IT@Intel White Paper

Intel IT
Data Center Efficiency
Software-Defined Networking
April 2014

Adopting Software-Defined Networking in the Enterprise

By virtualizing the network through a programmable interface, we can offer better support for Intel application developers

Executive Overview

Intel IT is adopting software-defined networking (SDN) to enable on-demand provisioning of networks and network services. By virtualizing the network through a programmable interface, we can offer better support for internal customers—Intel application developers—who work in a fast-paced agile development environment. These customers need to be able to access network resources without having to negotiate a bottleneck created by configuring networks and provisioning services.

SDN can help us increase business value from the virtual machines (VMs) in our data centers and can offer the following additional benefits:

- Reduced network-provisioning time
- Simplified network creation process in a self-service environment
- Reduced service costs through improved network management efficiency

Recently, the SDN industry made several performance and scalability improvements in SDN components. For the last two years we have been evaluating the following to help us determine the benefits of deploying SDN:

- Defining which use cases warrant the implementation of SDN
- Testing and analyzing different SDN architectures to understand the data flow between nodes
- Updating the roles and responsibilities for the employees who will be creating networks in a self-service environment but who might not have a background in network engineering

As we continue to adopt SDN and deploy it across our virtualization environment, we will continue to evaluate how SDN components and architectures can remove barriers to network services for our customers, enabling them to deploy applications more quickly.

Sridhar Mahankali
Cloud Network Engineer, Intel IT

Sanjay Rungta
Senior Principal Engineer, Intel IT

Contents

IT@INTEL

The IT@Intel program connects IT professionals around the world with their peers inside our organization—sharing lessons learned, methods, and strategies. Our goal is simple: share Intel IT best practices that create business value and make IT a competitive advantage. Visit us today at www.intel.com/IT or contact your local Intel representative if you'd like to learn more.

BUSINESS CHALLENGE

Intel IT operates approximately 55,000 servers in 64 data centers that support more than 104,000 employees.[1] For more than a decade, Intel IT has been virtualizing the servers in our office, enterprise, and services data center environments.

In 2000, it took more than 90 days to provision a virtual machine (VM). Today we can provision a VM on demand to speed time to market for business units (BUs) that are developing applications. More than 85 percent of the workload in office, enterprise, and services data centers is running on VMs. However, network resources have remained a bottleneck for on-demand provisioning, especially in our office and enterprise data centers. Therefore, we are seeking a solution to speed network services provisioning, enabling us to better serve the business.

Evolving Cloud Use in Our Data Centers: Storage, Compute, and Network

As illustrated in the timeline in Table 1, we can trace the use of virtualization at Intel back to 2000. By 2009, only 12 percent of the enterprise-computing environment was virtualized. Also during that time it took more than 90 days to provision a VM, and reliability of the virtualized environment varied.

In 2009, we began virtualizing most of our office and enterprise data center server environment. Just three years later in 2012, 75 percent of the enterprise-computing environment was virtualized, compute provisioning was on demand, and availability was 99.7–99.9 percent. In 2014 and beyond (with 85 percent of enterprise computing virtualized), we plan to continue virtualizing compute, storage, and network services, as well as support on-demand provisioning of these services. This will help us keep up with the dramatic increase in Intel's application development ecosystem.

With the legacy network-provisioning model that we used before the SDN model, provisioning a network required four to six hours of total work time—and with starts and stops, a backlog of requests to provision networks, and other complexities, the network-provisioning process could take as long as 13 days. The delay cascaded across other application-landing tasks such as security setup, local and global load balancing, and domain name system (DNS) configurations.

To keep up with the increasing volume of requests, we need to shift toward deploying programmable network components in our

[1] To define "data center," Intel uses IDC's data center size classification: "Any room greater than 100 square feet that houses servers and other infrastructure components."

Table 1. Since 2000, we have gone from a small percentage of virtual machines (VMs) to a private cloud in 2010 to a hybrid cloud today.

