CrossBow:
A vertically integrated QoS stack
Sunay Tripathi, Nicolas Droux, Thirumalai Srinivasan, Kais Belgaied, Venu Iyer
Aug 21st. 2009
Sigcomm WREN 2009, Barcelona

Sunay Tripathi,
Distinguished Engineer, Sun Microsystems Inc
Sunay.Tripathi@Sun.Com
Issues in Host based QoS solutions

• Performance
  > Additional Classification/Queuing for all packets
  > QoS layers typically high up in the stack (bulk of the work already done)
  > Packet needs to be DMA'd into the system before any policy can be applied

• QoS layers are typically bump in the stack

• Management complexities
Crossbow: Solaris Networking Stack

• 8 years of development work to achieve
  > Scalability across multi-core CPUs and multi-10gigE bandwidth
  > Virtualization, QoS, High Availability designed in
  > Exploit advanced NIC features

• Key Enabler for
  > Server and Network Consolidation
  > Resource partitioning
  > Cloud computing
Crossbow “Hardware Lanes”

Ground-Up Design for multi-core and multi-10GigE

- Linear Scalability using 'Hardware Lanes' with dedicated resources
- Network Virtualization and QoS designed in the stack
- More Efficiency due to 'Dynamic Polling and Packet Chaining'

Switch

<table>
<thead>
<tr>
<th>VLAN Separated</th>
</tr>
</thead>
<tbody>
<tr>
<td>Hardware Lane</td>
</tr>
</tbody>
</table>

Physical NIC

- Hardware Rings/DMA
- Hardware Rings/DMA

Physical Machine

- Kernel Threads and Queues
- Virtual NIC
- Virtual Machine/Zone
- Virtual Machine/Zone

Application

Squeue
Hardware Lanes and Dynamic Polling

- Partition the NIC Hardware (Rx/Tx rings, DMA), kernel queues/threads, and CPU to allow creation of “Hardware Lane” which can be assigned to VNICs & Flows

- Use Dynamic Polling on Rx/Tx rings to schedule rate of packet arrival and transmission on a per lane basis

- Effect of dynamic polling

  \[\text{Mpstat (older driver)}\]
  \begin{array}{cccccccccccc}
  \text{intr} & \text{ithr} & \text{csw} & \text{icsw} & \text{migr} & \text{smtx} & \text{srw} & \text{syscl} & \text{usr} & \text{sys} & \text{wt} & \text{idl} \\
  10818 & 8607 & 4558 & 1547 & 161 & 1797 & 289 & 19112 & 17 & 69 & 0 & 12 \\
  \end{array}

  \[\text{Mpstat (GLDv3 based driver)}\]
  \begin{array}{cccccccccccc}
  \text{intr} & \text{ithr} & \text{csw} & \text{icsw} & \text{migr} & \text{smtx} & \text{srw} & \text{syscl} & \text{usr} & \text{sys} & \text{wt} & \text{idl} \\
  2823 & 1489 & 875 & 151 & 93 & 261 & 1 & 19825 & 15 & 57 & 0 & 27 \\
  \end{array}

- Use Dynamic polling for B/W partitioning and isolation without any support from switches and routers

  \(~75\%\) Fewer Interrupts  \(~85\%\) Fewer Context Switches  \(~85\%\) Fewer Mutexes  \(~15\%\) More CPU Free
Crossbow Flows: Service Virtualization

Services and Protocols

Compute Resources

CPU 1 Virtual Squeue
- VOIP Squeue
- HTTPS Squeue
- DEFAULT Squeue
- Kernel threads/Qs
- Memory Partition

NIC 1
Flow Classifier

CPU 2 Virtual Squeue
- TCP Squeue
- UDP Squeue
- DEFAULT Squeue
- Kernel threads/Qs
- Memory Partition

NIC 2
Flow Classifier

CPU 'n' Virtual Squeue
- Kernel threads/Qs
- Memory Partition
Crossbow Flows based on:

> Services (protocol + remote/local ports)
> Transport (TCP, UDP, SCTP, iSCSI, etc)
> Remote and local IP addresses
> Remote IP Subnets
> DSCP labels

Following attributes can be set on each Flow:

> B/W limits
> Priorities
> CPUs

```
# flowadm create-flow -l bge0 protocol=tcp,local_port=443 -p maxbw=50M http-1
# flowadm set-flowprop -l bge0 -p maxbw=100M  http-1
```
Virtual Network Containers

Virtualization
- Flows
- Virtual NICs & Virtual Switches
- Virtual Wire

Resource Control
- Bandwidth Partitioning
- NIC H/W Partitioning
- CPUs/pri assignment

