New I/O Virtualization Techniques

Lyno Henrique G. Ferraz, Hugo E. T. Carvalho, Pedro S. Pisa, Otto Carlos M. B. Duarte
Universidade Federal do Rio de Janeiro - GTA/COPPE - Rio de Janeiro, Brazil

***Abstract*—The resurgence of virtualization has been motivated by the increased efficiency of resource utilization, because it is possible to share a machine among several virtual machines (VMs). The use of virtualized environments, however, brings performance drawbacks, because the virtualization layer introduces an extra task of virtual machine multiplexing. This overhead is particularly critical for I/O intensive loads. This paper investigates new techniques for I/O virtualization to realize the full benefits of virtualization for I/O intensive scenarios.**

I. Introduction

One of the most critical cases of heavy I/O utilization is the network intensive environment. In this case, it is necessary to provide high throughput and low delay at the same time. In addition, it is necessary to isolate the traffic of one VM from another. This scenario is required by several Internet services, such as web or game servers and, recently, virtual routers.

Today, I/O virtualization is under responsibility of the virtual machine monitor (VMM), which must multiplex outgoing data flow and demultiplex the incoming data flow. Considering the network example, the VMM must implement a way to share the link, divide its access, and classify incoming packets. At the same time, it must be fair to all VMs [1].

II. I/O Virtualization Models

There are several models of I/O virtualization [2]:

Emulation model: In this model, the VMM exposes a virtual device to the VM as an existing I/O device, often a legacy device. This model provides good compatibility with existing device drivers and require no change in the guest OS, although it introduces some overhead caused by translating instructions.

New software interfaces: Another existing model is the introduction of new software interfaces which are virtualization-friendly. A synthetic device driver is created to communicate with the new software interface, which reduces the overhead when compared with the emulation model. This model, however, has reduced compatibility due to the need of specialized support for the device driver by the guest OS.

Direct assignment: In this model, the driver that runs in the guest OS has direct access to the physical device. The VMM must provide access between the device driver and the physical device. Robust I/O assignment with minimal VMM involvement requires additional hardware support. Although this model gives full and isolated control of a physical device to a VM, it does not allow sharing of the physical device.

I/O device sharing: In this model, an I/O device supports multiple functional interfaces, each of them may be assigned to a VM. The device hardware must support multiple I/O requests of any VM.

III. Trends

In order to improve the overall performance of I/O sharing between VMs, more specific to network sharing, several techniques have been proposed.

Multiple Queues: The VMM has an important task of classifying packets. This task incurs in VMM processing overheads, because VMM besides delivering the packets has also to classify them. Modern network interface card (NIC) addresses this problem by doing itself the packet classification. To accomplish this feature, the NIC has several queues, which can be used by the VMs to isolate the traffic from each other [3].

Direct I/O Access: This is a functionality provided by the modern motherboards to safely use the direct device assignment model. This technique allows a device to access different address spaces, and hence it can access the VM address space directly. Memory accesses are assisted by the chipset of the motherboard. The chipset intercepts the device memory access and makes use of I/O page tables to determine whether the access is permitted and the actual location of the access. This mechanism, however, has problems of scalability since a physical device can not be shared between several VMs, it is assigned to only one VM [2].

Single Root I/O Virtualization (SR-IOV): This technique provides direct access between the queues and the VM, allowing the use of the I/O device sharing model. The devices can be implemented in numerous ways, but since most of them are accessed over PCI, the PCI-SIG created a standard approach. The standard provides a way of bypassing the VMM involvement in data movement. This standard approach also defines a way of sharing a NIC to several VMs [4].

IV. Conclusion and Future Work

We are currently developing a prototype to analyze the performance of the above-mentioned I/O virtualization techniques. These techniques provide natively device sharing support for virtual machines, minimizing overheads in network virtualization. Our final goal is to provide a high performance network infrastructure, which fulfill all the requirements of a virtual network scenario.

Acknowledgments

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

References

[1] S. Rixner, “Network virtualization: Breaking the performance barrier,” in *ACM Queue*, vol. 6, no. 1, Jan. 2008, pp. 36–52.

[2] Intel Corporation, *Intel Virtualization Technology for Directed I/O*, Sep. 2008. [Online]. Available: download.intel.com/technology/computing/vptech/Intel(r)_VT_for_Direct_IO.pdf

[3] ——, *Intel 82576 Gigabit Ethernet Controller Datasheet*, Oct. 2009. [Online]. Available: download.intel.com/design/network/datashts/82576_Datasheet.pdf

[4] Intel LAN Access Division, *PCI-SIG SR-IOV Primer*, Dec. 2008. [Online]. Available: download.intel.com/design/network/applnots/321211.pdf