WHITE PAPER

RIFT.io* and Intel – Taking Virtual Network Functions to Hyperscale

RIFT.io

benu

Introduction

Network functions virtualization (NFV) is a powerful technology that offers elasticity, scale, and reliability for networking applications. NFV is based on data center virtualization, which has produced web services consisting of hundreds of virtual machines (VMs). This proves that correctly engineered applications can scale to very large sizes. Communications service providers now want to see their virtual network functions (VNF) and apps have this same scalability.

Software companies working to build virtualized network functions or to transition existing applications from non-virtualized environments to NFV environments, however, have found that the transition can take a significant amount of engineering cycles to complete. Yet virtualization is essential if applications are to survive and scale. Application developers need efficient means of converting their software into VNFs, and then testing, deploying, scaling, and managing – all at web scale.

RIFT.io's* RIFT.ware* is an open source, hyperscale, NFV platform that simplifies development, deployment, and management of virtual network services, applications, and functions. With RIFT.ware, any application can be virtualized, tested, deployed at scale, and efficiently monitored to run in any cloud and at any scale.

The Challenge

Porting software designed for bare metal environments, sometimes with specialized hardware, into virtualized network functions (VNF) applications is not a straightforward task. Operating systems, hypervisors, and run-time environments often have inconsistent models – requiring developers to pick a particular ecosystem or build in compatibility with a number of alternatives.

Even though open source software such as OpenStack* provides a complete NFV environment, the integration process can take a long time. This is especially true for applications that depend on particular hardware features and special capabilities of their current environment. These restrictions and dependencies must be removed or matched to similar virtual functions. Integration with NFV/SDN requirements must be considered if the application is to scale in cloud environments – especially for stateful applications.

Table of Contents

A well-structured VNF takes full advantage of the underlying infrastructure to achieve several objectives:

- **Elasticity** – add compute, storage, and/or network resources as needed and to seamlessly integrate the resources into the running application.
- **Scalability** – expand the number of VNFs used without changing the attached network topology. For example, if a provider gateway (P-GW) scaled by creating additional VNFs, each with a new set of control, data, and management plane IP addresses, then neighboring systems would need to be reconfigured, load balanced, and perhaps replicated as well. This creates operational complexity and slows down service deployment.
- **Unlimited scale** – instantiate hundreds, if not thousands, of VMs within a single VNF and maintain the VNF as a single application.
- **Cloud support** – run in one or more cloud environments. Even when an application is hosted in a private data center, surges may need to overflow to the cloud ("cloud bursting").
- **Automation** – integrate with SDN controllers and service orchestration systems to support VNF features. It is extremely important to automate testing in support of rapid development.

Simply porting an application to run as a virtualized network function is only the first step. The real work starts with modifying the application to support elasticity, which often requires re-architecting the application. Additionally, going from one VM to tens of VMs in a VNF is already a major hurdle. Moving to hundreds and thousands of VNF instances presents further obstacles. Finally, hyperscale VNFs must also be monitored and managed.

The progression to full VNF scale (Figure 1) can be broken down into four levels, progressing from basic virtualization to hyperscale. Some carriers are planning 7 to 10 years for full migration, a time scale that is eons when measured in Internet years. What's needed is an approach to network virtualization that can accelerate NFV adoption and accelerate companies' progression along the maturity model.

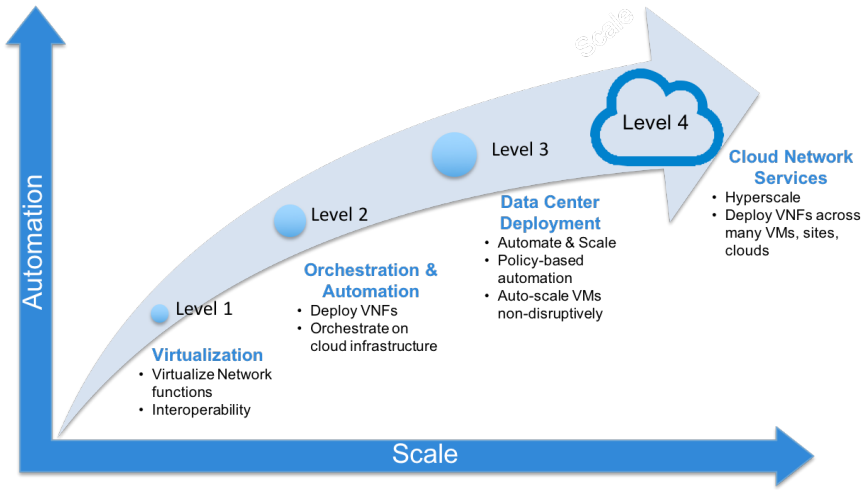

Figure 1. NFV Adoption Maturity Model

Solution

RIFT.ware is an open source NFV platform that simplifies development, deployment, and management of virtual network services, applications, and functions. It can:

