Server Automation Performance Team

HP Server Automation 9.14

Linux Patch Management Performance Characterization

Product	HP Server Automation (SA)
Functional Area	SW Management (Linux Patching)
Release	9.14 Pre-Release (build 45.0.30705.0)
Test Case Description	Patch RHEL Linux systems with representative set of RHN errata
PRD requirement ID	Based on SA PRD section 3.6.b.c: Run in 8 minutes.
Document ID	SA_9.14_linux_patching_v1.0.doc
Version Date	12/4/2012
Document Author	SA Performance Team (*SA_Perf@hp.com*)

Table of Contents

High-Level Performance Characterization Results Summary

HP Server Automation version 9.14 Linux Patch Management functionality was exercised in the performance lab to validate overall throughput, scalability and resource demand for a well-defined workload.

For the first phase of the analysis, using the stated hardware configurations (Appendix A and Appendix B) and representative Use Case, there were no performance regressions observed from the 9.10 to the 9.14 release versions of Server Automation. The Remediation job under test achieved a peak throughput of 4.9 servers per minute in a one slice SA core and 5.5 servers per minute in a two slice SA core at a workload level of 96 managed servers. Linux Patching Job Response time was better than the minimum acceptable time limit for the operation, as specified in the SA Performance Requirements Document (Ver. 7.0).

In a second phase of the analysis, the new SA Linux Patching code using Linux Yum Plug-in capability in place of legacy Yum Adapter functionality, performance and throughout gains were measured for the 9.14 Release. Some performance and job throughput gains are noted using the new Yum Plug-in capability of SA. A detailed comparison of Yum Plug-in vs. Yum Adapter capability and performance is presented.

Phase 1: Performance Regression Evaluation of Linux Patching

Test Case Requirements and Description

The SA Product Requirements Document Version 7.0 (PRD) specified the following test case:

> *Patch twenty RHEL Linux servers with a typical monthly erratum set including its dependencies and completes, returning its results back to the HP Server Automation client within 8 minutes of the user starting the job.*

The implemented RHEL patch management test case consists of the following:

1. Concurrently patches up to 96 RHEL managed servers with January 2008 Red Hat Network erratum
    a. Each managed server is an unpatched RHEL 32-bit AS4 Update 4 virtual machine running on VMware ESX 4.0.
    b. Managed server virtual machines were evenly distributed across 12 ESX hosts.
2. Uses a managed server device group with a single software policy attached which contains the January 2008 RHN erratum
    a. The policy is made up of 14 RHN patch sets which contain a total of 91 RPMs
    b. When run, the job installs 28 of the RPMs with a total file size of 52.79 MB

The test case submits the job via the Twist UAPI through pytwist. The test environment is driven through the SA Axis Test Manager.

Performance Characterization Results

Response Time

Response time is elapsed time required for a job to complete according to SA. For a job submission against a 20 managed servers, the average response time was 7 minutes and 34 seconds in a one slice core and 7 minutes and 40 seconds in a two slice core. This response time is better than the PRD requirement of 8 minutes for this task.

Load Levels and Throughput

Throughput is measured in number of servers processed per minute. This is computed from the total execution time of the job submission divided by the number of servers in the job. Total job throughput was averaged across multiple trials for managed server levels of 1, 5, 10, 20, 30, 40, 50, 75 and 96 managed servers.

# Concurrent Servers Under Test	Throughput 1-Slic e Core (servers/minute)	Throughput 2-Slice Core (servers/minute)	Response Time 1-Slic e Core (min:sec)	Response Time 2-Slice Core (min:sec)
1	0.2	0.2	4:42	4:48
5	1.0	1.0	5:10	5:10
10	1.6	1.6	6:11	6:17
20	2.6	2.8	7:34	7:40
30	3.3	3.5	9:00	8:37
40	3.7	4.0	10:54	9:50
50	4.2	4.8	11:56	10:30
75	4.6	5.4	16:25	13:58
96	4.9	5.5	19:43	17:36

Table 1: Throughput and Response Times for concurrent patching of managed servers

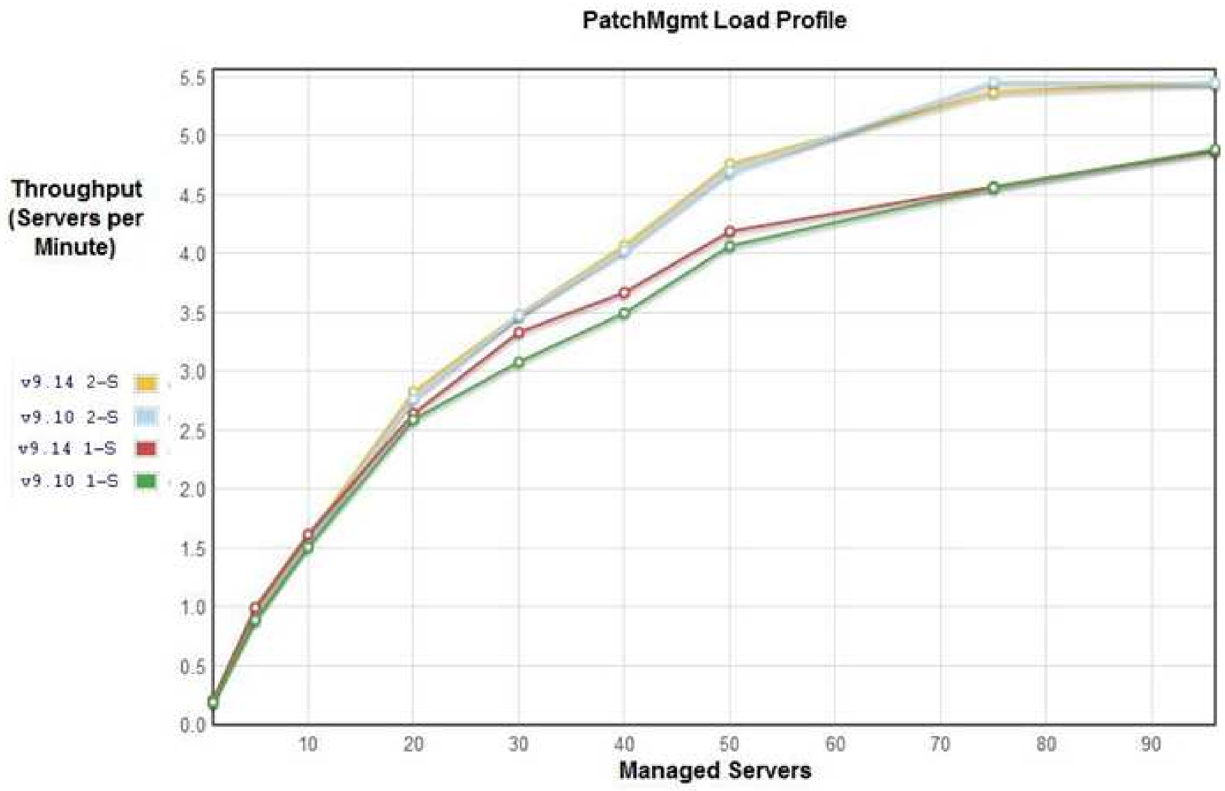


Figure 1: Patch Installation Job Throughput (Servers per Minute)

Performance Regression Analysis Summary

Under the same test conditions, no performance regressions are observed in the 9.14 release. SA code throughput and stability measurements are comparable to that for the 9.10 release.

Some amount of additional job throughout is obtained when a second Slice server is configured in the SA Core. For this workload and environment, a Slice scale-out factor of 1.12 is measured for 9.14 Linux Patching. This factor is similar to that measured for the 9.10 release.

Phase 2: Evaluation of 9.14 Linux Patching Yum Plug-in vs. Yum Adapter

SA Linux Patching uses the Linux *yum* utility to perform patch installation on managed servers. Starting with the 9.14 release, the yum workflow stages can use the newer Yum Plug-in capability in place of the legacy Yum Adapter capability. This new functionality was characterized in the Performance Lab to quantify the improvements provided by the Yum Plug-in capability. For a given Use Case, the operational, performance and scalability characteristics were measured for both the Yum Plug-in and Yum Adapter code paths.

Test Case Requirements and Description

For this investigation, managed servers were configured to RedHat 5 Update 6 operating systems, running inside Virtual Machines. RH AS 5 Update 6 is a supported managed platform for the SA 9.14 Release.

For this characterization, a larger and heavier footprint Software Policy was used to better expose the performance deltas of the different yum code paths.

