Computer Communications 35 (2012) 309–319

Contents lists available at SciVerse ScienceDirect

Computer Communications

journal homepage: www.elsevier.com/locate/comcom

Packet aggregation based network I/O virtualization for cloud computing

Manel Bourguiba *, Kamel Haddadou, Guy Pujolle

LIP6, Pierre & Marie Curie University, 4 Place Jussieu, 75005 Paris, France

ARTICLE INFO

Article history:
Received 27 April 2011
Received in revised form 4 October 2011
Accepted 6 October 2011
Available online 17 October 2011

Keywords:
Cloud computing
I/O virtualization
Driver domain
Networking performance
Xen

ABSTRACT

Virtualization is a key technology to enable cloud computing. Driver domain based model for network virtualization offers isolation and high levels of flexibility. However, it suffers from poor performance and lacks scalability. In this paper, we evaluate networking performance of virtual machines within Xen. The I/O channel transferring packets between the driver domain and the virtual machines is shown to be the bottleneck. To overcome this limitation, we proposed a packet aggregation based mechanism to transfer packets from the driver domain to the virtual machines. Packet aggregation, combined with an efficient core allocation, allows virtual machines throughput to scale up by 700%, while minimizing both memory and CPU consumption. Besides, aggregation impact on packets delay and jitter remains acceptable. Hence, the proposed I/O virtualization model satisfies infrastructure providers to offer Cloud computing services.

1. Introduction

Cloud computing is a new technology trend that is expected to reshape the information technology landscape. It is a way to deliver software, infrastructure and platforms as a service to remote customers over the Internet. Cloud computing reduces the cost of managing hardware and software resources by shifting the location of the infrastructure to the network. It offers high availability, scalability and cost-effectiveness since it is particularly associated with the provision of computing resources on-demand and with a pay-as-you-use model.

These resources are kept on the provider's servers which are located in various parts of the Internet. Their management is then shifted from the user to the provider. Cloud computing refers to both the applications delivered as services over the Internet and the hardware and systems software in the data centers that provide those services [12]. Along this paper we will use the terms "the cloud" and "clouds" interchangeably to refer to all the network-based services that run outside an organization, and the term "a cloud" and "cloud" to refer to a single cohesive service based in the cloud, while a cloud platform is used to refer to the data-centers hardware and software that provide these services. The cloud is also defined in [1] as a large pool of easily usable and accessible virtualized resources (such as hardware, development platforms and/or services). These resources can be dynamically reconfigured to adjust to a variable load (scale), allowing also for an optimum resource utilization. This pool of resources is typically exploited by a pay-as-you-use model in which guarantees are offered by the infrastructure provider by means of customized SLAs.

This definition introduces the virtualization as a key enabling technology for cloud computing. Advances in system virtualization make infrastructure-as-a-service a compelling paradigm. In fact, virtualization basically allows partitioning one physical machine to multiple virtual instances running concurrently and sharing the same physical resources. Virtualization is an enabling technology of cloud computing since it ensures cost effectiveness through resources sharing. It also offers flexibility through the ability of migrating virtual machines from one physical machine to another which helps reducing energy consumption. Furthermore, it enhances the cloud platform scalability and availability through the instantiation of new isolated virtual instances on demand.

Virtualization has been widely studied and deployed in recent years [28,29,31]. The virtual machine monitor (VMM), also called hypervisor is a software layer that presents abstractions of the underlying physical resources. It allows the different virtual machines to share the physical resources including the network device. Network I/O virtualization is essential to provide connectivity to the virtual machines. However, current implementation of VMMs [2,30] does not provide high enough throughputs especially when the applications running on the different virtual machines within the same physical machine are I/O intensive (web services, video servers, etc.) [5,6,15]. Network intensive applications are among the applications dominating the cloud-based data centers today [34].

In this paper, we focus on performance evaluation and analysis of network I/O applications in a virtualized cloud platform. The main contributions of this paper are summarized as follows: First

* Corresponding author. Tel.: +33 1 44 27 61 88; fax: +33 1 44 27 87 83.
E-mail addresses: manel.bourguiba@lip6.fr (M. Bourguiba), kamel.haddadou@lip6.fr (K. Haddadou), guy.pujolle@lip6.fr (G. Pujolle).

0140-3664/$ - see front matter © 2011 Elsevier B.V. All rights reserved.
doi:10.1016/j.comcom.2011.10.002

we evaluate the reception, transmission and forwarding performance of virtual machines in a virtualized cloud platform where the access to the network device is shared through the driver domain. Then we analyze the virtual machines performance limitations through profiling CPU and memory usage and tracking down the bottleneck. We show that the memory latency is behind the drastic degradation of the virtual machines networking performance. In order to make this performance scale up, we propose a packet aggregation mechanism to transfer packets between the driver domain and the virtual machines. We show through experimental evaluation that the proposed mechanism significantly improves virtual machines networking performance. We further investigate the core allocation impact on the achieved throughput in multi-core systems and study the aggregation mechanism impact on packets delay. We finally show that our proposal results in less per-byte memory transactions and CPU cycles consumption.

