

Network Virtualization and Resource Control -

Project Crossbow

White Paper
June 2009

Abstract

OpenSolaris Project Crossbow for Network Virtualization and Resource control brings virtualization to OpenSolaris networking, including resource management capabilities around the creation and management of virtual network interfaces.



Table of Contents

Executive Summary.....	1
Introduction.....	1
Architecture.....	3
Virtual Networking Scenarios	5
Summary.....	7
Other Resources.....	7

Executive Summary

The OpenSolaris Crossbow project implements a wide reaching re-architecture within the OpenSolaris networking stack to bring a variety of advantages:

- Providing a fully virtualizable network environment for more effective sharing of networking resources and increase the scope for server consolidation projects.
- Providing networking resource management capabilities to allows organizations to meet quality of service goals for networking.
- Decreasing latency and increasing throughput particularly as network load increases. Improve network performance in virtualized (and non-virtualized) environments.

The combination of open source software and industry standard hardware opens a variety of new avenues for OpenSolaris in embedded applications and in consolidation projects where the value of network virtualization and resource management offer capabilities other general purpose operating systems can not offer.

Introduction

The Crossbow Project is an ambitious effort to re-architect OpenSolaris networking to accomplish three aims

1. Crossbow network virtualization enhances the ability to consolidate server workloads. There are many facets to this virtualization.
 - By virtualizing the concept of the hardware Network Interface Controller (NIC) into Virtual NICs (VNICs), Crossbow promotes more effective sharing of networking resources. The VNIC construct allows dividing a physical NIC into multiple virtual ones to create OS-enforced isolated and dedicated network stacks from physical NIC to application.
 - Crossbow virtualizes a physical LAN switch through the “Etherstub” construct.
 - Virtualization is not only appropriate for dividing a physical NIC, but also for aggregation purposes. Aggregating two physical NICs to a single virtual one and dividing the result into, for example, 3 VNICs, allows better sharing of the network resources and provides redundancy should one of the physical links fail.
 - The industry standard Virtual LAN (VLAN) construct is supported, allowing NICs and/or VNICs to be assigned to a VLAN. This allows, in an environment with switches and routers that also support VLANs, end to end traffic isolation even though the traffic may be running on a shared physical link.

Other OS networking elements can be brought into play, particularly OpenSolaris router and firewall. Taken together, these elements enable building an entire networking topology within one computer system - useful for architecting/prototyping, testing and even deployments. For the latter case, a vender could offer a multi-system performance solution at the high end, and a less expensive consolidated solution offering the same capabilities, but at a reduced cost.

2. Crossbow network resource management allows organizations to meet quality of service goals for networking. Resource Management allows specifying guaranteed resource levels as well as resource maximums to system processes. Traditionally this has meant giving administrators control over CPU and memory resources to ensure that certain applications can get minimum resource levels no matter what other demand there is for those resources. Resource management can also be used to prevent applications from taking too much of CPU or memory resources. Crossbow extends Resource Management to networking in three ways:
 - Allow specifying the CPU resources assigned to a NIC port or Virtual NIC.
 - Allow specifying bandwidth limits for a Virtual NICs.
 - Allow specifying priorities for types of traffic.

These networking resource management capabilities enable enforceable organizational network sharing policies.

3. Crossbow can increase network throughput by more efficiently scheduling and handling packets. The best performance gains typically come with the latest generation intelligent NICs with packet classification and multiple receive and transmit ring buffers that Crossbow can manage. There are many aspects of the design of Crossbow that facilitate increased efficiency, but one of the major ones is dealing more efficiently with inbound packets. See the Architecture section below for details. The areas in which Crossbow shows potentially the best performance improvements are latency, scalability, high volume small packet traffic, and small packet forwarding. See the performance section below.

Crossbow also enhances the value of Sun's virtualization technologies for server consolidation using OpenSolaris as either the delivery OS or the underlying host for virtual machine guest operating systems.

Operating System Virtualization

Solaris Containers

Solaris Containers are a virtualization technology that allows one operating system instance to offer multiple virtual isolated OS environments. The key advantage of this approach is that while applications see an environment that looks like a dedicated OS, in reality these Container virtual OS environments are running on one OS. As a result Containers, when compared to hypervisor virtualization technologies, can make much more efficient use of system resources because one OS oversees the CPU, memory, and network resource allocation. Containers also have excellent scaling properties because of the extremely small system overhead they place on the OS. And finally creation and destruction of Containers is a light weight task that facilitates their use in dynamic environments.