1. Concurrently patches up to 96 RHEL managed servers with a larger representative set of Red Hat Network erratum, such as that in a RHN Monthly Errata set.
    a. Each managed server is an unpatched RHEL virtual machine running under VMware ESX 4.0
    b. Managed server virtual machines were evenly distributed across 12 ESX hosts
2. Uses a managed server device group with a single software policy attached which contains the representative set of RHN errata. (policy list in Appendix C)
    a. The policy consists of 34 RHN patch sets containing 120 RPMs of 225.5 MB total size
    b. When run, the job installs 42 of the RPMs with a total package size of 90.24 MB.

The test case submits the job via the Twist UAPI through pytwist.

Performance Characterization Results

Overall Job Throughput

Overall job throughput was measured for both 1-Slice and 2-Slice configurations of SA Core, for different combinations of the Yum Plug-in vs. Yum Adapter code paths. For all runs, the best throughout was obtained by using Yum Plug-in for the analyze phase and "rpm install" for the installation phase of the workflow.

The settings to control this behavior are controlled through the following SA tunable parameters:

way.analyze.yum: Determines what tool is used to analyze rpm packages
0 = use the yum bundled with SA (yum v2.6.1)
1 = use the native yum when available, else use the bundled yum (Default)
2 = use the native yum *(best performance)*

way.remediate.yum: Determines which tool is used to install rpm packages
0 = only use rpm (Default) *(best performance)*
1 = use yum when available else use rpm
2 = only use yum

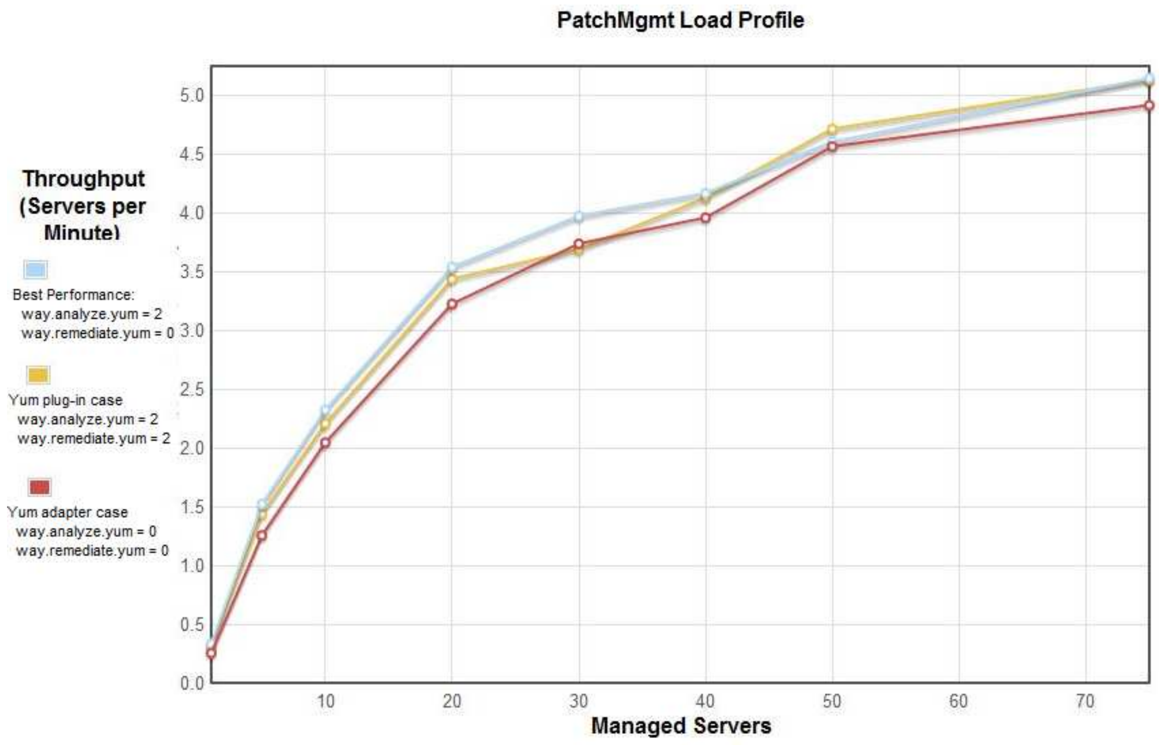


Figure 2: Overall Job Throughput, 1-Slice Core

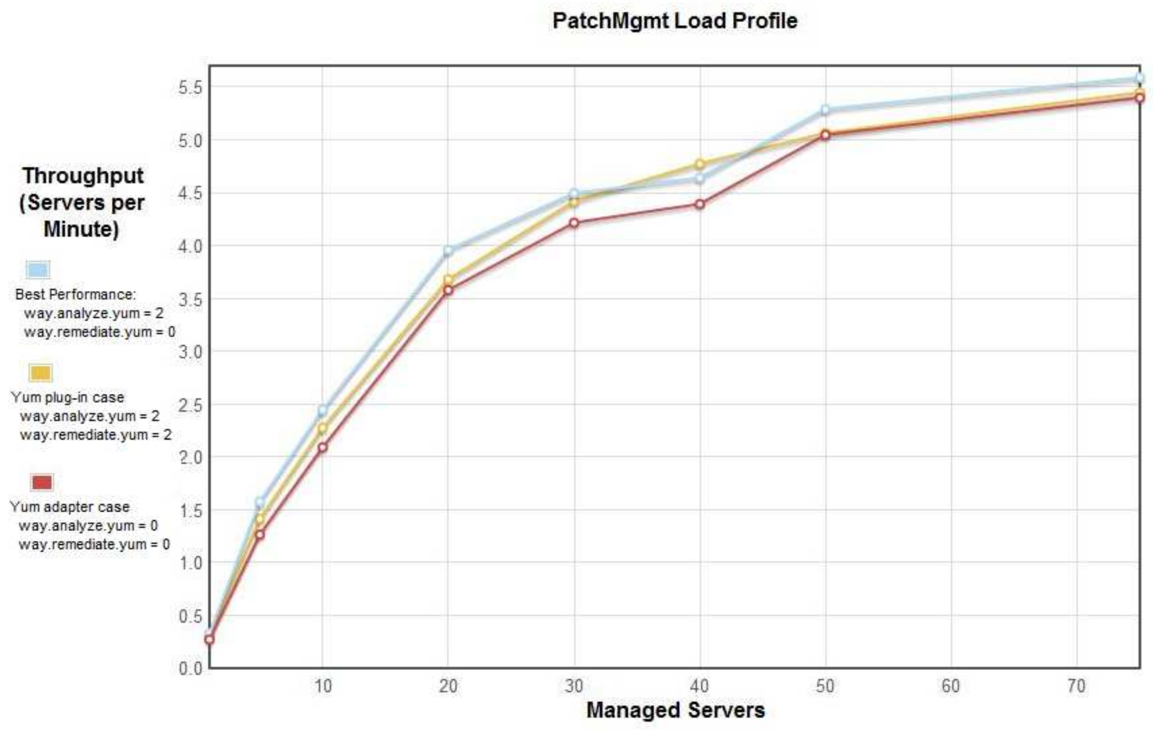


Figure 3: Overall Job Throughput, 2-Slice Core

Workflow Characterization for Yum Plug-in vs. Yum Adapter settings

The following plots illustrate the difference in the SA workflow stages (SA Wayscript thread states) for the Yum Adapter vs. Yum Plus-in code paths. Test conditions are kept the same between the runs: Policy size = 34 items, Managed Servers = 40, 2-Slice SA Core.

