IJARCCE

ISSN (Online) 2278-1021
ISSN (Print) 2319 5940

International Journal of Advanced Research in Computer and Communication Engineering
ISO 3297:2007 Certified
Vol. 6, Issue 9, September 2017

Performance Improvement of Smartphone Application using Lightweight Container Technology in Mobile Cloud

Nitish A. Shinde[1], Rekha A. Kulkarni[2]

P. G. Student, Department of Computer Engineering, Pune Institute of Computer Technology, Pune, India [1]

Assistant Professor, Department of Computer Engineering, Pune Institute of Computer Technology, Pune, India [2]

Abstract: Smartphones have become a part of our daily life. Smartphone users expect PC like functionalities over their phones. Making such vision is possible by offloading computationally intensive tasks from resource constrained smartphone to powerful cloud infrastructure. Much architecture like Cloudlet, Virtual Smartphone over IP (VSIP), CloneCloud, etc. are present in the literature, which uses the cloud based Virtual Machines (VMs) for computation offloading. However, such approaches require more time during startup and have high virtualization overhead, which results in increase in execution time of application. Also, VM based deployment requires more amount of disk space at cloud side. Container is latest, OS-level virtualization technique used in cloud environments for application deployments. Containers give better performance in terms of startup speed and size as compared to VMs. In the proposed system, we have used container based lightweight Docker technology for computation offloading over cloud. Our evaluation shows that, Docker based solution takes less amount of time during startup which in turn reduces response time to user. Also Docker based solution requires less amount of disk space for application deployment as compared to VM based approach, which in turn saves storage resources over cloud.

Keywords: Cloud computing, Container, Computation offloading, Virtualization, Smartphones, Docker.

I. INTRODUCTION

Smartphones have become a part of our daily life. It is gaining much popularity due to the support for wide range of applications like speech recognition, online games, image processing, face and object recognition, etc. These are complex kind of applications which requires high end configuration in smartphone to execute. But the way smartphones are evolving are not up to the mark to match these high end requirements. Therefore, smartphones are still unsuitable for such kind of applications due to constraints like low processing power, limited battery life and limited memory. In order resolve this problem, Mobile Cloud Computing (MCC) area comes into picture. It is a combination of Mobile Computing, Cloud Computing and Wireless Networks. It provides rich set of cloud resources to smartphone (mobile) devices to overcome its limitations. Computation offloading is practical software level approach to overcome computational limitations of mobile devices. It is the process of migrating resource-intensive computation of application from mobile device to the powerful cloud environment. Computation is done at cloud side and results are sent back to smartphone device. This will help in increasing speed of application execution and reducing battery consumption of smartphone device.

Many efforts are done in the literature to improve performance at smartphone (mobile) side by offloading computational codes to the cloud. Existing computation offloading systems use VM based approach for offloading computations over cloud server nodes. In this approach, virtual machines are used to execute offloaded components in the cloud server. It is found that, VM based offloading mechanism has overheads involved in terms deployment and management of VM. It involves overhead of VM management in its different states from creation to deletion, application deployment in VM, etc.; which requires additional computing resources and increases the execution cost of the application [6]. The main drawback of using VMs over cloud is, it requires more time during startup and it has high virtualization overhead. Responsiveness is important in mobile environment. Smartphone user expects, system should behave interactively, but VM based approach is the bottleneck for it.

Container is latest OS-level virtualization technique used in cloud environments for faster execution and lightweight deployment of applications. But still containers are not considered in mobile cloud domain. So the main aim of this dissertation work is to design computation offloading system based on container instead of VMs. In this paper, we have presented Docker based container solution for computation offloading. Section II describes related work done in this area. Section III describes design of our system. Section IV discusses the results. Section V discusses the future work and section VI concludes the paper.

IJARCCE

ISSN (Online) 2278-1021
ISSN (Print) 2319 5940

International Journal of Advanced Research in Computer and Communication Engineering
ISO 3297:2007 Certified
Vol. 6, Issue 9, September 2017

II. RELATED WORK

Various application models are proposed in order to improve performance of smartphone application using computation offloading. Some emphasized the importance of intelligent access scheme; some discussed generic issues of mobile cloud, etc. In our survey, we studied the basic models of Mobile Cloud Computing (MCC) that are available and studied the impact of use of the VM on execution time of application.

