
intel
Leap ahead
MULTIPLY YOUR PERFORMANCE
AND MAXIMIZE YOUR EFFICIENCY
MUTIPLY YOUR KNOWLEDGE
Intel Developer
FORUM

開
放
的
열린
مفتوح
libre
मुक्त
ముక్త
livre
libero
ಮುಕ್ತ
开放的
açık
open
nyílt
פתוח
オープン
livre
ανοικτό
offen
otevřený
öppen
открытый
வெளிப்படை

USE

IMPROVE

EVANGELIZE

OpenSolaris™ Virtualization

Greg Lavender and David Edmondson
Sun Microsystems, Inc.

Talk Outline

- Motivation
- Key technology enablers
- Virtualization
- System-level virtualization
  - Processor virtualization
  - Storage virtualization
  - Network virtualization
- System-level config and management

It would be nice if….

- You had access to all of your files, email, music, photos, videos, and application services…
  - using any suitable device: PDA/Phone, tablet PC, laptop, desktop, Internet café computer, friend's computer, etc.,
  - without having to always sync everything
  - with no concern for computing capacity
  - with practically unlimited persistent storage
  - that is encrypted and always available securely
  - from any network location
  - with adequate bandwidth and performance
  - that is automatically replicated when needed
- Nobody provides the infrastructure…yet

Key Technology Enablers

- Throughput computing & large fast memories
  - multi-core processors with Intel® Virtualization Technology
  - per core caches, lots of RAM, fast SSD
- Storage density and access speed
  - high capacity, high speed, low cost per disk for SAN/NAS
  - multi-path I/O throughput using PCI Express* and SAS/SATA-II
- Network bandwidth
  - Multi-port PCI Express* 1-to-10 Gigabit Ethernet with jumbo frames & 802.3ad link aggregation
  - multi-flow bandwidth & QoS management
- OS processor, storage and network stack virtualization software

The First Step - Virtualization

- Free applications and their live state from specific physical machine resources
  - machine resource virtualization
  - live virtual machine migration
- Free files from physical disks and locations
  - storage virtualization, file system snapshots, on-demand replication over the LAN/WAN
- Free communication from bandwidth contention
  - enable more throughput via higher data rate LANs (10GigE)
  - network virtualization, QoS and bandwidth management
  - broadband wireless (e.g., WiMAX/MIMO/OFDMA)
    - enable higher data rate/throughput for last-mile WMAN

Virtualization

- Abstraction of physical machine resources
- Three kinds (broadly speaking)
  - software virtualization
  - Paravirtualization (PVM)
  - hardware virtualization (HVM)
- Reasons for virtualization
  - better utilization
  - sandboxing & isolation
  - reliability & manageability

System-level Virtualization

- Multi-processor, multi-core, multi-path, multi-flow, multi-port, multi-* …
- Exploit virtualization of system resources
  - processor resources
  - storage resources
  - network resources
- Good software is the key (of course)
  - requires a well-integrated operating system platform that can align and exploit the various virtualized resources to enable managed high throughput computing

