TECHNICAL WHITE PAPER

Nimble Storage for Splunk on Oracle Linux & RHEL 6

Document Revision

Table 1.

Date	Revision	Description
11/9/2014	1.0	Initial Draft
01/15/2015	1.1	Internal Review
1/26/2015	1.2	Released

TABLE OF CONTENTS

Introduction

The purpose of this technical white paper is to walk through the steps for tuning a Linux operating system for Splunk running on Nimble Storage.

Audience

This guide is intended for Splunk solution architects, storage engineers, system administrators, and IT managers who analyze, design, and maintain a robust Splunk environment on Nimble Storage. It is assumed that the reader has a working knowledge of iSCSI SAN network design, and basic Nimble Storage operations. Knowledge of Oracle Linux and Red Hat operating system is also required.

Scope

Splunk's reference server hardware comprises two CPU 6-cores at 2Ghz per core, 12GB of memory, and a few 10K or 15K rpm drives that can sustain 1200 IOPS. To save storage capacity, and with the data stored on local disk, Splunk provides application-level compression. To prevent a SPOF, index replication can be configured in a cluster.

This white paper explains Nimble technology as well as how it can lower the TCO of your Splunk environment and still achieve better performance. This paper also discusses the best practices for implementing the Linux operating system for Splunk on Nimble Storage.

Solution Overview

Splunk Architecture

Splunk, headquartered in San Francisco, California, provides the leading software platform for real-time Operational Intelligence. Splunk solutions enable organizations to search, monitor, analyze, and visualize machine-generated data coming from websites, applications, servers, networks, sensors, and mobile devices. Splunk software helps organizations deepen business and customer understanding, mitigate cybersecurity risk, prevent fraud, improve service performance, and reduce cost.

Common Splunk Deployment Architectures

- Departmental
- Small enterprise
- Medium enterprise
- Large enterprise

Departmental Deployment

- A single Splunk instance, combining the functionality of both an indexer and a search head.
- Indexing volume of under 20GB/day
- A relatively small number of forwarders sending data to the instance, typically less than 10 and rarely exceeding 100
- Updates are handled either manually or via a deployment server resident on the indexer
- A few users, typically less than 10

Small Enterprise Deployment

- Several Splunk instances; for example, two or three indexers and a single search head that allows users to run combined searches across all the indexers
- Indexing volume between 20-100GB/day
- Up to several hundred forwarders feeding data to the indexers. The forwarders typically use load balancing to distribute the data across the indexes
- Updates are handled either manually or via a deployment server resident on the search head
- A large number of users, but generally well under 100

Medium Enterprise Deployment

- A large number of Splunk instances; for example, five or more indexers and a couple of search heads
- Indexing volume between 100-300GB/day
- Up to a few thousand forwarders feeding load-balanced data to the indexers
- Updates are handled by a separate configuration management tool, which is either a stand-alone deployment server or a third party tool like Puppet or Chef
- A large number of users, possibly numbering a hundred or more

Large Enterprise Deployment

- A large number of Splunk instances; for example, several dozen indexers and as many as 10 search heads
- Indexing volume ranging from 300GB to many TBs per day
- Many thousands of forwarders
- Updates are handled by a separate configuration management tool, which is either a stand-alone deployment server or a third party tool like Puppet or Chef
- A large number of users, potentially numbering in several hundreds

Nimble Storage Architecture

Adaptive Flash Storage Solutions Overview

Whether you're a small business or a large enterprise, Nimble Storage delivers the right mix of performance and capacity that your applications need, at a price you can afford. Engineered for efficiency, Nimble Storage allows you to consolidate storage for multiple workloads onto a single array. Fast and efficient storage snapshot-based backup/restore and replication functionality also come standard.

With Nimble Storage's Adaptive Flash platform, you can:

- Boost the performance of all your applications, with sub-millisecond latencies
- Maximize capacity and store more data with inline compression, thin provisioning, and efficient cloning
- Eliminate silos as well as scale storage performance and capacity to fit application requirements and budgets by adding compute, cache, or capacity independently and non-disruptively
- Keep storage and applications up and running with frequent snapshots and consistent backups, fast restores, and efficient replication for disaster recovery
- Simplify management with push-button deployment, integration with familiar tools such as VMware vCenter and Microsoft (SCVMM), and proactive monitoring with InfoSight

Cache Accelerated Sequential Layout (CASL™)

Nimble Storage solutions are built on its patented Cache Accelerated Sequential Layout (CASL™) architecture. CASL leverages the unique properties of flash and disk to deliver high performance and capacity – all within a dramatically small footprint.

CASL and InfoSight™ form the foundation of the Adaptive Flash platform, which allows for the dynamic and intelligent deployment of storage resources to meet the growing demands of business-critical applications.

Flexible Flash Scaling

Flexibly scale flash to satisfy the changing performance demands of today's business-critical applications.

Dynamic Flash-Based Read Caching

CASL caches "hot" active data onto SSD in real time—without the need to set complex policies. This way, it can instantly respond to read requests—as much as 10X faster than traditional bolt-on or tiered approach to flash.

Write-Optimized Data Layout:

CASL collects or coalesces random writes, compresses them, and writes them sequentially to disks. This results in write operations that are as much as 100x faster than traditional disk-based storage.

Inline Compression

CASL compresses data as it is written to the array with no performance impact. It takes advantage of efficient variable block compression and multicore processors. A recent measurement of our installed base shows average compression rates from 30 to 75 percent for a variety of workloads.

Scale-to-Fit Flexibility

CASL allows for the non-disruptive and independent scaling of performance and capacity. This is accomplished by upgrading the storage controller (compute) for higher throughput, moving to a larger flash SSD (cache) to accommodate more active data, or by adding storage shelves to boost capacity. This flexible scaling eliminates the need for disruptive forklift upgrades.

