

SCHOOL OF COMPUTING, UNIVERSITY OF
UTAH

INDEPENDENT STUDY REPORT

UserSpace Device Access in Linux

Author:
Sriraam APPUSAMY
SUBRAMANIAN

Supervisor:
Robert Ricci
Anton Burtsev

May 8, 2014

Abstract

Operating system allows processes in the system to share the computing, storage and peripheral resources with safety and isolation. Much of the mechanisms involved in achieving the above capabilities trades off performance/latency for flexibility which remains acceptable for most of the day-to-day applications. However, for certain applications like memcached, SDN controller, etc., whose performance is tied to their disk/peripheral device access capabilities, this flexibility is less desirable than performance. The goal of this independent study was to evaluate and explore existing systems that allow for direct peripheral device assignment to applications, exclusive or not. Intel's DataPlane Development Kit (DPDK) is a software library that uses modified userspace drivers, polling, prefetching/pre-allocation of buffers/queues and other enhancements to allow upto 80Mpps packet processing capabilities on Intel Architecture platforms. However DPDK supports restrictive number of NIC types. Intel VT-d is a hardware virtualization technique that provides dedicated I/O Memory management unit and Interrupt remapping capabilities that could allow for exclusive device usage by an application. SUD is an existing system, developed using User-mode Linux(UML) that aims to test malicious device drivers by running them in userspace. Since SUD was implemented in linux kernel 2.6.x, it became necessary to port the project to linux kernel 3.10.x to allow for further testing and benchmarking. Although most of the porting is done, testing still remains incomplete due to failed porting of broadcom driver BNX2 into UMLinux.

1 DPDK

Intel's DataPlane Development Kit aims to improve packet processing performance in Intel platforms. This is achieved by providing applications with customized software library that significantly improves peripheral device access performance. This library provides a simple API interface for buffer management, queue management, poll-mode capable userspace drivers and packet-flow classification. Further several abstraction layers are added to both userspace and kernel-space to help communicate with the peripheral device in traditional Linux-based operating systems. An experiment was setup in Emulab with gpu2 node and 2 Network Interface Cards(NICs) were assigned to DPDK. Following this, a sample application was run to verify DPDK. After a quick analysis of the existing openflow controllers and their performance metrics, NOX was chosen to be modified to support DPDK. However, limited types of hardware supported by DPDK warranted continued exploration for other mechanisms.

2 SUD

SUD is a system that aims to protect system resources against potentially malicious device drivers. This is achieved by moving the device driver out of the kernel-space and using User Mode Linux (UMLinux) to provide access to hardware features. In this system, a safe and generic proxy driver is added for each class of peripheral device. This proxy driver registers itself to the kernel as the device driver for the device of interest. A Message passing mechanism is setup between the proxy driver and the unmodified driver running in the userspace. This bidirectional message passing is supplemented by PCIe emulation and Message Signalling Interrupts(MSI) to provide safe access to devices. Further, SUD relies on reading/writing physical memory via DMA, a capability allowed by IOMMU (Intel or AMD) that supports isolation and memory mapped I/O for certain devices.

3 Porting to 3.x kernel

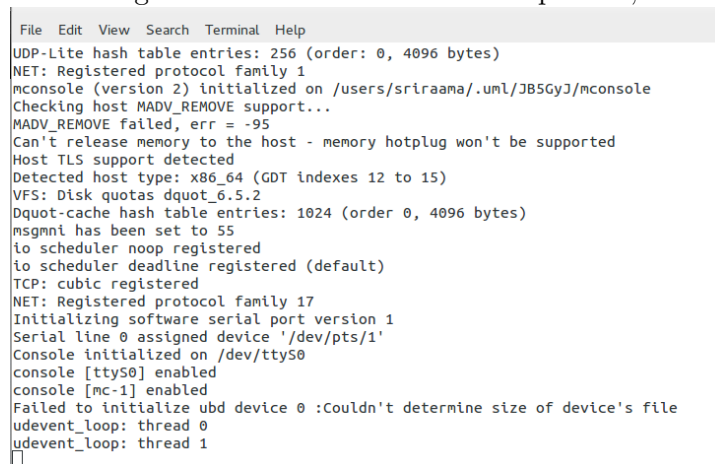
The primary task was to port the implementation of SUD to currently used linux kernel versions i.e. 3.10.x. This task involved modifying the linux kernel to add proxy driver and other patches. The port was performed to a d710 node in Emulab that has 4 Broadcom NetXtreme II BCM5709 Gigabit Ethernet chipsets. Following this, UMLinux was built with PCI emulation and unmodified device driver. Much of the implementation was ported directly to the new kernel with significant number of tweaks for compatibility. Although PCI emulation feature port was direct, recent broadcom device driver port in UMLinux has been so far unsuccessful since the device driver

is more recent than the substrate it integrates with. One approach might be to use an older version of the device driver. However, this remains untested and just a temporary solution.

