VNET/P: Bridging the Cloud and High Performance Computing Through Fast Overlay Networking

Lei Xia, Zheng Cui, John Lange, Yuan Tang, Peter Dinda, Patrick Bridges

Northwestern University
University of New Mexico
University of Pittsburgh
University of Electronic Science and Technology of China

http://v3vee.org
Overview

• **Motivation**: Bridging the cloud and HPC resources through virtual networking for HPC applications
  – Current virtual networking performance is NOT sufficient

• Design and optimization of **VNET/P**, a fast virtual overlay networking for such model
  – Applicable to other VMMs and virtual network systems

• Performance evaluation of VNET/P
  – Native/Near-native performance on 1Gbps/10Gbps networks

• *Possible* to extend software-based overlay networks into tightly-coupled environments
Outline

• Model and motivation
• VNET/P: design & optimization
• Performance evaluation
• Conclusions and future work
VNET Model

• A layer 2 virtual overlay network for the user’s virtual machines
  – Provide location independence to VMs
  – Carry VMs’ traffic via configurable overlay network
• Virtual machines on virtual networks as the abstraction for computing
• Virtual network as a fundamental layer for measurement and adaptation
  – Monitor application communication/computation behavior
  – Adaptive and autonomic mapping of virtual components to physical resources

A. Sundararaj, A. Gupta, P. Dinda, Increasing Application Performance In Virtual Environments Through Run-time Inference and Adaptation, HPDC’05
Bridging the Cloud and HPC

• Hosting HPC applications in VMs is possible
  – Low overhead in CPU/memory virtualization
• Extend *virtual overlay network* from loosely-coupled environments to *tightly-coupled* environments
• Seamlessly bridge the cloud and HPC resources
  – Applications can dynamically span to additional cloud resources
  – Virtual networking provides connectivity and mobility

• Performance of virtual overlay network is critical
  – How can it provide high performance inter-VM traffic while VMs are located on the *same* data center/cluster?
VNET/U

- VNET implemented at user-level
  - Among the fastest user-level overlay systems
    (78MB/s, 0.98ms)
  - Sufficient for wide-area/loosely-coupled applications
  - Throughput/latency limited by kernel/user transitions
  - Not sufficient for tightly-coupled applications running on cluster/supercomputer with gigabit or 10 gigabit networks

J. Lange, P. Dinda, *Transparent Network Services via a Virtual Traffic Layer for Virtual Machines*, HPDC’07
VNET/P

• High performance virtual overlay network
  – Targeting for HPC applications in clusters and supercomputers with high performance networks
  – Also applicable to data centers that support IaaS cloud computing

• High level approach
  – Move virtual networking directly into VMM
  – Enable optimizations that can only happen inside VMM
Outline

• Model and motivation
• VNET/P: design & optimization
• Performance evaluation
• Conclusions and future work
Palacios VMM

• OS-independent embeddable virtual machine monitor
• Open source and freely available
• Host OS: Linux, Kitten, Minix …
• Successfully used on supercomputers, clusters (Infiniband and Ethernet), and servers
  • VNET/P is in Palacios code base and is publicly available
  • Techniques general applicable to other VMMs

Palacios
An OS Independent Embeddable VMM
http://www.v3vee.org/palacios
VNET/P Architecture
Data Path
(packet transmission)

Guest
- TCP/IP stack
- Device Driver

VM Exit
- vNIC Exit Handler
- VNET/P Core: Routing/encapsulation
- VNET/P Bridge: Send packet to host network

VM Entry
- Device Driver

Palacios

VNET/P
- Ether hdr
- IP hdr
- TCP/UDP hdr
- Data

Time
Transmission/Reception Modes

**Guest-driven:** Enhance latency

![Diagram showing transmission/reception modes in a guest-driven scenario.]

**VMM-driven:** Enhance throughput/Reduce CPU cost

![Diagram showing transmission/reception modes in a VMM-driven scenario.]

Guest

Palacios+VNET/P
Dynamic Mode Switching

• VNET/P switches between two modes dynamically
  – Depends on the arrival rate of packets from VMs
    • Detected by exit rate due to virtual NIC accesses
  – Low rate: guest-driven mode to reduce the single packet latency
  – High rate: VMM-driven mode to increase throughput

rate = # of exits for virtual NIC from last 10ms
if (rate >= THRESHOLD && current-mode == GUEST-driven)
current-mode= VMM-Driven;
else if (rate < THRESHOLD && current-mode == VMM-driven)
current-mode= GUEST-Driven;
else
do-nothing;
endif
Packet Process Offloading Using Dedicated Thread

Baseline

Dedicated thread + guest-driven

Separate Core

Time

13
VMM-driven + Dedicated Thread on Separate Core

- High throughput mode avoids most exits
- VNET/P and Guest process packets in parallel
Large MTU

• Larger MTU improves throughput and reduces CPU cost
  – Fewer packets are processed for a given amount of data.
  – VNET/P adds to the per-packet processing cost

• Guest MTU
  – Virtio NIC supports up to 64KB MTU
  – Most of other para-NICs support large MTU

