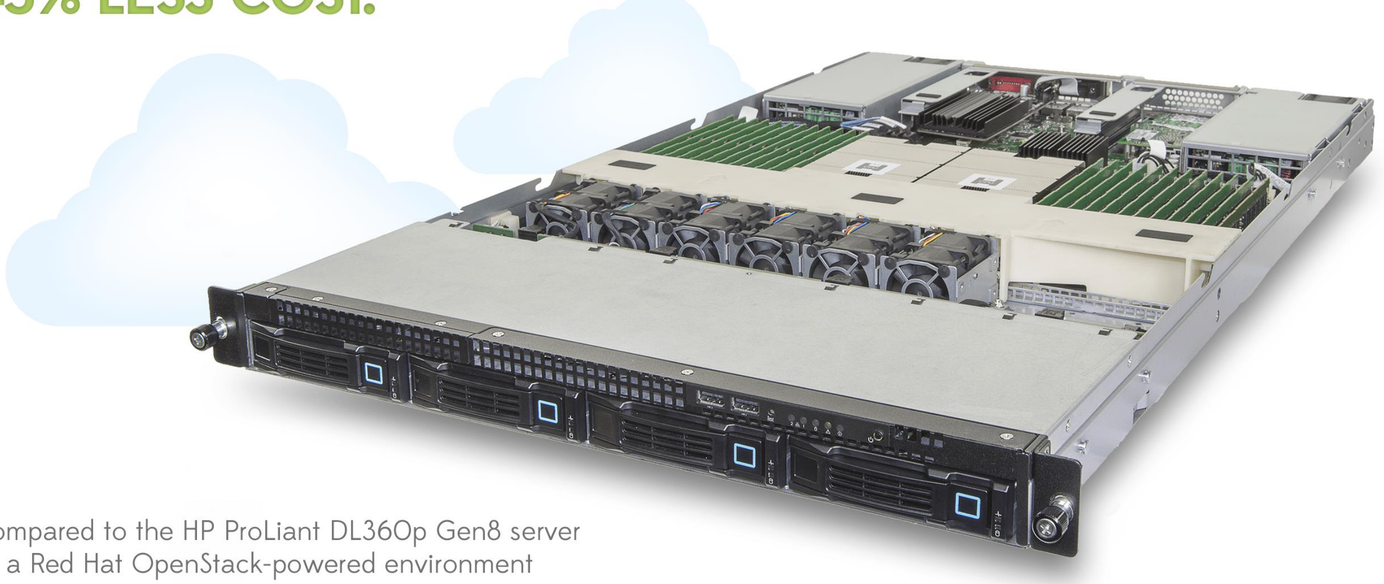
CLOUD WORKLOAD PERFORMANCE AND COST COMPARISON: AMD-BASED OPEN COMPUTE 3.0 SERVER VS. HP PROLIANT DL360P GEN8

AMD-BASED OPEN COMPUTE 3.0 SERVER

26% GREATER CLOUD DATABASE PERFORMANCE.
43% LESS COST.

AMD

OPEN Compute Project



compared to the HP ProLiant DL360p Gen8 server
in a Red Hat OpenStack-powered environment

With the growing space of cloud computing, both public and private, enterprises, service providers, and other large organizations are looking for ways of achieving equal or better performance with minimal capital expense outlay. The AMD-based Open Compute 3.0 platform is comprised of servers that meet this need directly. Offered at a lower cost, these servers have the potential of greatly increasing performance per dollar spent for large datacenters.

Here in the labs at Principled Technologies, we tested two servers, the AMD Opteron™ 6378-based Open Compute 3.0 server and the Intel® Xeon® processor E5-2640-based HP ProLiant DL360p Gen8, to see how much of an online transaction processing (OLTP) workload they could each handle in a Red Hat Enterprise Linux OpenStack platform-powered environment. Since many cloud services use open source database software, we configured the virtual machine (VM) test image to use PostgreSQL and tested multiple VMs on each server running an OLTP workload.

The AMD-based Open Compute 3.0 server, which costs only $6,311, handled 26.2 percent more orders per minute (OPM) running 12 VMs than the Intel-based HP ProLiant DL360p Gen8, a server that costs $11,193. That means an organization could have better cloud database performance while spending 43.6 percent less. For businesses and consumers using either private or public cloud environments to run database workloads, the AMD-based Open Compute 3.0 server is an excellent, cost-effective choice.

AUGUST 2013
A PRINCIPLED TECHNOLOGIES TEST REPORT
Commissioned by Advanced Micro Devices, Inc.

GREATER PERFORMANCE

In our test scenario, we installed a small Red Hat OpenStack environment, consisting of the OpenStack controller server, a server running the OpenStack Cinder block storage server backed by SSD storage, and using two compute nodes: the AMD-based Open Compute server, and the HP ProLiant server. We used DVD Store 2.1, a benchmark that simulates an online store, to generate the cloud database load. Using OpenStack, we spawned 12 VMs from the same OpenStack image running Red Hat Enterprise Linux 6.4 and PostgreSQL 9.2. First, we enabled the AMD-based Open Compute server as an OpenStack node, then spawned 12 VMs, and ran the test. We repeated the process on the HP ProLiant server, each time running 12 VMs on only the active OpenStack compute node, allowing us to measure cloud performance on one node at a time with all other cloud components being equal. Each VM was comprised of two vCPUs and 12GB RAM.

We focused our results on OPM. We ran the tests three times for both configurations and report the median results. As Figure 1 shows, the AMD-based Open Compute 3.0 server handled 26.2 percent more orders per minute than the Intel-based HP server.