	2000-2009	2010	2012	2014+
	TRADITIONAL HOSTING	MAINSTREAM VIRTUALIZATION	INTEL CLOUD 1.0	HYBRID CLOUD 2.0 CONVERGED CLOUD
Virtualized Servers	12%	42%	75%	85%
Provisioning	More than 90 days	10 days	On-demand compute	On-demand compute, network, and storage
Service Requests	Manual	Manual	Some on-demand	Full self-service
Reliability	Variable server reliability	99.7% VM reliability	99.7%-99.9% availability	99.9% availability capable

office and enterprise data center server environment so we can dynamically provision networks on demand. While virtual LANs (VLANs) and virtual routing and forwarding (VRF) enable us to virtualize certain network resources on top of the physical network, those technologies by themselves do not have the programmability that is necessary for automation and self-service.

Expected Business Benefits of a Programmable Virtualized Network

Integrating programmable network services components into the data center network can help increase agility, simplify the provisioning process, and reduce total cost of ownership (TCO).

- **Agility.** Configuring networks and provisioning services such as load balancers and firewalls can create a bottleneck in the overall application-landing process. With a programmable virtualized network, we can dynamically provision network resources from a centrally managed console or build additional customized automation using APIs. We anticipate that most of our data center network services can be provisioned on demand without network administrator interaction.
- **Simplicity.** A virtualized network with on-demand provisioning can reduce the burden on the network administrator and make it easier for our customers to create and manage their own networks for application development.
- **TCO.** A virtualized network with the ability to automate the configuration of network equipment can reduce the need to manually configure individual network components. We expect this programmable global view of the network to help reduce the cost of our network operations.

SOLUTION

Software-defined networking (SDN) capabilities, illustrated in Figure 1, enable us to virtualize the enterprise data center network similar to the way we virtualize servers. SDN separates the control plane from the data plane to provide a centralized, programmable interface for the SDN-enabled portion of the network. By making network components programmable, SDN provides for newer multitenant models and distributed security access control, enhances scalability, and enables rapid service innovation through network applications.

The OpenStack*-based private cloud environment we have been engineering and adopting in our enterprise data center network provides a framework for integrating SDN elements. OpenStack's Neutron APIs enable the provisioning, modification, and deletion of network services through both open source and proprietary programmable network components.

SDN capabilities are relatively new to the industry and are continuing to mature. Intel IT is evaluating SDN to determine whether it will decrease network-provisioning time, simplify the process of provisioning network services, and lower service costs. Our evaluation has centered on testing and analyzing the performance of SDN controllers. The SDN controller acts as the core of SDN by presenting a logical view of each tenant's network and managing flow control between network hardware and tenants.

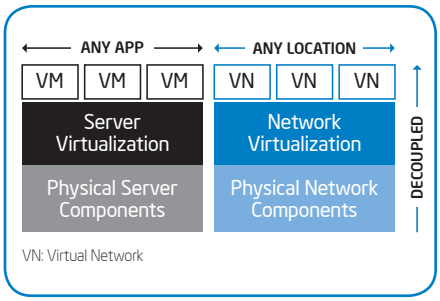


Figure 1. Just as server virtualization creates multiple virtual machines (VMs) decoupled from physical servers, network virtualization—that is, software-defined networking (SDN)—creates multiple virtual networks decoupled from the physical network components.

Networking Evolution: From Hardware Lock-In to Open Software

When computer networks first became popular, only a few suppliers manufactured both the hardware—including the silicon chips that powered it—and the software. This required companies to purchase the hardware, software, and accompanying services to provision a network from different suppliers..

As the industry matured, some suppliers started focusing their efforts on developing hardware components exclusively. As a result, the software market opened up and enabled third parties to develop and expand the capabilities of the software.

With the advent of software-defined networking (SDN), companies can regulate the development of software and make it more open. We believe that SDN will help eliminate supplier lock-in while simplifying and automating the network-provisioning process.

Choosing an SDN Architecture

Our evaluation has revealed that SDN can be implemented in two ways. One method uses OpenFlow*, an open-standard communications interface that allows direct access to and manipulation of switch-forwarding tables. The other method uses an overlay network, which consists of nodes and logical links built on top of an existing network. It creates a network service that is not available in the existing physical network, where each logical link may consist of numerous physical links in the underlying network.

