Monitoring and Analyzing Virtual Machine Interference via Host Kernel Tracing

Hani Nemati
Dec 10, 2015

Polytechnique Montréal
Laboratoire DORSAL
Agenda

Motivation

- Why tracing virtual machines?
- What are the challenges in tracing virtual machines?

Investigations

- Background
- Analyzing virtual machine interference via host kernel tracing
  - Resource Monitoring for virtual machines by host kernel tracing
    - Demo
  - Monitoring Processes and threads inside virtual machine with host kernel tracing
    - Demo

Conclusion and in-progress

References
Motivation

Why tracing virtual machines?

- Cloud User:
  - Identify performance anomaly
  - Find the root cause and correct it
  - What if the performance degradation is not from a process inside the VM?

- Cloud Administrator:
  - Identify performance degradation of each VM
  - Contention for shared resources
  - Find the root cause and solve the problem
Motivation

What are the main challenges in tracing virtual machines?
Investigations

Background

• The Paper that uses kernel level Information
  • PerfCompass [1]
    • Global and local anomaly detection tool

• The Papers that uses Hardware level Information
  • Resource contention detection in virtualized Environments [2]
  • DeepDive [3]
    • Identifies when interference occurs and what resources is causing it
    • Using Hardware Performance Counters
  • CPI2 [4]
    • Performance interference detector in shared resources
    • Using CPI and CPU usage
Investigations

Analyzing virtual machine interference via host kernel tracing

● **Interference** is what happens when VMs are concurrently competing for hardware resources
  ● Increase the latency and reduce the performance

● **Goal:**
  ● Identify the interference
    ● By applying Machine Learning
  ● Find the root cause of interference
    ● By finding most frequent events during the interference period
  ● Solve the interference
    ● By categorizing VMs based on workload by using Machine Learning
    ● Migrate the Vms , limitation on resource usage, add more resources
Investigations

Qemu threads in Control Flow View
Investigations

Virtual machine executed into ioctl()

Block I/O? Network? CPU processing?

Userspace?
Investigations

Resource Monitoring for virtual machine by host kernel tracing

• Disk I/O:
  • Using Qemu trace points
    • qemu:thread_pool_submit, qemu:thread_pool_complete, qemu:bdrv_co_io_em

• Network:
  • Using Host Kernel Trace
    • net_if_rx, net_dev_xmit, sched_switch and sched_wakup for vhost-$(VM-main-thread)

• CPU:
  • Using KVM Trace and Host Kernel Trace
    • sched_switch for qemu-system-x86, vm_entry, vm_exit, vcpu_guest_entry
Investigations

Resource Monitoring for virtual machine by host kernel tracing

• CPU Virtualization with intel-VT-x:
  • VMX transition
    • Between Guest and Virtual Machine Manager (VMM)
      • vm_entry, vm_exit
  • Virtual Machine Control Structure (VMCS)

```
vm_exit:
- Save VMCS area for Guest
- Load VMCS area for Host
- Check exit reason

vm_entry:
- Save VMCS area for Host
- Load VMCS area for Guest
```

Qemu Thread  Guest  VMM

<table>
<thead>
<tr>
<th>vCPU</th>
<th>vCPU</th>
<th>vCPU</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Handle exit reason

Time
Investigations

Resource Monitoring for virtual machine by host kernel tracing

Metrics for Disk I/O:

- Number of submitted disk request(s) (read/write) for each or all VM(s)
- How much read/write is submitted by each or all VM(s)
- Duration of completing a disk request
- Variance of submitting disk requests by each VM
- Variance of completing disk requests by each VM
Investigations

Resource Monitoring for virtual machine by host kernel tracing

- Metrics for Network:
  - Number of bytes received and transmitted
  - Variance of submitting network requests by each VM
  - Waiting time to receive/transmit packets
Investigations

Resource Monitoring for virtual machine by host kernel tracing

Metrics for CPU:

- How many vCPUs are active
- The amount of time qemu thread is in root mode and non-root mode
- Most frequent exit reasons and their frequency
- How much pCPU is used by each VM
- Variance of exiting non-root mode
- The amount of time vCPU is preempted in each or all VM(s)
- Variance of preemption for each VM
Investigations

Demo
Investigations

Monitoring Processes and threads inside virtual machine with host kernel tracing

• Guest Processor State is saved into VMCS [6]
  • Guest Register State
    • Control Register CR0, CR3, and CR4
    • Debug Register DR7
    • SP and IP and FLAGS and …
  • Guest non-Register State
    • Activity State
    • Interruptibility State and …
Investigations
Monitoring Processes and threads inside virtual machine with host kernel tracing

• New trace points:
  • `vcpu_enter_guest` to obtain unique tuple (CR3, SP, IP)
  • If you want more information:
    • `lttng_statedump_stack` to obtain thread stack range, thread ID, thread name, hostname
One More Thing ...
Investigations

Monitoring Processes and threads inside virtual machine with host kernel tracing

• What are these exit reasons?
  • Schedule out qemu-system-x86 on Host without sending HLT or Task switch
  • VM assume that the task is still running (Preemption)

vm_exit:
- Handle Interrupts (Internal/external)
- Read/Write MSR
- Handle Exceptions

HLT: VM CPU is Idle
Task Switch: Attempted a Task Switch
Investigations

Demo
Conclusion and in-progress

Inferences

● Fine-grained resource monitoring lets the cloud administrator identify the contention for resources.
● Monitoring processes and associated threads lets the cloud administrator find the amount of time a thread in the VM is actually running.
   - Preemption and too many switches between guest and VMM are causing latency in the VM.

Going Further

● Memory related usage view for each virtual machine by Host Kernel Tracing
● Add some performance indicators from hardware performance counter
● Identifying interference between resources automatically
● Building Control flow view of each virtual machine by Host Kernel Tracing
References


Questions?

Hani.nemati@polymtl.ca
https://github.com/Nemati/Trace-Compass
Demo
Demo
Demo
Questions?

Hani.nemati@polymtl.ca

https://github.com/Nemati/Trace-Compass