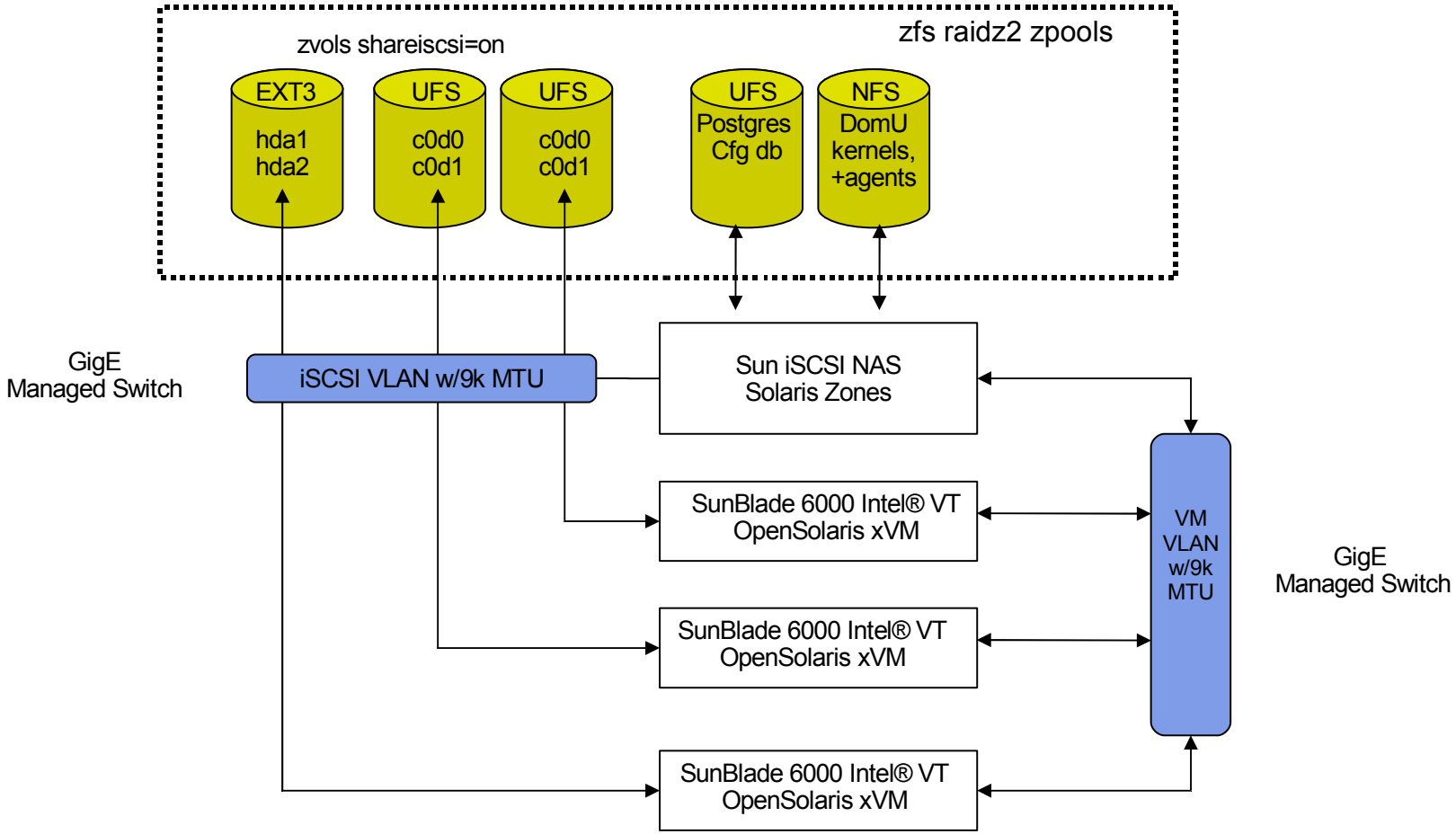
System-level Virtualization

- Processor virtualization with Intel® Virtualization Technology
  - VMware*, Parallels*, Linux* with Xen*
  - OpenSolaris™ with xVM
    - e.g., the Sun Blade 8000P has up to 240 x64 processor cores and 640GB of RAM
- Storage virtualization
  - SAN/NAS with virtualized storage pools & devices
  - OpenSolaris™ with the Zettabyte File System
    - ZFS + RAIDZ + NFSv4 + CIFS + iSCSI
- Network Resource Virtualization
  - managed bandwidth & QoS per physical NIC
  - OpenSolaris™ with Virtual NICs & 10 GigE NICs

Storage Virtualization

- Storage virtualization
  - virtualize disk storage using ZFS zpools and zvols
    - zpools are dynamic storage pools
    - zvols are virtual block devices
  - export zvols as iSCSI targets over switched GigE links
    - use VLANS or VNICs to isolate traffic and manage b/w
    - use jumbo frames and 802.3ad to do page sized transfers
  - configure VM file systems using zvols and iSCSI initiators
  - snapshot & live migrate zvols when VMs migrate, if needed
  - http://www.opensolaris.org/os/community/zfs/