A. CloneCloud
CloneCloud, proposed by Chun, Ihm, Maniatis, Naik, and Patti is based on partial offloading of application execution. For offloading it uses nearby infrastructure. It does not require programmers support for application partitioning for offloaded execution. It offloads part of application execution at fine granularity level (thread level). For partitioning, it makes the use of static analysis and dynamic profiling. Main goals of CloneCloud are to decide what to run where at runtime and make programmer free from application partitioning overhead [1].

B. Virtual Smartphone over IP (VSIP)
Virtual Smartphone over IP (VSIP), proposed by Chen and Itoh is based on providing cloud based smartphone image service to end user. In this system, user can able to use smartphone image present in the cloud over IP network. User can make use of such cloud image to run untrusted applications and keep smartphone free from vulnerabilities. This system helps in overcoming resource constraints of smartphone devices. It requires more bandwidth as we have to maintain continuous connection between end user and cloud and data transfer is in the form of client program events and graphical screen updates [2].

C. Cuckoo
Cuckoo, proposed by Kemp, Palmer, Kielmann, and Bal is based on partial offloading of the application execution. For offloading it uses nearby infrastructure. In this model, computation offloading framework for Android OS is implemented which helps developer in developing optimized applications. For offloading, it makes the use of underlying Android IPC mechanism. Computationally intensive tasks (services) are offloaded to server and light weight activities are executed at client side [3].

D. Mobile Capabilities Augmentation using Cloud Computing (MCACC)
Mobile Capabilities Augmentation using Cloud Computing (MCACC), proposed by Elgendy, Shawish, and Moussa. Similar to Cuckoo this framework is based on Android IPC mechanism, which divides applications into lightweight activities and complex services. Services either run locally or remotely based on a dynamic offloading decision model. In low bandwidth scenario, it makes the use of compression technique to reduce bandwidth utilization [4].

E. Cloudlet
Cloudlet, proposed by Satyanarayanan, Bahl, Caceres, and Davies is based on augmented execution of mobile application. For computation offloading it uses nearby resource-rich computer connected over Wi-Fi network. Mobile device acts as thin client with resource intensive applications running on nearby system known as Cloudlet. The system is based on the concept Virtual Machine (VM) migration, where VM state is saved from mobile device and offloaded to Cloudlet and resumed over there for further execution. Finally, results are merged back in the mobile device [5].

We have compared these models based on the technologies used for implementation as shown in Table 1.

Model	Objectives	Technology Used
CloneCloud [1]	Energy saving and reduction in execution time	Virtualization, DalvikVM
VSIP [2]	Energy saving and reduction in execution time	Virtualization, VMware ESXi
Cuckoo [3]	Energy saving and reduction in execution time	Service, Android
MCACC [4]	Energy saving, reduction in execution time and minimize bandwidth requirement	Service, Java VM
Cloudlet [5]	Reduction in execution time	Virtualization, VirtualBox

Table 1. Technology and tools used existing frameworks

IJARCCE

ISSN (Online) 2278-1021
ISSN (Print) 2319 5940

International Journal of Advanced Research in Computer and Communication Engineering
ISO 3297:2007 Certified
Vol. 6, Issue 9, September 2017

Most of the existing frameworks use virtualization technology for the execution of offloaded parts. We have studied impact of using Virtual Machines (VMs) on application execution in cloud environment. In mobile cloud computing, computationally intensive tasks are offloaded to cloud environment for faster execution. Practically almost all existing offloading systems use Virtual Machines (VMs) for such computation offloading. According to Shiraz, Abolfazli, Sanaei, and Gani, using such VM based approach requires additional computing resources utilization in the deployment and management of the VM on smartphone [6]. It is found that

- Creating VM instances on computing host requires additional resources and it increases application execution time.
- Application allocation to VM increases with the increase in number of VMs and applications.
- Average execution time for each application increases with increase in number of VMs and applications in none shared VMs.
- Average execution time of a single application increases with increase in number of VMs and applications, even VMs are shared.