Prior to Crossbow, applications running on Containers could access network interfaces but if a dedicated network stack for the Container's network interface was desired, a dedicated physical NIC or dedicated VLAN was required. With Crossbow a Container can be assigned as many Virtual NICs as needed and each will have it's own dedicated stack whose bandwidth, priority, and CPU allocation can be managed. Applications can be written to access the network without needing to be aware whether the interface is virtual or real. With Containers and network virtualization, one could consolidate multiple servers (and services) on to one instance of OpenSolaris.

Hypervisor Virtualization

xVM Hypervisor

xVM Hypervisor is a virtualization technology based on running various x86 Operating Systems on an OpenSolaris-based hypervisor. This capability will typically be provisioned through Sun xVM Ops Center scheduled to be released after OpenSolaris 2009.06.

Logical Domains

Logical Domains are a virtualization strategy for SPARC systems based on a hypervisor architecture. Each domain (guest OS) will have a virtual network device that will be used to access the underlying NIC and other domains. Associating networking properties (bandwidth, priorities, CPU resources) to LDOM guests from the service domain is currently not possible. Those features will be available in a future release of OpenSolaris.

Type 2 Virtualization

VirtualBox

VirtualBox allows running a variety of x86 operating systems as application on a variety of hosts, including OpenSolaris. Networking virtualization may not be interesting to the typical laptop user but for the enterprise, VirtualBox can be an attractive virtualization technology. VirtualBox can run in a 'headless' mode- not tied to a particular display device. Typically a single server would be set up as a test bed for a variety of Linux or Microsoft OS releases. The operator would interact with those test system through VNC sessions. If VirtualBox is run on OpenSolaris, all the advantages that Crossbow brings to network virtualization and network resource management can be passed to the VNICs used by the Linux and Microsoft guest operating systems.

From the preceding it should be clear the impact the Crossbow architecture has on network virtualization and resource management. Let us now turn to details about the architecture to understand more about how key features are delivered and to understand why Crossbow is about more than just virtualization. Crossbow is a redesign that impacts the entire network stack and particularly the data link layer to provide the foundation for the next generation network stack.

Architecture

The fundamental building blocks of this new architecture are Virtual NICs or VNICs- a construct for dividing a physical NIC into multiple virtual ones. To summarize, a VNIC device is accessed, from the applications viewpoint, exactly like a physical NIC. Applications do not need to be aware they are accessing the network though a VNIC. The characteristics of a VNIC- the bandwidth, what CPU resources are assigned to handle it, and priorities can be dynamically controlled.

Crossbow is designed as a fully parallelized network stack structure. Think of a physical network link as a road, then Crossbow allows dividing that road into multiple lanes. Each lane represents a flow of packets, and the flows are architected to be independent of each other- no common queues, no common threads, no common locks, no common counters. Tying this architecture to a modern NIC gives a block diagram as below

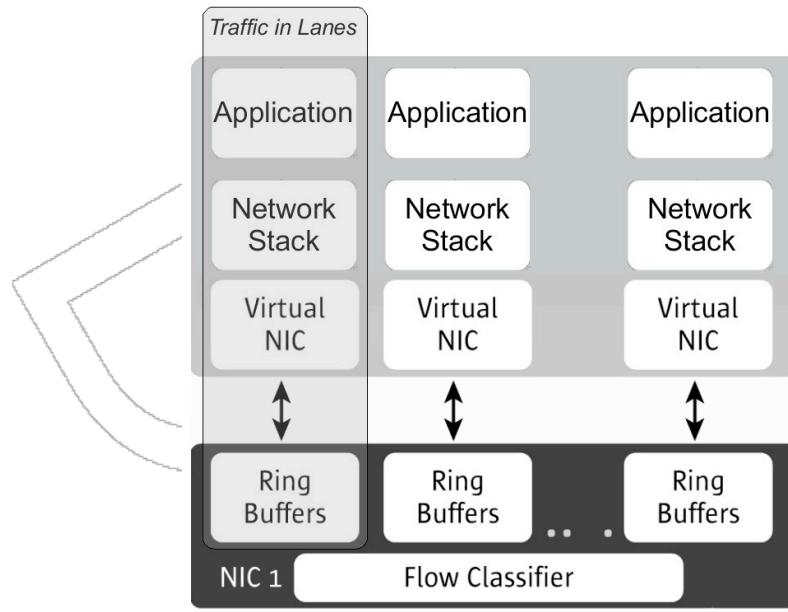


Diagram 1 – Traffic Lanes with Classification by Intelligent NIC

A key element of the design is flow classification, and in the case of diagram 1, the flow classification is done in the NIC. Once the incoming flow has been classified, it enters its own private lane indicated by the shaded area. The NIC has a Transmit (Tx) and Receive (Rx) ring buffer dedicated to each lane. The data is structured as a ring because Rx data that is not transferred into the OS will eventually get overwritten by new incoming data. Packets are not supposed to be dropped but we'll see below that if the load is so great that packets must be dropped, it's much better to drop them in the NIC before they further tie up CPU resources.