OpenFlow requires new hardware to support the OpenFlow forwarding table. An overlay network doesn't require any additional hardware so the investment in new equipment is not necessary. Additionally, there are different versions of OpenFlow, and not all SDN controller and network hardware suppliers support the same versions. OpenFlow is appropriate when the flow of data is volatile and networks need reprogramming based on the change of flow. However, we do not have a homogenous hardware environment, and because the data flows between our source and destination nodes are generally predictable after being established, an overlay model provides more value for us than an OpenFlow model.

From 2012 to mid-2013 we tested four SDN controllers with multiple architectures, ranging from an overlay-based architecture to OpenFlow-based architecture. Test criteria included network bandwidth and performance benchmarks. Figure 2 illustrates our test system.

We conducted our tests in three phases:

- **Phase 1: SDN controller performance issues.** This phase of testing in early 2012 supported the validity of the SDN concept and identified some performance issues common to all the SDN controllers we tested. These issues would need to be addressed by the industry before we could adopt the technology. For example, results from one test scenario of a 10 gigabit Ethernet (GbE) network without the overlay network showed a 9.39 Gbps throughput across two VMs on two hypervisors. The same setup using an overlay network showed only a 3.3 Gbps throughput. The performance did not improve when we added more VMs to the hypervisors, which indicated that the overlay drivers were not optimized.
- **Phase 2: Scalability challenges.** During this phase of testing in late 2012 and early 2013, throughput improved, but we encountered scalability challenges. We discovered that the architecture could not scale for a large number of VMs and flows. For each communication, new flows were initiated and did not expire, so the number of tunnels also increased across a large number of VMs.
- **Phase 3: Optimized flows.** The final phase of testing in mid-2013 revealed that product suppliers had addressed the proliferation of flows and that the software had matured and stabilized. Although various suppliers addressed the issue differently, the result was that each VM was able to maintain communication without needing to broadcast to other VMs; the establishment of flows was optimized based on the source and destination addresses.

Overlay Network Architecture

The SDN overlay network configuration provides the most value in terms of optimizing data flow in our data centers. In the second half of 2013, we deployed an overlay network in our production environment. Figure 3 illustrates a high-level view of our SDN architecture. The three primary components of this architecture are the SDN controller(s), gateway nodes, and hypervisor nodes, which connect to the VMs using virtual switches.

