CSCE 664 Wireless and Mobile Systems Project Proposal

Yukun Zeng, Bingqian Jiang
Department of Computer Science and Engineering
Texas A&M University
College Station, TX 77840
Email: {yzeng, jiangbingqianj}@tamu.edu

***Abstract*—We consider the problem of scaling mobile cloud benchmark on pay-as-you-go cloud computing platforms like AWS and Google Cloud. The intuition behind such an idea is that a large amount of physical mobile devices are hard to manage and impractical in terms of infrastructure cost. In this paper, we propose to virtualize mobile devices directly on top of Xen, a common virtualization system used in typical public cloud like AWS, this would eradicate the overhead of nested virtualization used in other virtualization solutions like Ravello Systems. As we virtualized mobile cloud in this way, the network connection (virtual I/O in this case) between different virtual devices is totally different from real world scenario where mobile devices are connected through wireless networks, typically WLAN. To obtain a more realistic virtualized environment, we further introduce a virtual router that sits on top of the virtual I/O and is responsible for forwarding, dropping, retransmitting data packets so that we can emulate the wireless network behavior on virtual I/O.**

***Index Terms*—Virtual Router, Wireless Network Emulation, Virtualized Networking, Virtualization, Mobile Cloud**

I. Introduction

A. Mobile Cloud

With the development of computing capability, we can now achieve performance simular to personal computer on a portable mobile device. This on the one hand eased our life by providing more complex and useful applications on mobile devices while it also created some dilemmas where massive computing resources fitted into phones, smart bands, etc are left idle and wasted most of the time. People have proposed many mobile cloud solutions that leverage the idle computing resource on those devices and use them to fill in high computing resource demand elsewhere. A lot of efforts have been put into mobile cloud and people have developed different platforms to solve different problems by leveraging idle mobile computing resources, we refer readers to [1] [2] [3] for more information about the emerging mobile cloud computing trend. In this paper, we will focus on a mobile cloud platform called Mobile Storm [4], which is a state-of-the-art platform for distributed stream processing and it is a previous work by LENSS lab.

B. Scalibility Benchmark

Naturally we can think of scalability as a important metric in evaluation of such platforms, however it's unpractical to probe the platform scalability using dedicated mobile device clusters as it's too expensive and even harder to manage than server clusters. To solve such a problem, we proposed to benchmark those platforms by leveraging the on-demand computing resources provided by state-of-the-art cloud computing platforms like AWS and Google Cloud, i.e., we could build mobile cloud by virtualizing any amount of devices based on our demand thereby testing the platform scalability. This approach does not only fit into the benchmarking of mobile cloud platforms but also is suitable for scaling different tests like probing the max load of an application server by adding more and more mobile devices as clients. However, new problems emerge if we are proceeding this way. In most common scenarios of mobile cloud, devices are usually connected through wireless networks, the backend can be either one server or a cluster and those servers are usually connected by wired networks. A typical example of the Mobile Storm cluster can be found in Fig. 1 where we have multiple mobile devices connected to each other as well as the master server through wireless router. However, things get totally different if we tries to virtualize the Mobile Storm cluster in public clouds. As shown in Fig. 2, the virtual mobile devices communicates with each other using virtual I/O. Different virtual devices might be running on the same physical machine or different physical machine, thus the virtual I/O can be either I/O of certain machine or wired network between different machines. Anyhow, such connectivity between virtual devices differs a lot from the actual scenario and we have to find a way to emulate wireless network in the virtualized environment so that we can benchmark in a realistic and scalable fashion.

C. Existing Solutions

Though there is a huge trunk of research papers in the field of cloud computing, virtualization as well as mobile clouds, there are few works dedicated to scaling mobile cloud benchmark by virtualizing mobile devices on public clouds, most highly relevant systems are commercialized and available now on market place of AWS. Ravello Systems and Genymotion are two commercial platforms that enable the scalability benchmark for mobile cloud. Ravello Systems enables us to run any virtual machine on any cloud without making any changes through adding a new hypervisor called HVX on top of VMs provided by public cloud platforms. Breaking down the stack of Ravello's solution, as shown in