A hardware classifier is not a requirement as we will see below, but it does offer the best performance. The classifier can be programmed to classify on a range of OSI Layer 2, Layer 3, or Layer 4 attributes-

- MAC address
- Source or Destination IP address
- Protocol
- Port

Operationally this means that HTTPS traffic can be handled by one lane, HTTP traffic by another, FTP by third and additional or alternate controls based on source or destination address can be added. Diagram 2 illustrates assigning specific traffic to the Queues identified in Diagram 1.

In an environment where security is important, hardware classification is generally perceived as a more secure solution. One could create a Solaris Container to house the application, assign a VNIC to that Container, and force all the Container traffic through dedicated hardware classified 'lanes'. From an OS perspective, whether hardware or software enforced the isolation in the lanes is preserved, but as noted, hardware assisted classification is generally a more appealing solution.

Not every NIC has a flow classifier, and even if it did, the number of lanes required by the OS may be more than can be supplied by the NIC. For that reason Crossbow also includes a software layer (see Diagram 2 below) for dealing with dumb NICs.

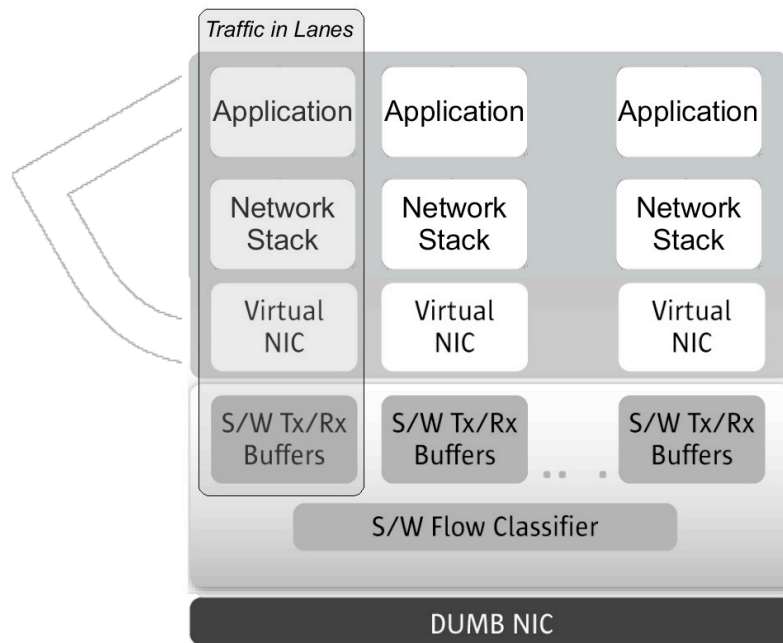


Diagram 2 – Traffic Lanes with Classification by the OS

In this case all the architecture for supporting VNICs is part of the OS- a software classifier with dedicated Transmit and Receive rings to create the lanes entirely in software. This same approach allows supporting applications with requirements for say, 20 VNICs on a system with a classifier that only feeds 16 hardware rings. The software enables creating real lanes from the hardware as well as virtual ones that share a real lane.

Crossbow is both network virtualization and network resource management. The ability to specify the characteristics of these lanes is an important element of the design. Crossbow allows managing resources through three mechanisms-

- Setting bandwidth limits
- Setting traffic priorities
- Assigning number of CPUs to handle the traffic

Bandwidth limits can be used to guarantee minimum resource levels or to prevent a VNIC (or NIC for that matter) from using more than it should, from an operations policy perspective. One could break up a physical 1Gb port into 3 virtual ones, and assign the VNICs 100Mb, 100Mb, and 800Mb rates. Assigning a higher bandwidth to the last VNIC reflects the operational importance of the applications using that interface and therefore can be used very effectively to help applications meet Quality of Server (QoS) goals.

These resource management capabilities tie cleanly to the architecture. Here are two examples:

Pushing flow control as close to the NIC is important. By using the ring buffers in NIC, Crossbow can In traditional flow control implementations, just bringing a packet into an operating system queue from which

it will potentially be dropped expends a great deal of the total processing cost of packet handling. If the flow is metered at the NIC the less impact a dropped packet will have on the system. For NICs that manage their own Tx/Rx rings, dropped packets incur no CPU overhead.

