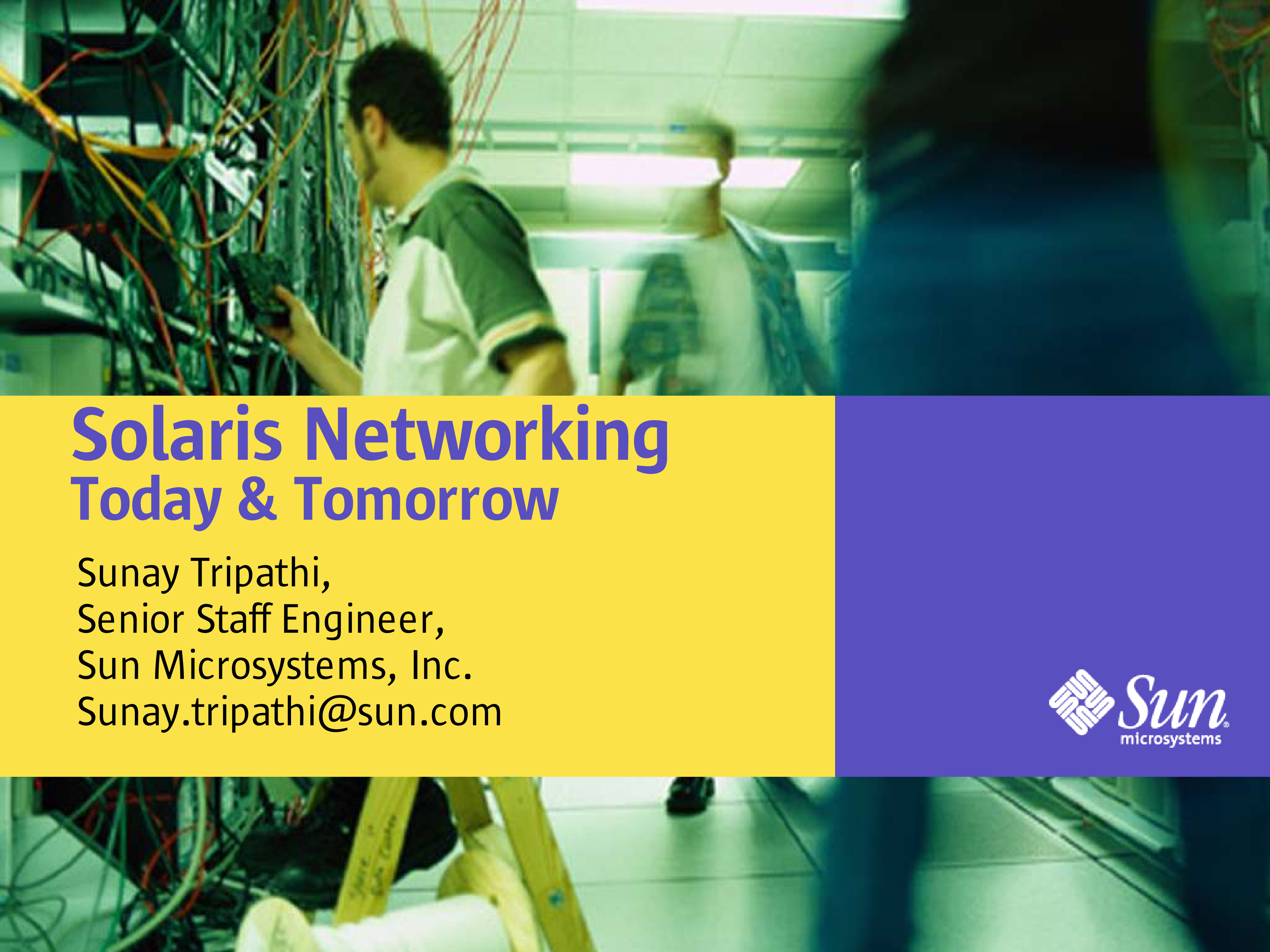


# Solaris Networking
## Today & Tomorrow

Sunay Tripathi,
Senior Staff Engineer,
Sun Microsystems, Inc.
Sunay.tripathi@sun.com

# The Building Block

- Solaris 10 delivered several ground breaking technologies and several useful projects in networking space
- Some of the key technologies/projects are:
  - FireEngine: A high performance architecture for networking stack
  - Wanboot: Remote boot machine using HTTP/HTTPS
  - SCTP
  - Fully deployable Ipv6
  - Bundled SIP stack for VOIP applications
- FireEngine created a ground breaking architecture which is used as a building block for current & future projects

# FE - Veritical Perimeter (Squeues)

- Squeue is a per CPU common serialization queue (FIFO) for all inbound/outbound packets
- Squeue provides the mutual exclusion to all TCP connections without locks (lockless design) by allowing only one thread to process it at any given time
- Packet once picked up for processing is taken all the way to socket (on inbound) or NIC (on outbound) giving it the property of Vertical perimeter

# FE - IP Classifier

- Use a connection classifier early in IP for incoming packet
- The connection structure ('connp') contains all the necessary information:
  - The CPU/squeue the packet needs to be processed on
  - The string of functions necessary to process the packet (event lists)

# FE - Squeue + Classifier

- Create a per CPU squeue index on cpuid
- Bind a connection to a particular squeue so packets for that connection are always processed on same squeue
- Bind each inbound connection to the squeue attached to the interrupted CPU for incoming connection to maintain data locality and vertical separation
- Use the classifier to direct packets to the CPU they need to be processed on