III. PROPOSED SYSTEM

Our system follows client-server based architecture, with client running mobile device and server in the cloud system. System is composed of 3 components: mobile client, front-end server and the Docker subsystem. Mobile client is responsible for interacting with the user. Mobile application is developed to support the both mobile and cloud execution mode. Based on the modes selected in mobile app, application is executed either locally or remotely on the cloud. The front-end server is responsible for handling the requests made by client and forwards these requests to appropriate Docker container through Docker engine. Docker subsystem contains the application images per user. When request is made to execute application remotely over the cloud container, appropriate image is selected from the available images using image id and request is forwarded to it for further execution. Once the computation is done results are sent back to client through front end server. Figure 1 shows high level architecture view our proposed system.

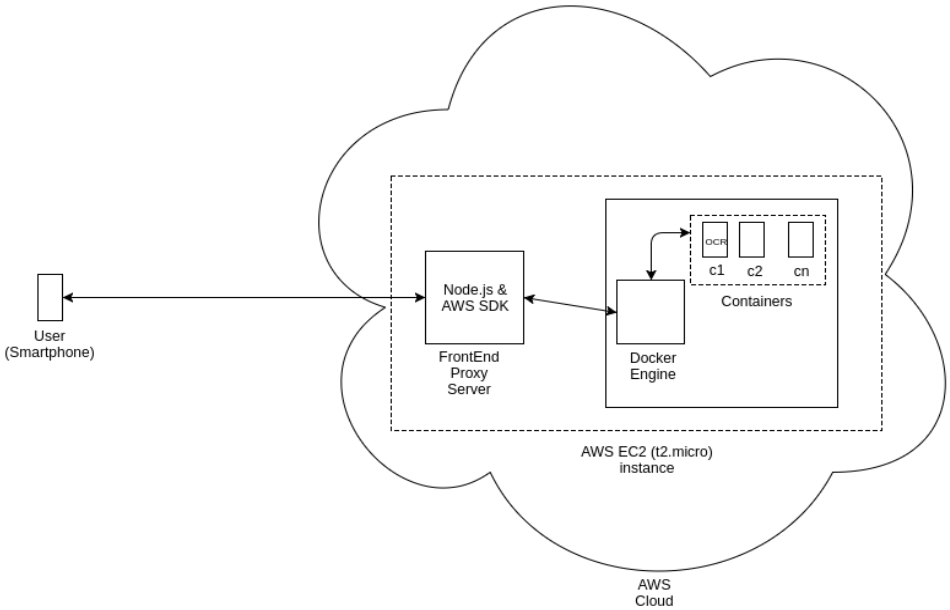

Figure 1. The high level architecture view of our proposed system

IV. IMPLEMENTATION AND RESULTS

For demonstration purpose we have implemented prototype application for image processing. Application is developed using open source technologies for better support and future enhancements. Application is online dictionary for English language. User is expected to submit image containing text characters through mobile app developed. Based on execution mode selected in app, image is processed either locally or on the cloud. And recognized word's meanings are shown to user. To support major mobile operating systems, client app is developed using cross platform Ionic framework. The front-end server runs on Express JS server with supporting Node modules. Front end server along with Docker engine is deployed in AWS EC2 Ubuntu instance (t2.micro). For image processing, we have used Google's Tesseract OCR library. Docker image contains Tesseract library API. When user submits the image to the cloud Docker image, it is processed by Tesseract module to get recognized text. Docker image contains 2 other sub modules i.e. pre-

IJARCCE

ISSN (Online) 2278-1021
ISSN (Print) 2319 5940

International Journal of Advanced Research in Computer and Communication Engineering
ISO 3297:2007 Certified
Vol. 6, Issue 9, September 2017

processing and dictionary API module. Pre-processing module performs stop word removal and dictionary API takes individual unique words and fetches the meanings for each recognized word. Results are then sent back to client mobile app in JSON format. Figure 2 shows the screenshot of our mobile app.

