INTERNATIONAL JOURNAL OF
RESEARCH IN COMPUTER
APPLICATIONS AND ROBOTICS
ISSN 2320-7345

RESHAPING THE VIRTUAL MACHINES IN SELF ORGANIZING CLOUDS

M.Pradeep[1], Dr.S.Uma[2]

[1]P.G. Scholar, PG Department of CSE, Hindusthan Institute of Technology, Coimbatore-32
[2] Head of the Department, PG Department of CSE, Hindusthan Institute of Technology, Coimbatore-32s
dce.pradeep@gmail.com

Abstract

Virtualization is a key technology underlying cloud computing platforms, where applications encapsulated within virtual machines are dynamically mapped onto a pool of physical servers. In this paper, we argue that cloud providers can significantly lower operational costs, and improve hosted application performance, by accounting for affinities and conflicts between co-placed virtual machines. The estimated VM size is the basis for allocating resources commensurate with demand. In contrast to the traditional practice of estimating the size of VMs individually, we propose a joint-VM provisioning approach in which multiple VMs are consolidated and provisioned together, based on an estimate of their aggregate capacity needs. This new approach exploits statistical multiplexing among the workload patterns of multiple VMs, i.e., the peaks and valleys in one workload pattern do not necessarily coincide with the others. Thus, the unused resources of a low utilized VM can be borrowed by the other co-located VMs with high utilization. Compared to individual-VM based provisioning, joint-VM provisioning could lead to much higher resource utilization. This paper presents three design modules to enable such a concept in practice. Specifically, a performance constraint describing the capacity need of a VM for achieving a certain level of application performance; an algorithm for estimating the aggregate size of multiplexed VMs; a VM selection algorithm that seeks to find those VM combinations with complementary workload patterns.

Keywords : VM's, Self clouds, Scheduling,

1. Introduction

Cloud platforms face two competing requirements from the perspectives of the cloud provider and the cloud user, respectively. The cloud provider would like to minimize power and cooling costs, which form a large portion of their operational costs. To achieve this goal, server consolidation can be used to minimize the number of physical servers required for hosting a set of applications. However, consolidation is often undesirable for cloud users, who are seeking maximum performance and reliability from their applications. Under provisioning of physical resources to applications may indirectly increase costs if frequent SLA violations result in lost business1. The key factor that determines the provisioning requirements of a virtual machine is its physical footprint: the amount of physical resources it consumes in terms of CPU time, storage (memory and disk), network bandwidth, and power. Existing consolidation and dynamic placement techniques have largely treated the physical footprint of a virtual machine as a location-independent measure. That is, it is generally assumed that the footprint of a VM will be the same regardless of which physical machine it is placed on. This assumption is reasonable for a homogeneous environment, where physical machines are identical and most VMs are running the same OS and applications.

However, in a cloud environment, we expect a diverse collection of applications to share a resource pool composed of heterogeneous resources. Applications will vary significantly in terms of resource requirements, access patterns and inter-dependencies; and the physical hosts will also vary in terms of their resource capacities and data affinities.

In such an environment, we contend that the physical footprint of a VM is highly malleable; that is, a VM's physical resource consumption is heavily dependent on its location, the characteristics of its physical host, and its interactions with other VMs. A key factor for achieving economies of scale in a compute cloud is resource provisioning, which refers to allocating resources to VMs to match their workload. Typically, efficient provisioning is achieved by two operations: static resource provisioning. VMs are created with specified size and then consolidated onto a set of physical servers. The VM capacity does not change; and dynamic resource provisioning. VM capacity is dynamically adjusted to match workload fluctuations. Static provisioning often applies to the initial stage of capacity planning. It is usually conducted in offline and occurs on monthly or seasonal timescales. Such provisioning functionality has been included in much commercial cloud management software. In both static and dynamic provisioning, VM sizing is perhaps the most vital step. VM sizing refers to the estimation of the amount of resources that should be allocated to a VM. The objective of VM sizing is to ensure that VM capacity is commensurate with the workload. While over-provisioning wastes costly resources, under-provisioning degrades application performance and may lose customers.