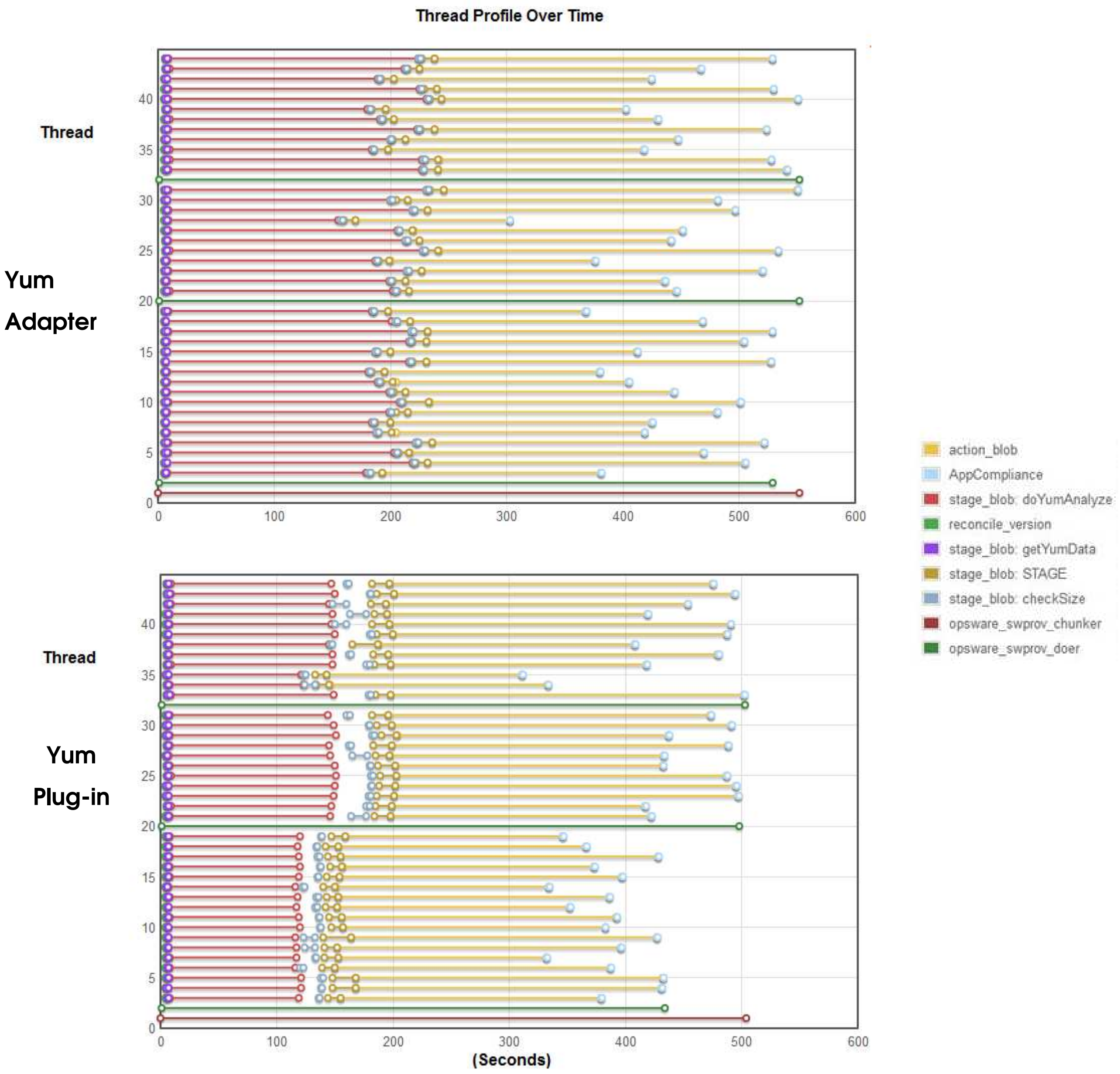


Figure 4: Overall Job Throughput, 2-Slice Core, 40 Managed Servers

The improvements in the earlier Patch Analysis stage that are provided by Yum Plug-in are readily apparent. Patch analysis is faster when executed by Yum Plug-in, resulting in an overall faster response time and improved job throughout for Linux Patching.

Workload Characterization in SA Core and Managed Servers

Linux Patching in SA presents varying resource demands over the lifetime of a job submission. The following series of graphs show the workflow and resource demands across selected parts of SA Core, for the same Job submission. This Job submission consisted of a path operation across 50 Managed Servers concurrently, in a 1-Slice SA Core configuration.

Workflow threads

The following graph shows 50 simultaneous patching operations in flight across the 50 target managed servers.

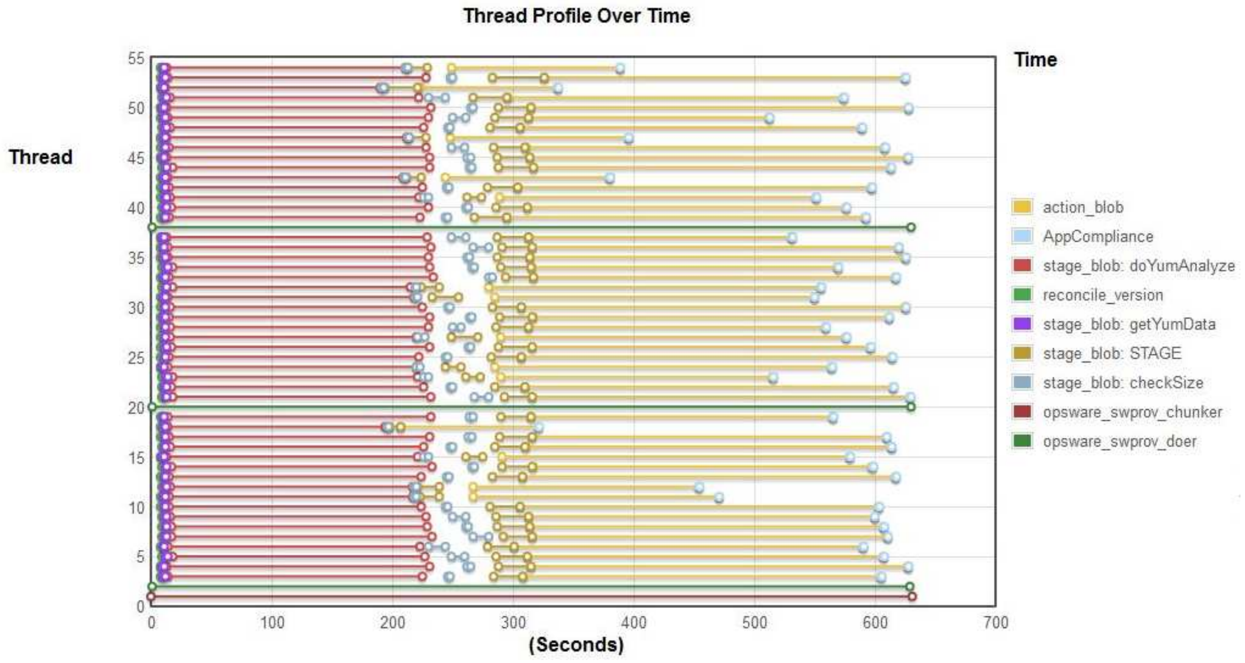


Figure 5: Concurrent patching operations against 50 managed servers in a one slice core

A Red Hat patch management job consists of a single "chunker" session which spawns one or more "doer" sessions. Each doer will operate on one or more managed servers. Several tunable parameters modify how the chunker session distributes and limits the load of the job across these doer sessions and the SA core as a whole. In the above figure, three doers operate on 16-17 managed servers each. Job sessions with more managed servers would have more doer sessions to handle them. If a SA core has more than one slice, then the doers will be evenly distributed across the slices and each slice will operate on a similar number of managed servers. Each managed server in the figure above is managed by the first doer session below it and all doers are managed by the chunker session at the bottom.

Resource Utilization on SA Truth Database

This graph shows that Database CPU demand increases in the early analysis stages and later when software registration runs. Network utilization has several small peaks across the job period.