Figure 1: Total database performance, in orders per minute, for the two servers. Higher numbers are better.

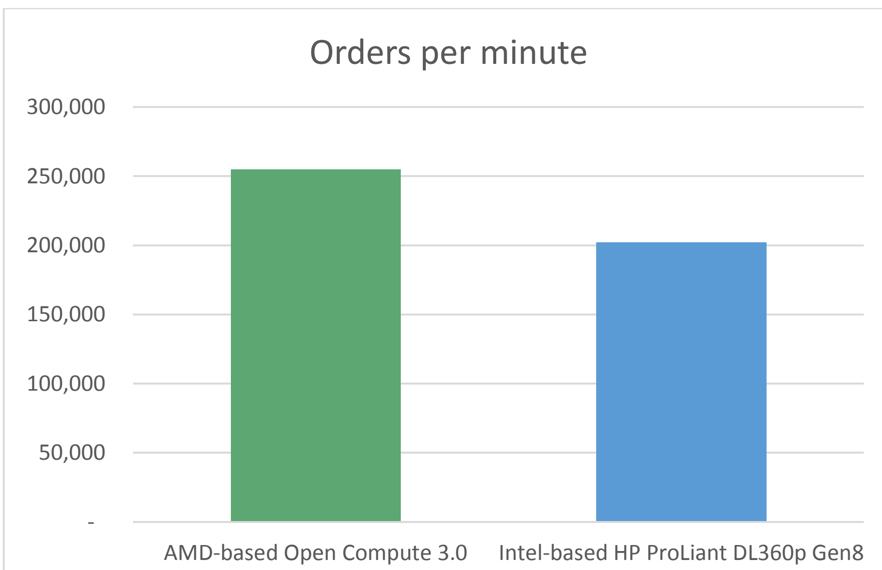


Figure 2 shows the OPM results of the benchmark that each VM achieved. When handling this workload of 12 VMs, the AMD Open Compute 3.0 solution ran 26.2 percent more transactions per minute than the HP solution. Both servers reported CPU utilization between 80 to 90 percent.

VM	AMD Open Compute 3.0 solution	HP ProLiant DL360p G8 solution
VM 1	21,477	16,675
VM 2	21,013	17,022
VM 3	21,582	16,887
VM 4	21,434	16,667
VM 5	20,927	16,852
VM 6	21,818	16,771
VM 7	21,163	16,921
VM 8	20,899	16,838
VM 9	21,551	16,722
VM 10	21,056	16,694
VM 11	20,916	16,895
VM 12	21,161	17,126
Total OPM	**254,997**	**202,070**
AMD Open Compute advantage	26.2%	

For information about the servers we tested, see Appendix A. For complete details on how we tested, see Appendix B.

GREAT SAVINGS

In addition to testing, we also compared pricing of the two servers. We noted each component of both servers and obtained the price for the total solution. We found that the AMD processor-based solution had a 43.6 percent lower cost than the HP solution. These are the list prices for both solutions and do not include shipping or tax.

The pricing of the HP server was current as of August 6, 2013. After creating a list of its components, we obtained the list price of the HP server from the company's website.[1] The HP quote we obtained contained two hard drives and was listed for $11,532, but for pricing fairness we removed the cost of the second hard drive (list price $339[2]), as the AMD-based server had only one drive. Therefore, the total cost of the HP solution was $11,193.

The pricing of the AMD-based server was current as of August 13, 2013. After creating a list of its components, we obtained the list price of the AMD-based server from an AMD Open 3.0 OCP provider via AMD. The total cost of the AMD-based solution was $6,311. Both quotes for the solutions included three-year warranty and maintenance.

[1] www.hp.com
[2] h30094.www3.hp.com/product/sku/10389176

MAXIMIZING PERFORMANCE PER DOLLAR

When we took the cost of the servers into account, we found that the AMD Open Compute 3.0 server had 123.2 percent greater performance per dollar, as Figure 3 shows.

	AMD Open Compute 3.0	HP ProLiant DL360p Gen8	Percentage win for AMD Open Compute 3.0
Total OPM for 12 VMs	254,997	202,070	26.2%
Cost	$6,311	$11,193	43.6%
Performance/dollar	40.4	18.1	123.2%

Figure 3: Performance-per-dollar calculations.

WHAT WE TESTED

About AMD-based Open Compute 3.0 server

AMD recently unveiled a new server platform, AMD Open 3.0, as part of its work with the Open Compute Project Foundation and its ongoing commitment to open industry standards. According to AMD, "Open 3.0 is a feature correct platform that enables low-cost, low power and flexible configurations, and offers the following advantages over off-the-shelf OEM platforms:

- A targeted feature-set that eliminates unnecessary components and optimizes the most important ones
- Low acquisition cost and low power, enabling lower total cost of ownership (TCO)
- A common platform … to help drive down server cost."

Learn more about AMD-based Open Compute 3.0 at amd.com/opencompute.

About Red Hat Enterprise Linux OpenStack Platform

With continuous support in the open-source OpenStack development community, Red Hat recently introduced Red Hat Enterprise Linux (RHEL) OpenStack Platform for building a public or private infrastructure-as-a-service cloud. As part of the Red Hat Cloud Infrastructure, Red Hat Enterprise Linux OpenStack Platform delivers Red Hat OpenStack technology optimized for and integrated with Red Hat Enterprise Linux. The design of RHEL OpenStack Platform allows for scalability and fault-tolerance as well as easy integration with RHEL for a managed private or public cloud development for cloud-enabled workloads. RHEL OpenStack Platform is a single subscription offering that includes RHEL server.

