

MVR: A Mobile Edge Computing Framework for Performance Offloading

Ankitha S¹ and Roopesh A R²

^{1,2}Rajeev Institute of Technology/CSE; Hassan, India
Email: ankushekar44@gmail.com, udp.roopesh@gmail.com

Abstract—Mobile applications are getting more complicated as mobile devices communication and sensor capabilities grow. As a method to extend the lifespan of batteries of mobile devices and boost application efficiency computation offloading one of mobile edge computing key features—becomes more relevant. This article introduces the offloading software framework as well as the "MVR" design, an innovative technology which enables computation offloading edge computing for mobile.

Index Terms— Mobile Edge Computing, Mobile Virtual Reality, Internet of Things, Design, Performance Offloading.

I. INTRODUCTION

Mobile gadgets are getting more and more frequent. According to a February 3, 2016 update of the Index for Cisco Visual Networking - Worldwide Mobile Data Traffic Projection [1], worldwide mobile data usage would increase from 3.7 megabytes monthly at the end of 2015 to 30.6 megabytes monthly by 2020. There will be 50 billion gadgets with internet access by 2020, promoting the goal of the Internet of Things. Thus, the battery life of mobile devices is not advancing at the same rate as the unparalleled expansion [2]. One of the biggest challenges facing the mobile sector is overcoming the energy constraints given the rapid rise in the use of mobile devices. Numerous studies on the topic of energy consumption have been executed from various angles. Although programming efficiency [5], vulnerability detection, and features optimization [6] of energy usage during application system view and energy efficiency of the device are improved, these strategies only minimize the significant energy use relying on the gadget itself. A totally distinct scientific strategy for energy consumption efficiency is offered by interface and server fusion. This server-interface fusion approach makes use of cloud assets that are already accessible to help the terminal software run well. A recently introduced concept known as Mobile Edge Computing (MEC) places Infrastructure and cloud computing technologies in the Radio Access Network (RAN) close to mobile clients. In fact, this is a feasible approach to resolving the battery issue and enabling mobile devices to carry their most energy-intensive functions to MEC environments when practical and feasible. This idea will shorten the time it takes for a program to run on a mobile device, which will save power usage.

II. MVR OFFLOADING

A network of mobile devices and Edge Network called MEC enables offloading computing. It's not essential to move all of the software implementation to the Edge Cloud in order to offload compute. A software is typically

broken into sections first, some of which require remote execution. In this case, all the modules in charge of the peripheral input and output. A choice must be made regarding which of the remaining modules is better suited for offloading. How should the offloading program be divided, then? And what should the decision be?

We introduce MVR, a cutting-edge architecture that utilizes the Edge Cloud platform for smooth offloading operation. MVR allows for the usage of virtual resources (VRs) in Edge Cloud, which boosts performance of applications while reducing the mobile device's own energy load and energy consumption. Theoretically MVR converts a mobile device's single-machine implementation into a distributed implementation that takes place on both the mobile gadget and Edge Network.

A. The MVR System Model

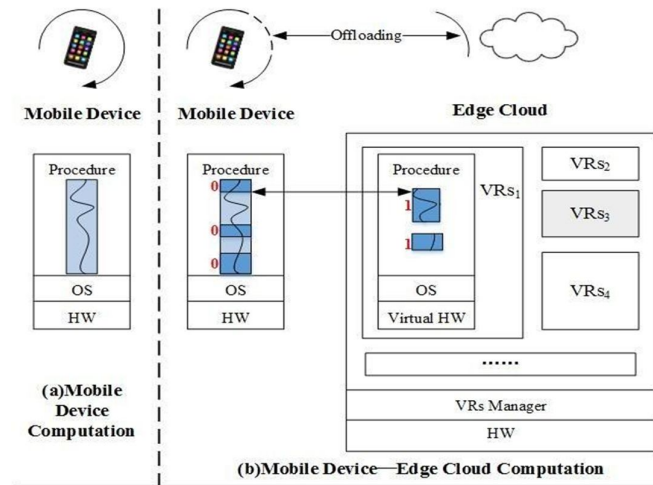


Fig 1. The MVR System Model

According to the Fig. 1 single-machine implementation model, the process (one activity of the program) is carried out consecutively on the mobile device. While using a scattered execution strategy, the mobile device's method is broken up into five tasklets, which includes three which weren't transferred to Edge Network (referred to "0" through the process) and two that were (referred to "1" in the procedure). The "1" tasklets are executed by a variety of VRs on the Edge Cloud (including VMs, containers, and others).