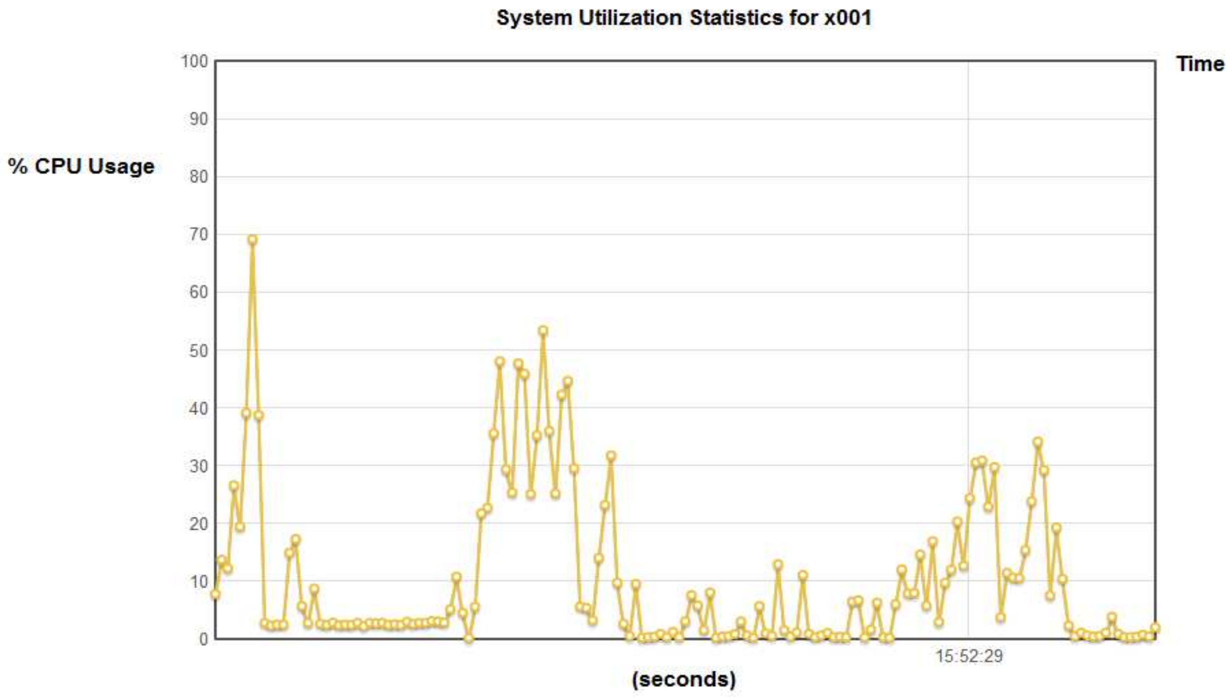


Figure 6: CPU Utilization on Truth Database Server

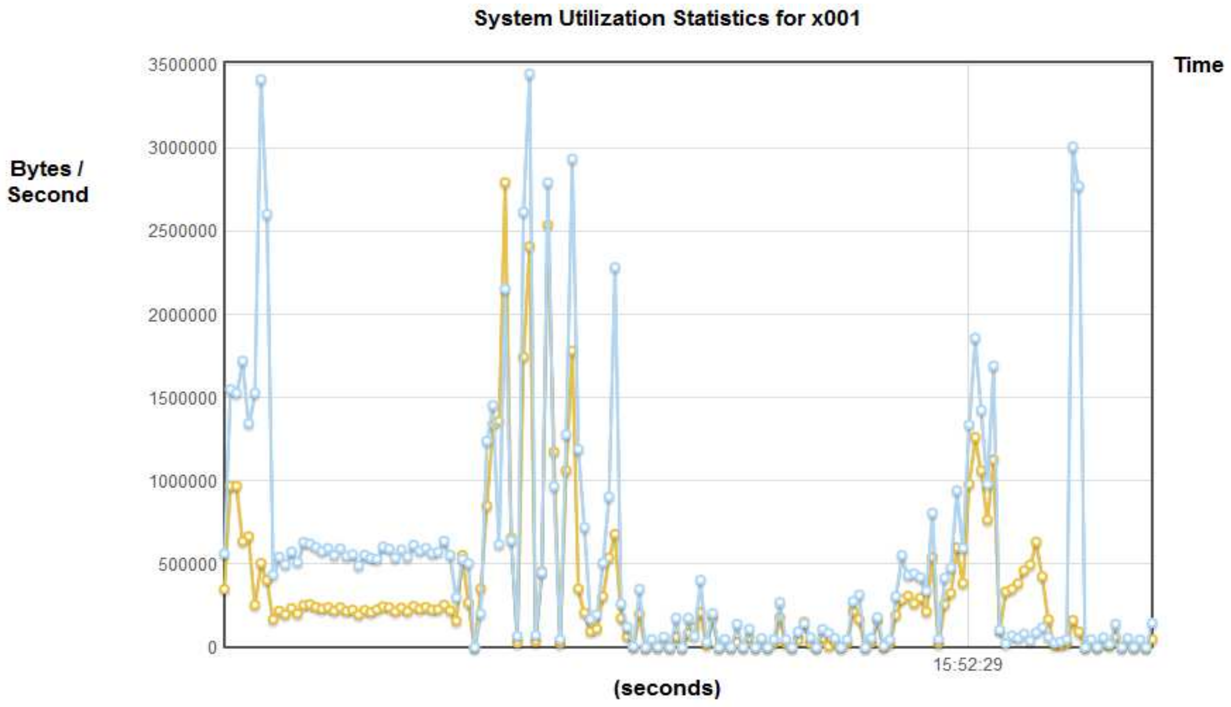


Figure 7: Network Utilization on Truth Database Server

Resource Utilization on SA Slice Server

The following graphs show CPU demand and network utilization on the Infrastructure/Slice server. CPU utilization stays high during the analysis and download stages of the operation, and is lower while the patching operations are being performed on the managed servers.