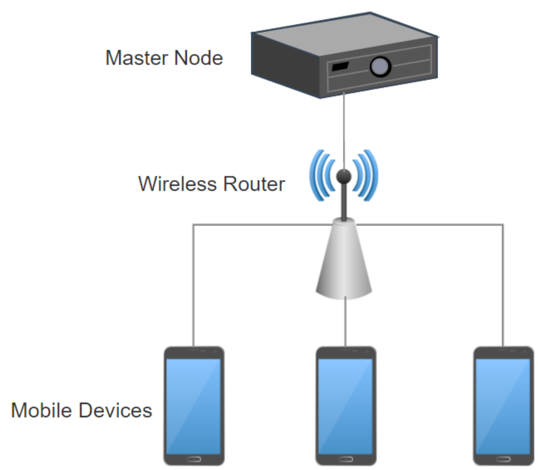

Fig. 1. Networking in Actual Mobile Storm Cluster

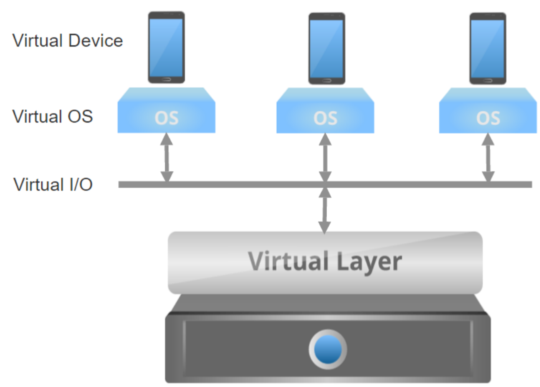

Fig. 2. Networking in Virtualized Mobile Storm

3, we can see that we can further run hypervisor on VMs provided by AWS and then run any Virtual Machine on top of the hypervisor. The solutions provided by Genymotion incorporate two aspects, one is to emulate Android x86 OS on top of a host OS in a near-native manner (basically running Android x86 on VirtualBox with all hardware acceleration), the other is to emulate Android OS on demand in public cloud. Genymotion on Demand could similarly provide easy pay-as-you-go Android device virtualization while it's close source and its internal mechanism is rarely discussed. While both solutions can be used for scaling mobile cloud benchmark, none of them can restore the actual wireless network scenario thus cannot guarantee realistic benchmark.

D. *Observation and Thoughts*

As shown in Fig. 2, the network of virtual Mobile Storm cluster is based on virtual I/O which is totally different from the wireless network behavior in actual scenario. We assume the virtual I/O is totally reliable. To acquire a wireless-alike environment in the virtualized cluster, our thought is that we might add a router (the router term might not be exactly accurate) on top of virtual I/O to control the traffic. The virtual router is responsible of forwarding data packets to target

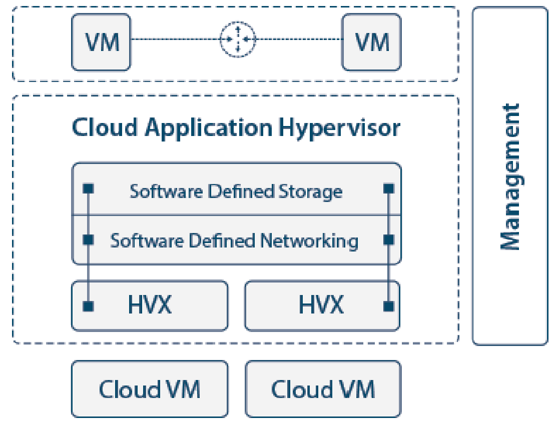

Fig. 3. Destack Ravello System Architecture

virtual devices like a wireless link that might generate packet loss, delay, retransmission, basically we wanna incorporate all features of a wireless link as data packets pass through the virtual router.

E. *Contribution*

We expect the contributions of this paper as follows:

- Propose a new approach of virtualizing mobile cloud directly on top of the Xen virtualization layer, i.e., be able to run android virtual devices cluster on Xen environment, this might also require efforts in virtualized networking as we need to figure out a way for building the interconnectivity between different virtual devices.
- Develop a virtual router sitting on top of the virtual I/O to forward data packet and control the transmission between different virtual devices, the final goal of the virtual router is to emulate a wireless network behavior on virtual I/O so that we obtain a realistic environment when virtualizing mobile devices cluster on public clouds.

II. Related Works

Current solutions for virtualizing mobile devices on cloud mainly include Genymotion [5] and Ravello Systems. They can both provide pretty good performance for virtualized devices and are highly scalable, but none of them takes common wireless communication paradigms between mobile devices into consideration. This is far from ideal as it is proved in [6] the wireless network differs a lot from wired network. Both wireless and wired network simulation or emulation have been studied thoroughly in the last decades [7] [8]. Network virtualization can be realized directly through virtualization systems like VMWare, VirtualBox or Xen, common networking provided by those virtualization systems include NAT, Host-only and bridged. Related works in network virtualization area include [9] [10]. However, few researches are dedicated in wireless network emulation in virtualized environment, we need a physical wireless network card on the host machine to acquire wireless behavior for VMs, which is impractical in public cloud. Among all related works, we focus on the following researches as they are the most related ones to our topic:

A. Empower: a network emulator for wireline and wireless networks

Empower[7] is a distributed network emulation system. It is able to facilitate the emulation of either wireline or wireless networks. In the case when network topology is critical to the underlying network protocol, the emulator could provide specific mechanisms to emulate network topology. Empower can also generate user-defined network conditions and traffic dynamics at packet level. However, Empower is specialized in dealing with emulation for wire and wireless networks, which does nothing with cloud service computing. Therefore, we need to move forward for other applications which would fit our concern more.

