

Virtual Putty: Reshaping the Physical Footprint of Virtual Machines

Jason Sonnek and Abhishek Chandra
Department of Computer Science and Engineering
University of Minnesota
Minneapolis, MN 55455
{sonnek,chandra}@cs.umn.edu

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. We show how these affinities can be inferred using location-independent VM characterizations called *virtual footprints*, and then show how these virtual footprints can be used to *reshape the physical footprint* of a VM—its physical resource consumption—to achieve higher VM consolidation and application performance in a cloud environment. We also identify three general principles for minimizing a virtual machine’s physical footprint, and discuss challenges in applying these principles in practice.

1 Introduction

Virtualization is a key technology underlying cloud computing platforms [5, 12], where applications encapsulated within virtual machines are dynamically mapped onto a pool of physical servers. Virtual machines provide several benefits in a cloud computing environment, including increased physical resource utilization via resource multiplexing, as well as flexibility and easy scale-up/scale-down through migration and fast restarts.

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 [10]. To achieve this goal, server consolidation [2, 16] 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 perfor-

mance and reliability from their applications. Underprovisioning 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 [17, 19, 11]. 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. By accounting for affinities and conflicts with physical hosts and other VMs, we argue that *the physical footprint of virtual machine can be reshaped* to achieve higher application performance and consolidation (Figure 1).

For instance, if two virtual machines have a lot of common memory pages, then their physical memory footprints could be reduced significantly by placing them on the same server and using content-based memory sharing [18, 20, 9] to achieve higher consolidation without

¹These concerns may be overlapping if the cloud provider and user are the same, e.g., for private clouds [12].

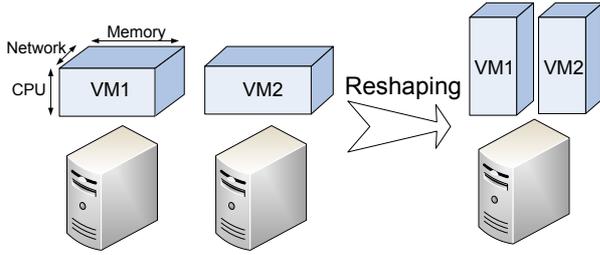


Figure 1: Reshaping the physical footprints of VM1 and VM2 by co-placing them on the same server.

impacting application performance. Similarly, by putting two communicating VMs on the same physical server, we can reduce the network footprint and improve the application performance without reducing the degree of consolidation in the system.

In this paper, we discuss how we can incorporate knowledge of potential affinities and conflicts between VMs to dynamically reshape their physical footprints through intelligent placement and migration. Our goal is to build control systems for cloud environments that can employ such footprint reshaping to achieve higher-level system objectives such as power savings, reliability and performance.

We first discuss potential benefits and challenges of footprint reshaping. Then, we discuss techniques for identifying potential affinities and conflicts between VMs, and present three general principles for minimizing a VM’s physical footprint. Finally, we discuss challenges associated with applying these principles to satisfy system-wide goals in a cloud.

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 [18, 20, 9] 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.

From a cloud user’s perspective, if the physical foot-

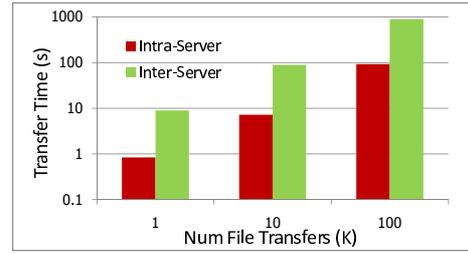


Figure 2: Impact of network footprint reduction on application performance.

prints of VMs running a distributed application can be reshaped to improve the application’s performance, without requiring additional resources or support from the cloud provider, 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 microbenchmark 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 (Figure 2). 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 [13] 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.

Despite the potential benefits shown above, reshaping of VM physical footprints is challenging, because the footprint itself is multi-dimensional, consisting of a different footprint along each resource dimension (CPU, memory, bandwidth, etc.). If two virtual machines are each contending for a shared resource, such as the CPU, a reduction in the physical footprint along the memory or network dimension may come at the cost of an increase along the CPU time dimension, as shown in Figure 1. This implies that while the physical footprint of a VM is elastic, *it need not have a uniform “elasticity” along different dimensions*; compressing the resource footprint along one dimension may result in a non-proportional in-

crease along other dimensions. To effectively employ footprint reshaping in practice, we must estimate the shaping potential of a VM along each resource dimension and the impact of reshaping on other dimensions.