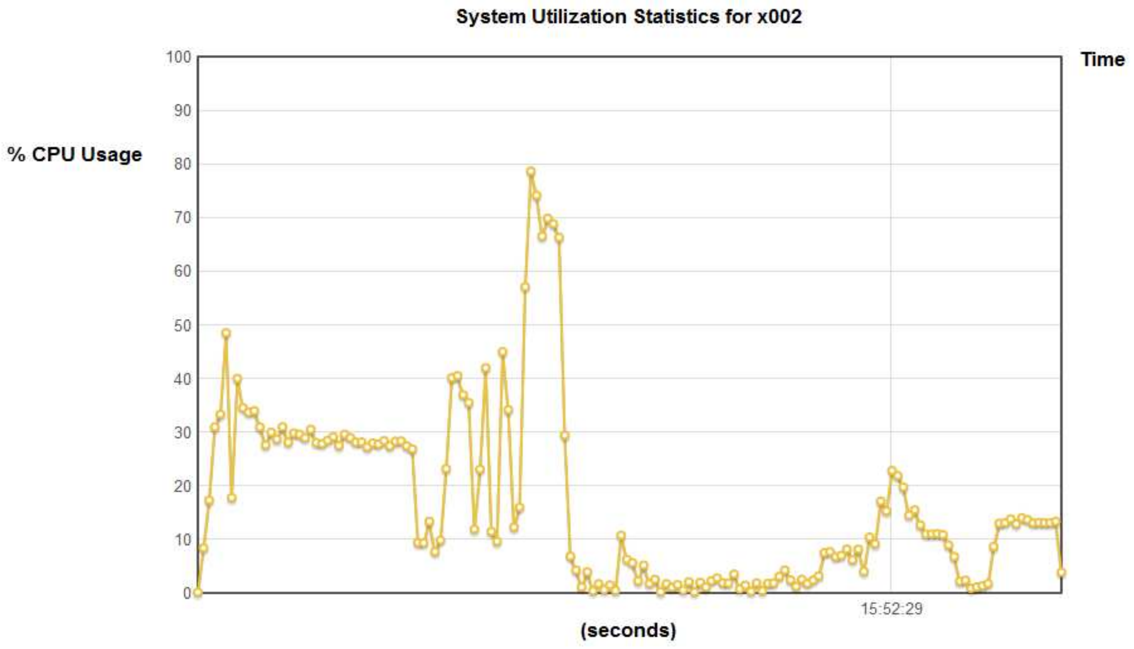


Figure 8: CPU Utilization on Infrastructure/Slice Server

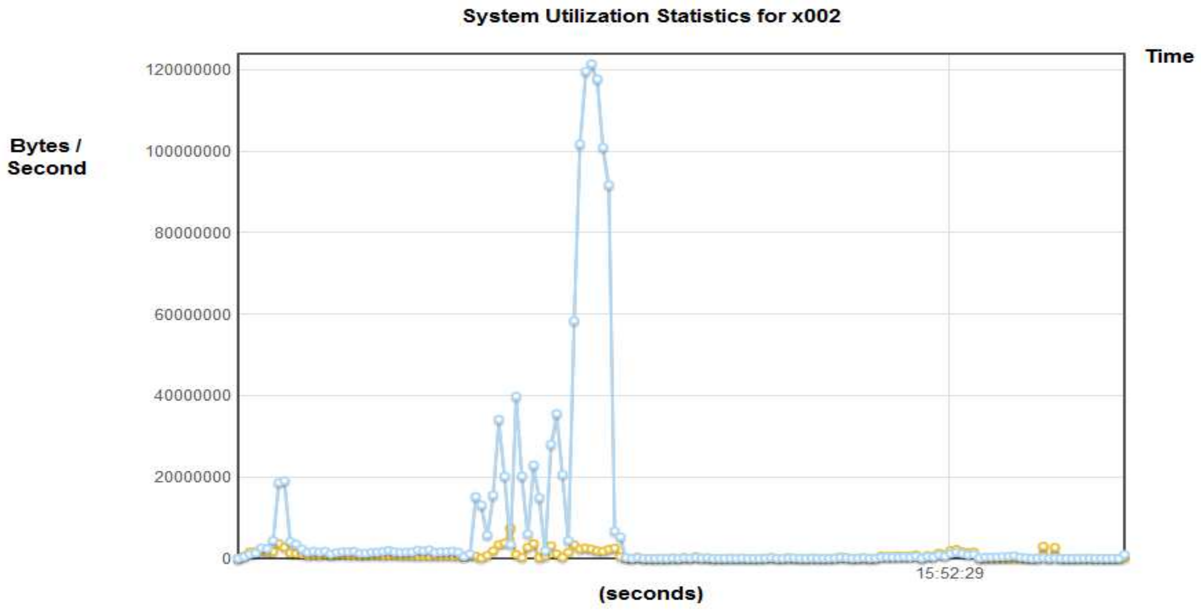


Figure 9: Network Utilization on Infrastructure/Slice Server

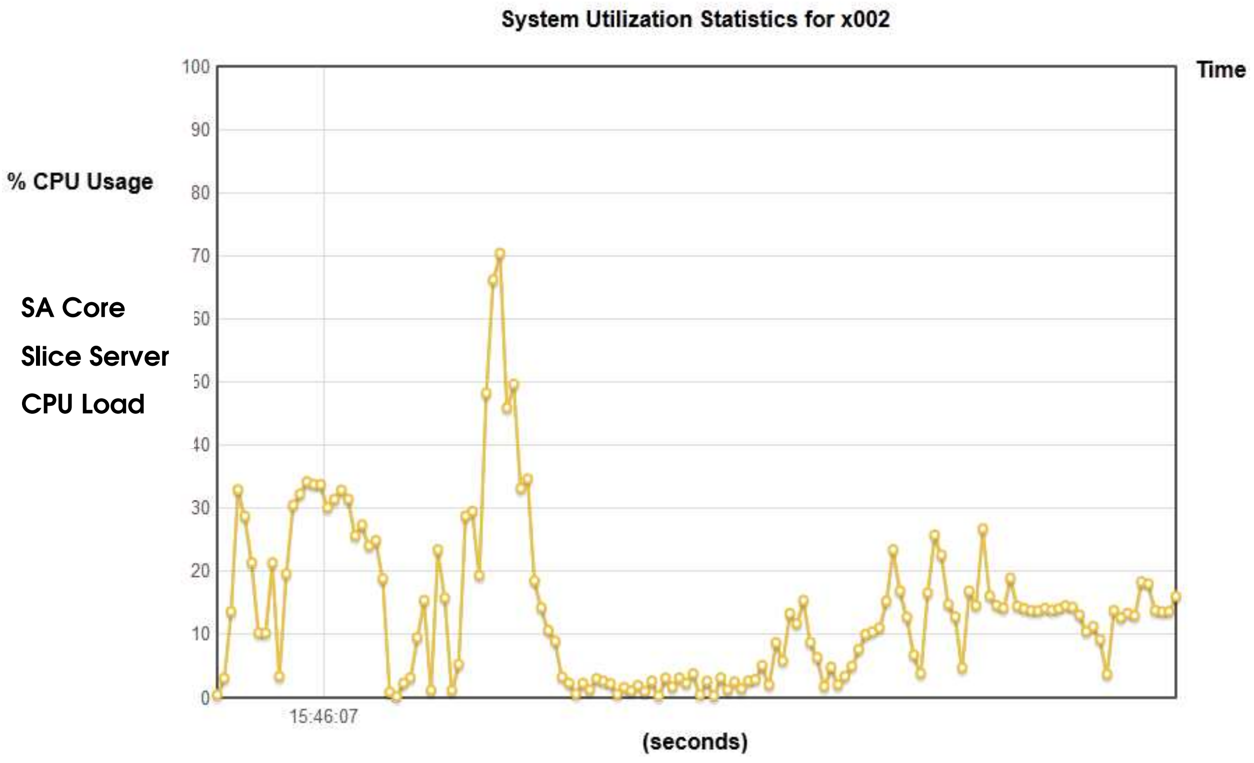
Impact on Managed Server

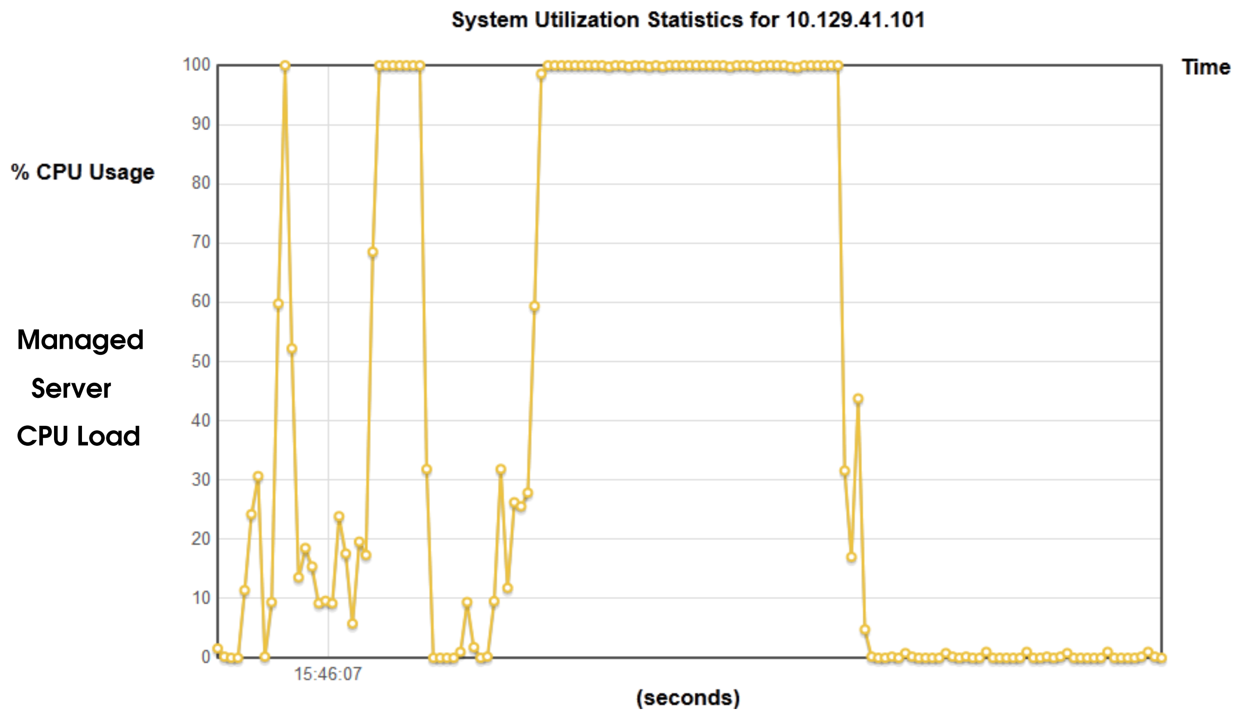
Patch management increases CPU workload on the managed server for the network download and installation stages. This use case puts load on the managed server and its network, disk, and CPU resources, as the patches are moved across the network and installed on the target server. For this characterization, the managed servers were small-footprint 1x virtual CPU servers at 2.66 GHz with 1 GB of memory, and 1 Gb LAN connections to the SA Core.

The single-core managed server tends to have a busy CPU load over the course of most of the task. During the initial Analyze phase and the final Install phase, the CPU usage maintains a high sustained rate, often hitting 100% usage. This high usage rate is primarily due to yum, the Red Hat rpm utility, and the impact of the Opsware HP Server Automation Agent is comparatively minimal. (Reference: SA Performance Team Engineering Whitepaper on SA Agent Performance "HPSA_SA_Agent_Characterization", which is published on the HP Software Self-Solve portal site. During later stages of the patching operation, patch installation on the managed server dominates the operation. Please note that patch installation on the managed server uses exactly the same mechanisms and system resource requirements (CPU, Memory, and Network) as would manual operation of Linux Patching on the Managed Server – SA Agent overhead itself is minimal.