# FE - TCP/ IP Merge

- Use function calls between TCP and IP to reduce per packet processing cost
- Separate and optimize the hot paths
- Merge TCP/ IP in one STREAM module (fully MT)
- The STREAM entry points are manipulated based on whether someone opens /dev/tcp or /dev/ip
- TCP/ IP modules behaves the same (as pre FE) i.e. if someone opens /dev/ip he gets the IP behaviour and if someone open /dev/tcp, he gets TCP behaviour

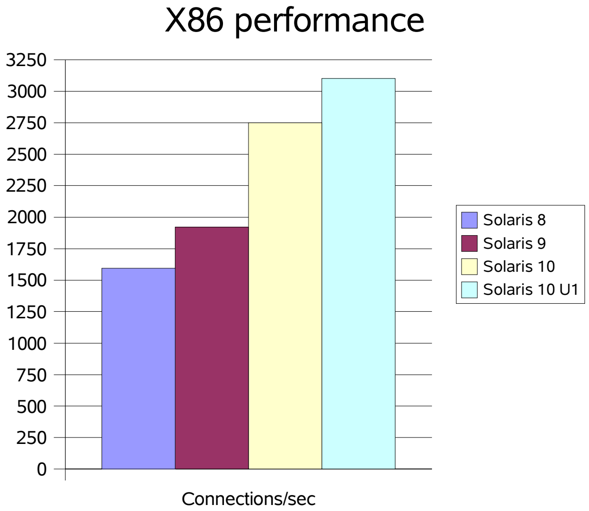
# FE - performance wins

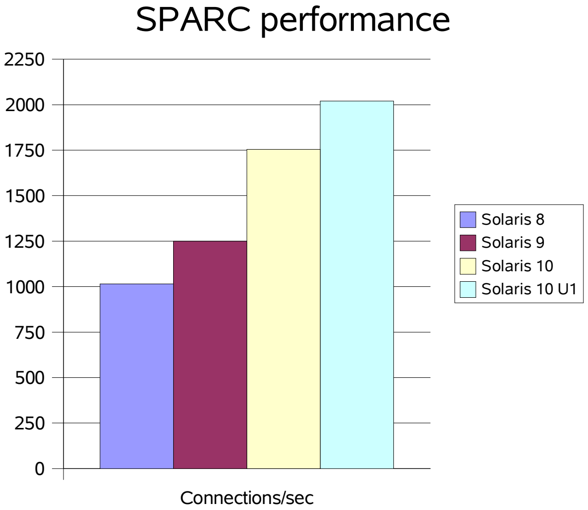
- FireEngine Phase 1 integrated in s10_41 and contained improvements for TCP/IP only
- Achieved 45% gain on web like workload on SPARC
- Achieved 43% gain on web like workload on x86
- Other gains (just due to FireEngine):
  - 25% SSL
  - 15% fileserving
  - 40% throughput (ttcp)
- On v20z, Solaris is faster than Linux by 10-20% using Apache or Sun One Webserver on a web based workload

# FE - Current Status (cont.)

- Webbench
  - Static - Solaris 10 outperforms Windows 2003 by 26%
  - Dynamic - Solaris 10 outperforms Windows 2003 by 29% and RHEL-AS3 by 3%
  - Ecommerce - Solaris 10 outperforms Windows 2003 by 18% and RHEL-AS3 by 14%
- Solaris 10 can fully saturate a 1Gb link with only 8% of 1x2.2Ghz Opteron and
- Solaris 10 can drive a 10Gb link at 7.3Gbps (limited by PCI-X bandwidth) using 2x2.2Ghz opteron CPUs utilized at less than 50%

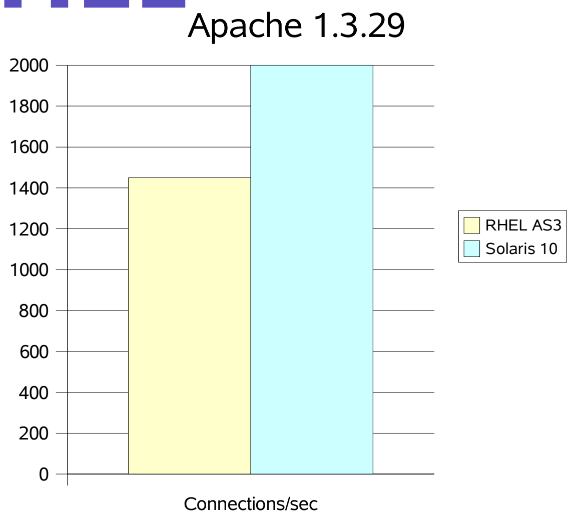
# Solaris web performance

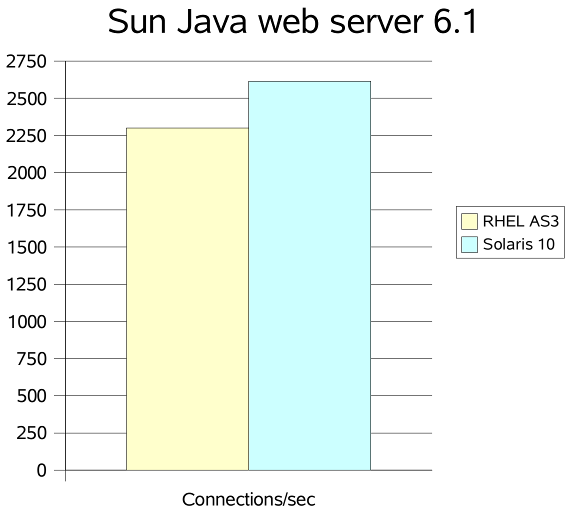




- Configuration
  - X86: v20z (2x2.2Ghz, 6Gb RAM, 2x1Gb NICs, Zeus 4.1r4)
  - SPARC: Sunblade 2500 (2x1.2Ghz USIII+, 8Gb RAM, 2x1Gb NIC, Zeus 4.1r4)
- Connections/sec are number of connections that can be handled at a certain bit rate (similar to SPECweb99 conn/sec)
- S10U1 numbers are based on project Nemo
- SPARC numbers are with single core Ultra Sparc III+ (US IV numbers should be double of these since we get linear scaling on dual cores)