- Support any network application on any cloud at any scale
- Scale on demand, with no impact on end users or surrounding networks and network elements
- Ensure application consistency regardless of number and type of environments as well as the number of VNFs
- Deploy on the most optimal, cost-effective infrastructure available, regardless of compute and network environment
- Improve lifecycle management
- Support rapid service deployment through test and deployment automation
- Eliminate vendor lock-in through use of open source software

RIFT.ware modules instantiate and terminate VNFs at multiple locations, in the thousands within multiple cloud environments. They ensure reliability through detection, isolation, and recovery mechanisms. Application state is propagated for stateful applications. Regardless of the number of VMs involved, RIFT.ware maintains a single IP address for the entire application, obviating networking and environmental changes. RIFT.ware collects, aggregates, and displays key performance indicators (KPIs) and key quality indicators (KQIs) in order to monitor and maintain large groups of VNFs.

RIFT.ware also supports application programming interfaces (APIs) that abstract away the differences between operating systems, hypervisors, and other VM infrastructure elements. Application developers use these APIs instead of vendor-specific interfaces, and RIFT.ware takes care of the mappings. Of particular value is RIFT.ware's ability to describe required hardware and software features so that the right platform can be selected for execution.

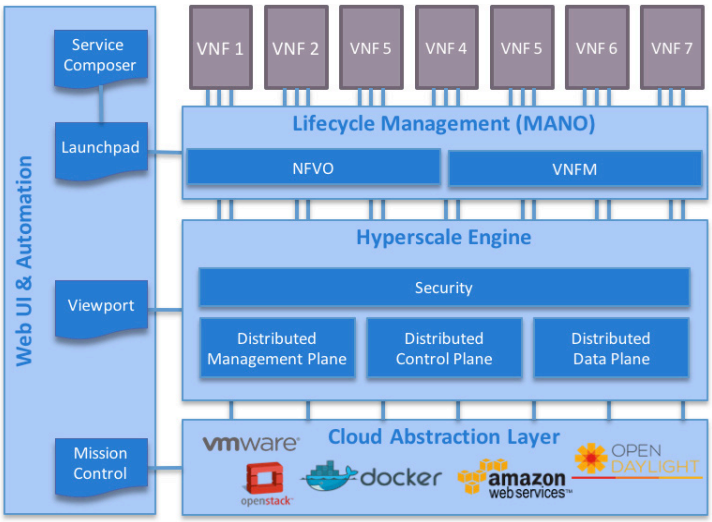

Figure 2. RIFT.ware Architecture

RIFT.ware is a next generation NFV platform that delivers management and orchestration (MANO) and automation of virtual network functions and applications at scale. RIFT.ware is available as open source software that is compliant with the latest open source communities and ETSI standards. The RIFT.ware platform is comprised of four modular components designed to virtualize, automate, and scale communications, network, and application functions. Service providers, network application builders, and enterprises can start with simple onboarding of unmodified functions and applications and create a roadmap that first virtualizes, then automates, then scales their network functions to deliver cloud-ready communications and applications and deliver cloud based services.

Lifecycle Management

RIFT.ware's Lifecycle Management module is responsible for management and orchestration (MANO) of network services and VNF lifecycle. It is responsible for starting and stopping network services and their constituent VNFs, allocation/deallocation of VMs to achieve elasticity, and fault detection and isolation to achieve reliability.

RIFT.ware's MANO functions are fully compliant with the European Telecommunications Standards Institute (ETSI)* NFV Management and Orchestration specification.[1] It provides the two functional blocks required for MANO:

- **NFV Orchestrator (NFVO)** – manages lifecycle and resource orchestration across multiple virtualized infrastructure managers.

