VMShadow: Optimizing The Performance of Virtual Desktops in Distributed Clouds

Tian Guo*, Vijay Gopalakrishnan†, K. K. Ramakrishnan†, Prashant Shenoy*,
Arun Venkataramani*, Seungjoon Lee†
*University of Massachusetts Amherst †AT&T Labs - Research

Abstract

We present VMShadow, a system that automatically optimizes the location and performance of applications based on their dynamic workloads. We prototype VMShadow and demonstrate its efficacy using VM-based desktops in the cloud as an example application. Our experiments on a private cloud as well as the EC2 cloud, using a nested hypervisor, show that VMShadow is able to discriminate between location-sensitive and location-insensitive desktop VMs and judiciously moves only those that will benefit the most from the migration. For example, VMShadow performs transcontinental VM migrations in ~ 4 mins and can improve VNC's video refresh rate by up to 90%.

1 Introduction

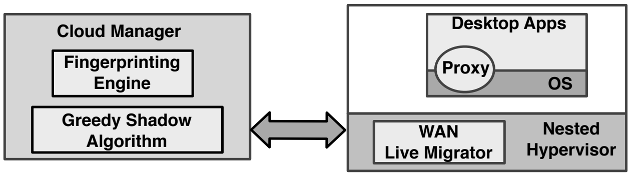

Figure 1: VMShadow Architecture.

Cloud computing has quickly become the paradigm for hosting applications ranging from multi-tier web applications to individuals desktops. Today, users manually determine which cloud location to run these applications based on broad estimates of their computation, bandwidth and latency needs. In this work, we argue that the *cloud platform*, rather than the user, is best suited for automatically determining the best location for hosting each application in a distributed cloud and transparently and seamlessly adjusting the mappings over time as application needs change. Towards this end, we present VMShadow, a system to transparently and dynamically manage the location and performance of virtualized applications in distributed clouds. Using *virtual desktop infrastructure* as an example, we show how VMShadow can optimize the performance of such desktop clouds. Desktop clouds run virtualized desktop machines that are accessed from thin clients using remote desktop protocols. Desktop clouds offer an interesting use-case for VMShadow, since desktops run a diverse set of applications, not all of which are location-sensitive. We prototype VMShadow in a nested hypervisor [2] and evaluate its efficacy using desktop VMs running on a Xen-based private cloud and Amazon's EC2.

2 VMShadow Overview

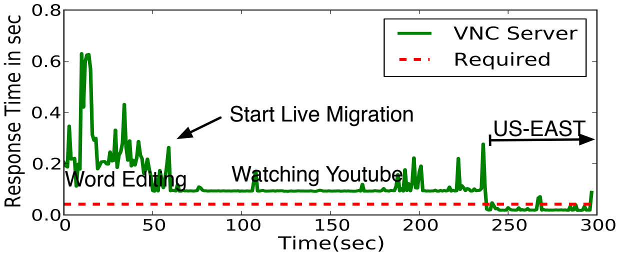

Figure 2: After migration, online streaming achieves lower VNC frame update times, directly improving user experience.

Figure 1 depicts the high level architecture of VMShadow: (i) *Fingerprinting Engine:* By periodically gathering traffic statistics from each VM, It infers location-sensitivity of VMs using port numbers and traffic volume. (ii) *Greedy Algorithm:* It uses cost-benefit metrics to judiciously pick highly location-sensitive VMs at the least cost to move, and triggers the necessary VM migrations. (iii) *Live migrator:* It provides optimized WAN-based live VM migration across different cloud providers using optimization techniques [3] and nested virtualization [2]. It can also limit concurrent live migrations intensity [1] to control the impact on foreground traffic and end-user applications. (iv) *TCP connection migration:* It provides live TCP connection migration to ensure seamless connectivity despite IP address changes with a connection proxy.

We have implemented a prototype of VMShadow using Linux, Xen 4.1.1 and Xen-Blanket [2]. Our prototype is written in C and Python and consists of several interacting components (refer Fig. 1) with a centralized Cloud Manager that manages the placement of VMs. Figure 2 shows a scenario of VMShadow detecting an application switch from word editing to watching YouTube and then triggering a WAN desktop migration, thereby improving the VNC refresh rate.[1]

SOCC '13, Oct 01-03 2013, Santa Clara, CA, USA
ACM 978-1-4503-2428-1/13/10.
http://dx.doi.org/10.1145/2523616.2525950

[1] Note that the VNC player only sends a refresh request after receiving

References

[1] S. Barker, Y. Chi, H. J. Moon, H. Hacigümüş, and P. Shenoy. ”Cut me some slack”: latency-aware live migration for databases. In *Proceedings of Conference on Extending Database Technology*, 2012.

[2] D. Williams, H. Jamjoom, and H. Weatherspoon. The xen-blanket: virtualize once, run everywhere. In *Proceedings of ACM EuroSys*, 2012.

[3] T. Wood, K. K. Ramakrishnan, P. Shenoy, and J. Van der Merwe. CloudNet : Dynamic Pooling of Cloud Resources by Live WAN Migration of Virtual Machines. In *Proceedings of ACM SIGPLAN/SIGOPS conference on Virtual Execution Environments (VEE)*, Mar. 2011.

a response to its previous request. Thus, the lower the response time and round-trip times, the higher the refresh rate.

This research was supported in part by NSF grant CNS-1117221 and a VURI award from AT&T.