# Solaris web performance vs RHEL





- Configuration
  - X86: v20z (2x2.2Ghz, 6Gb RAM, 2x1Gb NICs)
- Connections/sec are number of connections that can be handled at a certain bit rate (similar to SPECweb99 conn/sec)

# Solaris Networking: Today

Sunay Tripathi,
Senior Staff Engineer,
Sun Microsystems, Inc.
Sunay.Tripathi@sun.com

# Solaris Networking today

- GLDv3 (codename Nemo):
  - Dynamic switching between interrupt and polling
  - 10Gbps NIC support
  - Vlan and Trunking support for off the shelf NICs
- NCA merge to FireEngine (NL7C)
- UDP performance (codename yosemite)
- Forwarding performance (codename Surya)
- IPfilter performance

# Dynamically switch between Interrupt and Polling (and packet chaining)

- Networking interrupts are bad because writers gets pinned, context switches, etc.
- Bind a NIC to a Squeue and the let the Squeue own the NIC
- On backlog, Squeue turns the NIC interrupts off
- Squeue can retrieve packets from the ring (in chains) after the backlog is cleared (poll mode)
- If no backlog, Squeue switches the NIC back to interrupt mode

# More performance

- Another 25% improvement on x86 and 20% on SPARC platforms on web workloads
- Below is a sample mpstat output

***Mpstat (older driver)***

```
intr   ithr  csw   icsw  migr smtx  srw  sysc1  usr sys wt idl
10818  8607  4558  1547  161  1797  289  19112  17  69  0  12
```

***Mpstat (GLDv3 based driver)***

```
intr   ithr  csw   icsw  migr smtx  srw  sysc1  usr sys wt idl
2823   1489  875   151   93   261   1    19825  15  57  0  27
```

- Notice the decrease in interrupts, context switches, mutex contentions, etc. and increase in idle time

# 10 GbE

- We can do 7.3 Gbps on a v20z (with 50% utilization) using 9k frames
- We can do 7 Gbps on a v20z on recv with 1500 bytes frames
- Solaris 10 set new LAN record during *Internet 2: Land speed record* challenge by transferring 14Gbps over 2 x 10Gbps using a v20z
- Application to application round trip latency close to 40usec

# Trunking

- Create the trunk of 1Gb NICs or 10GB NICs
- Each member or the trunk is owned by individual Squeues which control the rate of arrival of packets
- We see pretty linear scalability for a trunk of 4 1Gb NICs – 3.6Gbps
- We plan to handle a combined bandwidth of 30Gbps from a trunk of 4 x 10Gb NICs on a v40z
- During Sunlabs Openhouse in 4/2005, transferred 12Gbps over a trunk of 2x10Gbps NIC (single IP address) on a v20z

# UDP Performance (Yosemite)

- Create FireEngine like architecture for UDP as well
- Improve applications like TIBCO which depend on UDP performance
- With IP fully multithreaded and UDP/IP merged into one STREAM module, alleviate problems like UDP packets getting dropped in kernel
- Tibco performance up by 90-130% on xmit and 70-80% on recv
- Available in Solaris 10 Update

# IP forwarding (Surya)

- Forward close to million pkts/sec using single opteron processor
- Have the ability to lookup the routing table from Nemo framework itself
- In future, forwarded packet will be turned around from Nemo layer itself
- Solaris will also have the ability to have multiple instances of routing table (per virtual stack)
- Available in Solaris 10 Update

# More information

- http://www.sun.com/2004-1012/feature
- http://wwws.sun.com/software/solaris/10/ds/network_performance.jsp
- http://www.sun.com/bigadmin/xperts/sessions/11_fireengine/
- http://www.sun.com/bigadmin/content/networkperf/
- http://www.sun.com/bigadmin/features/articles/meet_architects.html#sunay

# Solaris Networking: Tomorrow CrossBow – Stack Virtualization & Resource Control

Sunay Tripathi, Sr Staff Engineer,
Solaris Networking Technology
Sun MicroSystems Inc.