Example System Config



Intel e1000g NICs used for all VLANs
with jumbo frames enabled & 802.3ad

Example: ZFS iSCSI zvols

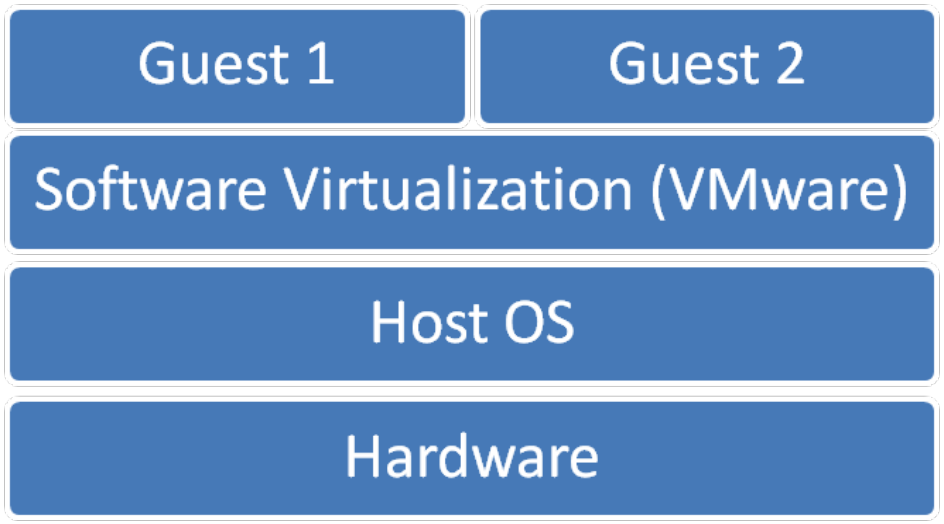
```
server# zpool create xenpool raidz2 /dev/dsk/c1d0s7 …
server# zpool list
NAME SIZE USED AVAIL CAP HEALTH ALTROOT
xenpool 424G 5.55G 418G 1% ONLINE -
server# zfs create -V 5G xenpool/vm1-disk
server# zfs set shareiscsi=on xenpool/vm1-disk
server# iscsitadm list target
server# mkfs.ext3 /dev/disk/by-path/ip-<address>:3260-<iSCSI Name>-lun-0
server# mount /dev/disk/by-path/ip-<address>:3260-<iSCSI Name>-lun-0 /some/path
server# debootstrap feisty /some/path http://us.archive.ubuntu.com/ubuntu
server# config the host and network config
server# umount /some/path
server# share -F nfs -o ro=<clients> /xenpool/kernels
client# mount server:/xenpool/kernels /xenpool/kernel

-- then configure the virtual machine to use the iSCSI enabled ZFS zvol
kernel = "/xenpool/kernels/vmlinuz-2.6-xen”
memory = 512
name = ”vm-ubuntu7.04”
disk =['phy:/xen/dsk/010000e0815926a100002a00469d2bde,hda1,w’,
       ‘phy:/xen/dsk/010000e0815926a100002a00469d2bdf,hda2,w']
root = "/dev/hda1 ro"
```

Network Virtualization

- OpenSolaris project Crossbow
  - http://www.opensolaris.org/os/project/crossbow/
  - Exploit multi-port, multi-core ethernet network interfaces
    - e.g., Sun's "Neptune" 10 GigE NIC
  - Virtual NICs (VNICs) instead of bridging
    - partition bandwidth based on service: e.g., iSCSI VNIC
    - mobile MAC addresses (supporting live VM migration)
    - better managed network resources per virtual machine
    - hardware flow classifiers, dynamic bandwidth management, QoS, per VNIC packet filters.
    - 802.3ad link aggregation
    - jumbo ethernet frames (9k MTU > vm page size)