[1] http://www.etsi.org/deliver/etsi_gs/NFV-MAN/001_099/001/01.01.01_60/gs_nfv-man001v010101p.pdf

- **VNF Manager (VNFM)** – performs instantiation, scaling VNFs up and down, and monitoring VNF health. The lifecycle manager module supports thousands of VNFs and network services. VNFs request particular hardware and software features through the use of VNF descriptors (VNFDs). VNFM/NFVO use enhanced platform awareness (EPA) to request VMs from cloud management systems that match workload requirements to platform capabilities. EPA is described more fully in the "Intel® Technology" section of this solution brief.

Hyperscale Engine

The hyperscale engine implements massive horizontal scalability while maintaining state consistency, management simplicity, and security. Multi-VM VNFs are managed as a single identity. Configuration and state updates are distributed to all VNFs in real time.

The hyperscale engine maintains a single IP address for the entire application with respect to the outside world through the use of programmable, distributed control and forwarding planes. When integrated with an application, the hyperscale engine acts as a distributed packet classifier, sending control and data messages to all VNFs to balance load and respond to traffic surges. A high performance data path is supported by state-of-the-art virtualization and acceleration technologies, including Open vSwitch, the Data Plane Development Kit (DPDK), and hardware offload. The use of a single IP address ensures that no network changes are needed in other applications.

While the lifecycle manager collects KPIs and KQIs from each VM, the hyperscale engine aggregates them into hierarchical records, with coordinated timestamps. KPIs and KQIs, which can be used to craft SLA policies, include:

- Packets per second
- Throughput per interface
- Aggregate bandwidth
- CPU and memory utilization per VM
- Round-trip latency
- VNF-specific statistics

Reliability is enhanced through the construction of hierarchical domains/clusters (Figure 3). VMs within a domain are responsible for detecting faults and recovering from within the domain (Figure 3a). If an entire domain fails or becomes inaccessible, then workload is transferred to other clusters at a higher level of control (Figure 3b).

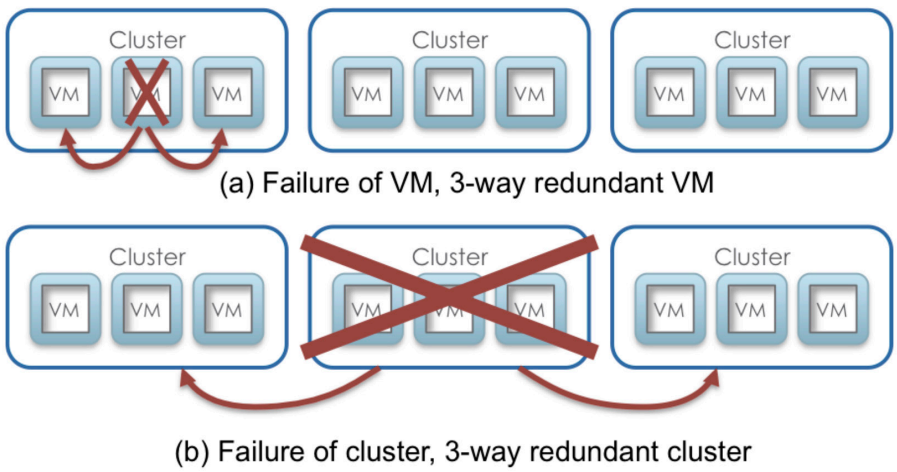

Figure 3. Fault Recovery with RIFT.ware

The hyperscale engine's security function ensures platform integrity through the selection of trusted execution platforms and use of crypto assistance for secure delivery. Sensitive information is kept private on the platform, and on control and bearer planes, regardless of the size of the network.

Cloud Abstraction Layer

The cloud abstraction layer makes it possible for VNFs to simultaneously run in multiple cloud environments by insulating VNFs from the vagaries of underlying cloud infrastructures. RIFT.ware supports a number of cloud environments based on OpenStack, VMware,* Amazon Web Services,* Microsoft Azure,* Linux Containers,* and others. Using the cloud abstraction layer, VNFs can take advantage of the latest advances in virtual technologies, such as hardware offload/assist, EPA, and SDN, without having to deal with each cloud's unique API.

Web UI and Automation

RIFT.ware simplifies management of large VNF deployments with a single web-based interface. Using the RIFT.ware web UI, computing environments that package virtual resources can be defined, launched, monitored, and managed based on an enterprise's organizational structure (sales, marketing, IT, etc.)

RIFT.ware's automation facilitates the creation of separate environments that can be used for development testing. A separate test API accelerates testing of new versions, facilitating quick deployment. Separate environments can also be used for A/B testing, with end users gradually migrated over to newer software releases.