# Real Scenarios

**Financial Services**

- *Trading house starts offering free financial information to attract customers*
- *Brokerage customers start complaining that trading site slows down*
- *The paying customers start deserting*

**Large ISP**

- *ISP wants to deploy virtual systems on same physical machines*
- *ISP sells each virtual system at different price levels to its customers*
- *Any virtual instance can overwhelmed the shared networking resource*

**Enterprise Computing**

- *A large company uses a workgroup server for day to day as well as critical traffic*
- *IT Ops doing non critical stuff started a backup over the network*
- *Users doing time critical work can't get bandwidth to do their job*

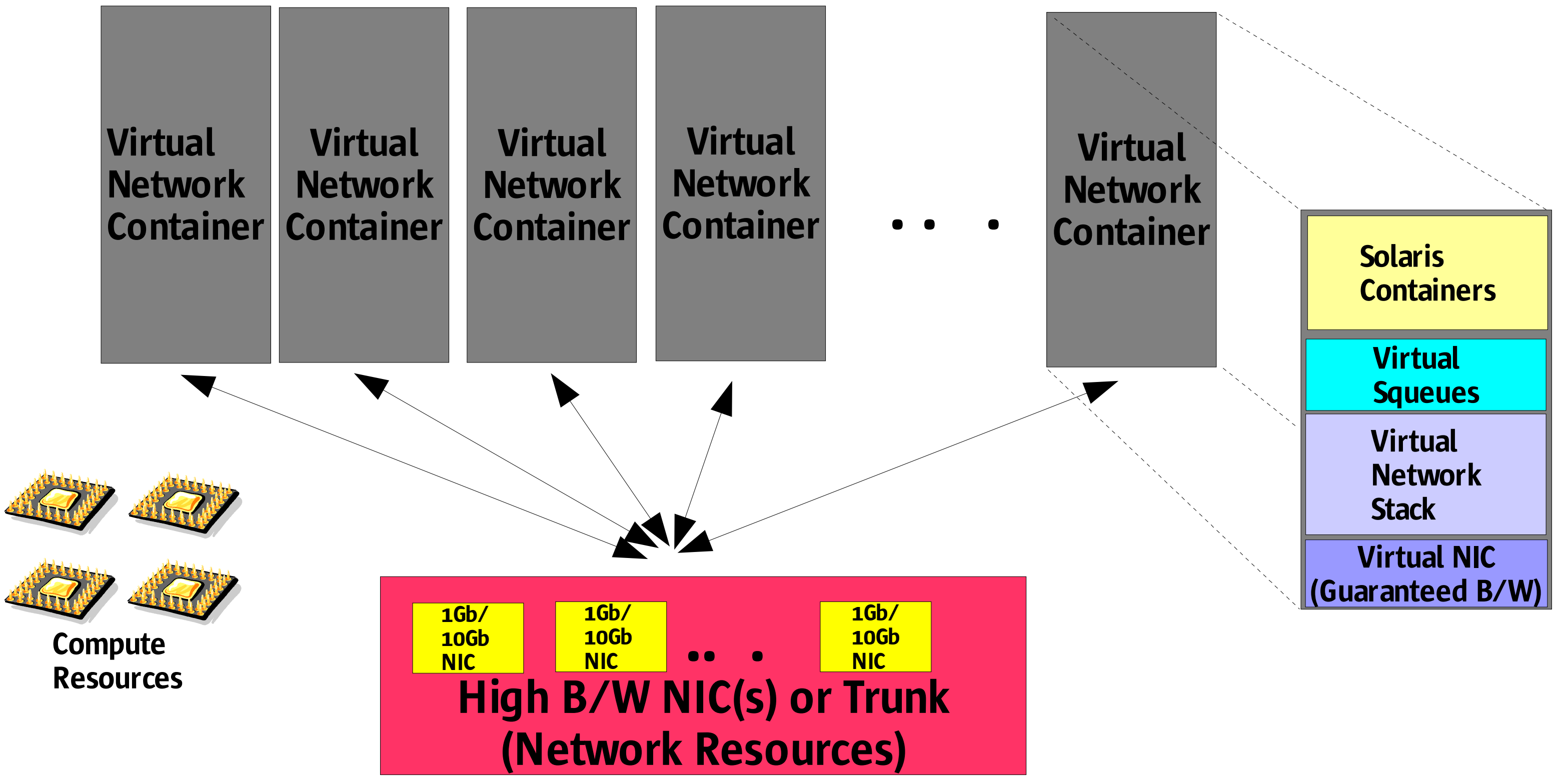
**What Happened?**

- *Critical services are overwhelmed by non-critical services, traffic types, or virtual systems*
- *No usable mechanism available for fairness, priority and resource control for networking bandwidth*

# Network Virtualization

- Virtualize the 1Gb and 10Gb NICs based on protocol, service, or container
- Requirements:
  - Specifiy priority and/or bandwidth relative to other virtual stacks on the system
  - Be able to choose protocol layers, firewalls rules, encryption rules, and any tuning specific to the virtual stack
- Constraints:
  - Virtual stacks isolated from each other
  - No performance overheads due to virtualization

# Virtual Network Stack


Virtual Network Container
Virtual Network Container
Virtual Network Container
Virtual Network Container
. . .
Virtual Network Container
Solaris Containers
Virtual Squeues
Virtual Network Stack
Virtual NIC (Guaranteed B/W)
Compute Resources
1Gb/ 10Gb NIC
1Gb/ 10Gb NIC
.. .
1Gb/ 10Gb NIC
High B/W NIC(s) or Trunk (Network Resources)