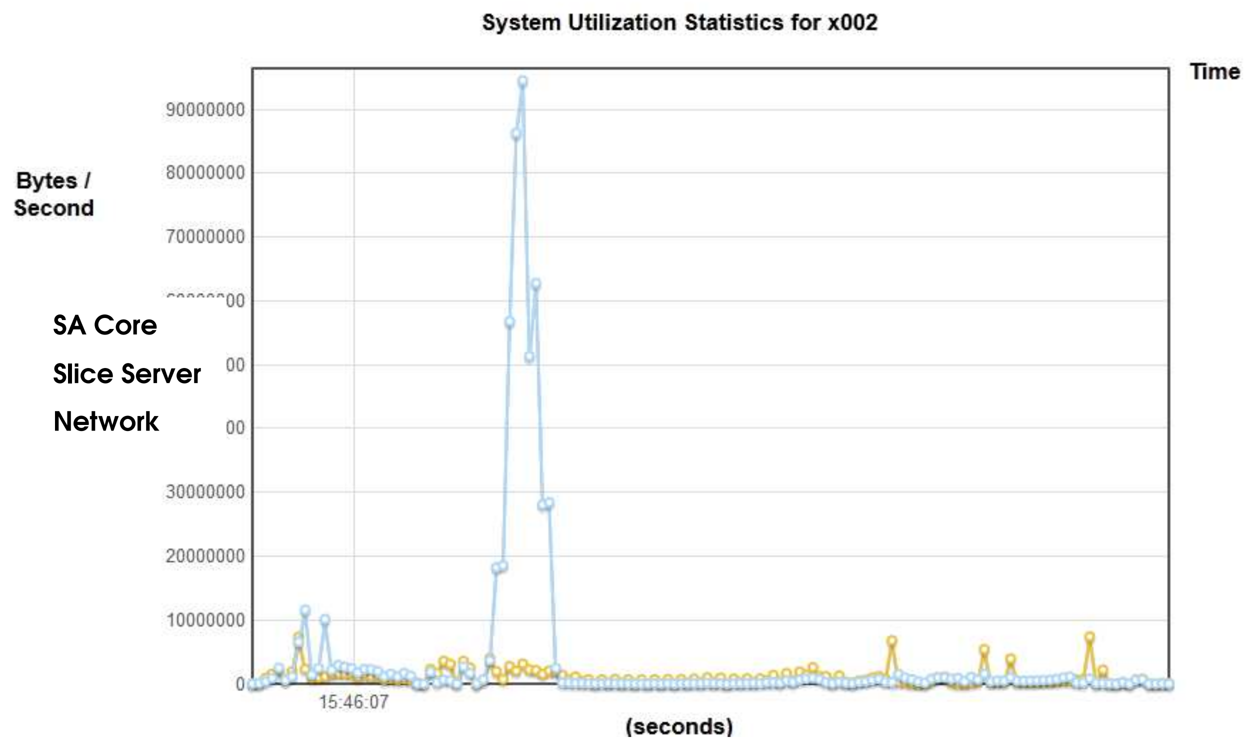
The managed servers used in these tests consisted of up to eight virtual machines sharing a single ESX physical host containing 8 CPU cores; thus the physical CPU, memory, network, and disk resources were all shared at higher load levels. Highly virtualized environments will exhibit competition for these resources if a large number of virtual machines are managed on a small number of physical hosts. Please refer to the best practices of the virtualization software to help reduce managed server performance problems.

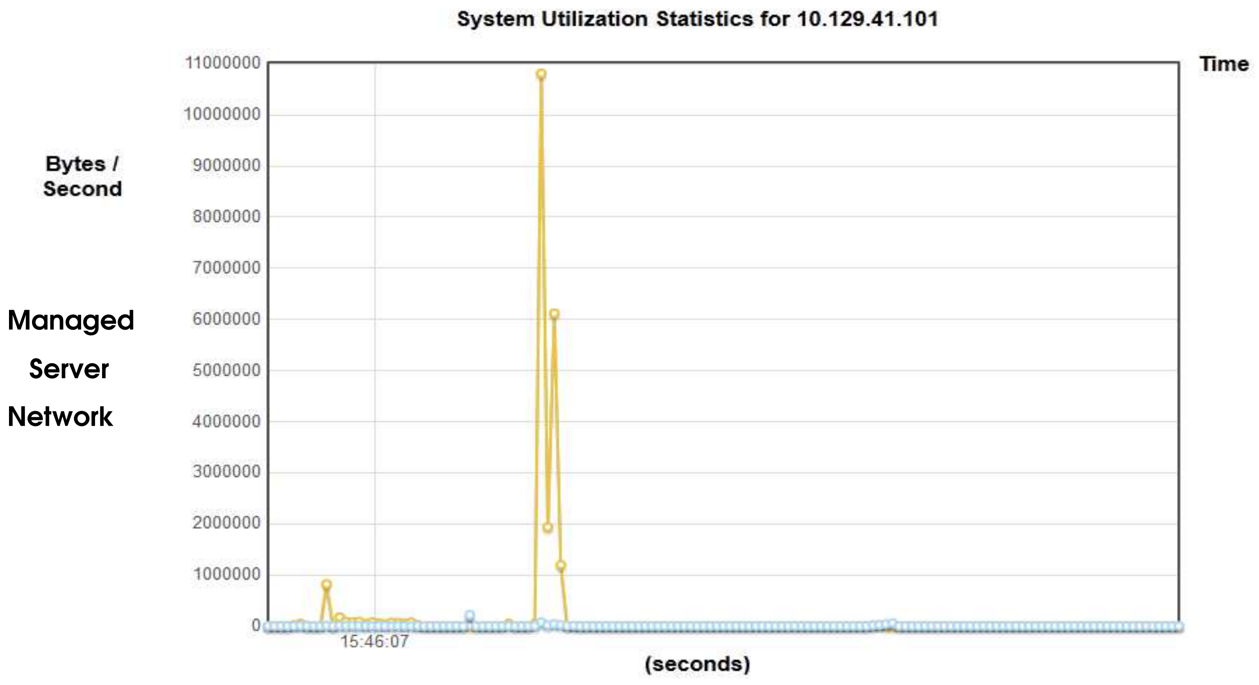
The following plots illustrate representative CPU and Network demands on SA Core and managed server during the various phases of operation of a Linux Patching job.





Representative plots for Network load are given here to illustrate the varying phases of the job.





Inter-operation with new SA Tsunami Remediation functionality

Operation of 9.14 Linux Patching was checked to verify that the Remediation portion of the workflow took advantage of the new SA Tsunami Remediation capability. SA Tsunami is a new version of the Remediation download functionality, which is measured to provide improved Remediation performance and throughout. Significant performance improvement factors of Tsunami Remediation versus legacy SA Remediation have been characterized in the SA Performance Lab. (Reference: SA Performance Team Engineering Whitepaper entitled "SA_9.05_TsunamiRemediationRearchitecture", which is published on the HP Software Self-Solve portal site).

9.14 Linux Patching is verified to use the new Tsunami code paths and so benefits from Tsunami performance improvements. During both the Analyze phase and the Download Phase of the Linux Patching Job, the files served from SA Core to Managed Server are verified to be served from the high-performance Tsunami webserver.

Utilization of the new Tsunami Remediation capability requires that new Tsunami-capable SA Agents are deployed on each of the Managed Servers. For these tests, SA Agents from 9.14 Release (SA build 45.0.30776.0) were installed on the managed servers.

Tunable Parameters

Red Hat patch management uses the Remediate framework to execute. Remediate can be adjusted with the SA web client interface by adjusting the tunable parameters for the Command Engine. All the way.remediate parameters affect the Remediate job type in some way.

In the default configuration, the maximum concurrency for Remediate actions is set by the way.max_remediations.action parameter at 100 servers. This means that up to 100 managed servers can all be operated on at once per datacenter. These actions will be distributed across all active slices in a SA core.

In addition, way.remediate.chunk_size and way.remediate.max_concurrency both affect job distribution and can also limit concurrency if chunk_size × max_concurrency is less than max_remediatetions.action. The chunk_size parameter indicates the maximum number of servers to operate on in a single doer session. By default the chunk_size is 20 managed servers. The chunker will create up to max_concurrency number of doers, which is 10 by default. Once the maximum number of doers has been created, additional managed servers beyond the chunk_size limit will be distributed evenly into the doers.

For example, a Remediate job against 250 managed servers will have 1 chunker that spawns 10 doers which each handles the remediate against 25 managed servers. Meanwhile a 100 managed server job would have 1 chunker that spawned 5 doers each operating against 20 managed servers. A 101 server job would have 1 chunker, 6 spawned doers, and each doer would handle 16-17 servers.

Tunable parameters to control the use of Yum Plug-in vs. Yum Adapter operation have already been documented in the paper. SA 9.14 default settings are configured to choose the best all-around combinations of performance of these settings.

Conclusions

For the given test configuration, the test case met the requirements stated in the PRD. With a job submission of 20 managed servers updated concurrently, a throughput of 2.6 servers per minute in a one slice configuration at a response time of 7 minutes and 34 seconds, and 2.8 servers per minute in a two slice configuration at a response time of 7 minutes and 40 seconds, was achieved which is under the stated 8 minutes in the PRD.

No performance regressions are observed for the given test cases when comparing the 9.14 to the 9.10 SA release.

For the 9.14 Release, the use of the new Yum Plug-in capability can provide performance improvements. For the given test cases, SA linux patching throughout is improved.

At the tested load levels, job times are heavily dependant on the complexity and size of the RPM set being installed. As the number of concurrent managed server operations is increased beyond the load levels tested, the Word component in the slice will become a more dominant limiting factor. Tsunami Remediation improvements are in use in this SA job workflow.