B. HVX: Virtualizing the Cloud

HVX [11] is a virtualization platform that enables complete abstraction of underlying cloud infrastructure from the application virtual machines. HVX allows deployment of existing VMs into the cloud without any modifications, mobility between the clouds and easy duplication of the entire deployment. HVX can be deployed on almost any existing IaaS cloud. Each instance of the HVX deployment packs in a nested hypervisor, virtual hardware, network and storage configuration. Combined with image store and management APIs, the HVX can be used for the creation of a virtual cloud that utilizes existing cloud provider infrastructure as the hardware rather than using physical servers, switches and storage.However, HVX's virtualization still exhibits the problem that it is unable to simulate the real world wireless communication, which may have higher latency and packets loss rate than wired connection, which HVX adopt at its end. On the other hand, HVX could be referred as running virtualization on VM, which would significantly slow down the speed. Thus, we may need to propose a better solution to optimize discussed problems.

C. Genymotion on Demand

Genymotion [5] is a new cloud-based virtualization platform aimed at Android developers. Running on Amazon Web Services, Genymotion On Demand offers access to the full Android operating environment online. However, Genymotion on Demand is a closed source software whose source code is not published. Therefore, we currently could not be guaranteed whether performance of Genymotion on Demand could satisfy our requirements.

D. Xen

Xen [12] is an x86 virtual machine monitor which allows multiple commodity operating systems to share conventional hardware in a safe and resource managed fashion, but without sacrificing either performance or functionality. This is achieved by providing an idealized virtual machine abstraction to which operating systems such as Linux, BSD and Windows XP, can be ported with minimal effort. The virtualization approach taken by Xen is extremely efficient: it allows operating systems such as Linux and Windows XP to be hosted simultaneously only with a negligible performance overhead — at most a few percent compared with the unvirtualized case.

Compared with Xen, many other virtualization systems expose a lot of disadvantages. Some require specialized hardware, or cannot support commodity operating systems. Some target 100% binary compatibility at the expense of performance. Others sacrifice security or functionality for speed. Few offer resource isolation or performance guarantees; most provide only best-effort provisioning, risking denial of service. Xen overcomes all above cons, and thus gains popularity with modern public cloud service like AWS. In this paper, we built our virtualized mobile cluster on XenServer, a commercial implementation of Xen. Our research will conducted based on Xen as well. With functions Xen provide, we will further extend and optimize the performance to achieve a faster virtualization environment for real-world wireless cloud service simulation.

III. Proposed Solution

A. Virtualizing Mobile Devices in Cloud

Ravello systems provided solution to run any virtual machine on any cloud without changes to operating systems through nested virtualization, however there will be additional overhead in the 2nd layer of virtualization in terms of processor power, network throughput, etc. Moreover, normal customers of public clouds are serviced directly from the virtualized OS level (i.e., an operating system is immediately installed per request) and they might not have access to nested virtualization as it's a low level setting in Xen which is not commonly exposed by cloud platforms for security considerations. Here in this paper we propose to virtualize Android system directly on top of Xen, which can greatly reduce the overheads. More specifically, we use XenServer as the hypervisor for the physical machine and install Android directly on VMs without any intermediate Linux or Windows system.

B. Wireless Emulator on Virtual I/O

As we have mentioned before, we can test the scalibility of a mobile cloud platform by leveraging virtualization techniques on public clouds, however, such an approach cannot provide a realistic wireless networking environment as the communication between different virtual devices goes through the virtual I/O, which is either the virtualized system I/O or wired network between different host servers. To better emulate the actual wireless networked mobile cloud platform, we propose a wireless network emulator on top of virtual I/O to virtually control the delivery of data packet between different virtual devices to acquire a network transmission pattern similar to real wireless network system. As shown in Fig. 4, the virtual wireless network emulator lies on the virtual I/O layer and is responsible for coordinating the network transmission between different virtual devices. At the current stage, we expect the emulator to be an application on Android, it might exposes some ports to receive or output data packets in the virtualized network, it works by dropping packets and retransmission in a wireless manner.