Traditionally, VM sizing is done on a VM-by-VM basis, i.e., each VM has an estimated size based on its workload pattern. In a significant departure from such an individual-VM based approach, we advocate a joint-VM provisioning approach in which multiple VMs are consolidated and provisioned based on an estimate of their aggregate capacity needs. Conceptually, joint-VM provisioning exploits statistical multiplexing among the dynamic VM demand characteristics, i.e., the peaks and valleys in one VM's demand do not necessarily coincide with the other VMs. The unused resources of a low utilized VM can then be directed to the other co-located VMs at their peak utilization. Thus, VM multiplexing potentially leads to significant capacity saving compared to individual-VM based provisioning. The savings achieved by multiplexing are realized by packing VMs more densely into hardware resources without sacrificing application performance commitment. While this increases the overall consolidation ratio, the additional virtualization overheads associated with scheduling somewhat higher number of VMs is generally minimal as long as the VM footprints fit in the provisioned capacity. The savings with our joint-sizing approach are up to 40% according to our analysis on the utilization data from a production data center.

In this work, we address these questions in detail. Specifically, the primary contributions of this work are:

- We introduce a Service-level-agreement (SLA) model that map application performance requirements to resource demand requirement. We propose a systematic method to estimate the total amount of capacity for provisioning multiplexed VMs. The estimated aggregate capacity ensures that the SLAs for individual VMs are still preserved.
- We present a VM selection algorithm that seeks to find those VMs with the most compatible demand patterns. The identified VM combinations lead to high capacity savings if they are multiplexed and provisioned together.
- We illustrate effective and feasible applications of the proposed technique for capacity planning and for providing resource guarantees via VM reservations. Both applications can be easily employed in existing cloud and virtualization management infrastructures with minimal intrusion and substantial benefits in return.

2. Benefits and Challenges

We envision benefits from both a cloud provider's perspective as well as a cloud user's perspective. From a cloud provider's perspective, reshaping the physical footprint to allow higher consolidation can help reduce hardware, energy and cooling costs. In addition, if such consolidation can be done dynamically without affecting the performance of applications, then maintaining user SLAs and achieving further cost savings in a time dependent manner (e.g., more aggressive consolidation at times of peak demand) can help substantially reduce provider costs and increase their profits. As an example, existing work in memory consolidation has shown that co-placing VMs with similar memory profiles can lead to substantial reduction in server memory usage (about 10-50%). Note that this consolidation can be achieved in a manner that is completely transparent to applications hosted within the VMs.

The VMs running a distributed application can be reshaped to improve the application's performance, without requiring additional resources or support from the cloud provider, and then the user can achieve higher performance at no additional cost. To demonstrate the potential benefit of footprint reshaping for a cloud user, we ran a motivating experiment to show the impact of network footprint reduction on application performance. In this experiment, we used a micro benchmark to measure the time to send a number (1K- 100K) of 100 KB files between two virtual machines hosted on different physical servers, and compared it to the time to transfer the same amount of data between virtual machines co-located on the same server. This experiment was conducted on a pair of identically-configured workstations running Xen 3.3.0, connected via 100 Mbps Fast Ethernet. As expected, our results show that as the quantity of data grows, the disparity between the two placements grows, with a reduction in transfer time of about 82.17s (a 92% savings) for 10K file transfers. Clearly, the benefit of co-placement in this case is highly dependent on the traffic volume and link bandwidth between the physical servers. In another experiment, we found that the makespan of the Intel MPI benchmarks on a 1 Gbps network dropped from 646s, when the 3 VMs participating in the computation were distributed amongst 3 distinct servers, to 195s when the 3 VMs were co-located on the same physical server.