Some horizontal scalability of the slice server component is seen at these load levels.

For the measured results, it is believed that the resource capability of the test system itself is limiting the observed results. Managed server resources tend to be the dominant factor for larger patch bundle sizes. The managed servers run on virtual machines that run on a number of physical servers. There are limits to the resources of CPU cycles and bandwidth to SAN (hosting the root disk images) available to the managed servers. At higher levels of concurrent managed server operations, test system limitations may further skew the observed results.

Appendix A: System Configuration

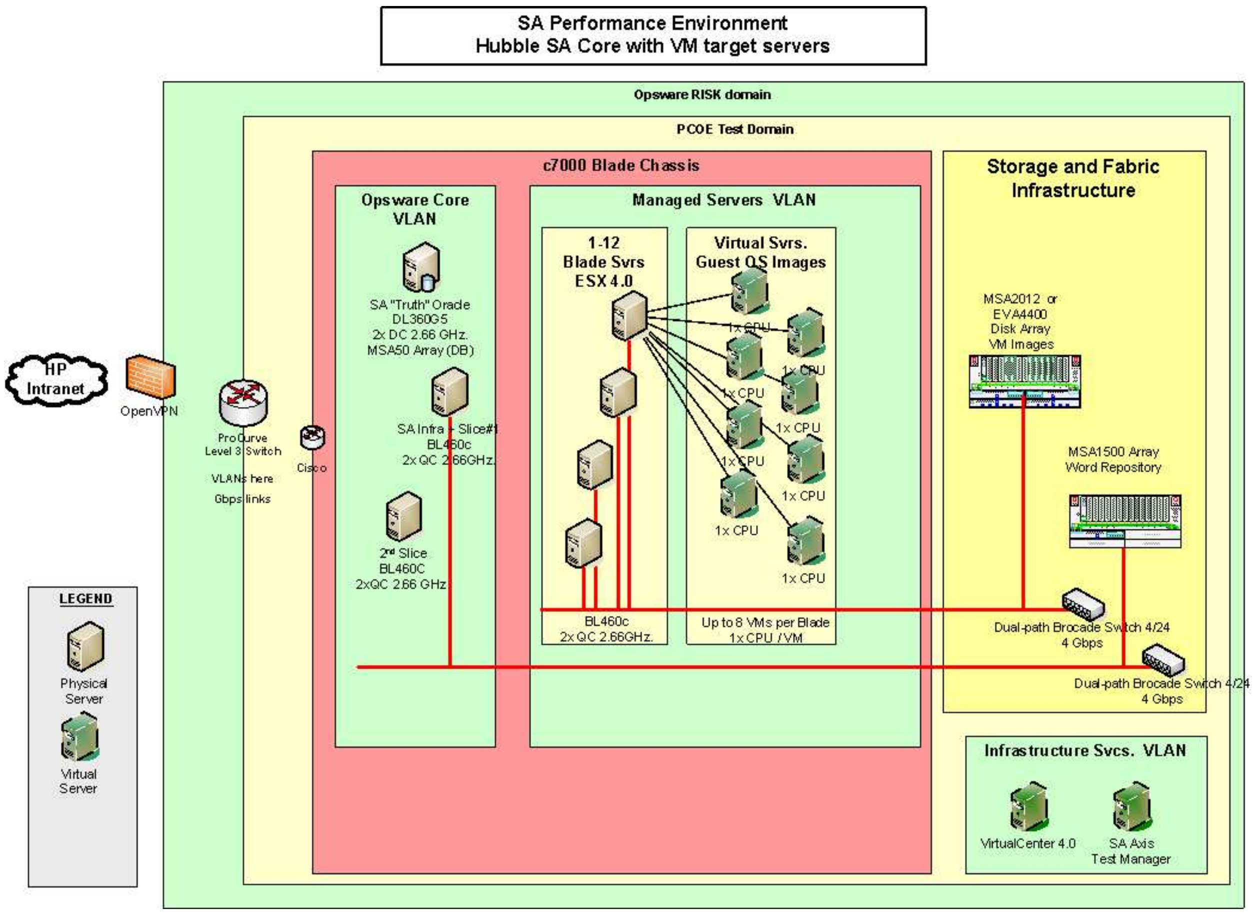
HP Server Automation Core

Server Role	Model Repository Database (Truth)
Hardware Specs	Local Disk: 2x 72GB 10K SA RAID-1+0 (Linux boot) Local Array (Oracle): MSA50 12x 72GB 10K SAA, Ultra-SCSI connect Memory: 32GB OS: RHEL AS 4 64-bit CPU: 2x Dual Core 2.66 GHz. Intel Xeon 5150 Model: HP Proliant DL360G5
Network Config	Network: 1 Gbps LAN, dedicated VLAN
Software Specs	Oracle 11 Standard Edition
Additional Notes	SA 9.14 - Build 45.0.30705.0
Server Role	Infrastructure and Slice services (First Slice) Multimaster (Vault) Data access engine (Spin - primary) Media Repository (Word storage on NFS, SMB) Gateways (mgw, cgw, agw) Command Engine (Way) Web service API (Twist) Opsware Global File System (Hub) Word Build Manager
Hardware Specs	Local Disk: 2x 72GB 10K SA RAID-1+0 (Linux boot) SAN Attach: 4Gbps single path FC, MSA Array Memory: 16GB OS: RHEL AS 4 64-bit CPU: 2x Quad-Core 2.66 GHz Intel Xeon 5355 Model: HP BL460cG1
Network Config	Network: 1 Gbps LAN, dedicated VLAN
Software Specs	SA 9.14 - Build 45.0.30705.0
Server Role	Slice Services (Second slice) Data access engine (Spin - secondary) Gateways (cgw, agw) Command Engine (Way) Web service API (Twist) Opsware Global File System (Hub) Word
Hardware Specs	Local Disk: 2x 72GB 10K SA RAID-1+0 (Linux boot) Memory: 16GB OS: RHEL AS 4 64-bit CPU: 2x Quad-Core 2.66 GHz Intel Xeon 5355 Model: HP BL460cG1
Network Config	Network: 1 Gbps LAN, dedicated VLAN
Software Specs	SA 9.14 - Build 45.0.30705.0

Managed Servers

Server Types	Blade servers, hosting VMware VMs
Hardware Specs	Local Disk: 2x 72GB 10K HP Server Automation RAID-1+0 (ESX boot) SAN Attach: 4Gbps dual path FC, EVA4400 Array (VM images) Memory: 32GB PC2-5300 OS: VMware ESX Server 4.0 CPU: 2x Quad-Core 2.66 GHz Intel Xeon 5355 Model: HP BL460cG1 VMware VM Images: 1x Virtual CPU 1 GB Virtual Memory
Network Config	Network: 1 Gbps LAN, dedicated VLAN
Software Specs	HP Server Automation 9.14 Agent - Build 45.0.30776.0
Additional Notes	RHEL VM load evenly distributed across 12 VMware ESX hosts

Appendix B: Performance Characterization Environment


SA Performance Environment
Hubble SA Core with VM target servers
Opsware RISK domain
PCOE Test Domain
c7000 Blade Chassis
Opsware Core VLAN
SA "Truth" Oracle DL360G5 2x DC 2.66 GHz. MSA50 Array (DB)
SA Infra – Slice#1 BL460c 2x QC 2.66GHz.
2nd Slice BL460C 2xQC 2.66 GHz
Managed Servers VLAN
1-12 Blade Svrs ESX 4.0
BL460c 2x QC 2.66GHz.
Virtual Svrs. Guest OS Images
1x CPU
1x CPU
1x CPU
1x CPU
1x CPU
1x CPU
1x CPU
1x CPU
Up to 8 VMs per Blade 1x CPU / VM
Storage and Fabric Infrastructure
MSA2012 or EVA4400 Disk Array VM Images
MSA1500 Array Word Repository
Dual-path Brocade Switch 4/24 4 Gbps
Dual-path Brocade Switch 4/24 4 Gbps
Infrastructure Svcs. VLAN
VirtualCenter 4.0
SA Axis Test Manager
HP Intranet
OpenVPN
ProCurve Level 3 Switch
VLANs here
Gbps links
Cisco
LEGEND
Physical Server
Virtual Server

Appendix C: Software Policy (Detailed List)