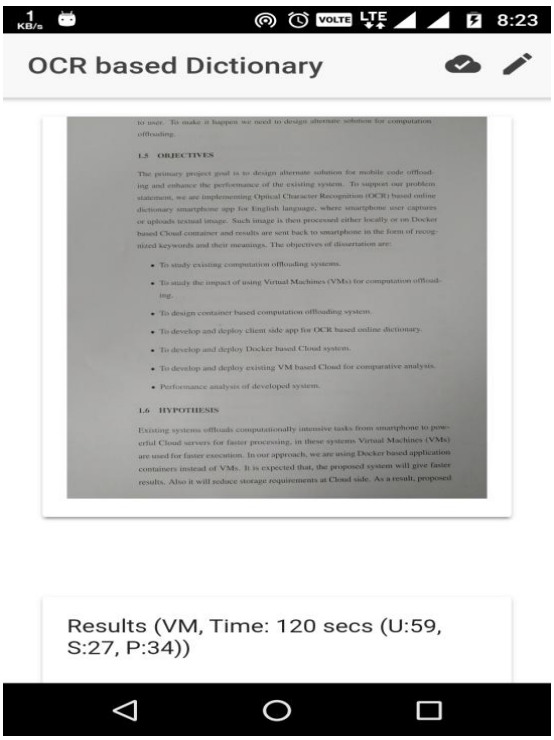

Figure 2. Screenshot 1 of our prototype system

We have tested the system in 2 different execution modes i.e. in VM over the cloud, and in Docker over cloud. For each mode, we have calculated execution time required to process input image. One run is not sufficient to get accurate execution time. For that 10 different run have done in both modes with variable size input images, as shown in table 2.

Table 2. Execution time for different test images on VM based cloud and Docker based cloud

Sr. No.	Input Image Size	VM (secs)	Docker (secs)
1	6.8 MB	89	87
2	6.6 MB	64	49
3	6.3 MB	70	74
4	6.2 MB	77	76
5	6 MB	64	50
6	840 KB	28	27
7	1.3 MB	9	8
8	1.9 MB	85	85
9	6.1 MB	45	30
10	302 KB	63	62

In our evaluation, we found that in order to perform OCR operation in smartphone it took more time for input images of size larger than 5 MB such that, we were not able to get output results. Our main focus is comparison of VM based cloud and Docker based cloud. As mentioned in table 2, it is found that our developed system requires less time as compared to existing (Virtual Machine) VM based system in terms of execution time. These results are calculated when VMs are in running mode. If we compare startup time of VM with Docker, Docker outperforms VM. So we can further reduce application execution time in Docker environment during startups and we can send faster response to smartphone user. Also cloud storage required to deploy Docker system is less as compared to VM, in our case we have 754 MB of Docker image, whereas VM is of minimum 1 GB.
Figure 3 shows the graph for input test cases. As shown in figure, our Docker based system executes in lesser time than VM based system.

IJARCCE

ISSN (Online) 2278-1021
ISSN (Print) 2319 5940

International Journal of Advanced Research in Computer and Communication Engineering
ISO 3297:2007 Certified
Vol. 6, Issue 9, September 2017

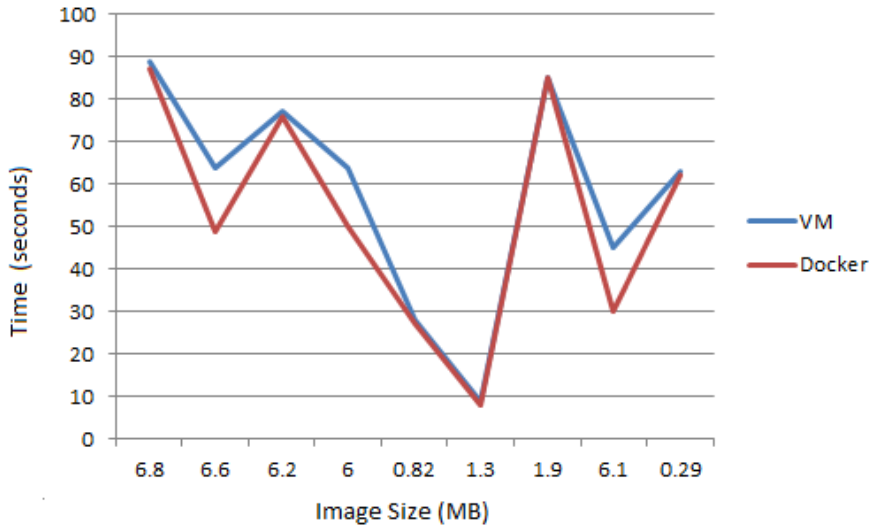

Figure 3. Performance Comparison of VM based cloud vs Docker based cloud

V. FUTURE WORK

We just developed prototype application for demonstration purpose without considering any security issues. During our survey, we found that security during authentication is overlooked in mobile cloud environment. Most of the attacks happen at authentication time [12]. Also we have not considered associations among keywords during pre-processing the recognized text, so we are going to address these issues in our future work