Observability
- Real time usage for each Link/flow
- Finer grained stats per Link/flow
- History at no cost
Defense against DOS/DDOS

- DDOS have the ability to cripple entire server farms and all services offered by them.
- Only the impacted services or virtual machine takes the hit instead of the entire grid.
- Under attack, impacted services start all new connections under lower priority flow with limited bandwidth.
- Connections transition to appropriate priority stacks after application authentication.
- IDS systems can use Crossbow APIs to create '0' B/W flows based on remote IP addresses or subnets of the attackers and minimize their impact.

www.opensolaris.org/os/project/crossbow
BACKUP
Solaris Core Network Functionality

- Networking Services
  - Routing Protocols using Quagga
  - L3/L4 Load Balancer kernel modules
  - IP Firewall (IPFilter)
  - DNS, DHCP, NTP, SIP, VOIP, etc
- Scalable & Virtualized Network Stack
  - Kernel Socket & Socket Filter
  - Modernized TCP/IP Stack
  - QoS: B/W limits, Priorities, CPU bindings
  - IP Multi Pathing (IPMP)
  - IP Tunneling
  - Defense against DDoS attacks
- Crossbow: Virtual Networking
  - VNICs, VSwitches, VWire
  - Service Virtualization (Flows)
  - L2 Services: Classification, Filtering
- Generic LAN Driver v3 – GLDv3
  - Aggregation
  - Vanity Names
  - Drivers (1GbE and 10GbE, FCoE, IPoIB)

www.opensolaris.org/os/project/crossbow
Virtual NIC (VNIC) & Virtual Switches

Virtual NICs
> Functionally physical NICs:
  > IP address assigned statically or via DHCP and snooped individually
  > Appear in MIB as separate 'if' with configured link speed shown as 'ifSpeed'
  > VNICS can be created over Link Aggregation on can be assigned to IPMP groups for load balancing and failover support
> VNICS Can have multiple hardware lanes assigned to them
> Can be created over physical NIC (without needing a Vswitch) to provide external connectivity with switching done in NIC H/W
> VNICS have configurable link speed, CPU and priority assignment
> Standards based End to End Network Virtualization
  > VLAN tags and Priority Flow Control (PFC) assigned to VNIC extend Hardware Lanes to Switch
> No configuration changes needed on switch to support virtualization

Virtual Switches
> Can be created to provide private connectivity between Virtual Machines
Virtual NIC & Virtual Switch Usage

# dladm create-vnic -l bge1 vnic1
# dladm create-vnic -l bge1 -m random -p maxbw=100M -p cpus=4,5,6 vnic2
# dladm create-etherstub vswitch1
# dladm show-etherstub

```
LINK
vswitch1
```

# dladm create-vnic -l vswitch1 -p maxbw=1000M vnic3
# dladm show-vnic

```
LINK  OVER  MACTYPE  MACVALUE       BANDWIDTH     CPUS
vnic1  bge1  factory  0:1:2:3:4:5    -             -
vnic2  bge1  random   2:5:6:7:8:9    max=100M      4,5,6
vnic3  vswitch1 random  4:3:4:7:0:1  max=1000M     -
```

# dladm create-vnic -l ixgbe0 -v 1055 -p maxbw=500M -p cpus=1,2 vnic9
**Physical Wire w/Physical Machines**

Client

- Port 6
  - 20.0.03
  - 1 Gbps

Router

- Port 9
  - 20.0.01
  - 1 Gbps

- Port 3
  - 10.0.03
  - 1 Gbps

Host 1

- Port 1
  - 10.0.01
  - 1 Gbps

Host 2

- Port 2
  - 10.0.02
  - 1 Gbps

Switch 3

Switch 1

**Virtual Wire w/Virtual Network Machines**

Client

- VNIC6
  - 20.0.03
  - 1 Gbps

Router (Virtual Router)

- VNIC9
  - 20.0.01
  - 1 Gbps

- VNIC3
  - 10.0.03
  - 1 Gbps

Host 1

- VNIC1
  - 10.0.01
  - 100 Mbps

Host 2

- VNIC2
  - 10.0.02
  - 1 Gbps

EtherStub 3

EtherStub 1
Crossbow extends H/W Lanes to Switch

- Dedicated path from switch to the Virtual Machine
- VNIC A can send PFC pause to switch forcing the traffic from client A to slow down
- Incoming traffic for Virtual machine B (who has higher configured link speed) does not suffer

Client A
(Sending traffic to Virtual Machine A faster than 100 Mbps)

Pause Frame sent by VNIC-A to switch asking it to slow the incoming traffic for VM-A
Virtual Machines