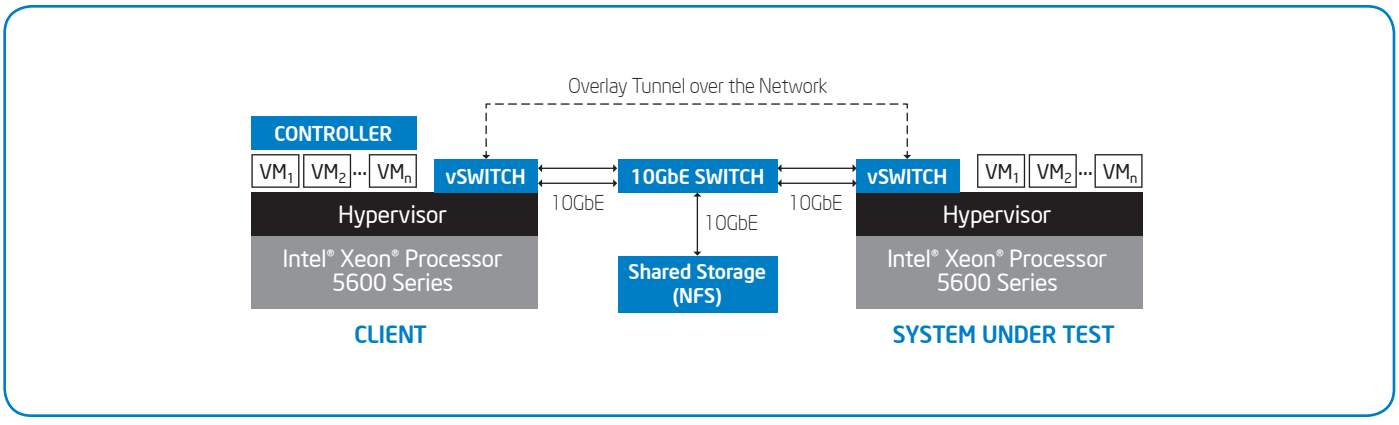


Figure 2. Intel's test configuration. During our evaluation of software-defined networking (SDN), we used an overlayed-based architecture to test and monitor SDN controller performance, scalablity issues, and flow optimization.

SDN CONTROLLER

The SDN controller makes an SDN network possible by abstracting the network from the hardware. The controller—in reality a software application, not a piece of hardware—manages the communication between applications and network devices. In a large implementation, there can be more than one SDN controller, which creates an SDN controller cluster.

The centralized-console approach results in intelligent networking characterized by optimized communication flows and automated configuration, along with an overall view of the network from a single console:

- **Optimized data flows.** A traditional network has a single path from the source of communication flow to its destination. The SDN controller can identify multiple paths for a flow and can split the flow's traffic across multiple nodes. The SDN controller optimizes the network path for a particular data flow based on the source and destination nodes. These capabilities increase network performance and scalability.
- **Automated configuration.** In contrast to a traditional, manual, device-by-device network configuration, the SDN controller saves time and increases configuration accuracy and consistency by automating the configuration of network devices. This approach helps to easily adjust configurations when network conditions change. Essentially, the SDN controller helps to manage the entire network architecture as if it were a single device.
- **Overall view of the network.** The SDN controller's console provides network administrators with a global view of the entire network—improving decision making and management efficiency.

The SDN controller's programmable interface gives network administrators greater control over network traffic than is possible in a traditional network. For example, our security policies may require inbound traffic for a particular server to pass through a firewall. But outbound traffic does not pose a security threat and would not necessarily have to pass through the firewall. In a traditional network, this level of granular control is difficult to achieve. With SDN, an employee who is not a network engineer can easily program the controller to redirect outbound traffic around the firewall. These policies can be applied through the SDN controller at the session, user, device, and application levels.

GATEWAY/SERVICES NODES

Gateway nodes provide a way to integrate VLANs' physical network fabric with SDN-enabled cloud fabric, which is configured and managed by the SDN controller. Gateway capability allows the integration of the two environments, allowing the SDN-enabled hypervisor to communicate with the non-SDN-enabled network. Services nodes process the broadcast, multicast, and flow establishment between VMs.

HYPERVISOR NODES

The hypervisor nodes host the VMs. As illustrated in Figure 3, each hypervisor node runs the virtual switching module, which is where the SDN controller builds the virtual networks for tenants. A single hypervisor might host VMs from several virtual networks. Policy control is managed by a virtual switch between the hypervisor and the VMs.

