Network virtualisation using Crossbow Technology

Uroš Nedić, MSc
OpenSolaris Contributor
• Crossbow Technical Background
• Network Machines
• Network in a Box
Crossbow Features

• Virtualization
  > IP Instances
  > NIC Virtualization - VNICS
  > Service Virtualization - Flows

• Resource partitioning
  > QoS/Diffserv (without performance penalties)
  > SLA on a per connection basis
  > Class of Service Support

• Performance & Observability
  > Dynamic polling
  > H/W and S/W fanout to multiple cores
  > Real Time Usage and History

Better Defense against DDOS attacks
Virtual Stacks

Services and Protocols

Compute Resources

CPU 1 Virtual Squeue
- VOIP SQUEUE
- HTTPS SQUEUE
- DEFAULT SQUEUE
- Virtual NIC
- Memory Partition

NIC 1
- Flow Classifier

CPU 2 Virtual Squeue
- TCP SQUEUE
- UDP SQUEUE
- DEFAULT SQUEUE
- Virtual NIC
- Memory Partition

NIC 2
- Flow Classifier

CPU 'n' Virtual Squeue
- Virtual NIC
- Memory Partition
Virtual Stacks

Services and Protocols

CPU 1 Virtual Squeue
- VOIP SQUEUE
- HTTPS SQUEUE
- DEFAULT SQUEUE

CPU 2 Virtual Squeue

CPU 'n' Virtual Squeue

The Squeue switches the MSI interrupt per stack between interrupt and polling mode and controls the rate of packet arrival for the virtual stack.

The VNICS are in the control path only. The data link layer is bypassed.

Memory Partition

Flow Classifier

NIC 1

NIC 2

Flow Classifier

Memory Partition

Memory Partition

Memory Partition
Crossbow Architecture

- Partition the NIC Hardware, kernel queues, and CPU and allow creation of Virtual NICs
- Use dynamic polling on Virtual NICs to schedule rate of packet arrival per VNIC
- Effect of dynamic polling

\textbf{Mpstat (older driver)}
\begin{tabular}{cccccccccccc}
intr & ithr & cs & ic & migr & smtx & srw & syscl & usr & sys & wt & idl \\
10818 & 8607 & 4558 & 1547 & 161 & 1797 & 289 & 19112 & 17 & 69 & 0 & 12 \\
\end{tabular}

\textbf{Mpstat (Crossbow based driver)}
\begin{tabular}{cccccccccccc}
intr & ithr & cs & ic & migr & smtx & srw & syscl & usr & sys & wt & idl \\
2823 & 1489 & 875 & 151 & 93 & 261 & 1 & 19825 & 15 & 57 & 0 & 27 \\
\end{tabular}

- Use Dynamic polling for B/W partitioning and isolation without any support from switches and routers
## Parallelized Stack: Made for Cores/Threads

<table>
<thead>
<tr>
<th>Network</th>
<th>NIC</th>
<th>Kernel</th>
<th>CPU</th>
<th>Application</th>
</tr>
</thead>
<tbody>
<tr>
<td>1000’s of Logical Connections</td>
<td>Neptune 2 x 10 Gbe and/or 4 x 1Gbe</td>
<td>Crossbow Virtualized Network Streams &amp; VNICS</td>
<td>Multi-Core Processor</td>
<td>Application Threads Or Solaris Containers Or Virtual Machines</td>
</tr>
</tbody>
</table>

- **CLASIFIER**
  - VNICS
    - Rx/Tx DMA
    - Kernel Qs and threads
  - Squeue

- **Kernel Qs and threads**
  - Rx/Tx DMA
  - Kernel Qs and threads

- **CPU**
  - Dedicated Lane
Virtual Network Containers

Virtualization
- Exclusive IP Stack
- Virtual NICs
- Virtual Switches

Resource Control
- Bandwidth Limits & Guarantees
- Priority
- Dedicated CPUs

Observability
- Watch real time usage for each VNIC
- Snmp and Kstat per VNIC
- History at no cost
Virtual Machines

Solaris Host OS

Host OS VIRTUAL SQUEUE
All Traffic

Host OS VNIC

Solaris Guest OS 1

Guest OS 1 VIRTUAL SQUEUE
HTTP SQUEUE
HTTPS SQUEUE
DEFAULT SQUEUE

Virtual NIC

Solaris Guest OS 2

Guest OS 2 VIRTUAL SQUEUE
All Traffic

Guest OS 2 VNICE

All Traffic

H/W Flow Classifier

NIC
Network Machines

Solaris
- DMZ Apache
  - TCP/UDP
  - IP
  - Virtual NIC A
- Router/Firewall
  - TCP/UDP
  - IP
  - Virtual NIC A
  - Virtual NIC B
- DNS Server
  - TCP/UDP
  - IP
  - Virtual NIC B

N2/NIU
- Flow Classifier
  - NIC A
  - Rx/Tx DMA
  - Internet
- Flow Classifier
  - NIC B
  - Rx/Tx DMA
  - Intranet

Dedicated CPUs

APIs for ISVs at each layer
Anatomy of a Typical Router

Major cost paid on bringing the packet in and route lookup etc.

Packets are dropped on Xmit Queues
Anatomy of Crossbow Router

Policy-based packet drop on ingress—zero cost drop.

- Explicit mapping of Compute Threads or Cores to Network Threads creating virtual “express lanes”
- Dedicated BW, Priority, and CPU/Threads assigned to each lane
- High performance packet classification
Physical Network

Crossbow Network in a Box
Example VNIC Usage

• Done using `dladm(1M)`, as with other data-link interface administration

```
# dladm create-vnic -d bge1 vnic1
# dladm create-vnic -d bge1 -m random -p maxbw=100M -p cpus=4,5,6 vnic2
# dladm create-etherstub vswitch1
# dladm show-etherstub
LINK
  vswitch1
# dladm create-vnic -d vswitch1 -p maxbw=1000M vnic3
# dladm show-vnic
```

```
<table>
<thead>
<tr>
<th>LINK</th>
<th>OVER</th>
<th>MACTYPE</th>
<th>MACVALUE</th>
<th>BANDWIDTH</th>
<th>CPUS</th>
</tr>
</thead>
<tbody>
<tr>
<td>vnic1</td>
<td>bge1</td>
<td>factory</td>
<td>0:1:2:3:4:5</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>vnic2</td>
<td>bge1</td>
<td>random</td>
<td>2:5:6:7:8:9</td>
<td>max=100M</td>
<td>4,5,6</td>
</tr>
<tr>
<td>vnic3</td>
<td>vswitch1</td>
<td>random</td>
<td>4:3:4:7:0:1</td>
<td>max=1000M</td>
<td>-</td>
</tr>
</tbody>
</table>
```
Join Us. ...

• Our communities and projects are open on OpenSolaris.org:
  > CrossBow: http://opensolaris.org/os/project/crossbow
  > VNM: http://opensolaris.org/os/project/vnm
  > Networking: http://opensolaris.org/os/community/networking

• Where you will find:
  > Lively discussions, design docs, FAQs, source code drops, binary releases, etc...