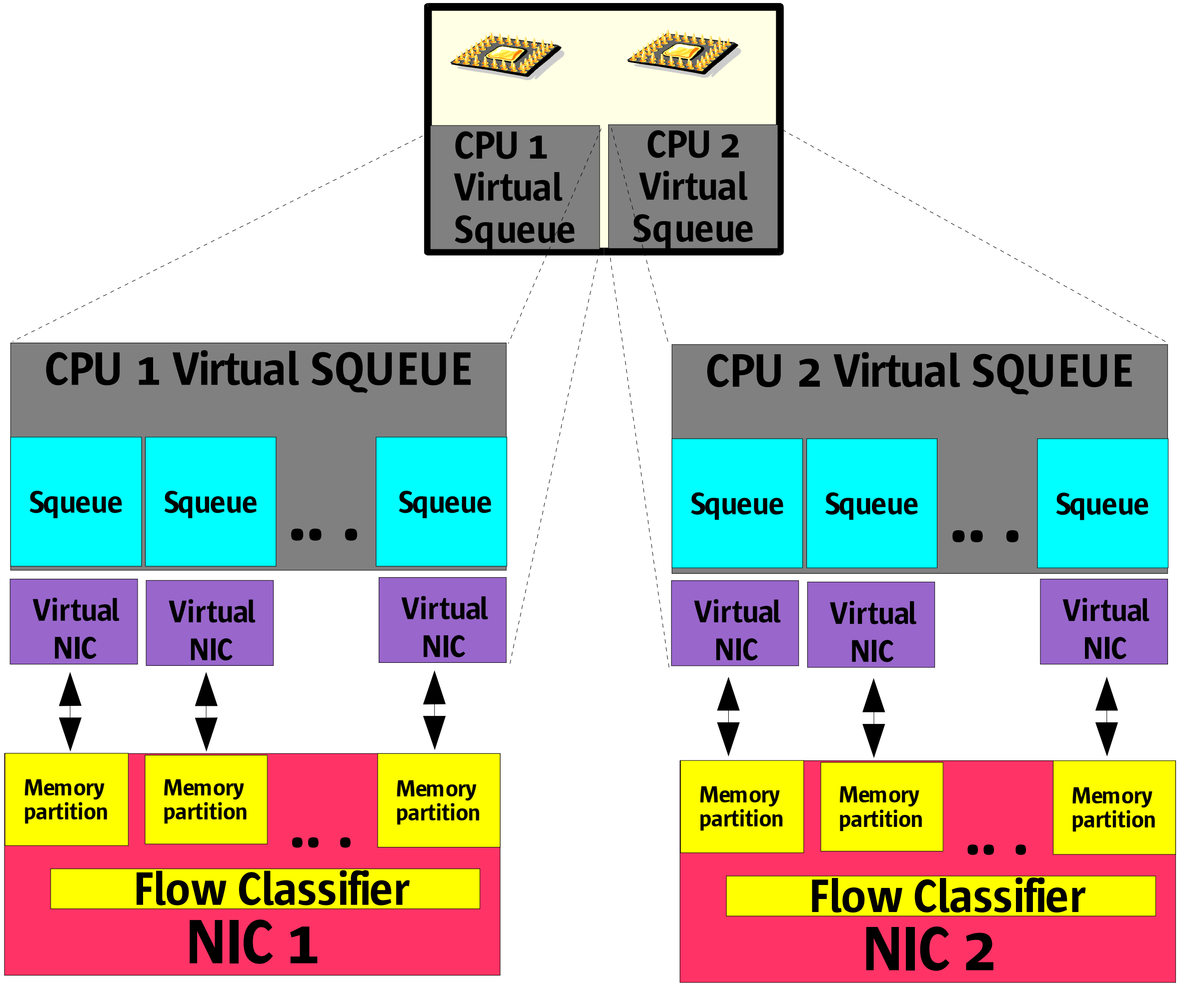
# Technical Obstacles

- Obstacles to achieving network virtualization:
  - Network processing in interrupt context
  - Anonymous packet processing in kernel
  - Common queues
- Performance can be degraded by the extra processing to enforce fairness, resource control or network virtualization