Intel® Technology

Intel, in cooperation with the open source community, has innovated to maximize VNF performance obtained from Intel® architecture-based servers. These servers are now fully capable of matching the performance of the dedicated hardware.

RIFT.ware facilitates use of OS, hardware, and software features that may be present in cloud hosts. A key component is enhanced platform awareness (EPA). EPA contributions from Intel and others to the OpenStack Kilo* cloud operating environment enable fine-grained matching of workload requirements to platform capabilities prior to launching a virtual machine.

For example, EPA can automatically launch a cryptographic workload on a platform with a hardware-based crypto-accelerator. For workloads requiring particular CPU and/or I/O capabilities, EPA helps OpenStack assign virtual machines to run on crypto-accelerator. For workloads requiring particular CPU and/or I/O capabilities, EPA helps OpenStack the optimal platforms. EPA also enables cloud service providers to offer premium, revenue-generating services based on specific hardware features.

Among the requirements that may be included in an EPA request are:

- Specific network interface cards (NICs)
- Non-uniform memory access
- CPU and thread pinning
- Huge memory pages
- Data Plane Development Kit (DPDK)
- Open vSwitch* – with or without offloading
- Data direct I/O
- Single root I/O virtualization (SR/IOV)
- Trusted execution technology (TXT)
- Intel® QuickAssist Technology (QAT)
- PCI-Pass Through
- Cache Management Technology (CMT)
- Cache Allocation Technology (CAT)

A number of these EPA attributes are particularly important to the RIFT.ware platform. These include:

- **Cache Monitoring Technology (CMT)** – allows an OS, hypervisor, or virtual machine monitor (VMM) to determine cache usage by applications running on the platform.
- **Cache Allocation Technology (CAT)** – allows an OS, hypervisor, or VMM to control allocation of a CPU's shared last-level cache. Once CAT is configured, the processor allows access to portions of the cache according to the established class of service (COS).
- **Intel® Communications Chipset 89xx (8955, 8950, 8926, 8925)** – a communications co-processor that accelerates cryptography and data compression.
- **Data Plane Development Kit (DPDK)** – the DPDK software library is used by RIFT.ware to route network packets around the Linux OS kernel. DPDK is a library of network drivers and an optimized run-time environment that has demonstrated network throughput of up to 80 million packets per second (Mpps). [2]
- **Intel QuickAssist Technology** – offers easy integration for built-in accelerators, employing a hardware-assisted security engine for implementing major security processes. It accelerates and compresses cryptographic workloads by offloading the data to hardware capable of optimizing those functions.
- **SR-IOV** – is a technique by which hardware resources, such as I/O interfaces, are shared by multiple virtual machines (Figure 4). NICs and servers that support SR-IOV are designed to replicate the resources necessary for each VM to be directly connected to the I/O device, so that main data movement can occur without hypervisor, virtual machine management, or virtual switch involvement.

[2] Results based on 1S Intel® Xeon® Processor E5-2600 (B0 stepping) 1x8Core, 2.0 GHz, PCIe* Gen2 Performance; More details about the performance of DPDK can be found at http://www.intel.com/content/dam/www/public/us/en/documents/solution-briefs/communications-packet-processing-brief.pdf.