3 Estimating the Virtual Footprint

While the physical footprint of a VM will depend on its location, potential opportunities for exploiting affinities or avoiding conflicts with other VMs and physical servers can be identified by representing a VM in terms of its *virtual footprint*: a location-independent characterization of its resource usage, execution behavior, and data dependencies. For instance, the virtual footprint of a VM may capture its memory contents, so that we can determine which other VMs it can share pages with. Similarly, the virtual footprint of a VM could capture its communication patterns and data dependencies, so that we can determine which other VMs or physical servers exchange maximum traffic with the VM. There are several challenges associated with generating such a virtual footprint for a VM. We illustrate possible solutions to these approaches through a prototype implementation we are building, which is focused on VM network footprints.

Characterizing the virtual footprint: To compile an accurate virtual footprint, we need information about a VM’s resource consumption. Ideally, we’d gather location-independent measures of VM behavior, such as the VM’s working set, file accesses and communication patterns, though location-dependent measures such as the VM’s CPU usage and traffic volume will also play a role. For instance, knowledge of a VM’s file access patterns may lead us to relocate the VM closer to its data. In our prototype, we have focused on characterizing the network behavior of a virtual machine using measures of VM-to-VM traffic volume.

Estimating the virtual footprint: Ideally, a VM’s virtual footprint would be estimated by monitoring its behavior online in a passive, non-invasive manner from within the VMM. This approach has the advantage of being application/OS-agnostic, and doesn’t require a priori planning or instrumentation of the guest operating system. Our current implementation uses VMM-layer monitoring mechanisms to measure the bidirectional traffic for each VM and collect statistics on the volume of traffic between pairs of VMs. However, it is difficult to collect some information in this manner, due to the semantic gap between the VMM and the virtual machines. Specifically, the virtual machine monitor lacks knowledge of OS abstractions such as caches and processes. Recent research has proposed mechanisms for inferring OS state from within the VMM by using gray-box techniques [14]. Other approaches to characterizing VM behavior include instrumentation of the guest OS [19] to

bridge the semantic gap and profiling/modeling of virtual machine workloads [7].

Fingerprinting the virtual machine: The main goal of estimating the virtual footprint of a VM would be to use it to determine potential affinities and conflicts with other VMs and physical servers. As a result, it should be represented as a compact *fingerprint* that can be efficiently exchanged and compared to other VM fingerprints. In our prototype, we represent a virtual footprint as a vector of traffic volume measures for each IP address, which enables us to infer data dependencies easily. Other promising techniques for computing compact fingerprints include representing memory virtual footprints using Bloom filters [20], and we expect that a similar approach could be used to derive disk virtual footprints.

Capturing dynamic virtual footprints: The virtual footprint of a VM is not a static quantity, but changes over time in response to the behavior of applications running inside it. Since a virtual footprint typically represents a snapshot in time, we will need techniques to generate aggregate footprints or to efficiently manage and compare multiple fingerprints per VM. In our prototype, we use exponential averaging of the traffic volumes to track changes in the communication pattern of a VM due to changes in the data dependencies of its applications.

4 Reshaping the Physical Footprint

Given a set of virtual footprints, we first discuss how we can use them to reshape the physical footprints of individual VMs, presenting three principles that apply to different resources. Then, we describe how a control system in the cloud platform can dynamically reshape VM physical footprints, through intelligent placement and migration, to improve performance and consolidation.

4.1 Minimizing Individual VM Footprints

Principle 1: Reduce Data Redundancy

By eliminating redundant copies of data, we can minimize a VM’s footprint along many space-constrained dimensions (memory, disk, etc.). For instance, by scanning memory, the VMM can identify redundant memory pages and reclaim physical memory by mapping each identical VM memory page to a single, copy-on-write physical page [18, 20]. Recent work [9] has proposed condensing the memory footprint even further using sub-page sharing and compression.