About our DVD Store 2.1 workload

To create our real-world e-commerce workload, we used the DVD Store Version 2.1 benchmarking tool. DS2 models an online DVD store, where customers log in, search for movies, and make purchases. DS2 reports these actions in orders per minute that the system could handle, to show what kind of performance you could expect for your customers. The DS2 workload also performs other actions, such as adding new customers, to exercise the wide range of database functions you would need to run your e-commerce environment.

For more details about the DS2 tool, see www.delltechcenter.com/page/DVD+Store.

IN CONCLUSION

A powerful server with strong performance that can handle demanding OLTP workloads is a smart choice for organizations interested in cloud computing. With 26.2 percent more transactions in an OLTP workload running on OpenStack cloud software, and at 43.6 percent less cost than an Intel-based HP solution, the AMD processor-powered Open Compute 3.0 server using Red Hat Enterprise Linux OpenStack Platform delivers powerful and cost-effective cloud performance.

APPENDIX A – SYSTEM CONFIGURATION INFORMATION

Figure 4 provides configuration information about the servers we used in our tests.

System	AMD-based Open Compute 3.0 server	HP ProLiant DL360p Gen8
General		
Number of processor packages	2	2
Number of cores per processor	16	6
Number of threads per processor	**16**	**12**
CPU		
Vendor	AMD	Intel
Name	Opteron™	Xeon
Model number	6378	E5-2640
Socket type	G34	LGA2011
Core frequency (GHz)	2.4	2.5
Bus frequency	6.4 GT/s	7.2 GT/s
L1 cache	64 KB (per core)	32 KB + 32 KB (per core)
L2 cache	1,000 KB (per core)	256 KB (per core)
L3 cache	16 MB (shared)	15 MB (shared)
Platform		
Vendor and model number	Quanta™	HP
Motherboard model number	S215-X1M2ZS	2M412456CN
BIOS name and version	American Megatrends 2.15.1236	HP P71
BIOS settings	C6 State disabled	Power setting set to performance
Memory module(s)		
Total RAM in system (GB)	256	256
Vendor and model number	Samsung® M393B1G73BH0-YH9	Samsung® M393B1G73BH0-YH9
Type	PC3-10600	PC3L-10600
Speed (MHz)	1,333	1,333
Speed running in the system (MHz)	1,333	1,333
Timing/Latency (tCL-tRCD-tRP-tRASmin)	9-9-9-36	9-9-9-36
Size (GB)	16	16
Number of RAM module(s)	16	16
Chip organization	Double-sided	Double-sided
Rank	Dual	Dual
OS/hypervisor		
Name	Red Hat Enterprise Linux 6.4	Red Hat Enterprise Linux 6.4
File system	ext4	ext4
Kernel	2.6.32-358.114.1.openstack.el6.x86_64	2.6.32-358.114.1.openstack.el6.x86_64
Language	English	English
RAID controller		
Vendor and model number	AMD SB700 SATA	HP Smart Array P420i
Firmware version	N/A	3.22
Cache size (MB)	N/A	0

System	AMD-based Open Compute 3.0 server	HP ProLiant DL360p Gen8
Hard drives		
Vendor and model number	Western Digital® WD800AAJS	HP 652605-B21
Number of drives	1	2
Size (GB)	80	146
Type	7.2K SATA	15K SAS
Ethernet adapters		
Vendor and model number	Mellanox® Technologies ConnectX-3 MCX342A-XCAN	QLogic® Corp NC523SFP
Number of ports	2	2
Type	10Gb adapter	10Gb adapter
USB ports		
Number	4	6
Type	2.0	2.0

Figure 4: System configuration information for the test systems.

APPENDIX B – HOW WE TESTED

We installed a small Red Hat OpenStack environment, consisting of the OpenStack controller server, a server running the OpenStack Cinder block storage server backed by high performance SSD storage (24 disks), and using two compute nodes: the AMD-based Open Compute server, and the HP ProLiant server. The client application traffic used a 1Gb network and the Cinder storage network used a 10Gb network. See Figure 5 for a visual representation of the test topology.

