Packet Forwarding Using Xen

Hugo E. T. Carvalho, Rodrigo S. Couto, Lyno Henrique G. Ferraz, Natalia C. Fernandes, Marcelo D. D. Moreira,
Igor M. Moraes, Miguel Elias M. Campista Luís Henrique M. K. Costa, Otto Carlos M. B. Duarte
Universidade Federal do Rio de Janeiro - GTA/COPPE - Rio de Janeiro, Brazil

***Abstract*—Network virtualization allows the development of systems with several virtual routers (VR) over the same physical router. Each virtual router can execute, at the same time, different protocol stacks without disrupting the coexisting routers. The main purpose of this work is to analyze Xen capabilities and limitations and compare its efficiency with native solutions. Tests are performed in order to isolate possible bottlenecks and see how each mechanism provided by Xen can impact in performance.**

I. Overview

Xen provides an architecture capable of supporting virtual routers. It offers an high flexibility framework to build virtualized systems, with the price of sacrificing some of its performance. In the Xen architecture, a layer called Virtual Machine Monitor (VMM) is spawned between the VRs and the real hardware. This layer is responsible for multiplexing the hardware access among all VRs, by the advent of an scheduler, that must decide which VR can access the hardware at each time. An abstraction of network access is created by a complex I/O mechanism. All these implementations are susceptible to bugs and misbehaviors that can lead to performance loss [2]. Xen architecture also includes a privileged virtual machine, called Domain 0, that has direct access to disk and network hardware and aids the VMM on I/O procedures. Domain 0 develops an important role in the behavior of the virtual routers, because it mediates the access between them and the real I/O device.

II. Testbed

We develop a testbed to evaluate the performance of different forwarding elements. Our testbed is composed of a traffic generator machine (TG), a traffic receiver machine (TR) and a traffic forwarder machine (TF) that interconnects TG and TR. Tests are performed in order to compare TF forwarding performance running either Xen or native Linux.

III. Partial Results

The first performed test analyzes the influence of Xen mechanisms on the response time of TR. For different background traffic being forwarded by TF, we measure the round trip time (RTT) between TG and TR, with packets passing through TF. All configurations, i.e., native Linux and Domain 0 in bridged and routed modes, but the virtual router configuration show constant results (up to 0.25 ms). The maximum obtained RTT is 1.25 ms, achieved with background traffic of 500 Mb/s and 64-byte packets. This might seem a huge gap, but when we are aware that a 150 ms delay is fully acceptable in VoIP applications [1], we can see that this difference is not significant, even if there are more hops in the end-to-end path between TG and TR. The test performed in Fig. 1 aims at finding how forwarding is affected when Xen deals with parallel networks. We can see that there are two main bottlenecks in the Xen. The first one is the I/O mechanism that must combine CPU usage

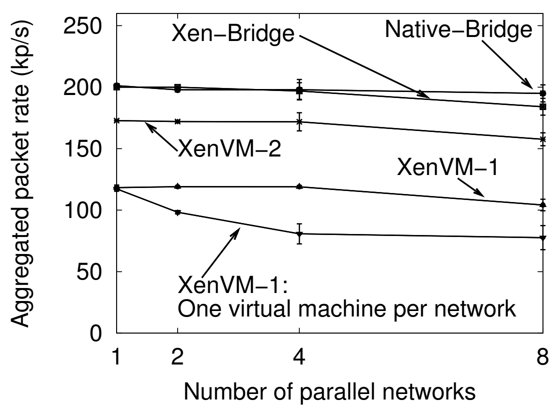

Figure 1. Forwarding capabilities with 1, 2, 4 and 8 parallel networks. The generated packet rate is 200 kp/s and packet size is 64B.

and memory page mappings, which degrades performance. The second one is the Xen CPU scheduler. As the number of parallel networks increases, the dispute for CPU augments. If VRs cannot have enough CPU, they cannot give Domain 0 enough free pages for receiving packet, and hence Domain 0 starts dropping packets. If we give an exclusive core for the VR (XenVM-2 configuration), the forwarding packet rate enhances in up to 50 kp/s, when compared to XenVM-1 (core shared with Domain 0) configuration, proving that there was a lack of CPU availability.

IV. Future Directions

Our test show that VRs can be used to forward latency-sensitive traffic without disrupting reliability for end users. Despite that fact, there are limitations in the packet forwarding rate, mainly due to the CPU scheduler and the I/O mechanism. Our next tests will be focused on new technologies such as multiple queue NICs, Netchannel2, and Intel's SR-IOV, that will bring lots of enhancements upon Xen. Those technologies can both alleviate the charge over Domain 0 and allow VRs to interact directly with the underlying hardware without disrupting isolation and security.

Acknowledgments

This work was supported by FINEP, FUNTTEL, CNPq, CAPES, and FAPERJ.

References

[1] H. Fathi, R. Prasad, and S. Chakraborty. Mobility management for VoIP in 3G systems: evaluation of low-latency handoff schemes. *IEEE Wireless Comm.*, 12(2):96–104, 2005.

[2] A. Menon, J.R. Santos, Y. Turner, G.J. Janakiraman, and W. Zwaenepoel. Diagnosing performance overheads in the Xen virtual machine environment.