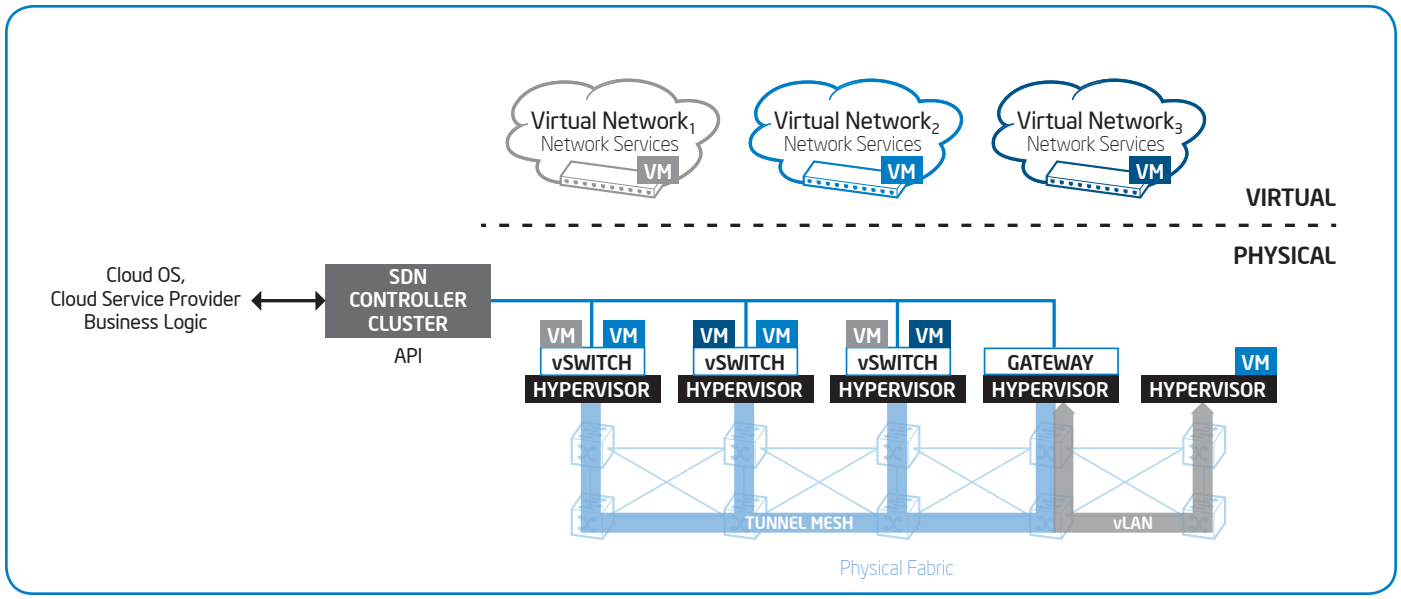


Figure 3. Our overlay software-defined network (SDN) architecture consists of three primary components: a cluster of SDN controllers, gateway/services nodes, and hypervisor nodes. In essence, an SDN controller enables management of the entire network architecture as if it were a single device.

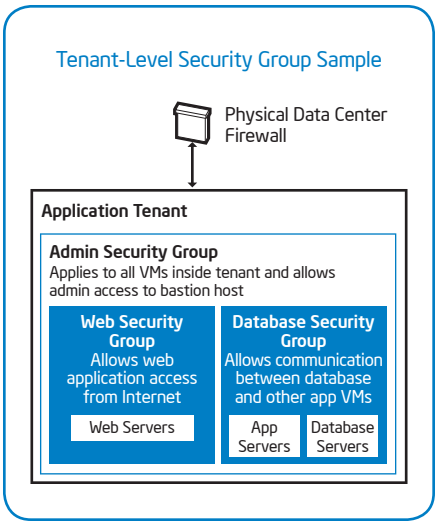


Figure 4. This security group model depicts the flexibility of network access control we have with our OpenStack*-based private cloud environment combined with software-defined networking (SDN) capabilities.

Changing the Security Architecture

Consumerization and other industry trends have contributed to the increase in applications that are being provisioned in our Internet-facing virtualization environments. We have traditionally used network-based segmentation and network access control as the primary mechanisms for securing VMs. In our legacy network-provisioning model, we relied on monolithic external firewall appliances to provide granular network access controls across these VMs and applications. This type of security model presented challenges to scalability, supportability, and risk. For example, in some of our environments, the external firewall had to enforce thousands of security rules, which created security manageability issues and increased potential for security breaches.

Our OpenStack-based private cloud environment combined with SDN capabilities enables us to shift to a more distributed and tenant-focused security framework. Each application tenant is provisioned in their own private virtual network environment on top of our private cloud. Network access control is managed within the tenant's virtual network environment using security groups.

Security groups are essentially a collection of access control rules that can be switched on inside each tenant. Multiple sets of security groups can be applied to the VMs to define the network security for that VM, as is illustrated in Figure 4.

By using application tenant-level security groups enforced on the hosts, we are able to implement a more manageable set of access control rules for each tenant and enable self-service for managing those rules. Physical firewall appliances can then be used to secure the perimeter of the data center and require a smaller, more manageable set of security rules.