The same principle can be applied to minimize local disk consumption. Both VMware and Xen 3.0 support copy-on-write (CoW) filesystem images, allowing virtual machines to share a common base filesystem image. However, utilizing this mechanism requires some

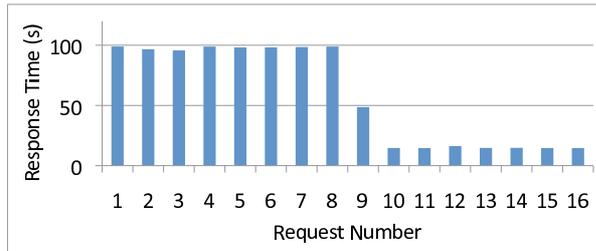


Figure 3: Change in a network service’s response time as its host VM is migrated from a remote server to the physical server holding its data.

amount of foresight, since CoW images must be allocated a priori. Higher levels of physical footprint compression could be achieved by extending the techniques employed in content-based memory sharing to disk blocks. By scanning the disk for shared, infrequently-written blocks, the system could construct a shared CoW image dynamically. The disk footprints can be further compressed by compressing idle blocks and eliminating dead blocks [1].

Principle 2: Place VMs close to their data

A virtual machine’s I/O and network footprint can often be reduced by placing the VM as close as possible to the data required by its applications. If all the data required by a VM is already present on a given physical server, the virtual machine’s physical network footprint can be effectively eliminated by placing the VM on that server. For example, many distributed computing platforms utilize a shared, global data store composed of individual disks attached to each physical server [4, 5]. A virtual hosting platform built on top of such a data store can condense a given VM’s network footprint by placing the VM on or near the physical server hosting its data. This principle is demonstrated by the experiment shown in Figure 3. In this experiment, we ran a simple network service, hosted within a virtual machine, that scans a large data file stored in a distributed file system; Initially, the virtual machine is started on the server which does not store the data file, resulting in high access times. After the eighth request, the virtual machine is migrated to the physical server hosting the data file, resulting in substantially reduced client response times. Note that this improvement in application performance does not require any change in resource provisioning.

In the case that the data is generated by another virtual machine (i.e., co-communicating applications, such as scientific workflow applications), co-placing VMs can improve application performance (as shown for the MPI benchmark in Section 2). It also presents opportunities to employ intelligent CPU scheduling policies to minimize the CPU-time consumed by these virtual machines [8]. When co-placement is impossible or undesirable due to

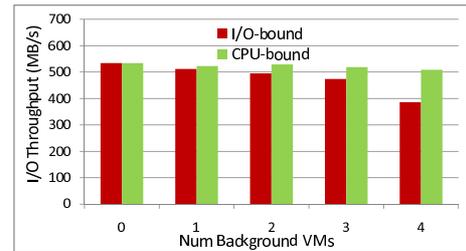


Figure 4: Statistical multiplexing of CPU and I/O-bound VMs provides higher I/O throughput.

conflicts along other resource dimensions, the network footprint can be minimized by ensuring that a VM is hosted on a physical server with a fast link to its data. Virtual hosting platforms deployed in wide-area environments, such as in a distributed Cloud platform [3], will see larger savings for lower data quantities, and thus are even more likely to benefit from this principle.

Principle 3: Exploit Statistical Multiplexing

Higher levels of consolidation can be achieved by attempting to co-place VMs which do not contend for the same resources. For instance, a CPU-bound virtual machine can be placed with an I/O-bound virtual machine. To illustrate the benefit of such multiplexing, we conducted an experiment where we measured the throughput achieved by a VM running dbench (an I/O intensive benchmark) along with a varying number of co-placed Fedora 9 VMs on a quad-core workstation running Xen 3.2.0. These co-placed VMs ran instances of dbench in one case, while they ran a CPU-bound benchmark in the other case. The results, shown in Figure 4, indicate a degradation in throughput due to increased I/O contention in the presence of other I/O-bound VMs, and no degradation in the presence of other CPU-bound VMs.

In some cases, it may be possible to explicitly trade performance for increased energy savings. For instance, by matching virtual machines which can tolerate lower CPU performance with physical servers that support hardware frequency-scaling, it is possible to significantly reduce power consumption. Furthermore, by considering the execution profiles and resource requirements of co-placed VMs, the VMM can schedule virtual machines so that resources such as CPU and disks, which feature low-power states, are idle for as much time as possible [15].