# The Crossbow Architecture

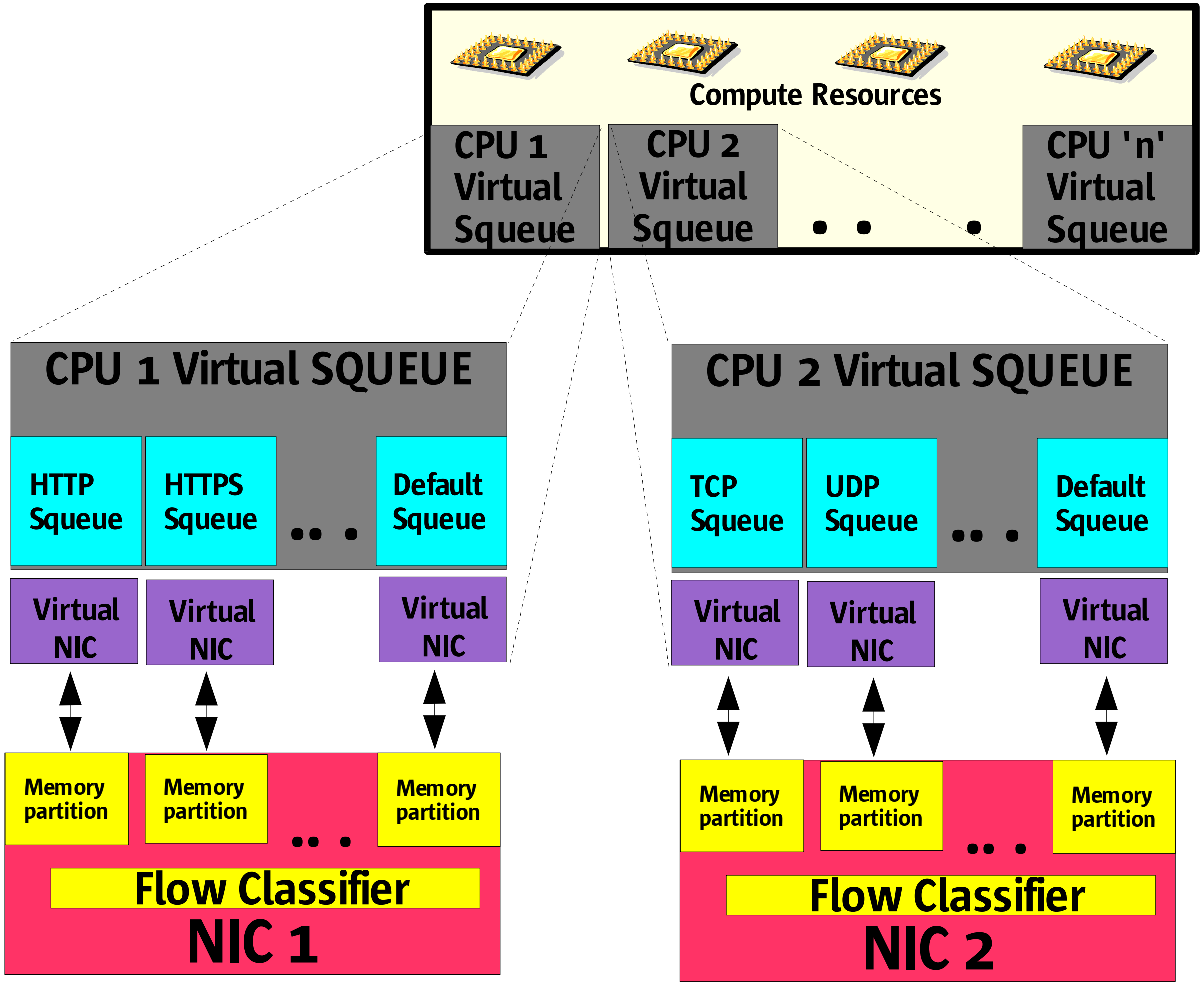
- Use the NIC to separate out the incoming traffic and divide NIC memory amongst the virtual stacks
- Assign MSI interrupt per virtual stack
- The FireEngine Squeue controls the rate of packet arrival into the virtual stack by dynamically switching between interrupt & polling
- Incoming B/W is controlled by pulling only the allowed number of packets per second
- Virtual stack priority is controlled by the squeue thread which does the Rx/Tx processing

# Virtual Stacks



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

# Virtual Stacks – Services & Protocols

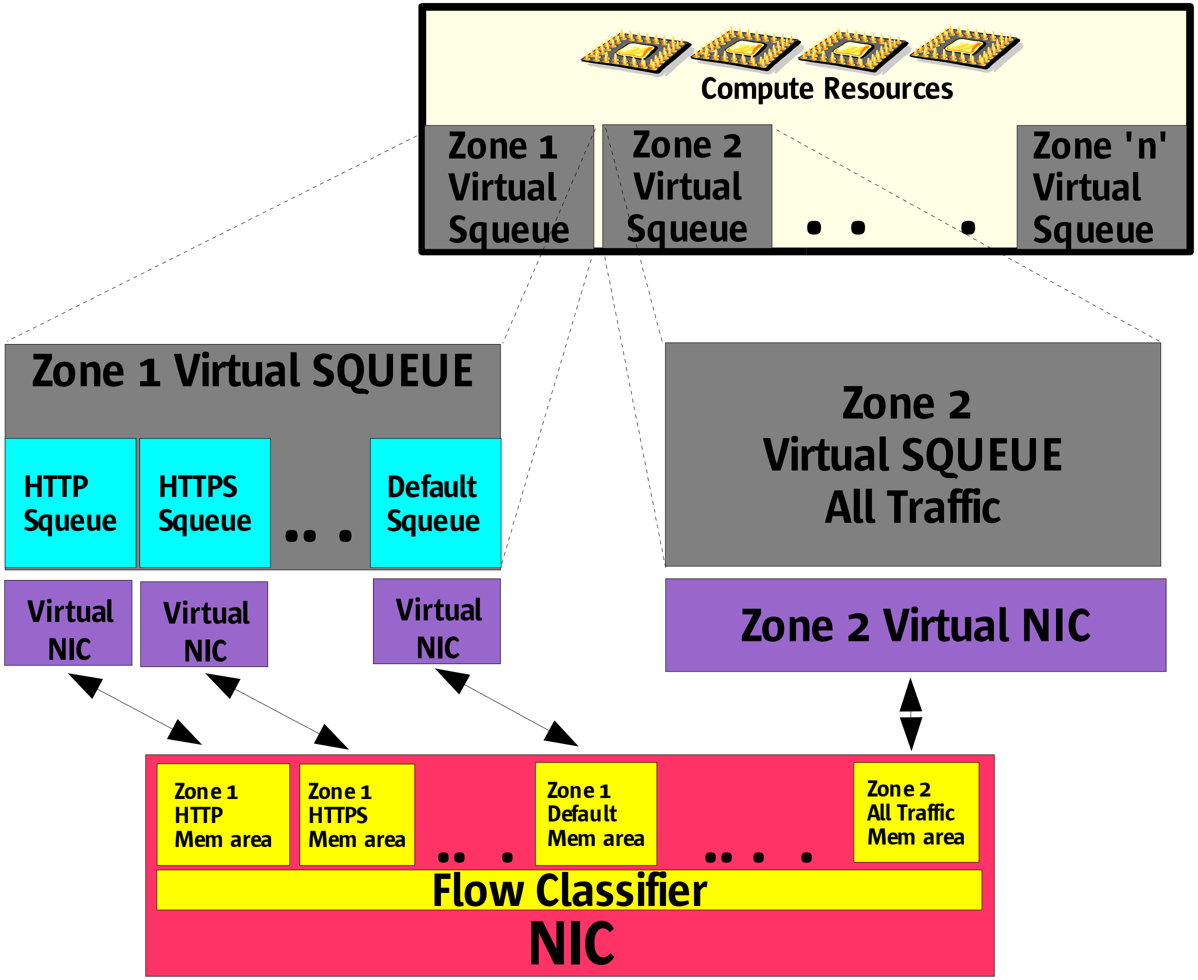


The Squeue switches the MSI interrupt per stack between interrupt and polling mode and controls the rate of packet arrival

