Towards Improved Provisioning and Utilization of Resources in Virtualized Environments

Thesis

Submitted in partial fulfillment of the requirements
for the degree of

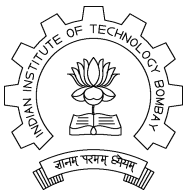
Doctor of Philosophy

by

Sujesha Sudevalayam
Roll No: 07305903

under the guidance of

Prof. Purushottam Kulkarni



Department of Computer Science and Engineering
Indian Institute of Technology, Bombay
Mumbai

Abstract

With widespread adoption of virtualization for hosting applications, service providers (like Amazon EC2 [**?**]) can facilitate better performance isolation, security, quicker deployment and elastic resource provisioning. Due to above benefits of virtualization, many hosting centers have transitioned from providing Hardware as a Service (HaaS) to Infrastructure as a Service (IaaS) instead. The primary difference between HaaS and IaaS is that the former involves use or leasing of physical hardware/machine whereas the latter involves leasing of virtual resources/machines.

When multiple virtual machines (VMs) are placed on a single physical machine (PM), they compete for various resources like CPU, memory, network and disk I/O and interact in many conflicting ways. In any given virtualized environment, the physical resources available can be broadly categorized into, (i) Resources allocated to the virtual machines—virtual CPU, memory, disk and (ii) Resources in the virtualized host—host CPU and cache. In this thesis, we address two important issues related to the management of both these types of resources more efficiently, towards the overall goal of optimizing the performance of virtualized applications.

The first problem of this thesis deals with managing the network usage of VMs and estimating CPU requirement on both the VM and its host PM. Since different tiers of an application require mutual network communication, *colocation* of communicating VMs on the same PM reduces physical network usage. *Network affinity* is the presence of network communication between a pair of VMs, and is *intra-PM* when the VMs are colocated, and *inter-PM* when they are dispersed onto different PMs. Thus, the nature of network affinity is *mutable* (i.e., changing between inter-PM and intra-PM) upon VM migrations. We make the case that since there is significant change in CPU resource usage when the VMs are colocated versus when they are dispersed, it is essential to capture such changes via a model, for server consolidation and VM placement decisions.

In our work, we explore the difference in CPU utilization due to *network affinity*, and propose models to estimate the changed CPU utilization. Specifically, we perform network benchmarking, which demonstrates effects of network affinity on CPU usage when VMs are colocated versus dispersed. Next, we develop VM *pair-wise* models that can estimate the "colocated" CPU usage, on being input their individual dispersed-case resource usages. We also build similar models to estimate the "dispersed" CPU usage based on the individual colocated-case resource usages. For the "colocation" and "dispersion" models, we first built models that predicted the total CPU usage upon migration—these CPU models use all resource (CPU, disk, mutable and immutable network) usage profiles as their input. However, these models had an error of around 4%. So, next we built enhanced models to predict only the difference in CPU usage—these models use only the *mutable* network traffic metrics as input, and have maximum error within 2%.

Finally, we demonstrated the application of *pair-wise* models to predict for multi-VM scenarios, with high accuracy.

The second problem of this thesis deals with managing the cache usage on a virtualized host to improve disk access performance of VMs. Due to increased permeation of virtualization-based systems, there is a lot of inherent content similarity in systems like email servers, web servers and file servers. Harnessing content similarity can help avoid duplicate disk I/O requests that fetch the same content repeatedly. In this work, we incorporate intelligent I/O redirection within the storage virtualization engine of the device to manage the underlying sector-based cache like a content-based cache.

We build a disk read-access optimization called DRIVE, that identifies content similarity across multiple blocks, and performs hint-based read I/O redirection. A metadata store is maintained and implicit caching hints are collected based on the VM's disk accesses. Using the hints, read I/O redirection is performed from within the VM's virtual block device, to manipulate the entire host-cache as a content-deduplicated cache. Our trace-based evaluation using a custom simulator, reveals that DRIVE achieves up to 20% better cache-hit ratios and reduces up to 80% disk reads. It also achieves up to 97% content deduplication in the host-cache.

Contents

List of Figures

List of Tables

Listings

List of Abbreviations

SLA: Service Level Agreement

Chapter 1

Introduction

With widespread adoption of virtualization for hosting applications, service providers (like Amazon EC2 [**?**]) can facilitate better performance isolation, security and elastic resource provisioning. A virtualization-based provisioning model is attractive for both providers—multiplex resources among several customers, and clients—*pay-per-use*, use and pay for only as much resource as required. Instances of both *public* [**?**] and *private* Clouds [**?**, **?**] exist, which leverage virtual machines for flexible provisioning.

Several issues, some of which are—mapping of resource requirements from physical to virtual environments [**?**], placement policies for virtual machines [**?**], dynamic resource provisioning [**?**], runtime consolidation and migration [**?**], storage provisioning and access management [**?**] need to be addressed to provision applications in virtual execution environments. Further, these problems need to be addressed in the context of meeting service level agreements (SLAs) and resource guarantees [**?**], and simultaneously maximizing the resource multiplexing potential. *Server consolidation* and *dynamic resource provisioning* [**?**], [**?**], [**?**], [**?**] are virtual machine migration-enabled techniques aimed to reduce provider-side resource sprawl and to address elastic resource requirements, respectively.

Since application demands are expectedly continuously varying, resource requirements will also be correspondingly elastic [**?**]. To support this, virtualization-based services require automated and dynamic resource provisioning [**?**]. Specifically, if a physical machine faces an explosion of resource requirements, one or more of its virtual machines may need to be *migrated* to other physical machines for load balancing [**?**, **?**] and meeting SLA guarantees [**?**]. Thus, *dynamic resource provisioning* is possible by scaling resources [**?**] on the same physical machine (when physical machine has sufficient resources to accommodate increased demands) or by migrating VM to another PM with sufficient resources (when source PM has insufficient resources).

It is widely acknowledged [**?**, **?**] that the average utilization levels in a datacenter is around 20%, that is to say, the peak-to-average utilization ratios are very high. Typically, under periods of high load, a VM may be allocated to a single PM of its own, and when the load falls back