VI. CONCLUSION

In this paper, we presented a container based lightweight computation offloading system for enhancing the performance of smartphone application in mobile cloud domain. Experimental results show that, the developed system enhances the performance of smartphone application in terms of less execution time using container approach. Also startup time required is less in containers, so we can further minimize application execution time.

REFERENCES

[1] Chun B. G., Ihm S., Maniatis P., Naik M., Patti A., "CloneCloud: elastic execution between mobile device and cloud," Proceedings of the Sixth Conference on Computer Systems (EuroSys'11), pp. 301-314, 2011.
[2] Chen E. Y., Itoh M., "Virtual Smartphone over IP," IEEE 10th International Symposium on a World of Wireless Mobile and Multimedia Networks (WoWMoM), 2010.
[3] Kemp R., Palmer N., Kielmann T., Bal H., "Cuckoo: a computation offloading framework for smartphones," Mobile computing application and service, Volume 76 of Lecture Notes in Computer Science (LNCS), pp. 59-79, Springer, 2012.
[4] Elgendy A., Shawish A., Moussa M., "An Enhanced Version of the MCACC to Augment the Computing Capabilities of Mobile Devices Using Cloud Computing," International Journal of Advanced Computer Science and Applications (IJACSA), 2014.
[5] Satyanarayanan M., Bahl V., Caceres R., Davies N., "The Case for VM-based Cloudlets in Mobile Computing," IEEE Pervasive Computing, pp. 14-23, 2009.
[6] Shiraz M., Abolfazli S., Sanaei Z., Gani A., "A study on virtual machine deployment for application outsourcing in mobile cloud computing," The Journal of Supercomputing, Volume 63, Issue 3, pp. 946-964, March 2013.
[7] Shiraz M., Gani A., Khokar H., Buyya R., "A review on distributed application processing frameworks in smart mobile devices for mobile cloud computing," IEEE Communications Surveys and Tutorials, Vol. 15, No. 3, 2013.
[8] Shiraz M., Abolfazli S., Sanaei Z., Gani A., Buyya R., "Cloud-Based Augmentation for Mobile Devices: Motivation, Taxonomies, and Open Challenges," IEEE Communications Surveys and Tutorials, Vol. 16, No. 1, 2014.
[9] Shiraz M., Abolfazli S., Sanaei Z., Gani A., Buyya R., "Heterogeneity in Mobile Cloud Computing: Taxonomy and Open Challenges," IEEE Communications Surveys and Tutorials, Vol. 16, No. 1, 2014.
[10] Rahman M., Gao J., Tsai W., "Energy Saving in Mobile Cloud Computing," International Conference on Cloud Computing, pp. 285-291, IEEE, 2013.
[11] Gajbhe M., Sakhare S., "Performance Augmentation of Mobile Devices using Cloud Computing," International Journal of Computer Science and Information Technologies, Vol. 6, pp. 3581-3587, 2015.
[12] Alizadeh M., Abolfazli S., Zamani M., Bahrun S., "Authentication in mobile cloud computing: A survey, Journal of Network and Computer Applications, Elsevier, 2015.
[13] Zhang X., Jeong S., Kunjithapatham A., Gibbs S., "Towards an Elastic Application Model for Augmenting the Computing Capabilities of Mobile Devices with Cloud Computing," The Journal of Mobile Networks and Applications, Volume 16, Issue 3, Springer, pp. 270-284, 2011.
[14] Klein A., Mannweiler C., "Access Schemes for Mobile Cloud Computing," Eleventh International Conference on Mobile Data Management, IEEE, 2010.
[15] https://www.docker.com/
[16] Dey S., Sampalli S., Ye Q., "Message Digest as Authentication Entity for Mobile Cloud Computing," in International Performance Computing and Communications Conference (IPCCC), IEEE, 2013.
[17] https://github.com/tesseract-ocr/