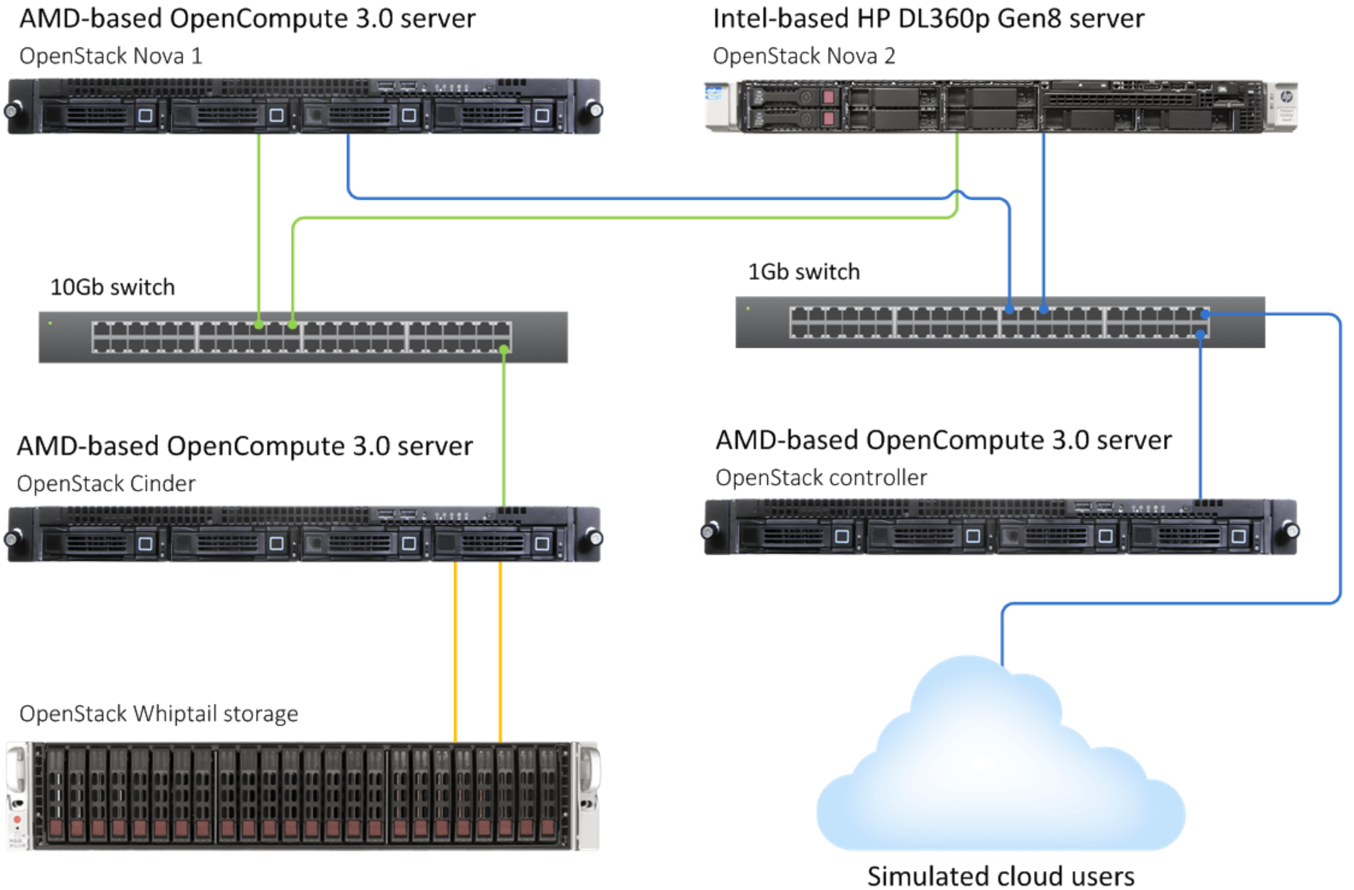


Figure 5: Test topology

Installing Red Hat Enterprise Linux 6.4 and OpenStack

Installing Red Hat Enterprise Linux 6.4

1. Insert and boot from the Red Hat Enterprise Linux 6.4 x86_64 installation DVD.
2. At the welcome screen, select Install or upgrade an existing system, and press Enter.
3. At the Media test screen, select Skip, and press Enter.
4. At the Red Hat Enterprise Linux 6 title screen, click Next.
5. At the Choose an Installation Language screen, select English, and click Next.
6. At the Keyboard Type screen, select U.S. English, and click Next.

7. At the Storage Devices screen, select Basic Storage Devices, and click Next.
8. If a warning for device initialization appears, select Yes, discard any data.
9. At the Name the Computer screen, enter the host name, and click Configure Network.
10. At the Network Connections screen, select the server's main or management network interface, and click Edit.
11. At the Editing network interface screen, check Connect Automatically.
12. On the same screen, select the IPv4 Settings tab, change the Method to Manual, and click Add.
13. On the same screen, enter the IP address, Netmask, Gateway, and DNS server. Click Apply.
14. Click Close on the Network Connections screen, and click Next on the Name the Computer screen.
15. At the Time zone selection screen, select the appropriate time zone, and click Next.
16. Enter the root password in the Root Password and Confirm fields, and click Next.
17. At the Assign Storage Devices screen, from the list in the left column, select the Linux disk, and click the arrow to copy the device to the right column. Next to the Linux disk, click the Boot radio button, and click Next.
18. At the Partition selection screen, select Replace Existing Linux System(s), and click Next.
19. If a warning appears, click Write changes to disk.
20. At the default installation screen, click Minimal, then click Next to begin the installation.
21. At the Congratulations screen, click Reboot.
22. After the system reboots, log in as root.
23. Ensure your system is updated via RHN. If not, register your system with RHN.

Preparing for OpenStack installation

- Install yum-utils.

```
# yum install -y yum-utils
```

- Prevent yum from pulling OpenStack information from the older RHOS repositories.

```
# yum-config-manager --disable rhel-server-ost-6-preview-rpms
# yum-config-manager --disable rhel-server-ost-6-folsom-rpms
# yum-config-manager --enable rhel-server-ost-6-3-rpms
```

- Run yum repolist to verify that you are not updating from the older RHOS repositories.
- Run the following commands to ensure that you are updating from the proper repositories:

```
yum install -y yum-plugin-priorities
yum-config-manager --enable rhel-server-ost-6-3-rpms \
        --setopt="rhel-server-ost-6-3-rpms.priority=1"
```

- Type `yum update –y` to get the latest updates for your system.
- Reboot your system after updating.

