Fast Virtual Machine Resume for Agile Cloud Services

Thomas Knauth, Christof Fetzer (TU Dresden)
The bigger picture

- turn off servers to save energy
- requires turning off and/or migrating services
- activate service when there is work again
- canonical example: web server
  - user does not want to wait, i.e., resume time critical
qemu: state of the art resume

For larger dump sizes the resume time is determined by the memory dump size. We observe two things in Figure 3: it illustrates the time it takes to resume a VM based on the times with suspend/resume than is currently possible. Figure 3 preserved across a suspend/resume cycle.

Besides reducing the startup latency, valuable information is about the working sets of programs running inside the VM. It is sensible to also give back the VM’s RAM belonging to the VM for longer periods of inactivity it is desirable to free up the memory resources. Alternatively, the virtual machine means giving it access to the CPU again. It does however still consume main memory. Unpausing a suspended instance does not consume any CPU or executing the VM anytime in the future. In contrast to a paused virtual machine, the persisted state allows to continue saying hours or days, it is sensible to also give back the VM’s RAM belonging to the VM.

Pausing a VM can, for example, be used to quiesce sources as soon as it is started until it is stopped. Like stopping and application services, the instance actively consumes referring to the resource consumption profile as well as the time it takes of operations with respect to the life cycle management of virtual machines. Starting a VM from scratch involves booting from a virtual machine image, and application services. The “startup cost” is paid every time a new virtual machine image is commissioned.
qemu: state of the art resume

Objective 1
qemu: state of the art resume

Objective 1

Objective 2

The diagram shows the relationship between dump size and resume time for different RAM configurations:
- 1 GB instance RAM configuration:
  - Resume time: 2 seconds
- 2 GB instance RAM configuration:
  - Resume time: 4 seconds
- 4 GB instance RAM configuration:
  - Resume time: 8 seconds

The diagram illustrates that resume time increases linearly with the dump size, and the on-demand resume feature reduces the time until the VM resumes execution drastically from ten seconds and more to 4 seconds.

The appeal of suspending and resuming virtual machines is that it is not only determined by the dump size, but also by the VM's memory configuration. For longer periods of inactivity, it is desirable to free up the memory resources.

For larger dump sizes, the resume time is determined by the memory dump size. We observe two things in Figure 3:
- The startup latency is paid every time a new virtual machine means giving it access to the CPU again.
- The entire memory content is written to disk before terminating the virtual machine.

To preserve the VM's memory state, it is sensible to also give back the VM's RAM belonging to the VM. For longer periods of inactivity, it is desirable to free up the memory resources.

For instance, a suspended instance does not consume any CPU or executing the VM anytime in the future. In contrast to a paused instance, a suspended instance does not have any memory resources.

The key insight is that restarting the virtual machine does not require enumerating all devices and starting various operating system and application services. The "startup cost" is paid every time a new virtual machine is started. However, there is potential to provide even faster startup times with respect to the life cycle management of virtual machines.
How does VM resume work?

• qemu implements eager resume, i.e., read guest’s entire memory image from disk into host’s RAM

• time to resume depends linearly on memory image size

• alternative: lazy resume
How is suspend implemented?

• on suspend, kvm writes a compressed memory image to disk

• single byte pattern page, i.e., zero page, is “compressed”

• suspend happens outside of virtual machine; no knowledge about used/unused memory
What happens on resume?

- allocate memory for guest’s RAM
- sequentially read suspend file and restore guest memory
- memset() compressed pages
Disabling gratuitous memory zeroing helps
How to implement lazy resume with minimal effort?

• Linux offers mmap() system call to read data from disk on demand

• qemu’s default, compressed suspend image not compatible with mmap()
mmap() idiosyncrasies

• MAP_SHARED vs MAP_PRIVATE
  • however, this also determines how updates are propagated to the underlying file
  • on FreeBSD mmap() separates sharing from update propagation, i.e., MAP_NOSYNC
First request latency

![Graph showing the relationship between request latency and suspend image size for stock and custom methods. The graph indicates that request latency increases with larger suspend image sizes.]
Resuming four instances simultaneously
Related work

- Zhang [VEE 2011, ATC 2013]
  - working set estimation and custom on-disk block layout
- VMware’s ESXi
- Twinkle [Infocom 2011]
  - Xen
Demo time

14:51:35 thomas@sedell01 ~/Downloads
% curl http://192.168.122.239/
<title>Welcome to nginx!</title>
</head>
<body bgcolor="white" text="black">
<center>Jeff Dean writes directly in binary. He then writes the source code as documentation for other developers.</center>
</body>
</html>

14:51:38 thomas@sedell01 ~/Downloads
% virsh save dreamserver-test dreamserver-test.virsh
Domain dreamserver-test saved to dreamserver-test.virsh

14:52:03 thomas@sedell01 ~/Downloads
% date; virsh restore dreamserver-test.virsh; curl http://192.168.122.239/; date
Mon Sep 16 14:52:10 CEST 2013
Domain restored from dreamserver-test.virsh
</html>
<title>Welcome to nginx!</title>
</head>
<body bgcolor="white" text="black">
<center>Jeff Dean writes directly in binary. He then writes the source code as documentation for other developers.</center>
</body>
</html>

Mon Sep 16 14:52:11 CEST 2013
14:52:11 thomas@sedell01 ~/Downloads
%
Summary

• resume speed of virtual machine has received little attention so far

• two strategies: eager vs lazy resume

• implemented lazy resume with minimal effort for qemu

• effectively reduces the time to first HTTP answer for a virtualized web server
Open science

- https://bitbucket.org/tknaught/cgc2013