Fast Booting Many Similar Virtual Machines

Xiaolin Wang\textsuperscript{1}, Zhenlin Wang\textsuperscript{2}, Shuang Liang\textsuperscript{1}, Zhengyi Zhang\textsuperscript{1}, Yingwei Luo\textsuperscript{1}, and Xiaoming Li\textsuperscript{1}

\textsuperscript{1} Dept. of Computer Science and Technology, Peking University, Beijing, China, 100871
\textsuperscript{2} Dept. of Computer Science, Michigan Technological University, USA
\{wxl,lyw,lxm\}@pku.edu.cn, zlwang@mtu.edu

Abstract. Virtual Machines have been commonly used for server consolidation in data centers, network classrooms, and cloud computing environments. Although booting up a virtual machine takes much less time than booting up a physical computer, booting up multiple virtual machines on a single physical server still takes a lot of time. We propose a method to speed up the booting process when a set of similar virtual machines share a snapshot enabled storage. Our experiments show that the second virtual machine may reduce the booting time by half.

1 Introduction

Virtualization has become a backbone technique in data centers supporting cloud computing \cite{1}\cite{2}\cite{3}. Since a virtual machine runs on the virtualized hardware, normally, booting up a virtual machine takes quite less time than booting up a physical computer, since, during the booting process, a physical computer usually needs to initialize all the hardware devices and wait for their response.

In many situations, the software stack of all virtual machines is quite similar to each other. Figure 1 shows a network classroom application, each terminal connects to a virtual machine, and their operation systems are the same and configurations are similar; they run the same set of applications; and they serve the same kind of users.

To speedup the booting process of those similar virtual machines, one effective way is to setup a base virtual machine on the snapshot-enabled storage. All other virtual machines are cloned from the base one by creating their own snapshots. Thus, all the clones may share the common blocks from the main storage image that remains unchanged. When booting up one virtual machine, shared blocks in the main image are cached via a page cache. As a result, physical disk accesses to the main image by other virtual machines can be greatly reduced.

We propose a novel method based on the snapshot method. Note that, when a virtual machine boots up the guest operation system, most disk blocks fetched to the memory will never be modified as long as the virtual machine is running. We can share such blocks each in a single physical memory page via the CoW (Copy on Write) mechanism. With the sharing, not only the disk accesses will be reduced, but also the total memory requirement is greatly reduced. Moreover, the CPU cache hit rates can be increased.
We implement our method in KVM [4] with Qcow [5], a snapshot-enabled disk image type. In the rest of the paper, section 2 discusses the design issues; section 3 describes the implementation details, section 4 presents the experiment results, and finally, section 5 concludes with some discussions on our future work.

2 Design

When a series of similar virtual machines are booting up, a later one will read many identical disk blocks that the first virtual machine has already read into its memory. The goal of our method is to identify such disk reads and find out correspond memory pages in the first virtual machine, then map those pages to the later virtual machine and mark the page as CoW (Copy-on-Write) [6]. We call the process as booting blocks sharing. We use the Qcow format disk image to support this process. Figure 2 illustrates the two major modifications on KVM.

There are five steps generally to fulfill booting blocks sharing. The first step is to prepare a snapshot enabled virtual disk. We choose the Qcow image file as the virtual disk. The second step is to startup the source virtual machine. Disk blocks read by the source virtual machine can be shared through memory page mapping by those following virtual machines when they startup. The third step is to startup virtual machines that may share disk blocks. We intercept the disk I/O read operations and check whether to map a disk block already read as the response content and return. The fourth step is to process those share memory page so that all virtual machines may run properly. And the last step is to free those shared memory pages when all virtual machines shutdown. We detail each step as following.

![Fig. 1. Network classroom](image-url)