B. A Top-Down Perspective of the MVR Design

Previously stated, Edge Network offers a range of VRs, and all execution should take place in these VRs. A top-down perspective of the MVR design is presented in Fig. 2

The MVR runtime by the mobile device comprises four parts:

- **Proxy Monitor:** in charge of finding efficient proxy, such as Edge cloud.
- **Decision Engine:** Facilitate offloading decisions by combining static and dynamic analysis. Based on a set of constraint issues, Static Analyzer is in charge of finding techniques which may be deployed to the Edge Cloud: In order to avoid the appearance of layered offloading, the tasklet using native materials must be operated on a mobile device, along with the tasklet sharing native data. With regard to the present cyber context of constant shifts like broadband, Variable Tester evaluates both the prices and rewards of compute offloading.
- **Tasklet Distributor:** assigns a value of "0" or "1" to the tasklet, distributing the computational tasklet to be processed by VRs on the Edge cloud or mobile devices.
- **Executor:** performs the action and communicates to Edge Cloud.

However, each VR in the Edge Cloud is made up of six main components:

- **Registration Director:** in charge of enrolling and validating mobile devices making service requests.
- **Predictor:** forecasts when the mobile user will arrive and when the tasklet will be completed. We commonly refer to the work being offloaded as a "Very Important Person (VIP)" and, in order to further enhance application performance, we give resources in preparation for these "VIP".
- **Partition Analyzer:** designs offloading overhead operation in conjunction using the mobile device's

decision-making engine after evaluating the Edge Cloud's existing network parameters.

- **Decision Making tool:** offers a specific approach regarding the computing tasklets for a least amount of energy computation with a quick turnaround time and chooses which portions of tasklet "1" should be offloaded to Edge Cloud while keeping the "0" tasklet local.
- **Virtual Resource Controller:** Virtual resources (VRs) are the foundational element of Edge Cloud. A specific VRs Controller (VRC) in the Edge Cloud is responsible for controlling each of these VRs. The necessary resources are allotted using Virtual Resource Controller when the offloaded computational tasklets emerge. One tasklet could require many VRs, therefore the virtual resource manager must combine the execution outcomes and provide them to the mobile gadgets.
- **Administrator:** identical to mobile gadgets, an administrator carries out operations during the execution of a procedure.

In contrast to the above elements, the operating system (OS), hardware (HW), and remote hardware on the edge network are some additional fundamental elements for system operation.

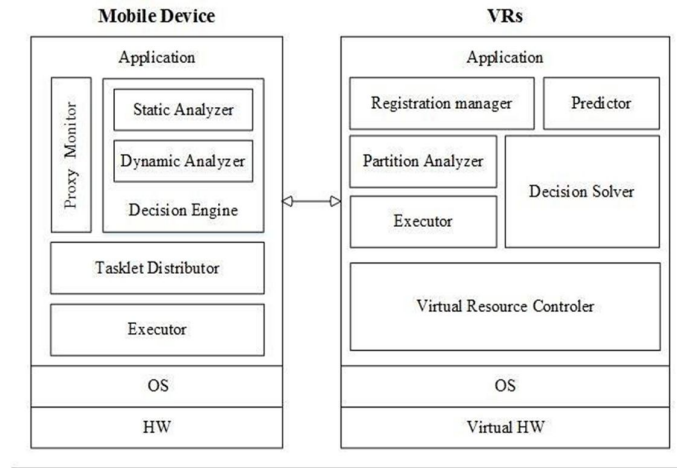


Fig 2. Prototype's MVR Design

C. MVR Offloading Procedure

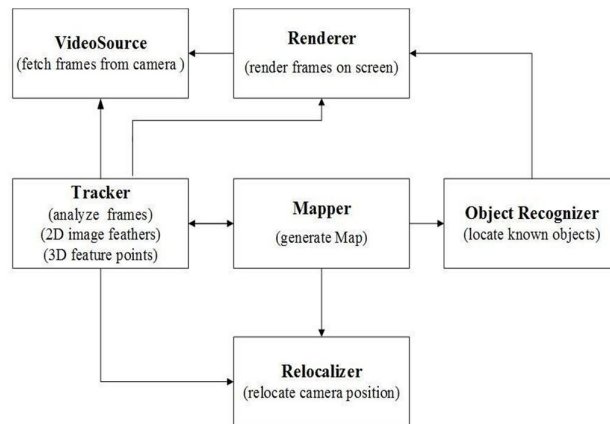


Fig 3. Tasklets related to the AR algorithm

Useful programmes that can use Edge Cloud often have time-sensitive requirements and need extremely low computation and communication delay in a setting with constrained bandwidth and processing power. In this section, we'll go over how MVR is applied to virtual reality (VR). Six tasklets- Video Source, Monitor, Image Recognizer, Mapper, Renderer, Relocalizer —are listed in [7], as shown were divided into an AR algorithm that used a video clip in grayscale for 3 dimensional things. Fig. 3 illustrates how these tasklets are related to one