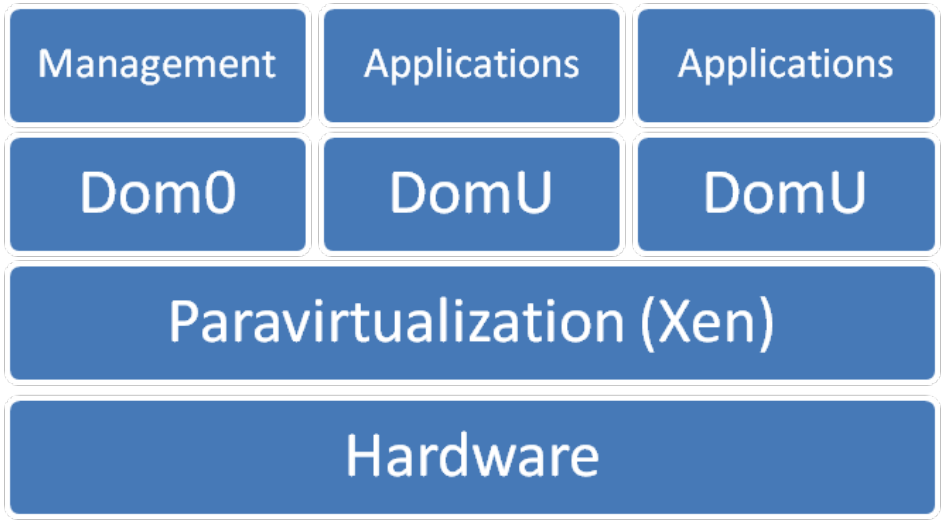
Software Virtualization

- Runs above the host OS
- Two methods:
  - Binary Translation
  - Trap-and-Emulate
- VMware* Workstation, Fusion* and Parallels*



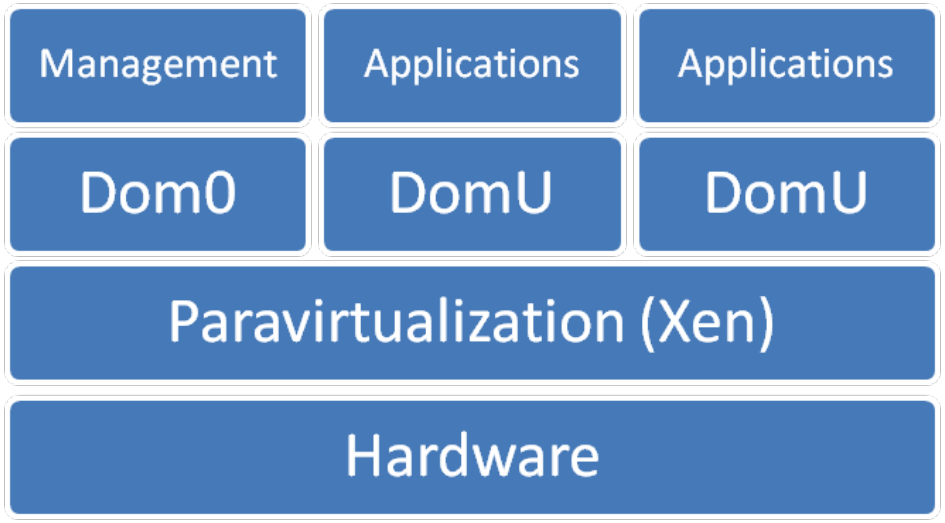
Paravirtualization

- Software that runs directly on the hardware
- Requires host (Dom0) and guest (DomU) operating systems to be modified
- Performance benefits
- Can take direct advantage of hardware virtualization features
  - Xen*, KVM, VMware ESX*



Host and Guest VMs

- Dom0 – Administrative
  - controls resources
  - admin tools for config, monitoring and management
- DomU – Guest
  - users generally unaware they are on a virtual machine
    - PVM vs HVM



Virtual Machine Live Migration

- Allows virtual machines to be migrated from one physical machine to another with minimal disruption of service
- Migrate entire VM state, including virtual memory pages and swap
  - No latent dependencies as with process migration
- Phases of Memory Migration
  - Push, Stop and Copy, Pull
- Remap network traffic to new machine
  - Remap IP traffic using unsolicited ARP reply