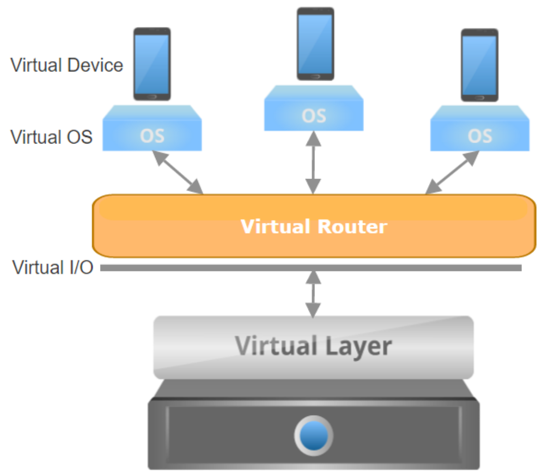

Fig. 4. Virtual Wireless Network Emulator

C. Expected Chanllenges

Our project is highly challenging in that it incorporates problems in both virtualization and networking, basically we need to find proper approach to achieve the following goals:

- Improving virtual device performance: virtual machines are typically much slower than physical machines with the same hardware and software configuration, even with the hardware speedup provided by newly introduced Intel VT-x or AMD-V CPU, which makes the benchmark harder. To reduce the virtualization overhead, we propose to virtualize mobile devices at the lowest level possible and we adopt Android for x86 to acquire better support for desktop CPUs (oppsed to typical Android device CPU like ARM).
- Networking in virtualized environment: as we mentioned above, we need to virtualize mobile device cluster using XenServer, however, to the best of our knowledge so far, native network is not provided directly by the virtualized Android system on XenServer, we might have to deal with device driver and figure out a way to network virtual devices in either the same machine or LAN.
- Wireless network emulation: though a huge volume of researches have been put into wireless network emulation, here we expect many challenges since we are doing emulation on Android platform and the whole system lies in a virtualized environment, the final implementation of wireless emulation should be a totally different story comparing to previous work.

REFERENCES

[1] K. Kumar and Y.-H. Lu, "Cloud computing for mobile users: Can offloading computation save energy?" *Computer*, vol. 43, no. 4, pp. 51–56, 2010.

[2] H. T. Dinh, C. Lee, D. Niyato, and P. Wang, "A survey of mobile cloud computing: architecture, applications, and approaches," *Wireless communications and mobile computing*, vol. 13, no. 18, pp. 1587–1611, 2013.

[3] N. Fernando, S. W. Loke, and W. Rahayu, "Mobile cloud computing: A survey," *Future generation computer systems*, vol. 29, no. 1, pp. 84–106, 2013.

[4] Q. Ning, C.-A. Chen, R. Stoleru, and C. Chen, "Mobile storm: Distributed real-time stream processing for mobile clouds," in *Cloud Networking (CloudNet), 2015 IEEE 4th International Conference on*. IEEE, 2015, pp. 139–145.

[5] M. Wilson, "Genymotion On Demand is a cloud-based virtualized Android emulator for the enterprise," https://betanews.com/2017/02/27/genymotion-on-demand-android/, 2017, [Online; accessed 20-March-2017].

[6] S. Cen, P. C. Cosman, and G. M. Voelker, "End-to-end differentiation of congestion and wireless losses," *IEEE/ACM Transactions on networking*, vol. 11, no. 5, pp. 703–717, 2003.

[7] P. Zheng and L. M. Ni, "Empower: A network emulator for wireline and wireless networks," in *INFOCOM 2003. Twenty-Second Annual Joint Conference of the IEEE Computer and Communications. IEEE Societies*, vol. 3. IEEE, 2003, pp. 1933–1942.

[8] G. Judd and P. Steenkiste, "Using emulation to understand and improve wireless networks and applications," in *Proceedings of the 2nd conference on Symposium on Networked Systems Design & Implementation-Volume 2*. USENIX Association, 2005, pp. 203–216.

[9] A. Menon, A. L. Cox, and W. Zwaenepoel, "Optimizing network virtualization in xen," in *USENIX annual technical conference*, no. LABOS-CONF-2006-003, 2006.

[10] N. M. K. Chowdhury and R. Boutaba, "Network virtualization: state of the art and research challenges," *IEEE Communications magazine*, vol. 47, no. 7, 2009.

[11] Oracle, "HVX: High performance nested virtualization," https://www.ravellosystems.com/technology/nested-virtualization, 2015, [Online; accessed 20-March-2017].

[12] P. Barham, B. Dragovic, K. Fraser, S. Hand, T. Harris, A. Ho, R. Neugebauer, I. Pratt, and A. Warfield, "Xen and the art of virtualization," in *ACM SIGOPS operating systems review*, vol. 37, no. 5. ACM, 2003, pp. 164–177.