

17-699 Independent Study Proposal

Title: GPU migration in Virtual Environment using log replay

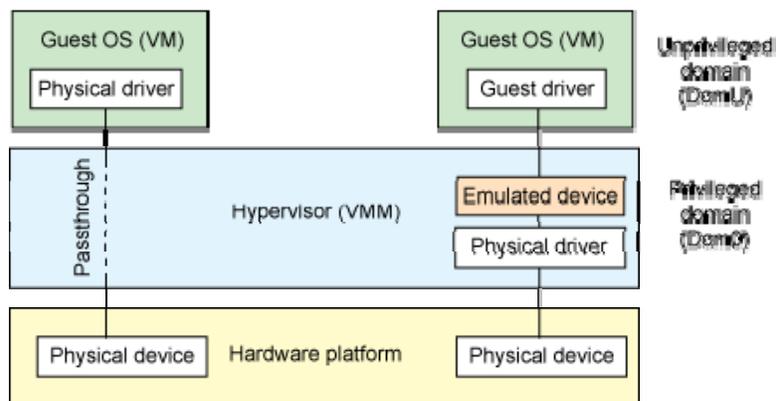
Student: Parth Mehta

Instructor: Mahadev Satyanarayanan

Motivation:

The difficulty of virtualizing graphical processing units (GPUs) has so far limited the use of virtual machines (VMs) for running interactive applications. For this class of applications, software rendering is the prevalent option for virtualized execution, and it is unacceptably slow. Virtualizing GPUs is difficult for a number of reasons details of which are mentioned in [1]. Broadly, the hardware interface to a GPU is proprietary, and keeps constantly changing. The device driver that is shipped with the GPU is closed source rendering it useless for a virtualization environment.

PCI pass-through provides a means to use PCI devices directly from the guest operating system. The figure below shows how pass-through enables the guest OS to directly use PCI devices rather than emulated device.



However when this virtual machine now needs to be migrated to another host platform state information about the GPU is needed. This is difficult for the above mentioned reasons. Thus an approach to migration is to recreate the state of the GPU in another host platform by performing the same operations that were performed on the GPU previously.

Most applications use a standard API like OpenCL or CUDA to use the computing power of the GPU. If logs of these API calls are taken and replayed in another host the GPU state could be recreated.

Goal:

The goal of this independent study is to demonstrate GPU migration in virtualized environment with replay of logs.

Learning Objectives:

- 1 To learn how graphics programming is done using openCL or CUDA to exploit GPU capabilities.
- 2 Analyze one of the OpenCL or CUDA api for taking logs of operations on the GPU

- 3 Explore how the logs can be used to recreate the state of the GPU for application to resume execution on another host platform.
- 4 Explore how the logs can be optimized to minimize the size of the logs.

Method:

- 1 Demonstrate an application that uses CUDA running on the host OS.
- 2 Demonstrate correct working of PCI passthrough and run that same application that uses CUDA in the guest OS.
- 3 Analyze the CUDA or OpenCL API for effective logging of operations and choose one of these API.
- 4 Analyze what kind of logging and at what level can the logging be done to recreate the state on another machine.
- 5 Explore how these logs can be used to recreate the GPU state on another host by replaying the operations that were done on the GPU.
- 6 Demonstrate the VM migration by replaying GPU logs on the guest OS.
- 7 Perform Log optimizations on the logs to speed up the recreation of GPU state and also minimize data transferred while shipping the logs.

Deliverables:

- 1 Report explaining how PCI pass-through can be used for using the GPU in a virtualized environment.
- 2 Sample application where the API calls are logged and GPU state is migrated to another host machine by replaying the logs.
- 3 Report containing analysis of API and rationale behind where and how logging can be done to recreate the state, how state of GPU can be recreated, methods used to obtain these results and optimizations to be performed on the logs.

TimeLine:

Start: 20st May 2013

End Date of this study: 6th August 2013 (Final deliverables due)

Total number of man hours = 12 weeks * 12 hours * 1 student = 144 man hours.

References and Sources:

- [1] <http://www.cs.cmu.edu/~satya/docdir/lagar-cavilla-vee-vmgl-2007.pdf>
- [2] <http://www.ibm.com/developerworks/linux/library/l-pci-passthrough/>
- [3] http://fedoraproject.org/wiki/QA:Testcase_Virtualization_KVM_PCI_Device_Assignment_libvirt_nodedev_operations
- [4] <http://www.khronos.org/opencl/>
- [5] http://www.nvidia.com/object/cuda_home_new.html