4.2 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 [19, 11, 17]. 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 (e.g., memory) can have adverse consequences along other resource dimensions (e.g., CPU), as demonstrated in Figure 4. 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 (owned by the cloud users themselves), 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 [6]. Finally, in a large-scale system, control decisions would likely need to be performed in a decentralized manner.

5 Conclusion

In this paper, 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, disk and power consumption. Finally, we discussed challenges associated with applying these principles in practice in a cloud environment.

References

- [1] D. Boutcher and A. Chandra. Practical Techniques for Purging Deleted Data Using Liveness Information. *SIGOPS Operating Systems Review*, July 2008.
- [2] J. Chase, D. Anderson, P. Thakar, A. Vahdat, and R. Doyle. Managing Energy and Server Resources in Hosting Centers. In *Proceedings of SOSP*, 2001.
- [3] K. Church, A. Greenberg, and J. Hamilton. On Delivering Embarassingly Distributed Cloud Services. In *Proceedings of the Seventh ACM Workshop on Hot Topics in Networks*, 2008.
- [4] J. Dean and S. Ghemawat. MapReduce: Simplified Data Processing on Large Clusters. In *Proceedings of SOSP*, 2004.
- [5] Amazon Elastic Compute Cloud (EC2). <http://aws.amazon.com/ec2/>.
- [6] M. Ehr Gott and X. Gandibleux. *Multiple Criteria Optimization: State of the Art Annotated Bibliographic Surveys*. Springer, 2002.
- [7] D. Gmach, J. Rolia, L. Cherkasova, and A. Kemper. Capacity Management and Demand Prediction for Next Generation Data Centers. In *Proceedings of IEEE Intl. Conference on Web Services*, 2007.
- [8] S. Govindam, A. Nath, A. Das, B. Urgaonkar, and A. Sivasubramaniam. Xen and co.: Communication-aware cpu scheduling for consolidated xen-based hosting platforms. In *Proceedings of the 3rd Intl. Conference on Virtual Execution Environments*, 2007.
- [9] D. Gupta, S. Lee, M. Vrable, S. Savage, A. C. Snoeren, G. Varghese, G. M. Voelker, and A. Vahdat. Difference Engine: Harnessing Memory Redundancy in Virtual Machines. In *Proceedings of OSDI*, 2008.
- [10] J. Hamilton. Cost of Power in Large-Scale Data Centers, Nov. 2008. <http://perspectives.mvdirona.com/2008/11/28/CostOfPowerInLargeScaleDataCenters.aspx>.
- [11] C. Hyser, B. McKee, R. Gardner, and B. Watson. Autonomic Virtual Machine Placement in the Data Center. Technical Report HPL-2007-189, HP Laboratories, Feb. 2008.
- [12] IBM Cloud Computing. <http://www.ibm.com/ibm/cloud>.
- [13] Intel MPI Benchmarks. <http://www.intel.com/cd/software/products/asm-na/eng/cluster/219847.htm>.
- [14] S. T. Jones, A. Arpaci-Dusseau, and R. Arpaci-Dusseau. Geiger: Monitoring the Buffer Cache in a Virtual Machine Environment. In *Proceedings of ASPLOS*, 2006.
- [15] R. Nathuji and K. Schwan. VirtualPower: Coordinated Power Management in Virtualized Enterprise Systems. In *Proceedings of SOSP*, 2007.
- [16] M. Rosenblum and T. Garfinkel. Virtual machine monitors: Current technology and future trends. *IEEE Computer*, 38(5), 2005.
- [17] A. Singh, M. Korupolu, and D. Mohapatra. Server-Storage Virtualization: Integration and Load Balancing in Data Centers. In *Proceedings of the ACM/IEEE Conference on Supercomputing*, 2008.
- [18] C. A. Waldspurger. Memory resource management in vmware esx server. In *Proceedings of OSDI*, 2002.
- [19] T. Wood, P. Shenoy, A. Venkataramani, and M. Yousif. Black-box and Gray-box Strategies for Virtual Machine Migration. In *Proceedings of NSDI*, 2007.
- [20] T. Wood, G. Tarasuk-Levin, P. Shenoy, P. Desnoyers, E. Cecchet, and M. Corner. Memory Buddies: Exploiting Page Sharing for Smart Colocation in Virtualized Data Centers. In *Proceedings of the 5th ACM Intl. Conference on Virtual Execution Environments*, 2009.