# Virt. Stack per container

- Each Solaris container can have its own virtual stack with private routing table etc.
- When container is created, the B/W, priority and number of possible virtual stacks within the container is specified
- The Container administrator can configure the allocated virtual stacks to its own taste
- Each Container can have its own routing table, firewall, etc and tune it according to its requirement

# Virtual Stacks - Containers

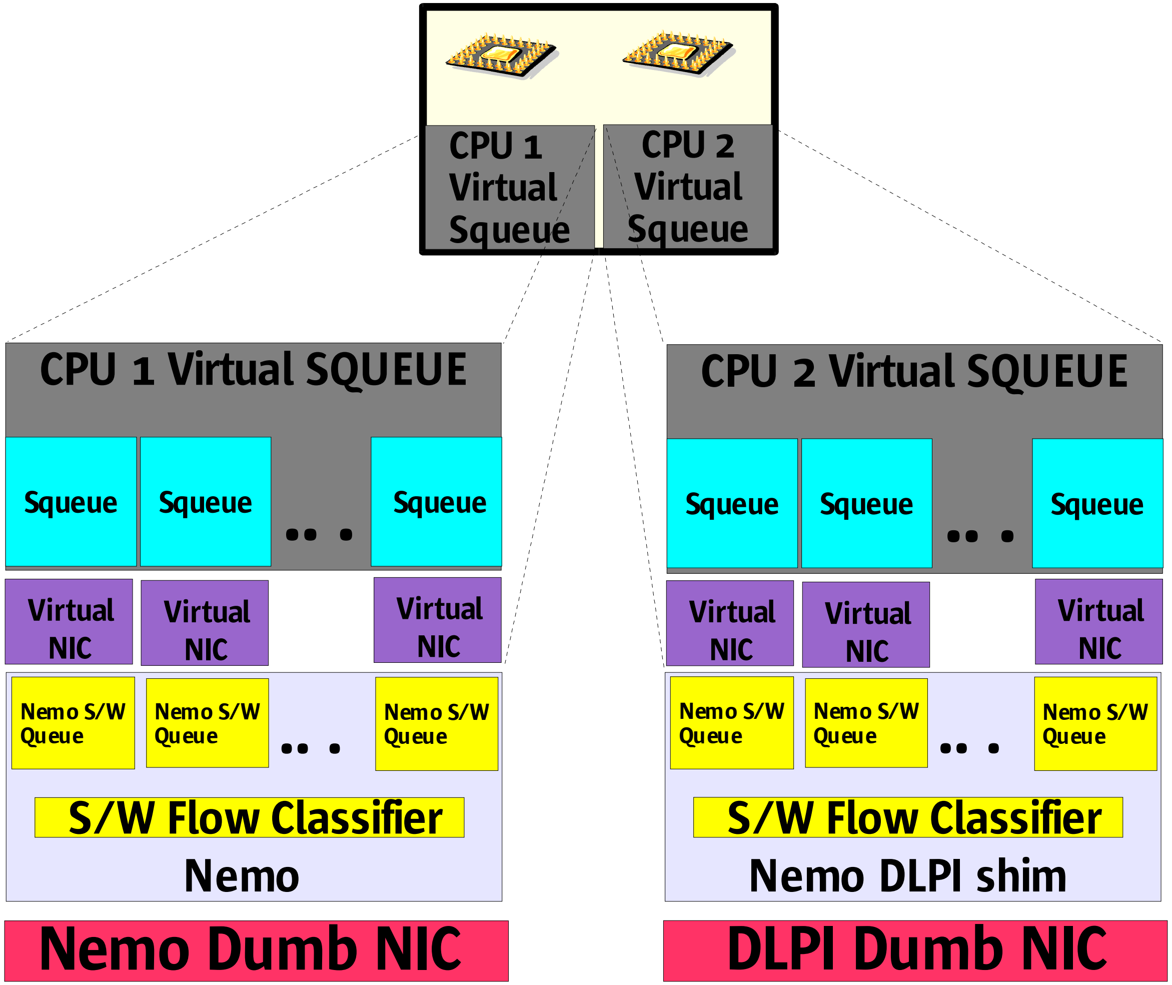


The Squeue switches the MSI interrupt per stack between interrupt and polling mode and controls the rate of packet arrival

# Dumb NICs

- The architecture supports non Nemo NICs as well as Nemo NICs which don't have flow classification capabilities
- We simulate multiple queues or memory area in the Nemo layer using a S/W flow classifier
- Nemo provides a DLPI shim layer for non Nemo drivers
- All the general 1Gb and 10Gb NICs (Sun's, Intel's, Broadcom, Neterion, etc) in future will support the flow classification and memory partitioning capability at no extra cost