Live Migration Times

Job	Total Migration Time	Downtime
Quake 3 Server	7 Seconds	70 ms
SPECweb99	71 Seconds	210 ms
Stop&Copy	30 Seconds	3.5 secs

System-level Config & Mgmt

- Simplify system-level config, monitoring & management
  - Java JMS/JMQ technology, Postgres, NetBeans
- Java Agents
  - Three types
    - System configuration agent
    - Physical server agent per physical machine
    - Virtual machine agent per virtual machine
  - Collect information and execute actions on physical and virtual machines
  - Communicate using the JMS/JMQ message bus

Current Status

- Processor + Storage virtualization
  - Software can scale to manage hundreds of virtual machines, just add more boxes/blades
- OpenSolaris* x64 as Dom0 has compelling features
  - planning to implement network virtualization
    - using dual/quad port 1GigE & 10GigE NICs
  - beginning to exploit VNICs and GigE for SAN/NAS and VM live migration
- Working towards mobile VMs
  - live migration to/from laptops
  - requires live zvol replication/migration for fully disconnected operations

Lightweight Virtual Machines

- Live migration of full OS is heavy weight
  - no residual dependencies because full state moves
  - but high dependence on full OS file system
- Suggests the idea of LVMs consisting of minimal run-time needed to support specific application(s)
  - e.g., virtualized JVM
- Virtualized network application-specific appliances
  - LVMs targeted at application-specific services
    - NAS, firewalls, routers, app servers, VoIP, IPTV, etc.
  - more horizontally scalable, faster migration, smaller footprint, well-defined file system dependencies
  - can more accurately characterize and predict system resource requirements and establish higher assurance

Network Virtualization

- OpenSolaris* xVM allows multiple virtual machines to run on a single physical machine
  - each virtual machine requires network access
- Network ports are becoming faster and more capable, yet there are limits on the number available:
  - 1G everywhere, 10G more common
  - 4 ports normal, >16 ports rare

Constraints are required

The ability to constrain virtual machines' use of the network is paramount:

- control access to physical resources
- control access to network services
- limit bandwidth use
- limit the ability to damage other hosts

Challenges

Difficult to associate activity with “billable” entities:

- protocol processing in interrupt context
- anonymous packet processing in the kernel

Difficult to segregate traffic:

- common packet queues

Performance suffers:

