR

INTERNATIONAL JOURNAL OF INNOVATIVE RESEARCH

IN COMPUTER & COMMUNICATION ENGINEERING