• Host MTU
  – 10G usually supports jumbo MTU (9000Bytes)
Implementation

• Code size

<table>
<thead>
<tr>
<th>Components</th>
<th>LoC</th>
</tr>
</thead>
<tbody>
<tr>
<td>VNET/P Core</td>
<td>1955</td>
</tr>
<tr>
<td>VNET/P Bridge</td>
<td>1210</td>
</tr>
<tr>
<td>VNET/P Control Backend</td>
<td>1080</td>
</tr>
<tr>
<td>Virtio NIC Backend</td>
<td>987</td>
</tr>
<tr>
<td><strong>Total</strong></td>
<td><strong>5232</strong></td>
</tr>
</tbody>
</table>

– Mostly VMM-independent code
– Easy to port to other VMMs
Outline

• Model and motivation
• VNET/P: design & optimization
• Performance evaluation
• Conclusions and future work
Performance Evaluation

- **Micro-benchmarks: Bandwidth and Latency**
  - End-to-end performance
  - Multi-node performance

- **Application Performance**
  - NAS and HPCC

- **Comparison**
  - **VNET/P**: VMs with Linux and overlay, testing in guests
  - **Native**: Linux on hosts, no VMs, no overlay
  - **VNET/U**: VMs with user-level overlay
Native Bandwidth on 1Gb Network

VNET/P achieves native bandwidth
Near-native Round-trip Latency on 1Gb Network

VNET/P achieves 2 times of native latency
High Bandwidth on 10Gb Network

VNET/P achieves around 65%-70% of native bandwidth
Low Latency on 10Gb Network

VNET/P achieves 3 times of native latency

VNET/P-10G
Native-10G

Message Size (Bytes)
IMB Pingpong Latency
Scalable High Bandwidths (10Gbps)

Overhead is independent of scale

Bandwidth by HPCC Communication Benchmark
Scalable Low Latencies (10Gbps)

Overhead is independent of scale

Latency by HPCC Communication Benchmark
## Application Performance

<table>
<thead>
<tr>
<th>Mop/s</th>
<th>$\frac{VNET/P-10G}{\text{Native}-10G}$ (%)</th>
</tr>
</thead>
<tbody>
<tr>
<td>ep.B.8</td>
<td>99.9%</td>
</tr>
<tr>
<td>ep.B.16</td>
<td>99.3%</td>
</tr>
<tr>
<td>ep.C.8</td>
<td>99.0%</td>
</tr>
<tr>
<td>ep.C.16</td>
<td>98.9%</td>
</tr>
<tr>
<td>mg.B.8</td>
<td>74.3%</td>
</tr>
<tr>
<td>mg.B.16</td>
<td>81.0%</td>
</tr>
<tr>
<td>cg.B.8</td>
<td>86.2%</td>
</tr>
<tr>
<td>cg.B.16</td>
<td>93.7%</td>
</tr>
<tr>
<td>ft.B.16</td>
<td>85.8%</td>
</tr>
<tr>
<td>is.B.8</td>
<td>99.8%</td>
</tr>
<tr>
<td>is.B.16</td>
<td>99.6%</td>
</tr>
<tr>
<td>is.C.8</td>
<td>99.8%</td>
</tr>
<tr>
<td>is.C.16</td>
<td>98.9%</td>
</tr>
<tr>
<td>lu.B.8</td>
<td>83.9%</td>
</tr>
<tr>
<td>lu.B.16</td>
<td>74.3%</td>
</tr>
<tr>
<td>sp.B.9</td>
<td>91.9%</td>
</tr>
<tr>
<td>sp.B.16</td>
<td>96.9%</td>
</tr>
<tr>
<td>bt.B.9</td>
<td>78.0%</td>
</tr>
<tr>
<td>bt.B.16</td>
<td>96.7%</td>
</tr>
</tbody>
</table>
Outline

• Model and motivation
• VNET/P: design & optimization
• Performance evaluation
• Conclusions and future work
Future Work

• Further performance improvements
  – More specific optimizations to achieve native performance *(in submission)*
    • Optimistic interrupts,
    • Cut-through forwarding
    • Noise isolation
  – Move VNET up to guest through guest code injection *(to appear in ICAC’12)*

• Extend VNET/P on other high performance interconnects (Infiniband, SeaStar, etc)
  – Provide Ethernet abstraction for HPC application on different physical networks
VNET on Various Interconnects

• VNET on InfiniBand
  – Already works
  – Currently via IPoIB framework
    • 4.0Gbps bw/Native IPoIB 6.5Gbps
  – Pursuing high performance and leverage advanced hardware nature

• VNET over Gemini
  – In progress
Summary

• Current virtual networking is not fast enough for tightly-coupled environments
  – Bridge cloud and HPC resources for HPC applications

• VNET/P: high performance virtual overlay networking for tightly-coupled parallel systems
  – Overlay networking directly implemented into VMM
  – Native performance on 1Gb network
  – Close to native performance on 10Gb network

• Software-based overlay network can be extended into tightly-coupled environments
• Thanks, Questions??

• **Lei Xia**
  - Ph.D candidate, Northwestern University
  - lxia@northwestern.edu
  - http://www.cs.northwestern.edu/~lxia990

• V3VEE Project: [http://v3vee.org](http://v3vee.org)