Changing Roles and Responsibilities

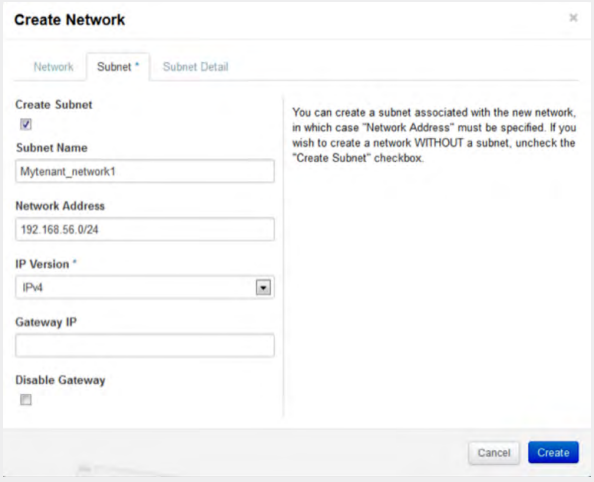
There is no longer a hard separation between our network, server, and compute teams' responsibilities when creating a network instance. With SDN, network capabilities that were once implemented on a physical network device are now implemented on a virtual network. Therefore, SDN adoption requires the skills that were previously divided between systems administrators and network administrators to be combined.

Self-Service Network in Action

Software-defined networking (SDN) makes network services provisioning a self-service activity for employees who are not network administrators. For example, Thomas Birch, the IT Incubator Program Manager at Intel, uses SDN to create networks that enable Intel business units (BUs) to launch a web service.

This screenshot is from the web interface that Birch uses to manage the networks he creates. For his use case, Birch sets up inbound connections into a virtual machine (VM) by establishing network rules. He looks at a VM, determines what the other VMs need to communicate to it and for what reason, and deploys and tests the network instance—all from the web interface.

Before we adopted SDN, Birch had to request all network services provisioning from a network administrator. Now he creates networks in real time. "Prior to SDN," says Birch, "there was limited automation, and setup involved a lot of manual modification. I had to ask an IT engineer to open specific ports, which turned me into a middle man...SDN puts me in control."



To create a self-service network, employees use a web GUI to provision a virtual network instance.

To close this gap, we have created the cloud systems administration function. Representatives from each team—network, server, and compute—are responsible for developing and operating the cloud environment. These representatives collaborate and share their skillsets.

There is a learning curve for employees who provision networks but do not necessarily have a background in networking. The network engineering team helps these employees understand some basic concepts and networking security practices, and then the provisioning GUI guides them the rest of the way. We are also building intuitive application environment templates to make it easier for customers to provision network components on demand without needing extensive network knowledge.

Preliminary Results

Service provisioning time is already improving. The SDN controller's global view of the network and the self-service benefit enabled by SDN are lowering the pressure on network administrators to meet service level agreements (SLAs) for network service requests.

SDN is simplifying the process of network provisioning. Creating networks requires some training, especially for employees who have little or no networking experience. However, after getting assistance from network engineers and working in the SDN management application, employees realize the benefits of a self-service approach. They appreciate the removal of constraints to support BUs working in agile development, increased control over network deployment, and the ability to deploy and test in real time.

Initial test results indicate that SDN can reduce service costs through improved network management efficiency. We also can retain our current infrastructure investment and rely less on proprietary hardware and dedicated appliances.

NEXT STEPS

In 2014, we will expand the deployment of the overlay network from our initial production deployment in 2013. We will also continue to explore ways to use OpenFlow to add value to the overlay network.

We plan to use APIs to virtualize network services such as load balancers and web application firewalls within our cloud environment. We will monitor the CPU overhead these capabilities will add on hosts and, when appropriate, offload the processing to lower-level hardware or to a network interface controller (NIC).

Although the migration from a hypervisor environment to an SDN environment presents many challenges, we are exploring methods to simplify this migration.

As we evolve from a monolithic security model to a distributed security model, we need to monitor scalability, especially in terms of host limitations. We plan to offload the processing whenever possible.