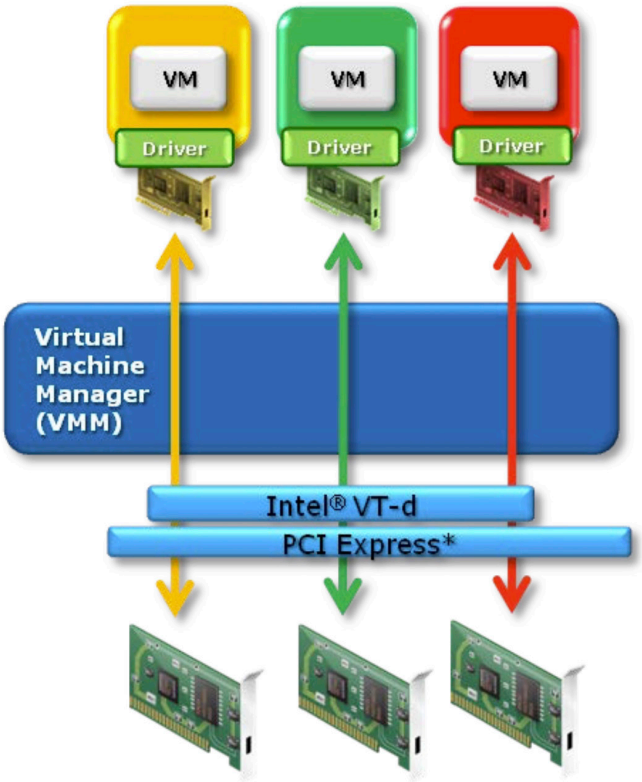

Figure 4. SR-IOV

- **Intel® Trusted Execution Technology (TXT)** – Intel TXT is specifically designed to harden platforms from the emerging threats of hypervisor attacks, BIOS, or other firmware attacks, malicious root kit installations, or other software-based attacks.

Case Study: Demonstrating EPA Optimization

Using its Burlington, Massachusetts, headquarters' data center lab, RIFT.io has successfully tested and demonstrated the performance benefits of using EPA. RIFT.io tested and demonstrated intelligent workload placement and the effect of features and capabilities on performance. Intelligent workload placement with EPA is documented in the white paper, "RIFT.ware Intelligent Workload Placement." [3]

In the demo, a network application that depended on high throughput encryption with low latency was used with a combination of EPA-selected features. Performance with various combinations of features was demonstrated and measured (Table 1).[4]

The three sets of demos leveraged different EPA features and were run on both Intel® Ethernet Server Adapter X520 NIC and Intel® Ethernet Multi-host Controller FM10000 products.

The results showed the advantages of EPA:

Demos 1A and 1B showcased two identical setups with three VNFs configured together to send TCP traffic in a loop. The demo measures performance via the throughput seen on the return path. With six EPA features turned on in demo 1B, the tests showed between 6 times and 7 times improvement in throughput on the return path compared to demo 1A.

Demos 2A and 2B showed intelligent workload placement of VMs using TXT and QAT EPA attributes. In this case, there were four VNFs configured together as a service designed to emulate IoT gateway performance, and to specifically highlight IPSec throughput. Demo 2B features a NIC with an Intel Communications Chipset 89xx to assist in IPSec computations. By placing the VMs on the server with the Intel Communications Chipset 89xx for IPSec, the test showed a 20 times improvement in IPSec re-keying. Simultaneously, the demo 2B tests resulted in better CPU utilization when compared to demo 2A (without the Intel Communications Chipset 89xx). Demo 2A showcased the security aspect of VM placement based on TXT requirement.

Demo 3A and 3B showcased intelligent workload placement based on TXT, CAT, and the Intel Ethernet Multi-host Controller FM10000. The network service created with the VNFs, specifically with traffic generator, was able to send approximately 60 Gbps of traffic over the Intel Ethernet Multi-host Controller FM10000 cards and, in the process, was injected with a noisy neighbor program that caused heavy latency. In this demo, CAT EPA was able to manage cache and allocate it in such a way that latency was dramatically improved, even with noisy neighbor enabled.

[3] http://riftio.com/wp-content/uploads/2015/12/RIFTware-and-Intel-EPA-White-Paper-1215.pdf

[4] For additional hardware and software configurations beyond those in Table 1 for the EPA Demonstrations, see Appendix A.

EPA Attributes	Demo 1A	Demo 1B	Demo 2A	Demo 2B	Demo 3A	Demo 3B
Intel® Ethernet Server Adapter X520 NIC	√	√	√	√	√	√
Intel® Ethernet Multi-host Controller FM10000 products NIC					√	√
Intel® Communications Chipset 89XX				√		
NUMA					√	√
CPU Optimization (CPU Pinning)		√	√	√	√	√
CPU Optimization (CPU Thread Pinning)		√	√	√	√	√
Memory Optimization (Huge Page)		√	√	√	√	√
DPDK	√	√	√	√	√	√
OvS Acceleration		√				
DDIO	√	√	√	√	√	√
TXT			√	√		
QAT				√		
PCI-Pass Through			√	√	√	√
CAT						√
CMT						√

Table 1. EPA Demonstration Cases

Case Study: Benu Networks* Virtual Service Edge*

Benu Networks'* next-generation Virtual Service Edge (VSE)* offers a new class of subscriber management, scale, and intelligence that can stitch network layer service logic into cloud service delivery in a programmatic way. VSE virtualizes CPE network functions into a virtual CPE (vCPE) in a unique architecture that uses a service overlay on SDN-based platforms. This moves complexity out of the CPE and into the network where it is more easily managed and upgraded. Benu's VSE addresses scale and service agility for network operators delivering cloud-based managed services for residential and business customer segments.