3. Methodology Overview

Our VM multiplexing and joint-VM provisioning approach is composed of three function modules, which collectively capture the necessary steps for defining the multiplexed VMs, and their individual and joint capacity requirements. These three modules include: (1) a general SLA imposed on VM capacity; (2) a joint-VM sizing algorithm that calculates the total capacity needs for multiplexed VMs; and (3) a VM selection algorithm that identifies *compatible* VM combinations for being consolidated and provisioned jointly. Below, we describe how these three modules cooperate within a general resource provisioning framework. Given a set of VMs, the VM selection algorithm identifies VM combinations that achieve high capacity savings if provisioned together. The selection criterion is how complementary the VM demand patterns are. Highly complementary VMs will be grouped together by the selection algorithm. Eventually the selection algorithm partitions VMs into multiple sets. Those VMs in the same set will be consolidated onto the same physical server and thus can be considered as a *super VM*. To provision such a super VM, we first need to calculate its aggregate capacity need. To this end, we introduce a SLA model and a joint-VM sizing algorithm.

The SLA model defines a relation between the VM capacity and the performance level of the applications running on the VM. Moreover, the SLA model makes it convenient to derive a constraint on the super VM capacity simply based on specified SLA for individual VM. Based on the derived constraint and the aggregate workload of the super VM, the joint-VM sizing algorithm proceeds to calculate the super VM's capacity need, which is the minimum amount of resources that should be allocated to the super VM without degrading individual VM's SLA. We apply such a VM multiplexing approach and demonstrate its benefits in two cloud management operations. The first application is VM consolidation. We identify compatible VMs, provision them jointly in a compute cloud and significantly reduce hardware requirement. Second, we define a *joint reservation* model to provide VM-level resource guarantees in a virtualized environment. By identifying compatible VMs and their SLA, we are able to derive a joint reservation level based on individual VM reservations. We group compatible VMs in resource pools, and enforce joint reservations at the resource pool level. All of these enable dramatically improved VM consolidation ratio in the cloud.

3.1 VM consolidation

VM consolidation is performed when a VM controller needs to create and deploy a set of VMs on a set of physical hosts. The goal of VM consolidation is to determine a mapping of VMs to physical hosts such that the minimum number of hosts is used. Existing VM consolidation schemes consist of two steps: estimating the future size for each VM, and placing VMs on physical hosts. The first step, estimating VM size, is usually solved by first forecasting the future workload, then finding a capacity size that can sufficiently cover the forecasted workload. The second step, VM placement, usually requires solving a bin packing type of problem. Specifically, since each VM carries a size and each physical host has fixed capacity, the VM placement problem is equivalent to packing items (VMs) into the smallest number of bins (hosts) without violating the size limit on each bin. In practice, VM placement is tackled by either heuristics or solving an integer programming problem. By exploiting VM multiplexing, it is possible to achieve even more compact consolidation. The only necessary change is to replace the first step in the above procedure with the proposed three building blocks. Briefly speaking, the VM controller first applies the proposed SLA model to describe the performance requirement for each VM. It then runs the VM selection algorithm to partition VMs into VM groups. For each VM group, the joint-VM sizing algorithm is employed to determine the capacity being allocated.

3.1. 1 Reshaping for System-wide Goals

A control system for cloud platforms can utilize these principles to achieve higher measures of system-wide objectives such as power-savings and performance, through affinity-aware intelligent placement and migration. Given a set of virtual footprints, identifying an “op-timal” placement across multiple resource dimensions is an instance of the bin packing problem, which is known to be NP-hard. Many placement and migration algorithms have been developed to solve this problem in the context of load-balancing. These algorithms would have to be tailored to incorporate the footprint reshaping principles described above, for instance, by including affinity/conflict information as additional constraints to the optimization problem. Alternatively, these principles can guide heuristics to enhance the initial solutions obtained from these algorithms. While existing algorithms coupled with the principles described above provide guidance for reducing the physical footprint along individual resource dimensions, reshaping the VM along one resource dimension can have adverse consequences along other resource dimensions.

