# Washington University in St. Louis



#### Performance Analysis of Packet Capture Methods in a 10 Gbps Virtualized Environment

Michael J. Schultz and Patrick Crowley ICCCN 2012 (2012 Aug 02)

#### Multi-core Proliferation





- Core counts continue to rise
  - Cores may idle in distributed workload
  - Move to virtualizing distributed workload
- Still want high-performance networking out of virtual appliances

## Virtualization Overview

- Modern extensions enable 'unprivileged' operation
  - i.e., anything that changes system state is trapped to the host O/S
- This trap-decode-execute routine can take time
  Result is slow I/O handling operations in VM

## Trap-Decode-Execute Example

Based on Intel VT-x extensions

- Control structure (VMCS)
  - Guest processor state
  - Host processor state
  - Control
  - Exit information



#### virtio Device Driver

- Paravirtualization is one attempt to improve performance
  - Both host and guest are virtual-aware



# Machine Configuration

	Bare-Metal	Virtual Machine
Linux Kernel	3.2.1	
Operating System	CentOS 5	
Processor Clock Rate	2.27 GHz	
Number of Cores	8	4
Private Cache Size	256 KiB	64 KiB
Memory	12 GiB	6 GiB
Data Bit Rate	10 Gbps	-
Data NIC	Intel 82599EB	Qumranet Virtio
Data Driver	ixgbe	virtio_net

## Networking Configuration



Washington University in St.Louis

#### Linux Network Bridge



Washington University in St.Louis

## Open Network Laboratory (ONL)



http://onl.wustl.edu/

# Experimental Design



# Testing Methodology

- Build software from source
- Begin logging and load processing software
- Start packet generators (discarding warm up)
- Data collection
- Stop packet generators (discarding cool off)
- Unload processing software and end logging
- Copy log file off system and reboot

Completely Automated

## Bare Metal Throughput



## Virtual Machine Throughput



#### Packet Rates



#### oprofile Breakdown

- oprofile is low overhead (1-3%) performance monitor
  - Shows time processor spends in each function
- When using VM, host spends substantial time
  - Interrupt handling (emulation)
  - Polling (getting packets from physical hardware)
  - I/O functions (moving/remapping memory)

#### Conclusions and Future Work

- Hardware improvements like SR-IOV will help – But software cannot rely on them
- Still need focus on scheduling and interrupt handling
- Revisit past ideas, recognizing paradigm shift from single-O/S to multi-O/S