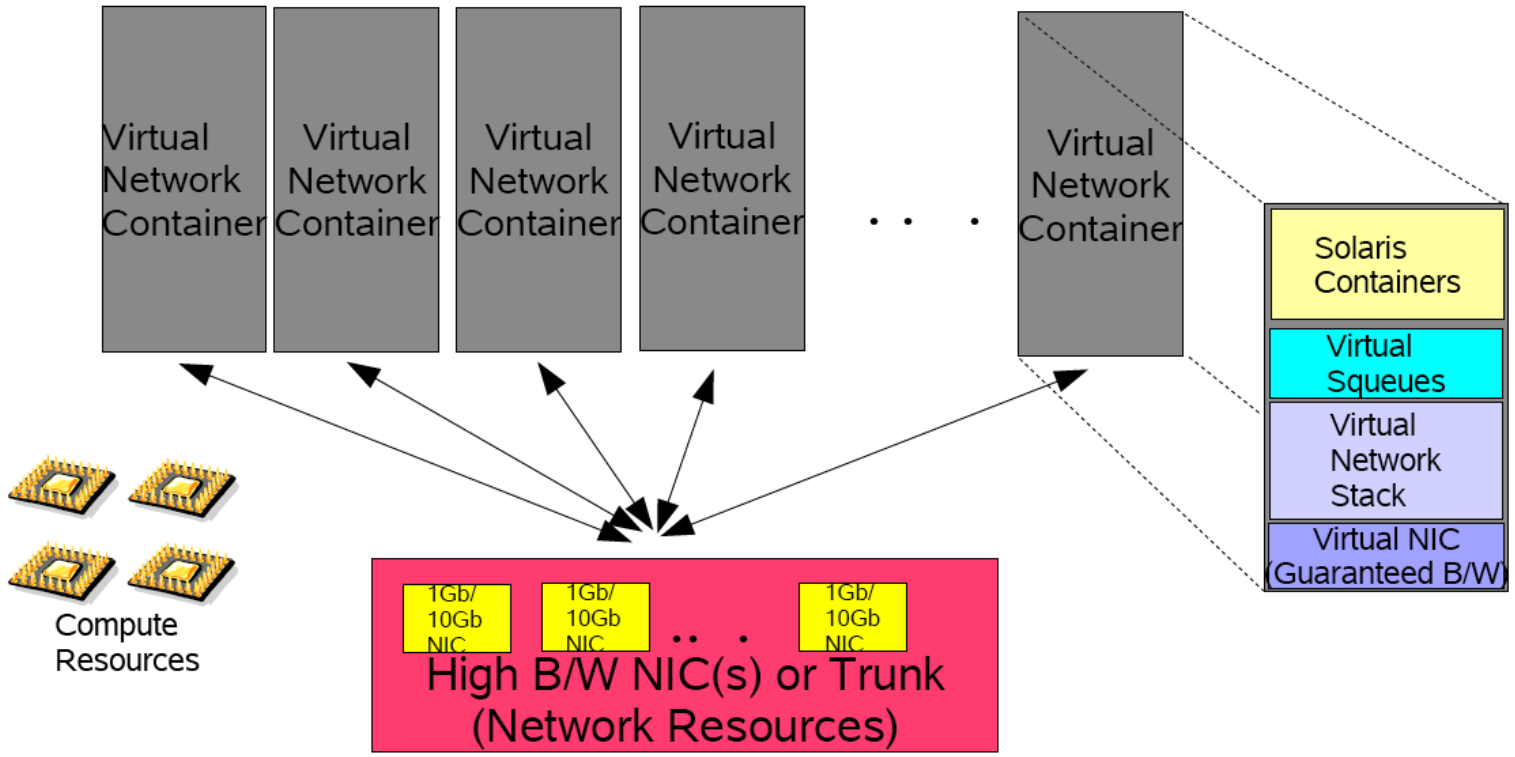
- extra processing to ensure fairness, resource control, etc.

Crossbow

An OpenSolaris* project to improve network virtualisation:

- partition NIC memory, DMA channels, etc. into multiple “Virtual NICs”
- use a flow classifier to build a virtual stack on each VNIC
- independently switch individual VNICs between interrupt and polling mode
- control the rate of packet arrival for a VNIC independently of all others

