Assignment: Extending KVM

Submitted By: Niveditha Bhandary (010820550)

Requirements:

Your assignment is to augment KVM with additional sysadmin-controlled functionality. You will be implementing statistics gathering functionality in the Linux KVM kernel. Each statistic gathered from the
list below will be awarded the indicated number of points, based on properly implementing the item described.

At a high level, you will need to perform the following:
• Configure a Linux KVM host machine using Ubuntu 16.04 (and update to the latest packages via apt-get)
• Download the Linux kernel source code
• Modify KVM in the Linux kernel source code to add the appropriate statistics gathering code
• Add a facility to KVM to print the requested statistics to the system log or console when requested. The output format is up to you but it should be easy to read and understand.
• Recompile the new Linux kernel and install
Next, create one (or several) VMs in KVM, and verify your statistics gathering code is working properly.

Statistics To Be Gathered
(5 points) – total number of exits of each type, for all CPUs
(5 points) – total number of events injected, per CPU
(5 points) – average number of exits per second, per CPU
(5 points) – average latency to process each exit (in cycles), per exit type, per CPU

Solution:

I. How do you obtain the statistics implemented?

1. Apply the git diff patch (cmpe283.diff) to kvm folder using "git apply cmpe283.diff"
2. Build and install the kernel
3. Build the kernel module in the module folder present in <kernel_source>/arch/x86/kvm path
4. Run the get_statistics.sh script from <kernel_source>/arch/x86/kvm/module
5. View the statistics by dmesg

II. Steps used to complete the assignment:

1. Open terminal on Ubuntu 16.04 machine and do all update and upgrades
    * sudo apt-get update
    * sudo apt-get upgrade
2. Clone kernel source code from the following link:
    * git clone git://kernel.ubuntu.com/ubuntu/ubuntu-xenial.git
3. Build the environment:

* sudo apt-get build-dep linux-image-$(uname -r)

4. Make appropriate changes to vmx.c and vmx.h files
5. Make changes in /etc/default/grub, and set a long enough timeout to ensure that you have time to access the new kernel during boot

> sudo edit /etc/default/grub
I used these settings:
GRUB_DEFAULT=0
#GRUB_HIDDEN_TIMEOUT=0
GRUB_HIDDEN_TIMEOUT_QUIET=true
GRUB_TIMEOUT=20
> sudo update-grub

6. Building the kernel

Change your working directory to the root of the kernel source tree and then type the following commands:

* fakeroot debian/rules clean
* fakeroot debian/rules binary-headers binary-generic binary-perarch

After build is successful we see *.deb in previous directory (on doing cd ..)

7. Testing the new kernel

Install the package set with dpkg -i and then reboot:

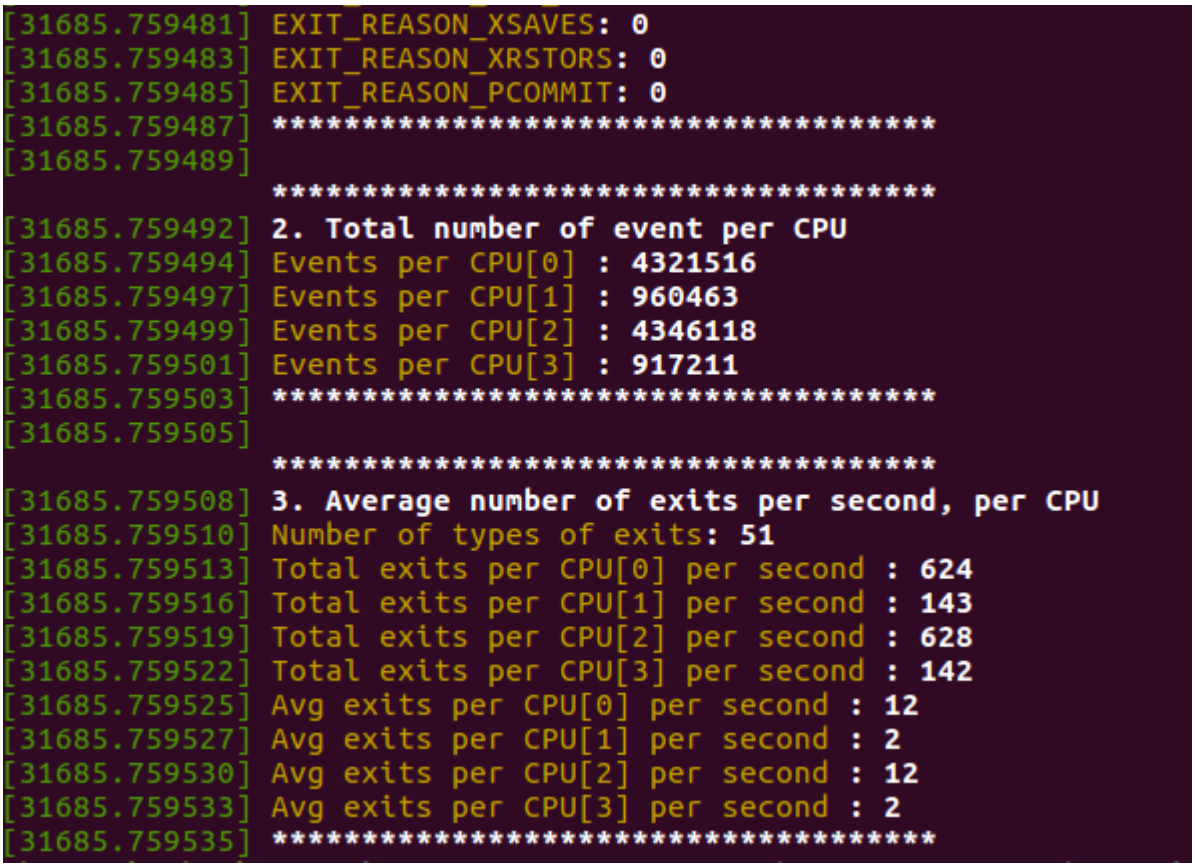
* sudo dpkg -i *.deb
* sudo reboot

8. In the GRUB select the newly created Kernel
9. Build the kernel module
10. The module folder contains the following files:
    * get_statistics.c
    * get_statistics.sh
    * Makefile
11. Run the following commands to build our helper module:
    * make

12. Install Virtual Machine Manager in host OS.
13. Install DSL inside the VM using iso image.
14. Once the Guest VM is running, go back to directory module and run statistics.sh
15. View the statistics by dmesg

III. Sample Output:

```
[                ] ************************************
[31685.759328] CMPE283 bonus assignment
[31685.759330] ************************************
[31685.759332]
[                ] ************************************
[31685.759335] 1. Total number of exits of each type
[31685.759339] EXIT_REASON_EXCEPTION_NMI: 28
[31685.759342] EXIT_REASON_EXTERNAL_INTERRUPT: 170896
[31685.759345] EXIT_REASON_TRIPLE_FAULT: 0
[31685.759348] EXIT_REASON_PENDING_INTERRUPT: 0
[31685.759352] EXIT_REASON_NMI_WINDOW: 74625
[31685.759354] EXIT_REASON_TASK_SWITCH: 0
[31685.759358] EXIT_REASON_CPUID: 556
[31685.759361] EXIT_REASON_HLT: 6730340
[31685.759364] EXIT_REASON_INVD : 0
[31685.759367] EXIT_REASON_INVLPG  : 0
[31685.759369] EXIT_REASON_RDPMC : 0
[31685.759372] EXIT_REASON_RDTSC : 0
[31685.759375] EXIT_REASON_VMCALL: 0
[31685.759378] EXIT_REASON_VMCLEAR: 0
[31685.759381] EXIT_REASON_VMLAUNCH: 0
[31685.759384] EXIT_REASON_VMPTRLD: 0
[31685.759387] EXIT_REASON_VMPTRST: 0
[31685.759389] EXIT_REASON_VMREAD: 0
[31685.759392] EXIT_REASON_VMRESUME: 0
[31685.759394] EXIT_REASON_VMWRITE: 0
[31685.759397] EXIT_REASON_VMOFF: 0
[31685.759399] EXIT_REASON_VMON: 0
[31685.759402] EXIT_REASON_CR_ACCESS: 1155641
[31685.759405] EXIT_REASON_DR_ACCESS: 28504
[31685.759409] EXIT_REASON_IO_INSTRUCTION: 34549618
[31685.759412] EXIT_REASON_MSR_READ: 47
[31685.759415] EXIT_REASON_MSR_WRITE: 93
[31685.759417] EXIT_REASON_INVALID_STATE: 0
[31685.759420] EXIT_REASON_MSR_LOAD_FAIL: 0
[31685.759424] EXIT_REASON_MWAIT_INSTRUCTION: 0
[31685.759426] EXIT_REASON_MONITOR_TRAP_FLAG: 0
[31685.759430] EXIT_REASON_MONITOR_INSTRUCTION: 0
[31685.759433] EXIT_REASON_PAUSE_INSTRUCTION: 0
[31685.759436] EXIT_REASON_MCE_DURING_VMENTRY: 0
[31685.759440] EXIT_REASON_TPR_BELOW_THRESHOLD: 0
[31685.759443] EXIT_REASON_APIC_ACCESS : 3130338
[31685.759445] EXIT_REASON_EOI_INDUCED: 0
[31685.759449] EXIT_REASON_EPT_VIOLATION: 1461181
[31685.759452] EXIT_REASON_EPT_MISCONFIG: 932047
[31685.759456] EXIT_REASON_INVEPT: 0
[31685.759458] EXIT_REASON_RDTSCP: 0
[31685.759461] EXIT_REASON_PREEMPTION_TIMER: 0
[31685.759464] EXIT_REASON_INVVPID: 0
[31685.759467] EXIT_REASON_WBINVD: 4
[31685.759470] EXIT_REASON_XSETBV: 0
[31685.759473] EXIT_REASON_APIC_WRITE: 0
[31685.759476] EXIT_REASON_INVPCID: 0
[31685.759478] EXIT_REASON_PML_FULL: 0
[31685.759481] EXIT_REASON_XSAVES: 0
[31685.759483] EXIT_REASON_XRSTORS: 0
```



References:

1. https://wiki.ubuntu.com/Kernel/BuildYourOwnKernel
2. http://askubuntu.com/questions/679251/switching-back-from-development-to-default-kernel
3. http://lxr.free-electrons.com/source/arch/x86/kvm/vmx.c
4. https://www.kernel.org/