4 Results

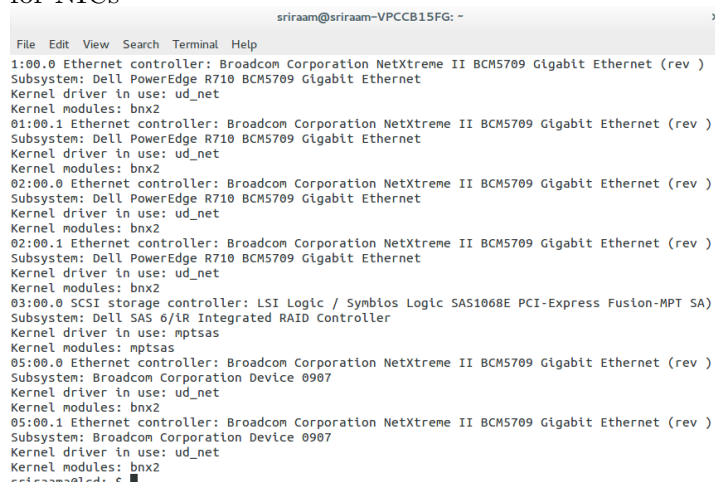
The SUD port was almost nearly complete with proxy driver(`ud_net`) being registered in the host kernel. The UMLinux device driver process was compiled without the necessary `broadcom` device driver and so it could not be thoroughly tested. However the following screenshots show the status,in detail.

Figure 1: UMLinux device driver process, waiting on events



```
File Edit View Search Terminal Help
UDP-Lite hash table entries: 256 (order: 0, 4096 bytes)
NET: Registered protocol family 1
mconsole (version 2) initialized on /users/sriraama/.uml/JB5GyJ/mconsole
Checking host MADV_REMOVE support...
MADV_REMOVE failed, err = -95
Can't release memory to the host - memory hotplug won't be supported
Host TLS support detected
Detected host type: x86_64 (GDT indexes 12 to 15)
VFS: Disk quotas dquot_6.5.2
Dquot-cache hash table entries: 1024 (order 0, 4096 bytes)
msgmni has been set to 55
io scheduler noop registered
io scheduler deadline registered (default)
TCP: cubic registered
NET: Registered protocol family 17
Initializing software serial port version 1
Serial line 0 assigned device '/dev/pts/1'
Console initialized on /dev/ttyS0
console [ttyS0] enabled
console [mc-1] enabled
Failed to initialize ubd device 0 :Couldn't determine size of device's file
udevent_loop: thread 0
udevent_loop: thread 1
```

Figure 2: Host kernel 'lspci -k' showing the `ud_net` proxy driver registered for NICs



```
sriraam@sriraam-VPCCB15FG: ~
File Edit View Search Terminal Help
1:00.0 Ethernet controller: Broadcom Corporation NetXtreme II BCM5709 Gigabit Ethernet (rev )
Subsystem: Dell PowerEdge R710 BCM5709 Gigabit Ethernet
Kernel driver in use: ud_net
Kernel modules: bnix2
01:00.1 Ethernet controller: Broadcom Corporation NetXtreme II BCM5709 Gigabit Ethernet (rev )
Subsystem: Dell PowerEdge R710 BCM5709 Gigabit Ethernet
Kernel driver in use: ud_net
Kernel modules: bnix2
02:00.0 Ethernet controller: Broadcom Corporation NetXtreme II BCM5709 Gigabit Ethernet (rev )
Subsystem: Dell PowerEdge R710 BCM5709 Gigabit Ethernet
Kernel driver in use: ud_net
Kernel modules: bnix2
02:00.1 Ethernet controller: Broadcom Corporation NetXtreme II BCM5709 Gigabit Ethernet (rev )
Subsystem: Dell PowerEdge R710 BCM5709 Gigabit Ethernet
Kernel driver in use: ud_net
Kernel modules: bnix2
03:00.0 SCSI storage controller: LSI Logic / Symbios Logic SAS1068E PCI-Express Fusion-MPT SA)
Subsystem: Dell SAS 6/IR Integrated RAID Controller
Kernel driver in use: mptsas
Kernel modules: mptsas
05:00.0 Ethernet controller: Broadcom Corporation NetXtreme II BCM5709 Gigabit Ethernet (rev )
Subsystem: Broadcom Corporation Device 0907
Kernel driver in use: ud_net
Kernel modules: bnix2
05:00.1 Ethernet controller: Broadcom Corporation NetXtreme II BCM5709 Gigabit Ethernet (rev )
Subsystem: Broadcom Corporation Device 0907
Kernel driver in use: ud_net
Kernel modules: bnix2
```

5 Conclusion

Although porting SUD from older to newer kernel version was exhausting and tedious, it provided great insight into programming IOMMU and device drivers ,proving to be a rich learning experience. In retrospect, it might have been better to build SUD for the recent linux kernel from scratch, with support for newer features might have been a rewarding experience.