Vagabond: Dynamic Network Endpoint Reconfiguration in Virtualized Environments

Kallol Dey, Debadatta Mishra, Purushottam Kulkarni
Email: {kallol,deba,puru}@cse.iitb.ac.in

Systems and Network Research Group (SYNERG)
http://www.cse.iitb.ac.in/synerg/

Indian Institute of Technology, Bombay

SoCC 2014
I/O virtualization techniques

Software-based
- Better resource control and management
- Higher overheads
- Example: xen split driver model

Hardware-assisted
- Low overheads
- Inflexible, VMM control compromised
- Example: pci pass-through assignment (SRIOV)
Efficient use of multi-function devices

- Static assignment of VF to VM
  - Inflexible. Once assigned remains with the VM
  - In-efficient—does not allow dynamic re-arrangements
- What is needed?
  - Dynamic assignments—allows assignment and preemption of VFs
  - No service disruption because of preemption
Contributions

- Flexible network device virtualization
  - vagabond—runtime switching between S/W and H/W IO virtualization modes
  - Design and implementation of vagabond in Xen
  - Evaluation and cost-benefit comparison with other solutions

- Usage of vagabond
  - Live migration of VM
  - Resource controller for fairness
- **vagabond** controller enables two paths—software bridge and VF direct
- **software bridge mode** is similar to split-driver mode
- **VF mode** enables a direct path from device to the guest
Direct data path—VF mode

- Modifications to VF driver—packet receive and buffer request
- Receiver thread to batch notifications to guest VM
- Mapper thread maintains mapped guest page pool
Guest VM grants are mapped to dom-0 address space
- Mapped pages are assigned to VF receive descriptors—single DMA copy on receive
- Granted pages unmapped after receive notification to guest
CPU utilization is ~40% and ~85% less for UDP send and receive with vagabond VF mode (compared to split-driver)

Least CPU overhead in VF direct assignment — vagabond VF is the second best with 18% more CPU in UDP receive
Impact on throughput during switch is negligible

In VF mode, dom-0 CPU utilization reduced by 50%

In VF mode, dom-U overhead is less by ~20%
Migration is initiated at 90 sec when VM is in **VF** mode

vagabond controller switches the VM to **bridge** mode before stop-and-copy
Conclusions

- **vagabond** provides flexible network endpoint assignment with minimal software intervention in hardware mode.
- The CPU overheads of **vagabond** is up to 2x lower than split driver approach.
- Compared to VMM bypass mode, 17% more CPU required in the best case.
- Future work—Alternate network packet data path, coalescing to reduce CPU overheads.

Thank you.

www.cse.iitb.ac.in/synerg