Benu needed means of getting to market quickly, optimizing its application with new compute features, and scaling a highly reliable application. RIFT.ware helped Benu with all of these objectives.

RIFT.ware optimized Benu VSE workload placement with EPA and SDN so that it could utilize DPDK, QAT, SR-IOV, and PCI-pass through. This made it possible for Benu to offer 10 Gbps/interface throughput with the reliability, latency, and jitter SLAs that previously required purpose-built network equipment and over-provisioning.

Scaling and high availability are key to VSE's success. The RIFT.ware hyperscale approach helps balance the incoming load among VM instances and detects failed instances to route traffic. The VSE, in turn, maintains all required state information in a distributed fashion and quickly adapts to these internal VNF changes.

Conclusion

Creation of virtualized, web-scale network services is not an easy job for software developers, but must be done in order to be relevant in NFV networks. Network applications and VNFs built with RIFT.ware combine the economics and scale of hyperscale data centers with the security and availability of carrier-grade network services. With RIFT.ware, any company can start with a single VNF on a single VM on any cloud platform and scale to any number of VNFs, VMs, or locations on multiple clouds with no capacity constraints or distance limitations. RIFT.ware reduces complexity, accelerates service velocity, and promises to fundamentally change the economics of the networking industry.

Appendix: EPA Demonstration Hardware and Software Environment [5]

Software	Version	Hardware Packages	Version
Rift.ware	3.0	Intel® Ethernet Multi-host Controller FM10000 products NIC	Rev A0 (FM10840-Q28DA2)
OpenStack	Kilo 2015.1.0	Intel Xeon Processor E5-2699	Intel® Xeon® CPU E5-2699 v3 @ 2.30 GHz
DPDK Open vSwitch	2.0 2.3.1-git4750c96	Intel Xeon Processor E5-2658 with Cache Allocation Technology	Intel® Xeon® CPU E5-2658 v3 @ 2.20 GHz
Intel® Server BIOS	SE5C610.8 6B.01.01.0008.021120151325, SE5C610.8 6B.01.01.1008.031920151331 SE5C610.8 6B.01.01.0005.101720141054	Intel® Communications Chipset 89XX	Intel® DH895XCC Series QAT
Strong Swan	5.3.0	Intel® Ethernet Server Adapter X520 NIC	Intel® X520, copper interface (direct attach)
Device Army VNF	RIFT.ware 3.0	Intel® Server	S2600WT2
Traffic Sink VNF	RIFT.ware 3.0	Arista 100G Switch	Arista DCS-7504 Hardware version: 02.00 Software image version: 4.15.0FX Architecture: i386 Internal build version: 4.15.0FX-2407560.4150FX Internal build ID: 9e02bf4d-69e2-48f3-997d-4507a639bcf8
TraffGen VNF	RIFT.ware 3.0	Dell Force 10 Switch	System Description: 48-port E/FE/GE (SA) Revision: 0A Software Version: 2.5.3.7

[5] Other names and brands may be claimed as the property of others.

CAG VNF	RIFT.ware 3.0	Brocade 10G Switch	NAME: Chassis DESCR:System Chassis SID:BR-VDX6740 SwitchType:131 Firmware name: 6.0.1
Premise GW VNF	RIFT.ware 3.0	Cisco Nexus 3K 10G Switch	Cisco Nexus3064 Chassis ("48x10GE + 16x10G/4x40G Supervisor") Intel® Celeron® CPU P450 with 3665256 KB of memory. Processor Board ID FOC16027UVH
Hadoop	2.2.0		
ODL	distribution-karaf-0.2.3-Helium-SR3		
RIFT.visualization (WebUI)	3.0		
vLB	RIFT.ware 3.0		
Ejabberd XMPP server	15.0		
Sleek XMPP Application	1.3.1		

Disclaimers

Intel technologies' features and benefits depend on system configuration and may require enabled hardware, software or service activation. Performance varies depending on system configuration. No computer system can be absolutely secure. Check with your system manufacturer or retailer or learn more at intel.com.

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. For more complete information visit http://www.intel.com/performance.

Intel does not control or audit third-party benchmark data or the web sites referenced in this document. You should visit the referenced web site and confirm whether referenced data are accurate.

*Other names and brands may be claimed as the property of others.

333871-001