# Virtual Stacks with Dumb NICs



CPU 1 Virtual Squeue

CPU 2 Virtual Squeue

CPU 1 Virtual SQUEUE

Squeue Squeue .. . Squeue

Virtual NIC Virtual NIC Virtual NIC

Nemo S/W Queue Nemo S/W Queue .. . Nemo S/W Queue

S/W Flow Classifier

Nemo

Nemo Dumb NIC

CPU 2 Virtual SQUEUE

Squeue Squeue .. . Squeue

Virtual NIC Virtual NIC Virtual NIC

Nemo S/W Queue Nemo S/W Queue .. . Nemo S/W Queue

S/W Flow Classifier

Nemo DLPI shim

DLPI Dumb NIC

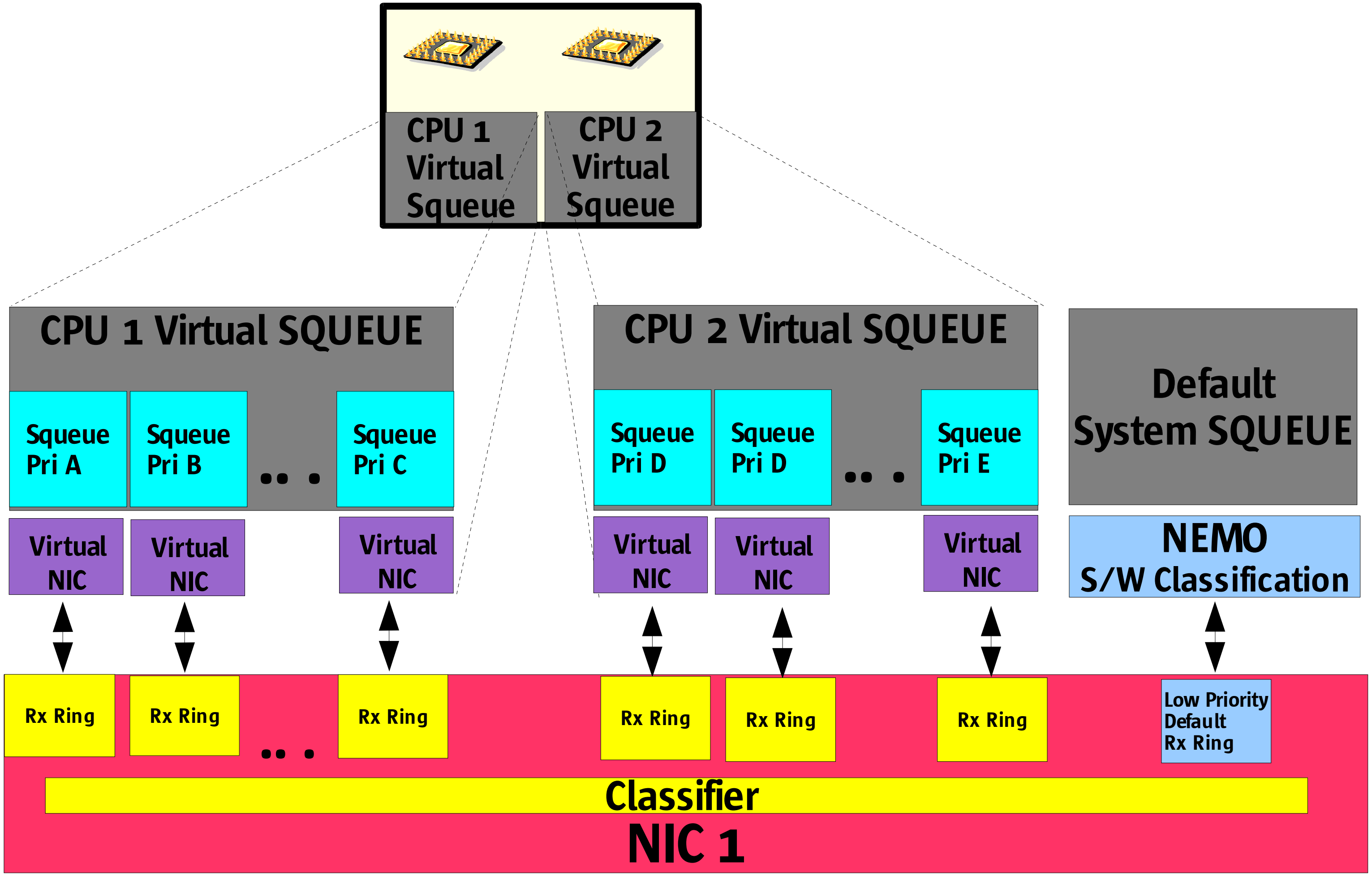
# Defense against DDOS

- Denial of Service attack (DOS) are a threat today
- DOS have the ability to cripple the entire grids 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 (limited resource) stack
- Connections transition to appropriate priority stacks after application authentication

# Application Level priority

- Crossbow allows administrators to prioritize virtual stacks based on application level priority
- The virtual stacks are programmed for a priority band and are not specific to traffic type
- During connection setup, the connection is mapped to appropriate priority virtual stack by dynamically programming the classifier (with application specifiying the priority)

# Differentiated Services



The Squeue switches the Rx Ring between interrupt and polling mode and controls the rate of packet arrival

# Fair Accounting System

- Finer grain accounting comes for free
- We can now do per squeue accounting to track usage by a container, service or protocol
- A userland daemon can pull the statistcis out at fixed interval and do accounting etc.

# Solaris Networking
## Today & Tomorrow

Sunay Tripathi,
Senior Staff Engineer,
Sun Microsystems, Inc.
Sunay.tripathi@sun.com