- Install the final remaining prerequisites for OpenStack:

```
yum install -y openssh-clients vim openstack-packstack
```

Installing Red Hat OpenStack on all systems

Before running Packstack, we created an LVM storage group on our future Cinder server (located on our external storage) named cinder-volumes. We ran Packstack on the system we intended to use as our OpenStack Controller node by typing the following command:

```
packstack --answer-file=answers.cfg
```

Below are the contents of the answers.cfg file:

```
[general]

# Set to 'y' if you would like Packstack to install Glance
CONFIG_GLANCE_INSTALL=y

# Set to 'y' if you would like Packstack to install Cinder
CONFIG_CINDER_INSTALL=y

# Set to 'y' if you would like Packstack to install Nova
CONFIG_NOVA_INSTALL=y

# Set to 'y' if you would like Packstack to install Quantum
CONFIG_QUANTUM_INSTALL=y

# Set to 'y' if you would like Packstack to install Horizon
CONFIG_HORIZON_INSTALL=y

# Set to 'y' if you would like Packstack to install Swift
CONFIG_SWIFT_INSTALL=n

# Set to 'y' if you would like Packstack to install the OpenStack
# Client packages. An admin "rc" file will also be installed
CONFIG_CLIENT_INSTALL=y

# Comma separated list of NTP servers. Leave plain if Packstack
# should not install ntpd on instances.
CONFIG_NTP_SERVERS=<IP Address of NTP server>
```

```
# Set to 'y' if you would like Packstack to install Nagios to monitor
# openstack hosts
CONFIG_NAGIOS_INSTALL=n

# Path to a Public key to install on servers. If a usable key has not
# been installed on the remote servers the user will be prompted for a
# password and this key will be installed so the password will not be
# required again
CONFIG_SSH_KEY=

# The IP address of the server on which to install MySQL
CONFIG_MYSQL_HOST=<IP Address of OpenStack controller>

# Username for the MySQL admin user
CONFIG_MYSQL_USER=root

# Password for the MySQL admin user
CONFIG_MYSQL_PW=<password>

# The IP address of the server on which to install the QPID service
CONFIG_QPID_HOST=<IP Address of OpenStack controller>

# The IP address of the server on which to install Keystone
CONFIG_KEYSTONE_HOST=<IP Address of OpenStack controller>

# The password to use for the Keystone to access DB
CONFIG_KEYSTONE_DB_PW=<password>

# The token to use for the Keystone service api
CONFIG_KEYSTONE_ADMIN_TOKEN=e1f149f7e9dd4e529f4fd21212507dd1

# The password to use for the Keystone admin user
CONFIG_KEYSTONE_ADMIN_PW=<password>

# The IP address of the server on which to install Glance
CONFIG_GLANCE_HOST=<IP Address of OpenStack controller>

# The password to use for the Glance to access DB
```

```
CONFIG_GLANCE_DB_PW=4cb21c0ccd2e4bec

# The password to use for the Glance to authenticate with Keystone
CONFIG_GLANCE_KS_PW=6536b85583ba4f77

# The IP address of the server on which to install Cinder
CONFIG_CINDER_HOST=<IP Address of Cinder server>

# The password to use for the Cinder to access DB
CONFIG_CINDER_DB_PW=c66f1c6aac9d4e12

# The password to use for the Cinder to authenticate with Keystone
CONFIG_CINDER_KS_PW=e0ee24d0255c4112

# Create Cinder's volumes group. This should only be done for testing
# on a proof-of-concept installation of Cinder.  This will create a
# file-backed volume group and is not suitable for production usage.
CONFIG_CINDER_VOLUMES_CREATE=n

# Cinder's volumes group size
CONFIG_CINDER_VOLUMES_SIZE=20G

# The IP address of the server on which to install the Nova API
# service
CONFIG_NOVA_API_HOST=<IP Address of OpenStack controller>

# The IP address of the server on which to install the Nova Cert
# service
CONFIG_NOVA_CERT_HOST=<IP Address of OpenStack controller>

# The IP address of the server on which to install the Nova VNC proxy
CONFIG_NOVA_VNCPROXY_HOST=<IP Address of OpenStack controller>

# A comma separated list of IP addresses on which to install the Nova
# Compute services
CONFIG_NOVA_COMPUTE_HOSTS=<IP Address of HP node>,<IP Address of AMD node>

# The IP address of the server on which to install the Nova Conductor
```

```
# service
CONFIG_NOVA_CONDUCTOR_HOST=<IP Address of OpenStack controller>

# The password to use for the Nova to access DB
CONFIG_NOVA_DB_PW=ba5aabad97d84cd1

# The password to use for the Nova to authenticate with Keystone
CONFIG_NOVA_KS_PW=2ac25599e9144cb3

# The IP address of the server on which to install the Nova Scheduler
# service
CONFIG_NOVA_SCHED_HOST=<IP Address of OpenStack controller>

# The overcommitment ratio for virtual to physical CPUs. Set to 1.0
# to disable CPU overcommitment
CONFIG_NOVA_SCHED_CPU_ALLOC_RATIO=16.0

# The overcommitment ratio for virtual to physical RAM. Set to 1.0 to
# disable RAM overcommitment
CONFIG_NOVA_SCHED_RAM_ALLOC_RATIO=1.5

# Private interface for Flat DHCP on the Nova compute servers
CONFIG_NOVA_COMPUTE_PRIVIF=eth1

# The IP address of the server on which to install the Nova Network
# service
CONFIG_NOVA_NETWORK_HOST=<IP Address of OpenStack controller>

# Public interface on the Nova network server
CONFIG_NOVA_NETWORK_PUBIF=eth0

# Private interface for Flat DHCP on the Nova network server
CONFIG_NOVA_NETWORK_PRIVIF=eth1

# IP Range for Flat DHCP
CONFIG_NOVA_NETWORK_FIXEDRANGE=192.168.32.0/22