The remainder of this paper is organized as follows: In Section 2, we present an overview of virtualized cloud platforms, and the virtualization technology we adopt. Section 3 introduces some works dealing with I/O virtualization in cloud platforms. We evaluate and analyze the networking performance of virtual machines in Section 4. Then we propose a packet aggregation mechanism to make the virtual machines throughput scale up in Section 5. Section 6 concludes the paper and introduces future work.

2. Overview and background

2.1. Cloud computing environment

Cloud computing refers to both the applications delivered as services over the Internet and the hardware and systems software in the data centers that provide those services[12]. Services can be: Software (SaaS), Platform (PaaS), or Infrastructure (IaaS). The distinction is made based on the level of abstraction and the level of management of resources presented to the client. To better understand the performance limitations of a cloud computing infrastructure, we first need to understand how cloud platforms are designed. A cloud platform basically consists in multiple data centers with a web portal, connected through a WAN. The data center is composed of multiple physical nodes connected through a LAN. Inside the data center, the infrastructure can be virtualized (Fig. 1) in which case each physical machine supports multiple isolated virtual machines. These virtual machines share the same hardware and storage, and can be migrated from one physical machine to another in the same data center or even in a remote data center. Different applications (game server, media server..) run over these virtual machines and users have direct access to those applications through the web portal. The user can either have access to only the application (SaaS), or the development platform (PaaS) or the whole machine (IaaS).

2.2. Virtualization technology

Virtualization allows multiple virtual machines to run concurrently on the same physical machine. Most deployed virtualization technologies include Xen, VMware, OpenVZ, and Linux VServer. OpenVZ and LinuxVServer offer operating system (OS) level virtualization referred to also as container-based virtualization, where the OS supports multiple isolated user-space instances called containers. They share the same kernel. VMware and Xen fall into the full and para-virtualization categories respectively. The main difference between these two categories is that the guest OS must be ported in para-virtualization and not in full-virtualization. In both categories, the virtual machine monitor (VMM) is responsible of managing the virtual machines (creating, migrating, scheduling the virtual machines, etc.). Full and para-virtualization offer a high level of flexibility compared to container-based virtualization since it is possible to customize data planes by modifying the network stack in the kernel. This is not possible in container-based virtualization since only the application level is virtualized and all virtual instances share the same kernel.

The VMM also ensures physical resources sharing (CPU, memory, etc.) and provides isolated shared access to the devices through a special virtual machine called driver domain (Fig. 2). The driver domain hosts the devices physical drivers. The driver domain is responsible for protecting I/O access and is trusted to transfer traffic to the appropriate virtual machine. With the driver domain based model, all the virtual machines share the same network interface and the driver domain demultiplexes incoming and multiplexes outgoing traffics. A great level of transparency is hence reached since the guest machines do not have to implement the eventually buggy device drivers. Besides, since all the traffic goes through the driver domain, this latter enjoys more traffic monitoring abilities like admission control or establishing priorities between the flows with regard to their types. Isolation is also achieved since the VMM strongly protects access to the memory space. However, the driver domain-based I/O model performance experiences performance limitations due to the overhead incurred by the communication between the driver domain and the guests. We will further analyze this limitation in Section 4.

Nevertheless, I/O virtualization can also be achieved by direct exclusive access to the device [3]. This model achieves high levels of isolation and good networking performance which only depends on the NIC capacity. However, this limits the system's scalability

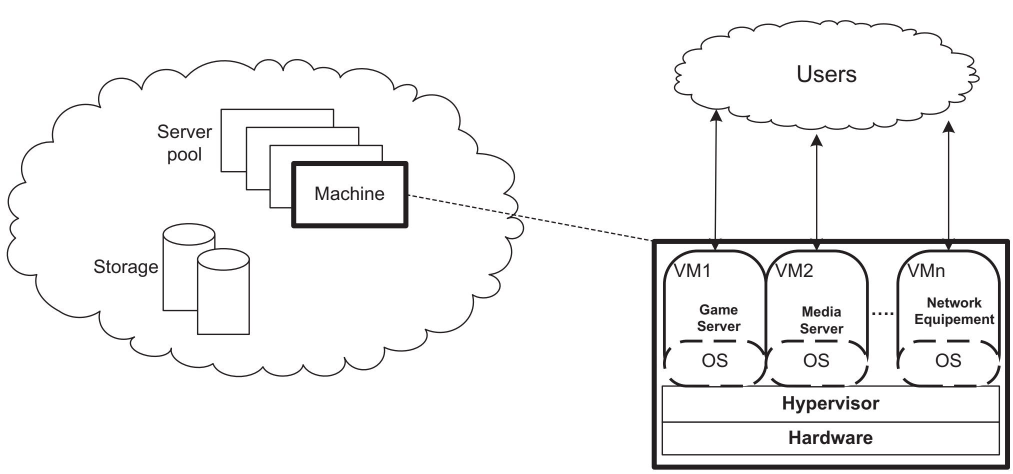

Fig. 1. Virtualized cloud platform.