CONCLUSION

SDN virtualizes the network for our cloud environments and allows employees other than network engineers to program the overlay network components so we can dynamically provision networks on demand. Network services that once took several days to configure can now be fulfilled in real time through a self-service approach.

We are realizing these SDN benefits:

- Faster service provisioning
- Simpler, self-service network creation
- Reduced service costs

Additionally, we have more control over our current infrastructure, and we rely less on proprietary hardware and dedicated appliances.

RELATED INFORMATION

Visit www.intel.com/IT to find content on related topics:

- "Intel IT's Data Center Strategy for Business Transformation"
- "Upgrading Data Center Network Architecture to 10 Gigabit Ethernet"

ACRONYMS

BU	business unit
DNS	domain name system
GbE	Gigabit Ethernet
NIC	network interface controller
SDN	software-defined networking
SLA	service level agreement
TCO	total cost of ownership
VLAN	virtual LAN
VM	virtual machine
VRF	virtual routing and forwarding

For more information on Intel IT best practices, visit www.intel.com/IT.

Software and workloads used in performance tests may have been optimized for performance only on Intel® microprocessors. Performance tests, such as SYSmark* and MobileMark*, are measured using specific computer systems, components, software, operations and functions. Any change to any of those factors may cause the results to vary. You should consult other information and performance tests to assist you in fully evaluating your contemplated purchases, including the performance of that product when combined with other products.

Configurations:

Server (System Under Test)	Model: Intel® S5520UR (Urbanna) ▪ Processor: Intel® Xeon® processor X5680 @ 3.33 GHz (2 sockets, 6 cores/socket), 96 GB RAM, Hyper-threading disabled ▪ BIOS: S5500.86B.01.00.0060 ▪ Network Adapters: 2 x Intel® Ethernet Sever Adapter X520-2 connected @ 10GbE to the 10GbE switch ▪ Network Driver: ixgbe 2.0.84.8.2-10vmw-NAPI ▪ Hypervisor: VMware ESX* 5.1 RC (build 716794) ▪ Guest VM Configuration: 1 vCPU, 4 GB RAM, RHEL 6.2 (2.6.32-220.el6.x86_64), vmxnet3 and ixgbevf, 24 VMs ▪ Big Switch Controller (Corona) – beta release
Client Configuration	Model: Intel® S5520UR (Urbanna) ▪ Processor: Intel® Xeon® processor X5670 @ 2.93 GHz (2 sockets, 6 cores/socket), 96 GB RAM, Hyper-threading disabled ▪ BIOS: S5500.86B.01.00.0060 ▪ Network Adapters: 2 x Intel® Ethernet Sever Adapter X520-2 connected @ 10GbE to the 10GbE switch ▪ Network Driver: ixgbe 2.0.84.8.2-10vmw-N ▪ Hypervisor: VMware ESX* 5.1 RC (build 716794) ▪ Guest VM Configuration: 1 vCPU, 4 GB RAM, RHEL 6.2 (2.6.32-220.el6.x86_64), vmxnet3, 24 VMs
Network Configuration	Model: Extreme Networks Summit* X650 10GbE switch ▪ Network Connectivity: – 2 x Intel® Ethernet Sever Adapter X520-2 connected @ 10GbE (from server) – 2 x Intel® Ethernet Sever Adapter X520-2 connected @ 10GbE (from client) – 1 x 10 Gigabit Ethernet Controller IX1-SFP+ connected @ 10GbE (from NetApp FAS6240)
Storage Configuration	NetApp FAS6240 ▪ Version 8.0.1 P1 ▪ Intel® Xeon® processor E5540 @ 2.53 GHz (8 proc), 48 GB RAM ▪ 512 GB of PAM-II ▪ NFS storage @ 10GbE
Application	Netperf* 2.5
Data Collection Tools Used	▪ Captured throughput from Netperf ▪ Captured system utilization on the VMware ESX* host using "esxtop" utility

Printed in USA ♻ Please Recycle 0414/WWES/KC/PDF 330118-001US