# IP Range for Floating IP's
```

```
CONFIG_NOVA_NETWORK_FLOATRANGE=10.3.4.0/22

# Name of the default floating pool to which the specified floating
# ranges are added to
CONFIG_NOVA_NETWORK_DEFAULTFLOATINGPOOL=nova

# Automatically assign a floating IP to new instances
CONFIG_NOVA_NETWORK_AUTOASSIGNFLOATINGIP=n

# The IP addresses of the server on which to install the Quantum
# server
CONFIG_QUANTUM_SERVER_HOST=<IP Address of OpenStack controller>

# Enable network namespaces for Quantum
CONFIG_QUANTUM_USE_NAMESPACES=y

# The password to use for Quantum to authenticate with Keystone
CONFIG_QUANTUM_KS_PW=7f5b62e37a344ce4

# The password to use for Quantum to access DB
CONFIG_QUANTUM_DB_PW=83f0378ddfb446ee

# A comma separated list of IP addresses on which to install Quantum
# L3 agent
CONFIG_QUANTUM_L3_HOSTS=<IP Address of OpenStack controller>

# The name of the bridge that the Quantum L3 agent will use for
# external traffic
CONFIG_QUANTUM_L3_EXT_BRIDGE=br-ex

# A comma separated list of IP addresses on which to install Quantum
# DHCP agent
CONFIG_QUANTUM_DHCP_HOSTS=<IP Address of OpenStack controller>

# The name of the L2 plugin to be used with Quantum
CONFIG_QUANTUM_L2_PLUGIN=openvswitch

# A comma separated list of IP addresses on which to install Quantum
```

```
# metadata agent
CONFIG_QUANTUM_METADATA_HOSTS=<IP Address of OpenStack controller>

# A comma separated list of IP addresses on which to install Quantum
# metadata agent
CONFIG_QUANTUM_METADATA_PW=2d3827d6a26243d5

# The type of network to allocate for tenant networks
CONFIG_QUANTUM_LB_TENANT_NETWORK_TYPE=local

# A comma separated list of VLAN ranges for the Quantum linuxbridge
# plugin
CONFIG_QUANTUM_LB_VLAN_RANGES=

# A comma separated list of interface mappings for the Quantum
# linuxbridge plugin
CONFIG_QUANTUM_LB_INTERFACE_MAPPINGS=

# Type of network to allocate for tenant networks
CONFIG_QUANTUM_OVS_TENANT_NETWORK_TYPE=vlan

# A comma separated list of VLAN ranges for the Quantum openvswitch
# plugin
CONFIG_QUANTUM_OVS_VLAN_RANGES=physnet1:1000:2000

# A comma separated list of bridge mappings for the Quantum
# openvswitch plugin
CONFIG_QUANTUM_OVS_BRIDGE_MAPPINGS=physnet1:br-net1

# The IP address of the server on which to install the OpenStack
# client packages. An admin "rc" file will also be installed
CONFIG_OSCLIENT_HOST=<IP Address of OpenStack controller>

# The IP address of the server on which to install Horizon
CONFIG_HORIZON_HOST=<IP Address of OpenStack controller>

# To set up Horizon communication over https set this to "y"
CONFIG_HORIZON_SSL=n
```

```
# PEM encoded certificate to be used for ssl on the https server,
# leave blank if one should be generated, this certificate should not
# require a passphrase
CONFIG_SSL_CERT=

# Keyfile corresponding to the certificate if one was entered
CONFIG_SSL_KEY=

# The IP address on which to install the Swift proxy service
CONFIG_SWIFT_PROXY_HOSTS=<IP Address of OpenStack controller>

# The password to use for the Swift to authenticate with Keystone
CONFIG_SWIFT_KS_PW=ad152bdb90a04d23

# A comma separated list of IP addresses on which to install the
# Swift Storage services, each entry should take the format
# <ipaddress>[/dev], for example 127.0.0.1/vdb will install /dev/vdb
# on 127.0.0.1 as a swift storage device(packstack does not create the
# filesystem, you must do this first), if /dev is omitted Packstack
# will create a loopback device for a test setup
CONFIG_SWIFT_STORAGE_HOSTS=<IP Address of OpenStack controller>

# Number of swift storage zones, this number MUST be no bigger than
# the number of storage devices configured
CONFIG_SWIFT_STORAGE_ZONES=1

# Number of swift storage replicas, this number MUST be no bigger
# than the number of storage zones configured
CONFIG_SWIFT_STORAGE_REPLICAS=1

# FileSystem type for storage nodes
CONFIG_SWIFT_STORAGE_FSTYPE=ext4

# To subscribe each server to EPEL enter "y"
CONFIG_USE_EPEL=n

# A comma separated list of URLs to any additional yum repositories
```

```
# to install
CONFIG_REPO=

# To subscribe each server with Red Hat subscription manager, include
# this with CONFIG_RH_PW
CONFIG_RH_USER=

# To subscribe each server with Red Hat subscription manager, include
# this with CONFIG_RH_USER
CONFIG_RH_PW=

# To subscribe each server to Red Hat Enterprise Linux 6 Server Beta
# channel (only needed for Preview versions of RHOS) enter "y"
CONFIG_RH_BETA_REPO=n

# To subscribe each server with RHN Satellite,fill Satellite's URL
# here. Note that either satellite's username/password or activtion
# key has to be provided/
CONFIG_SATELLITE_URL=

# Username to access RHN Satellite
CONFIG_SATELLITE_USER=

# Password to access RHN Satellite
CONFIG_SATELLITE_PW=

# Activation key for subscription to RHN Satellite
CONFIG_SATELLITE_AKEY=

# Specify a path or URL to a SSL CA certificate to use
CONFIG_SATELLITE_CACERT=

# If required specify the profile name that should be used as an
# identifier for the system in RHN Satellite
CONFIG_SATELLITE_PROFILE=

# Comma separated list of flags passed to rhnreg_ks. Valid flags are:
# novirtinfo, norhnsd, nopackages
```