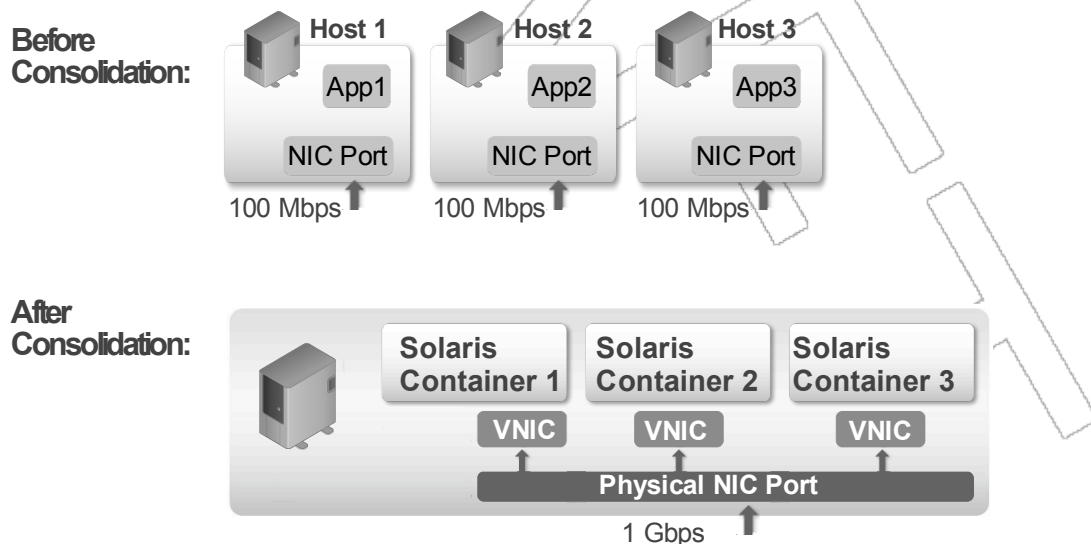
Other QoS implementations tend to be a layer inserted into the network stack- a choke point that doesn't take into account contention for resources elsewhere. With OpenSolaris Resource Management it is possible to provide an application with the minimum CPU, memory, and networking resources to holistically enforce Quality of Service requirements- just setting bandwidth limitations is not enough.

Network resource controls are not limited to VNICs. They can also be used to manage physical NICs- or more accurately since a NIC may have multiple physical ports- to manage each of those ports by setting bandwidth, CPU, and priorities.

One other element of the architecture is worth highlighting, the way Crossbow manages interrupt handling. In low utilization mode, packets are handled in the traditional interrupt manner. In lower speed, large packet traffic this is an adequate approach but in high speed, high load networks this can have a negative effect on overall system throughput. There are many techniques to mitigate this, but fundamentally an interrupt per packet is simply not efficient on a busy network. Crossbow automatically switches from interrupt mode to polling mode when the packet arrival rate exceeds a threshold. Polling has a key advantage for busy networks - one poll by the driver can potentially return a chain of many packets in one operation, far more efficient than one packet per interrupt.