Crossbow Architecture

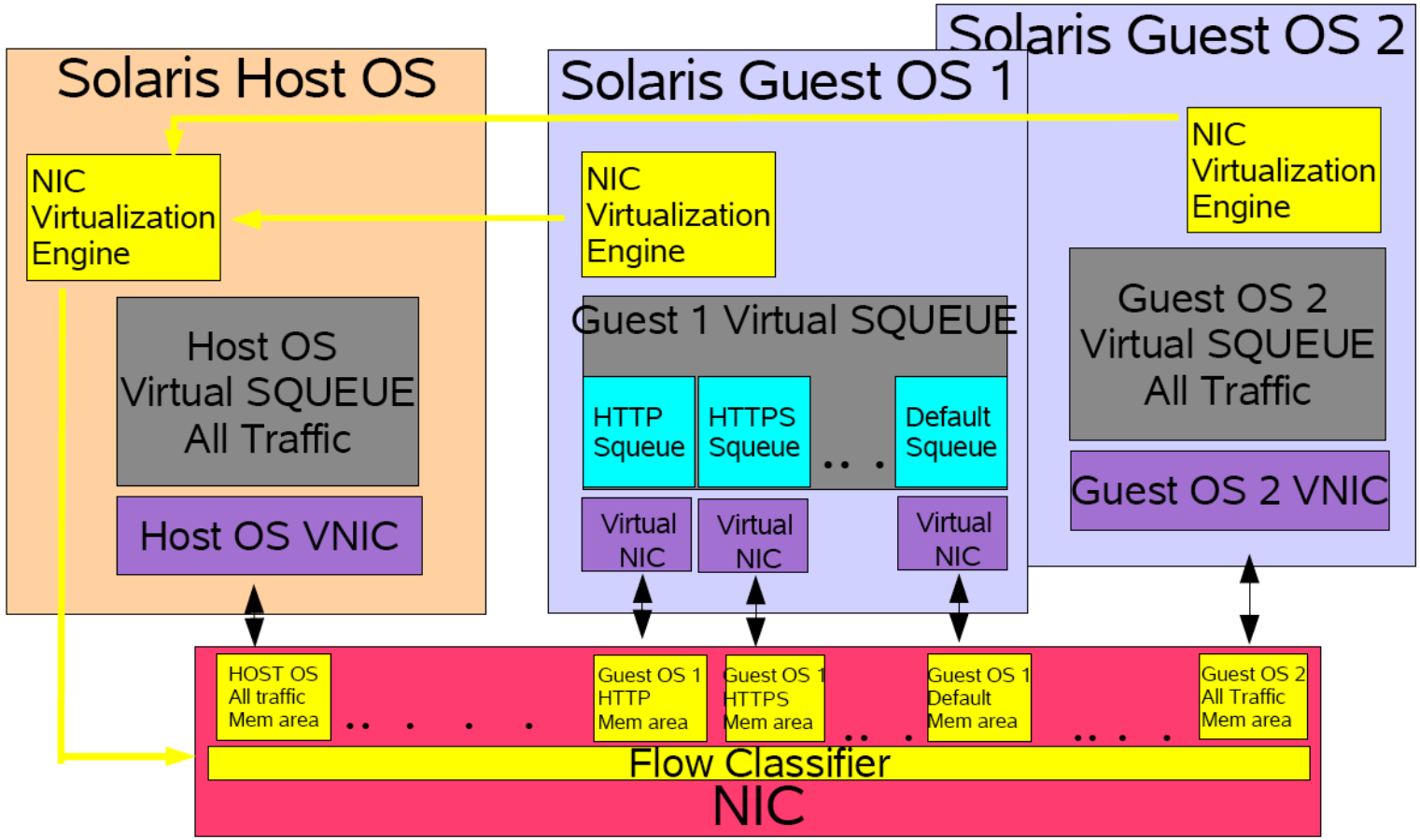


Crossbow and xVM

Provide network access to guest domains via a VNIC

- guest domain traffic is segregated from that of other domains
- hardware traffic classification

Crossbow and xVM



Status

Basic VNIC functionality used by xVM in OpenSolaris* build 75

- 1G throughput for guest domains on a par with underlying physical machine
- CPU cost is higher, latency is higher
- 10G testing just started

Future

- Resource control for VNICs soon
- Improved inter-domain protocol implementation:
  - hypervisor based copy rather than page flippping
  - multicast control
- Improved domain 0 implementation:
  - copy-on-write access to guest domain packets in control domain
- Hybrid IO:
  - direct access to virtualisable hardware in guest domains

開
放
的
열린
مفتوح
libre
मुक्त
ముక్త
livre
libero
ಮುಕ್ತ
开放的
açık
open
nyílt
פתוח
オープン
livre
ανοικτό
offen
otevřený
öppen
открытый
வெளிப்படை

USE

IMPROVE

EVANGELIZE

Thank you!

Greg Lavender and David Edmondson
Greg.Lavender@sun.com and dme@sun.com

“open” artwork and icons by chandan:
http://blogs.sun.com/chandan

Please fill out the Session Evaluation Form to win $100 Gift card!

How?

- Use your IDF Flash Drive
- Go to a IDF Survey Stations
- Go to Intel.com/go/myidfevals

There will be daily drawings for 5 Gift cards – The more evaluations you fill out the more chances to win!

Please note: There will be one gift card per person per day!
Please see terms and conditions for drawing in Program Guide

Thank You for your input, we use it to improve future Intel Developer Forum events