Solaris Host OS
- NIC Virtualization Engine
- Host OS VNIC
  - VIRTUAL SQUEUE
    - All Traffic

Solaris Guest OS 1
- NIC Virtualization Engine
- Guest OS 1 VIRTUAL SQUEUE
  - HTTP SQUEUE
  - HTTPS SQUEUE
  - DEFAULT SQUEUE
  - Virtual NIC

Solaris Guest OS 2
- NIC Virtualization Engine
- Guest OS 2 VIRTUAL SQUEUE
  - All Traffic

H/W Flow Classifier
- Guest OS 1 HTTP
- Guest OS 1 HTTPS
- Guest OS 1 DEFAULT
- Guest OS 2 All Traffic
- Guest OS 2 VNIC
Dynamic Polling: Effect on Throughput

High Load TCP Read/Write Test
5 Clients (pktsz=1500; wrtsz=8k)

Config Details:
5 Client; 1 Server – 10GigE Links
3 Clients reading (10 thread each)
2 Clients writing (10 thread each)
All Client/Server:
x4150 dual soc 8x2.8Ghz Intel CPU
10 GigE NIC – Intel Oplin (ixgbe)
Dynamic Polling: Effect on Latency

UDP 66byte pkt Low Load Latency Test

![Graph](image1.png)

**Pkts Received via Interrupt/Poll Ratio**

![Graph](image2.png)

**Number of Chains < 10 pkts**

![Graph](image3.png)

UDP 66byte pkt High Load Latency Test

![Graph](image4.png)

**Pkts Received via Interrupt/Poll Ratio**

![Graph](image5.png)

**Pkt Chain Lengths**

![Graph](image6.png)
Virtual Network Machines
Networking as a Service (NaaS)

- Virtual Network Machines – Networking as a Service
  - Monetize via the subscription model in cloud using virtualized networking services like vRouter, vloadbalancer, vFirewall, vDHCPserver, vDNSserver, etc

- Virtualized Networking Services wrapped in Solaris Zone/Xen/VB running on dedicated Networking blades/appliance
  - Open Source Virtualized Networking Services
  - VNICs and Vswitches provide the virtualized ports similar to physical ports
  - Enable Virtual Networks with configurable link speeds using Virtual Wire

- Management for a Virtual Network Machines
  - Solaris command line
  - Cisco Style 'cli'
  - Web based
Virtual Network Machine Appliance for the cloud

The network is the computer

Crossbow

Networking as a Service (Naas) Subscription based or dedicated appliance

www.opensolaris.org/os/project/crossbow
Virtual Network Machines over 10Gbe

OpenSolaris
- vFirewall, vVPN
- vRouter
- vNTP, vDHCP, vDNS, vLDAP, ..

N2/x64 Server/Blades
- Virtual NIC A
- Virtual NIC B

10Gbe NIC/NIU
- NIC A
- NIC B

Flow Classifier & Offload Eng.

Dedicated CPUs

APIs for ISVs at each layer

Rx/Tx DMA

WAN

Data Center VLAN'd ETH Fabric

TCP/UDP

IP
Cloud Virtual Machines over 40Gbs IB

OpenSolaris

Dom0

iSER, NFS, ...

RDMA, IPoIB

DomU

Apps

TCP/IP

DomU

.....

Apps

TCP/IP

.....

N2/x64 Server/Blades

IBTF

VNIC/ EoIB

VNIC/ EoIB

.....

VNIC/ EoIB

APIs for ISVs at each layer

Rx/Tx Q-Pair

Rx/Tx Q-Pair

 Rx/Tx Q-Pair

Rx/Tx Q-Pair

Rx/Tx Q-Pair

Rx/Tx Q-Pair

Infiniband Firmware

HCA A

......

Dom0 IB Partition

Infiniband Firmware

HCA B

DomU IB Partition

Dedicated CPUs
Open Storage Networking

- Priority Based Flow Control (PFC)
  > 8 ethernet virtual lanes with their own pause mechanism
  > Extend the Crossbow H/W Virtualized Lanes to the switch
- Enhanced Transmission Selection (ETS)
  > Add Class of service support within the ethernet virtual lane
  > Extend the Crossbow flow based QoS to the switch
- Link Layer Discovery Protocol (LLDP) and Congestion notification (optional)
- PFC and ETS is useful in normal virtualization and server QoS scenarios
- PFC, ETS, and LLDP are necessary to implement Data Center Bridge Exchange protocol (DCBX) and FCOE
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:
  - Active discussions, design docs, FAQs, source code drops, preliminary binary releases, etc...