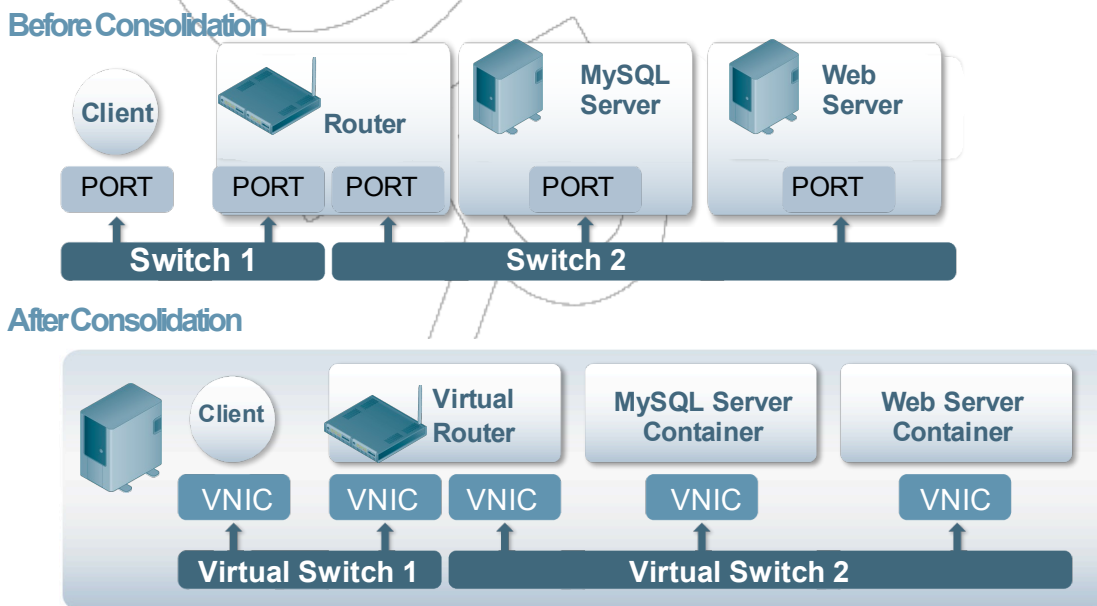
Virtual Networking Scenarios

The value of virtual networking as described above, should be apparent. The combination virtual NICs and the ability to manage those resources makes an excellent match to virtual server technologies. For example, the ability to carve up a 10Gb or 1Gb physical interface into smaller 'lanes' and assign those to an xVM Hypervisor or Solaris Container is compelling because the hypervisor (in the case of xVM Server) or the Solaris operating system (in the case of Containers) determines bandwidth, priorities, and or CPU resources assigned to the virtual entity.



In this case three systems and their applications have been consolidated on one OpenSolaris system with three Solaris Containers. In this example the consolidated server was moved to a 1Gigabit per second (Gbps) network, so the host bandwidth limits prior to consolidation are enforced after consolidation even though on a new higher speed network. Many different scenarios are possible. If the consolidation system had enough ports, the network topology of the environment before consolidation could be duplicated after consolidation. Or alternatively different bandwidths could be assigned based on importance of the application's bandwidth requirements. The main point is that Crossbow gives an administrator much more control of the network resources while still enjoying the advantages of server consolidation.

A more sophisticated virtualization project is pictured below



4

In this case one host runs a MySQL database, another runs a web server, and the third is the client accessing a web service that relies on the database. A developer might consolidate this environment on one system to work on some element of the interaction between these three system. The testing organization might also use this consolidation to simplify the hardware requirements for testing the entire application environment. Note that we've added two new virtualization elements for this example, a switch and a router. The virtual switch makes it possible for the Containers to communicate directly between each other. In the first simple example of this section the Containers could not communicate directly within the host. Either a virtual switch or an external physical switch would need to be added. The virtual router is the open source Quagga project, included with OpenSolaris 2009.06. OpenSolaris also includes the IPFilter firewall, another element that could be used in server consolidations.

In general OpenSolaris and Crossbow offer a wide range of capabilities for building a "network in a box". Firewalls, virtual switches, and routing- all those building blocks are available. This is particularly valuable in

consolidation projects where not only does one consolidate multiple services on one system, but one can eliminate physical network interconnects by virtualizing the networking.

A final example of the power of Project Crossbow is its use as part of a defense against Denial of Service (DoS) attacks. As pointed out above, Crossbow allows creating communication lanes from the physical network port on the NIC to the application. When the NIC hardware supports both flow classification and receiver ring buffers the Crossbow architecture enables much more efficient way to mitigate the effects of a Denial of Service attack. Bandwidth limits will pace the kernel in polling the VNIC (or NIC) for packets. When packet rates are very high as in a DoS attack, packets will inevitably be dropped in the receive buffer. Not involving the kernel in dropping packets has a significant impact on decreasing the load on the CPU that in turn diminishes the impact on other services of the DOS attack. And the dynamic nature of VNICs means that it would be possible, having identified a DOS attack, to change the characteristics of the classifier to drop packets based on a variety of filtering rules or further limit the bandwidth of the flow.

We've mentioned a number of examples of the use of Project Crossbow, particularly in conjunction with Virtual Machines, e.g. Sun xVM Server, and Solaris Container environments. Project Crossbow is not dependent on working only in virtualization environments. One could easily see the resource control aspects of Project Crossbow coming into play in other application scenarios. OpenSolaris itself could be configured to be a router and the parallel network stack processing enabled by Project Crossbow enhances the value of such a configuration.

Summary

Project Crossbow is next step in the evolution of Solaris networking stack, bringing bandwidth resource control and virtualization as key elements of the overall design. The architecture dovetails with current trends in NIC design to extract the best performance, and also dovetails with current trends in server virtualization to enhance the ability to create virtual computing environments that can coexist on one hardware platform, but behave like independent systems with their own dedicated resources.

Other Resources

See www.OpenSolaris.org/os/projects/crossbow, the OpenSolaris project page for Crossbow.

The www.opensolaris.com/networking site has specific information related to OpenSolaris 2009.06 Crossbow and other networking projects. You will also find pointers to additional Crossbow resources,

See the

for more information on Project Crossbow.