CONFIG_SATELLITE_FLAGS=

Specify a HTTP proxy to use with RHN Satellite
CONFIG_SATELLITE_PROXY=

Specify a username to use with an authenticated HTTP proxy
CONFIG_SATELLITE_PROXY_USER=

Specify a password to use with an authenticated HTTP proxy.
CONFIG_SATELLITE_PROXY_PW=

# The IP address of the server on which to install the Nagios server
CONFIG_NAGIOS_HOST=<IP Address of OpenStack controller>

# The password of the nagiosadmin user on the Nagios server
CONFIG_NAGIOS_PW=7e1eea0871ee4394

Additional OpenStack configuration

Configuring the network

- Create a bridge on the OpenStack Nova Manage node for instance traffic:

```
ovs-vsctl add-port br-net1 eth2
```

- Use the bridge you created as a router for all instance traffic through Quantum:

```
source ~/keystonerc_admin
quantum router-create router01
quantum net-create public01 --provider:network_type flat --
provider:physical_network physnet2 --router:external=True
quantum subnet-create --name public01_subnet01 --allocation-pool
start=10.38.15.100,end=10.38.15.199 --gateway 10.38.15.1 --disable-dhcp
public01 10.38.15.0/24
quantum router-gateway-set router01 public01
quantum net-create net01 --provider:network_type vlan --
provider:physical_network physnet1 --provider:segmentation_id 1000
quantum subnet-create --name net01_subnet01 net01 192.168.101.0/24 --
dns_nameservers list=true 10.38.15.1
quantum router-interface-add router01 net01_subnet01
```

Configuring the flavor

Using the Horizon dashboard, create a new OpenStack flavor specifiying 2 vCPUs and 12GB RAM. Use this flavor in creating VMs.

Adjusting quotas

By default, the quotas for the first OpenStack project are too low. Either increase the quotas on the first project or create a new one with the following quotas:

- 100 instances
- 60 vCPUs
- 1,024,000 MB RAM
- 100 volumes
- 100,000 GB of storage

Configuring the VM image

Creating the PostgreSQL instance

We did our initial configuration of the PostgreSQL VM on a separate KVM server, and then imported it into our OpenStack configuration as a volume. We then took a snapshot of that volume to allow us to create clones of the image.

Installing Red Hat Enterprise Linux on the PostgreSQL VM

1. Insert and boot from the Red Hat Enterprise Linux 6.4 x86_64 installation DVD.
2. At the welcome screen, select Install or upgrade an existing system, and press Enter.
3. At the Media test screen, select Skip, and press Enter.
4. At the Red Hat Enterprise Linux 6 title screen, click Next.
5. At the Choose an Installation Language screen, select English, and click Next.
6. At the Keyboard Type screen, select U.S. English, and click Next.
7. At the Storage Devices screen, select Basic Storage Devices, and click Next.
8. If a warning for device initialization appears, select Yes, discard any data.
9. At the Name the Computer screen, type the host name, and click Configure Network.
10. At the Network Connections screen, select the server's main or management network interface, and click Edit.
11. At the Editing network interface screen, check Connect Automatically.
12. On the same screen, select the IPv4 Settings tab, change the Method to Manual, and click Add.
13. On the same screen, enter the IP address, Netmask, Gateway, and DNS server. Click Apply.
14. Click Close on the Network Connections screen, and click Next on the Name the Computer screen.
15. At the Time zone selection screen, select the appropriate time zone, and click Next.
16. Enter the root password in the Root Password and Confirm fields, and click Next.
17. At the Assign Storage Devices screen, from the list in the left column, select the Linux disk, and click the arrow to copy the device to the right column. Next to the Linux disk, click the Boot radio button, and click Next.
18. At the Partition selection screen, select Replace Existing Linux System(s), and click Next.
19. If a warning appears, click Write changes to disk.
20. At the default installation screen, click Minimal, then click Next to begin the installation.

21. At the Congratulations screen, click Reboot.
22. After the system reboots, log in as root.
23. Ensure your system is updated via RHN. If not, register your system with RHN.
24. Type yum update to get the latest updates for your system.

Preparing to install PostgreSQL

- Edit the file /etc/selinux/config and change the line SELINUX=enforcing to SELINUX=disabled
- Install additional software:

```
# yum install tuned
```

- Disable these unused daemons with the following script:

```
# for s in auditd autofs avahi-daemon bluetooth cpuspeed crond cups
dnsmasq \
     fcoe firstboot ip6tables iptables irqbalance kdump libvirt-guests
lldpad \
     mdmonitor netconsole netfs nfs nfslock ntpdate portserve postfix
     qpidd \ restorecond rhnsd rhsmcertd rpcbind rpcgssd rpcidmapd
     rpcsvcgssd; do
     chkconfig $s off
     service $s stop
done
```

- Ensure these services are enabled using the following script:

```
# for s in sshd sysstat tuned; do
     chkconfig $s on
     service $s start
done
```

- Synchronize the time source.
- Edit /etc/ntp.conf, adding a relevant IP address as a time source.

  a. Run the command `chkconfig ntpdate on`
  b. Run the command `chkconfig ntpd on`
  c. Run the command date, and ensure that the time is synchronized with the domain controller.
  d. Configure the VM's IP address on interface eth0.

