CPU Allocation on Xen Virtual Networks

Rodrigo S. Couto, Hugo E. T. Carvalho, Lyno Henrique G. Ferraz,
Miguel Elias M. Campista, Luís Henrique M. K. Costa and Otto Carlos M. B. Duarte
Grupo de Teleinformática e Automação - PEE/COPPE - DEL/POLI
Universidade Federal do Rio de Janeiro - Rio de Janeiro, Brazil
Email:{souza,hugo,lyno,miguel,luish,otto}@gta.ufrj.br

***Abstract*—In this work we investigate virtual network performance using Xen. We perform experiments using scheduling parameters and static CPU allocation. Results indicate that the forwarding packet rate of each virtual router can be controlled by adjusting CPU scheduling parameters.**

I. Introduction

The growing need to develop pluralist architectures for the future Internet leads to the use of virtual routers in platforms such as Xen [1]. Xen implements a software layer called VMM (Virtual Machine Monitor) which controls how virtual machines, called Domains, access hardware resources. Domain 0 is a special Domain that implements an interface between abstract devices within virtual machines and physical devices. In the case of networking, packets destined to unprivileged Domains must pass through Domain 0 before reaching virtual interfaces. This communication between Domain 0 and other Domain imposes limitations to network tasks due to high CPU consumption. Moreover, in Xen environment the Domains share physical resources, including CPU. In this paper, we aim at analyzing the impact of CPU resource allocation on virtual routers performance.

II. Xen Credit Scheduler

For each Domain running on a Xen environment is assigned a number of Virtual CPUs (VCPUs). These VCPUs correspond to the CPUs that the running processes in a Domain can see. Xen implements the Credit Scheduler to manage CPU time given to each Domain. This scheduler dynamically allocates a physical CPU core to each VCPU for a certain amount of time. In Xen platform the administrator can also manually allocate a CPU core to a VCPU of a certain Domain but cannot guarantee if another VCPUs will share the same core. The basic idea of Xen Credit Scheduler is to assign credits to each Domain based on two parameters: weight and cap. Each time a Domain executes, i.e. each of its VCPUs, it consumes credits. Based on credit accounting, the scheduler determines which VCPU can run. The weight determines the share of CPU time each Domain can get and doesn't represent an upper-limit, i.e Domains can use more than its share when CPU is idle. The cap, in turn, is an absolute value that represents a percentage of CPU a Domain can use, i.e cap=50 is half CPU. The cap gives more control to the system administrator about the CPU usage by the Domains.

III. Performance Evaluation

We aim at analyzing how Domain 0 CPU allocation impacts forwarding performance and how can a system administrator manage CPU resources to guarantee a certain throughput to a virtual router. To accomplish this, we conduct two experiments. These experiments were carried out in a testbed with traffic generators (TGs) sending packets to traffic receivers (TRs) passing through a traffic forwarder (TF). All machines run Debian Linux and TF runs Xen 3.4.2 system.

A. Domain 0 CPU cores

All packets destined to virtual routers must pass through Domain 0. Thus, we need first to evaluate the performance of Domain 0 adjusting the number of CPU cores allocated to it. In this experiment, we send packets from one of the TGs to one of the TRs at a fixed rate and vary the number of CPU cores allocated to Domain 0. We run this test for up to four virtual machines forwarding packets and sharing the same TF physical network interfaces. On the one hand, the performance of packet forwarding is severely impacted when all Domains share one CPU core because of the high contention for CPU resources. On the other hand, the best performance is achieved when Domain 0 has one exclusive core. Surprisingly, assigning more cores to Domain 0 results in performance degradation in our scenario. This result shows that the tasks that Domain 0 executes to forward packets through virtual machines are single-threaded and that these tasks are not well suitable for multi-core environments. The behavior observed in this test is independent of the number of virtual machines concurrently forwarding packets.

B. Cap Adjustment

Credit Scheduler parameters can be used to offer different service levels to virtual routers. In our case, we use the cap to show how packet rate can be adjusted when the system administrator has control of CPU resource sharing among virtual routers. The experiment consists in sending packets from a TG to TR at a fixed rate through each virtual router in TF. In this experiment, TF has two virtual routers with each one having its own network interface pair and sharing the same CPU core. Thus, we ensure that the main resource that the virtual routers are sharing is the CPU. Domain 0 in this case has its own CPU core.

In this experiment we assign a fixed cap of 100 to a virtual machine (VM2) and vary the cap of the other virtual machine (VM1). Results show that the performance of VM1 decreases when its cap is reduced whereas VM2 performance increases. Thus, we show that we can assure certain performance level to each virtual router, merely adjusting the cap. Our future work aims at building a mechanism to dynamically adjust caps depending on each Domain performance level.

References

[1] 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 SOSP*, Oct. 2003, pp. 164–177.

Acknowledgment

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