another. To determine the position of camera using 2D picture attributes and 3D detail lines whose position has been developed and altered by Monitor, the VideoSource retrieves video frames that have been produced by the Renderer and computed by the Tracker. When no feature points are detected, the Relocalizer attempts to move the position instead of attempting to locate recognized objects.

As an illustration, a mobile device operating an AR application looks for an innovative Edge Cloud, which recognizes and anticipates the smartphone device that defines it based on a fresh or normal customer, along with "VIP." Six tasklets comprise a finished AR technique, based on the Decision Engine and Decision Solver. In this augmented reality instance, the Tasklets of Video Source and the Renderer must be implemented on the cellphones since they utilise gadgets that is unique to that device. Therefore, these tasklets would be given a "0" rating. While the Relocalizer moves the camera position, the Tracker examines video clips and determines the position of camera, and both must be labelled as "1". The Tasklet Distributor will send each tasklet to a smartphone device or an Edge Network. When a tasklet is transferred to Edge Cloud, the Remote Resource Manager there starts assigning matched resources based on the user's identity and the needs of the task. The specific VRC in Edge Cloud will gather outcomes when each tasklet has been finished and return them to the smartphone. The following action is taken once the usage is finished by the smartphone, which combines the outcomes based on the connections between the tasklets.

III. CONCLUSION

In order to build an incredible connection between compute and interaction-expensive apps with the efficient Edge Network, the MVR architecture is described for processing offload in the edge of mobile computing in this paper. With the introduction of new technology like 5G, there are numerous difficult problems that need to be handled. For instance, the combined optimization issue involving runtime, energy consumption, rent costs, latency time, and other factors is not well understood in a multi-dynamic system. Can a reliable "Mobile Federation" be established and maintained using a system? In a scenario where there are several clouds and service suppliers, how useful is such an approach? Mobile user's level of privacy might swiftly and regularly change from one area to another. In the context of MEC, we anticipate how the rapid growth of security apps for smart phone users could render compute dumping harder to achieve.

REFERENCES

- [1] C. VNI, "Cisco visual networking index: Global mobile data traffic forecast update, 2015-2020," URL: <http://www.cisco.com/c/en/us/solutions/collateral/service-provider/visual-networking-index-vni/white-paper-c11-520862.pdf>.
- [2] M. R. Palacin, "Recent advance in rechargeable battery materials: a chemists perspective," *Chemical Society Reviews*, vol. 38, no. 9, pp. 2565–2575, 2009.
- [3] A. Zhou, S. Wang, B. Cheng, Z. Zheng, F. Yang, R. Chang, M. Lyu, and R. Buyya, "Cloud service reliability enhancement via virtual machine placement optimization," *IEEE Transactions on Services Computing*, 2016.
- [4] S. Kosta, A. Aucinas, P. Hui, R. Mortier, and X. Zhang, "Thinkair: Dynamic resource allocation and parallel execution in the cloud for mobile code offloading," in *Infocom, 2012 Proceedings IEEE*, pp. 945–953, IEEE, 2012.
- [5] D. Li and W. G. Halfond, "An investigation into energy-saving programming practices for android smartphone app development," in *Proceedings of the 3rd International Workshop on Green and Sustainable Software*, pp. 46–53, ACM, 2014.
- [6] K. H. Kim, A. W. Min, D. Gupta, and P. Mohapatra, "Improving energy efficiency of wi-fi sensing on smartphones," vol. 34, no. 17, pp. 2930–2938, 2011.
- [7] T. Verbelen, P. Simoens, F. De Turck, and B. Dhoedt, "Cloudlets: Bringing the cloud to the mobile user," in *Proceedings of the third ACM workshop on Mobile cloud computing and services*, pp. 29–36, ACM, 2012.
- [8] X. Lin, Y. Wang, Q. Xie, and M. Pedram, "Task scheduling with dynamic voltage and frequency scaling for energy minimization in the mobile cloud computing environment," *IEEE Transactions on Services Computing*, vol. 8, no. 2, pp. 175–186, 2015.