- Type the following command to restart networking to effect these changes:

```
# service network restart
```

- Reboot the system:

```
# shutdown -r now
```

Installing PostgreSQL

We used the PostgreSQL database server, version 9.2, as the database software on the RHEL VMs.

1. Download the rpm for PostgreSQL 9.2 and install it:

```
     # yum install http://yum.postgresql.org/9.2/redhat/rhel-6-
x86_64/pgdg-redhat92-9.2-7.noarch.rpm
```

2. Log onto the system as root.
3. Install the PostgreSQL database server and client:

```
# yum install postgresql92-server postgresql92-contrib
```

- Run the command initdb to initialize the PostgreSQL database cluster for the first time, and make sure that PostgreSQL will start with boot.

```
# service postgresql-9.2 initdb
# chkconfig postgresql-9.2 on
```

- Tune the database engine. Modify the following lines of the postgresql.conf file:

```
checkpoint_segments = 1024
wal_buffers = 16MB
shared_buffers = 2GB
effective_cache_size = 5GB
checkpoint_timeout = 1h
checkpoint_completion_target=0.9
```

4. Add one line to the PostgreSQL configuration file pg_hba.conf to permit SQL queries from the client network:

```
host all <test bed IP subnet>.0/24    trust
```

5. Modify the listen_addresses line in the PostgreSQL configuration file postgresql.conf:

```
listen_addresses = '*'
```

6. Run service postgresql-9.2, restart to restart PostgreSQL with the new settings.

Final tuned configuration

Run the following commands to ensure that tuned is configured properly for each system:

```
Cinder node: tuned-adm profile enterprise-storage
Nova compute node: tuned-adm profile virtual-host
PostgreSQL instances: tuned-adm profile virtual-guest
```

Configuring the clients

Our client machines containing the DVDStore executable were virtual machines running Windows Server 2008 R2 SP1 Enterprise Edition. These VMs had one vCPU and 4GB vRAM assigned to them and ran on a separate four-socket hypervisor machine.

Installing Windows Server 2008 R2 SP1 Enterprise Edition on the VMs

1. Power on the VM and open the console.
2. Right-click the machine, and choose Open console.
3. At the Language Selection Screen, click Next.
4. Click Install Now.
5. Select Windows Server 2008 R2 SP1 Enterprise (Full Installation), and click Next.
6. Click the I accept the license terms check box, and click Next.
7. Click Custom.
8. Click Drive options (advanced).
9. Ensure you select the proper drive, and click New.
10. Click Apply.
11. Click Next.
12. At the User's password must be changed before logging on warning screen, click OK.
13. Set the Administrator password, and click the arrow to continue.
14. At the Your password has been changed screen, click OK.
15. Install the hypervisor tools and drivers on the VM

Final client configuration.

We installed all updates available on July 31, 2013 on each client virtual machine. We also assigned a static IP address to each client VM on the test subnet.

Running the test

We created a series of batch files and shell scripts to automate the complete test cycle. We tracked the DVD Store orders-per-minute metric with Windows Performance Monitor, which is a running average calculated through the test. In this report, we report the last OPM reported by each client/target pair.

Each complete test cycle consisted of the general steps listed below. For each scenario, we ran three test cycles, and chose the median outcome.

1. Clean up prior outputs from the host system and all client driver systems.
2. Delete all running instances, with their respective Cinder storage.
3. Reboot the OpenStack controller node, the Cinder node, the compute nodes, and all client systems.
4. Wait for a ping response from the server under test (the Nova compute node), the infrastructure systems, all client systems, and all VMs.
5. Spawn 12 new instances from the gold image snapshot, specifying boot from volume snapshot (create a new volume).
6. Let the test server idle for 10 minutes.
7. Start Performance Monitor on all clients.
8. Start the DVD Store driver on all respective clients.

   We used the following DVD Store parameters for testing the virtual machines in this study:

```
ds2sqlserverdriver.exe --target=<target_IP> --run_time=10 --
nthreads=16 --dbsize=5GB --detailed_view=Y --warmup_time=10 --
think_time=0 –ramp_rate=10
```

ABOUT PRINCIPLED TECHNOLOGIES

Principled Technologies, Inc.
1007 Slater Road, Suite 300
Durham, NC, 27703
www.principledtechnologies.com

We provide industry-leading technology assessment and fact-based marketing services. We bring to every assignment extensive experience with and expertise in all aspects of technology testing and analysis, from researching new technologies, to developing new methodologies, to testing with existing and new tools.

When the assessment is complete, we know how to present the results to a broad range of target audiences. We provide our clients with the materials they need, from market-focused data to use in their own collateral to custom sales aids, such as test reports, performance assessments, and white papers. Every document reflects the results of our trusted independent analysis.

We provide customized services that focus on our clients' individual requirements. Whether the technology involves hardware, software, Web sites, or services, we offer the experience, expertise, and tools to help our clients assess how it will fare against its competition, its performance, its market readiness, and its quality and reliability.

Our founders, Mark L. Van Name and Bill Catchings, have worked together in technology assessment for over 20 years. As journalists, they published over a thousand articles on a wide array of technology subjects. They created and led the Ziff-Davis Benchmark Operation, which developed such industry-standard benchmarks as Ziff Davis Media's Winstone and WebBench. They founded and led eTesting Labs, and after the acquisition of that company by Lionbridge Technologies were the head and CTO of VeriTest.