Furthermore, the optimal placement is dependent on higher-level system policies, which specify requirements in terms of potentially conflicting goals such as power-savings, performance and reliability. For example, a public cloud provider may desire to minimize hosted VMs' consumption of power, bandwidth and memory, even if it comes at the expense of increased CPU time and disk space (which the user is paying for). In contrast, a private cloud, featuring an abundance of hardware and fast interconnects, might seek to minimize power consumption and execution time, regardless of bandwidth, memory and disk utilization. To address these challenges, the footprint shaping algorithms will need to incorporate optimization techniques which account for inter-resource trade-offs and conflicting objectives. Finally, in a large-scale system, control decisions would likely need to be performed in a decentralized manner.

4. Conclusion

This paper advocates leveraging VM multiplexing to improve resource utilization in compute clouds. The benefit of VM multiplexing is that when the peaks and troughs in multiple VMs are temporally unaligned, these VMs can be consolidated and provisioned together to save capacity. This paper presents three design modules that enable the concept in practice. Specifically, a new SLA model reflects application performance requirements; a joint-VM sizing technique that estimates the aggregate capacity needs for multiplexed VMs; and a VM selection algorithm for identifying most compatible VM combinations. The proposed design modules can be seamlessly plugged into existing resource provisioning applications. VM multiplexing is evaluated with two example applications: VM capacity planning and providing VM resource guarantees via reservations. Experiments based on data from an operational cloud demonstrate that the proposed joint-VM provisioning significantly outperforms traditional approaches. We showed how cloud providers and users can benefit from increased consolidation and application performance by dynamically reshaping virtual machines' physical resource consumption through intelligent placement and migration. We described how to identify opportunities for reshaping by characterizing VMs in terms of a location-independent virtual footprint, and then presented three general principles for minimizing a virtual machine's physical footprint with respect to memory, network, and disk and power consumption. Finally, we discussed challenges associated with applying these principles in practice in a cloud environment.

REFERENCES

[1] J.E. Smith and R. Nair, Virtual Machines: Versatile Platforms for Systems And Processes. Morgan Kaufmann, 2005.

[2] D. Gupta, L. Cherkasova, R. Gardner, and A. Vahdat, "Enforcing Performance Isolation across Virtual Machines in Xen," Proc. Seventh ACM/IFIP/USENIX Int'l Conf. Middleware (Middleware '06), pp. 342-362, 2006.

[3] M. Armbrust, A. Fox, R. Griffith, A.D. Joseph, R. Katz, A. Konwinski, G. Lee, D.A. Patterson, A. Rabkin, I. Stoica, and M. Zaharia, "Above the Clouds: A Berkeley View of Cloud Computing," Technical Report UCB/EECS-2009-28, Feb. 2009.

[4] D.P. Anderson, "Boinc: A System for Public-Resource Computing and Storage," Proc. IEEE/ACM Fifth Int'l Workshop Grid Computing, pp. 4-10, 2004.

[5] P. Crescenzi and V. Kann, A Compendium of NP Optimization Problems. ftp://ftp.nada.kth.se/Theory/Viggo-Kann/compendium.pdf, 2012.

[6] O. Sinnen, Task Scheduling for Parallel Systems, Wiley Series on Parallel and Distributed Computing. Wiley-Interscience, 2007.

[7] O.H. Ibarra and C.E. Kim, “Heuristic Algorithms for Scheduling Independent Tasks on Nonidentical Processors,” J. ACM, vol. 24, pp. 280-289, Apr. 1977.

[8] X. Meng et al., “Efficient Resource Provisioning in Compute Clouds via vm Multiplexing,” Proc. IEEE Seventh Int’l Conf. Autonomic Computing (ICAC ’10), pp. 11-20, 2010.

[9] J. Sonneck and A. Chandra, “Virtual Putty: Reshaping the Physical Footprint of Virtual Machines,” Proc. Int’l HotCloud Workshop in Conjunction with USENIX Ann. Technical Conf., 2009.

[10] D. Gupta et al., “Difference Engine: Harnessing Memory Redundancy in Virtual Machines,” Proc. Eighth Int’l USENIX Symp. Operating Systems Design and Implementation, pp. 309-322, 2008.