Policy:	**"RHEL5 Server Patches"**
Total Package Size:	**225.5 MBytes**

Contained Policies: 34

RHBA-2012:1437
RHBA-2012:1411
RHBA-2012:1429
RHBA-2012:1406
RHBA-2012:1348
RHBA-2012:1342
RHBA-2012:1335
RHBA-2012:1303
RHBA-2012:1272
RHBA-2012:1271
RHBA-2012:1270
RHBA-2012:1260
RHBA-2012:1257
RHBA-2012:1249
RHBA-2012:1224
RHBA-2012:1193
RHBA-2012:1191
RHBA-2012:1160
RHBA-2012:1155
RHBA-2012:1126
RHBA-2012:1074
RHBA-2012:1071
RHBA-2012:1016
RHBA-2012:0714
RHBA-2012:0684
RHBA-2012:0674
RHBA-2012:0672
RHBA-2012:0580
RHBA-2012:0574
RHBA-2012:0560
RHBA-2012:0558
RHBA-2012:0553
RHBA-2012:0537
RHBA-2012:0527

Packages: 120 total

util-linux-2.13-0.59.el5_8.i386
firefox-10.0.10-2.el5_8.i386
perl-5.8.8-38.el5_8.i386
perl-suidperl-5.8.8-38.el5_8.i386
OpenIPMI-2.0.16-13.el5_8.i386
OpenIPMI-devel-2.0.16-13.el5_8.i386
OpenIPMI-gui-2.0.16-13.el5_8.i386
OpenIPMI-python-2.0.16-13.el5_8.i386
OpenIPMI-tools-2.0.16-13.el5_8.i386
OpenIPMI-libs-2.0.16-13.el5_8.i386
OpenIPMI-perl-2.0.16-13.el5_8.i386
cman-2.0.115-96.el5_8.4.i386

```
cman-devel-2.0.115-96.el5_8.4.i386
sssd-client-1.5.1-49.el5_8.3.i386
sssd-tools-1.5.1-49.el5_8.3.i386
libipa_hbac-1.5.1-49.el5_8.3.i386
libipa_hbac-devel-1.5.1-49.el5_8.3.i386
libipa_hbac-python-1.5.1-49.el5_8.3.i386
sssd-1.5.1-49.el5_8.3.i386
xorg-x11-server-Xdmx-1.1.1-48.91.el5_8.2.i386
xorg-x11-server-Xephyr-1.1.1-48.91.el5_8.2.i386
xorg-x11-server-Xnest-1.1.1-48.91.el5_8.2.i386
xorg-x11-server-Xorg-1.1.1-48.91.el5_8.2.i386
xorg-x11-server-Xvfb-1.1.1-48.91.el5_8.2.i386
xorg-x11-server-Xvnc-source-1.1.1-48.91.el5_8.2.i386
xorg-x11-server-sdk-1.1.1-48.91.el5_8.2.i386
perl-LDAP-0.33-4.el5_8.noarch
kexec-tools-1.102pre-154.el5_8.1.i386
linuxwacom-0.7.8.3-11.2.el5_8.i386
linuxwacom-devel-0.7.8.3-11.2.el5_8.i386
sudo-1.7.2p1-14.el5_8.4.i386
mod_nss-1.0.8-4.el5_8.2.i386
gdb-7.0.1-42.el5_8.1.i386
ipsec-tools-0.6.5-14.el5_8.5.i386
cyrus-sasl-md5-2.1.22-7.el5_8.1.i386
cyrus-sasl-sql-2.1.22-7.el5_8.1.i386
cyrus-sasl-gssapi-2.1.22-7.el5_8.1.i386
cyrus-sasl-lib-2.1.22-7.el5_8.1.i386
cyrus-sasl-devel-2.1.22-7.el5_8.1.i386
cyrus-sasl-ntlm-2.1.22-7.el5_8.1.i386
cyrus-sasl-plain-2.1.22-7.el5_8.1.i386
cyrus-sasl-2.1.22-7.el5_8.1.i386
cyrus-sasl-ldap-2.1.22-7.el5_8.1.i386
xorg-x11-server-Xdmx-1.1.1-48.91.el5_8.i386
xorg-x11-server-Xnest-1.1.1-48.91.el5_8.i386
xorg-x11-server-Xvfb-1.1.1-48.91.el5_8.i386
xorg-x11-server-Xephyr-1.1.1-48.91.el5_8.i386
xorg-x11-server-Xorg-1.1.1-48.91.el5_8.i386
xorg-x11-server-Xvnc-source-1.1.1-48.91.el5_8.i386
xorg-x11-server-sdk-1.1.1-48.91.el5_8.i386
pdksh-5.2.14-37.el5_8.1.i386
e2fsprogs-1.39-34.el5_8.1.i386
e2fsprogs-devel-1.39-34.el5_8.1.i386
e2fsprogs-libs-1.39-34.el5_8.1.i386
uuidd-1.39-34.el5_8.1.i386
sudo-1.7.2p1-14.el5_8.3.i386
initscripts-8.45.42-1.el5_8.1.i386
samba3x-common-3.5.10-0.110.el5_8.i386
samba3x-swat-3.5.10-0.110.el5_8.i386
samba3x-domainjoin-gui-3.5.10-0.110.el5_8.i386
samba3x-winbind-devel-3.5.10-0.110.el5_8.i386
samba3x-3.5.10-0.110.el5_8.i386
samba3x-client-3.5.10-0.110.el5_8.i386
samba3x-doc-3.5.10-0.110.el5_8.i386
samba3x-winbind-3.5.10-0.110.el5_8.i386
subscription-manager-0.98.16.3-1.el5_8.i386
```

```
subscription-manager-firstboot-0.98.16.3-1.el5_8.i386
subscription-manager-gnome-0.98.16.3-1.el5_8.i386
subscription-manager-migration-0.98.16.3-1.el5_8.i386
openldap-2.3.43-25.el5_8.1.i386
openldap-devel-2.3.43-25.el5_8.1.i386
compat-openldap-2.3.43_2.2.29-25.el5_8.1.i386
openldap-clients-2.3.43-25.el5_8.1.i386
openldap-servers-overlays-2.3.43-25.el5_8.1.i386
openldap-servers-sql-2.3.43-25.el5_8.1.i386
openldap-servers-2.3.43-25.el5_8.1.i386
mod_ssl-2.2.3-65.el5_8.i386
httpd-2.2.3-65.el5_8.i386
httpd-devel-2.2.3-65.el5_8.i386
httpd-manual-2.2.3-65.el5_8.i386
ipa-client-2.1.3-2.el5_8.i386
net-snmp-devel-5.3.2.2-17.el5_8.1.i386
net-snmp-perl-5.3.2.2-17.el5_8.1.i386
net-snmp-5.3.2.2-17.el5_8.1.i386
net-snmp-libs-5.3.2.2-17.el5_8.1.i386
net-snmp-utils-5.3.2.2-17.el5_8.1.i386
binutils-2.17.50.0.6-20.el5_8.3.i386
binutils-devel-2.17.50.0.6-20.el5_8.3.i386
tar-1.15.1-32.el5_8.i386
mod_dav_svn-1.6.11-10.el5_8.i386
subversion-devel-1.6.11-10.el5_8.i386
subversion-javahl-1.6.11-10.el5_8.i386
subversion-ruby-1.6.11-10.el5_8.i386
subversion-1.6.11-10.el5_8.i386
subversion-perl-1.6.11-10.el5_8.i386
cpp-4.1.2-52.el5_8.1.i386
gcc-objc-4.1.2-52.el5_8.1.i386
libgnat-4.1.2-52.el5_8.1.i386
gcc-gnat-4.1.2-52.el5_8.1.i386
gcc-4.1.2-52.el5_8.1.i386
gcc-gfortran-4.1.2-52.el5_8.1.i386
libmudflap-devel-4.1.2-52.el5_8.1.i386
libgcj-src-4.1.2-52.el5_8.1.i386
libgfortran-4.1.2-52.el5_8.1.i386
libstdc++-4.1.2-52.el5_8.1.i386
libgcj-4.1.2-52.el5_8.1.i386
libobjc-4.1.2-52.el5_8.1.i386
libstdc++-devel-4.1.2-52.el5_8.1.i386
gcc-java-4.1.2-52.el5_8.1.i386
libgcc-4.1.2-52.el5_8.1.i386
gcc-c++-4.1.2-52.el5_8.1.i386
gcc-objc++-4.1.2-52.el5_8.1.i386
libgcj-devel-4.1.2-52.el5_8.1.i386
libmudflap-4.1.2-52.el5_8.1.i386
openais-0.80.6-36.el5_8.1.i386
openais-devel-0.80.6-36.el5_8.1.i386
wget-1.11.4-3.el5_8.2.i386
openais-0.80.6-36.el5_8.2.i386
openais-devel-0.80.6-36.el5_8.2.i386
vsftpd-2.0.5-24.el5_8.1.i386
```