Scale Out

Scale capacity and performance beyond the physical limitations of a single array by seamlessly clustering any combination of Nimble Storage hybrid arrays. Eliminate capacity silos and performance hotspots, and easily manage all hardware resources across the cluster as a single storage entity.

Snapshots and Integrated Data Protection

CASL can take thousands of point-in-time instant snapshots of volumes by creating a copy of the volumes' indices. Any updates to existing data or new data written to a volume are redirected to free space (optimized by CASL's unique data layout). This means that snapshots have no performance impact and take little incremental space as only changes are maintained. This also simplifies restoring snapshots, as no data need to be copied.

Efficient Replication

Nimble Storage efficiently replicates data to another array by transferring compressed, block-level changes only. These remote copies can be made active if the primary array becomes unavailable. This allows easy and affordable deployment of disaster data recovery – especially over a WAN to a remote array where bandwidth is limited.

Zero-Copy Clones

Nimble Storage arrays can instantly create snapshot-based read/writeable clones of existing volumes. These clones benefit from fast read and write performance, making them ideal for demanding applications such as VDI or test/development.

InfoSight

InfoSight leverages the power of deep-data analytics and cloud-based management to deliver true operational efficiency across all storage activities. It ensures the peak health of storage infrastructure by identifying problems, and offering solutions, in real time. InfoSight provides expert guidance for deploying the right balance of storage resources — dynamically and intelligently — to satisfy the changing demands of business-critical applications.

Nimble Storage's Adaptive Flash platform provides a solution for some of the biggest storage challenges with Splunk deployments, including management, performance, scalability, availability, and data protection. Nimble Storage arrays help reduce the total cost of ownership, and increase the return on investment for Splunk Enterprise environments.

Tested Solution

Nimble Recommended Settings for Splunk Indexer

Nimble Array

- The Nimble OS should be at least 2.1.4 on a CS300 or CS500 or a CS700 series

Linux Operating System

- iSCSI Timeout and Performance Settings

Understanding the meaning of these iSCSI timeouts allows administrators to set these timeouts appropriately. These iSCSI timeouts parameters in the */etc/iscsi/iscsid.conf* file should be set as follows:

node.session.timeo.replacement_timeout = 120

node.conn[0].timeo.noop_out_interval = 5

node.conn[0].timeo.noop_out_timeout = 10

node.session.nr_sessions = 4

node.session.cmds_max = 2048

node.session.queue_depth = 1024

= = = NOP-Out Interval/Timeout = = =

node.conn[0].timeo.noop_out_timeout = [value]

The iSCSI layer sends a NOP-Out request to each target. If a NOP-Out request times out (default - 10 seconds), the iSCSI layer responds by failing any running commands and instructing the SCSI layer to re-queue those commands when possible. If dm-multipath is being used, the SCSI layer will fail those running

commands and defer them to the multipath layer. The multipath layer then retries those commands on another path. If dm-multipath is not being used, those commands are retried five times (node.conn[0].timeo.noop_out_interval) before failing altogether.

node.conn[0].timeo.noop_out_interval [value]

Once set, the iSCSI layer will send a NOP-Out request to each target every [interval value] seconds.

= = = SCSI Error Handler = = =

If the SCSI Error Handler is running, running commands on a path would not be failed immediately when a NOP-Out request times out on that path. Instead, those commands would be failed after replacement_timeout seconds.

node.session.timeo.replacement_timeout = [value]

Important: Control how long the iSCSI layer should wait for a timed-out path/session to reestablish itself before failing any commands on it. **The above recommended setting of 120 seconds allows ample time for controller failover**. Default is 120 seconds.

Note: If set to 120 seconds, IO will be queued for 2 minutes before it can resume.

The "***1 queue_if_no_path***" option in /etc/multipath.conf sets iSCSI timers to immediately defer commands to the multipath layer. This setting prevents IO errors from propagating to the application; because of this, you can set replacement_timeout to 60-120 seconds.

Note: Nimble Storage strongly recommends using dm-multipath for all volumes.

- Multipath configurations

  The multipath parameters in the */etc/multipath.conf* file should be set as follows in order to sustain a failover. Nimble recommends the use of aliases for mapped LUNs.

```
defaults {
    user_friendly_names yes
    find_multipaths yes
}
```

```
devices {
  device {
    vendor          "Nimble"
    product         "Server"
    path_grouping_policy  group_by_serial
    path_selector       "round-robin 0"
    features        "1 queue_if_no_path"
    path_checker       tur
    rr_min_io_rq       10
    rr_weight        priorities
    failback        immediate
  }
}
multipaths {
    multipath {
        wwid          20694551e4841f4386c9ce900dcc2bd34
        alias         splunk-index1
    }
}
```

- Disk IO Scheduler

  The IO Scheduler needs to be set at "*noop*"

To set the IO Scheduler for all LUNs online, run the below command. **Note**: the multipath must be setup first before running this command. Any additional LUNs added or server reboots will not automatically change to this parameter. Run the same command again if new LUNs are added or if a server reboots.

[root@mktg04 ~]# *multipath -ll | grep sd | awk -F":" '{print $4}' | awk '{print $2}' | while read LUN; do echo noop > /sys/block/${LUN}/queue/scheduler ; done*

To set this parameter automatically, append the below syntax to */etc/grub.conf* file under the kernel line.

elevator=noop

- CPU Scaling Governor

  CPU Scaling Governor needs to be set at "*performance*"

To set the CPU scaling governor, run the below command.

[root@mktg04 ~]# *for a in $(ls -ld /sys/devices/system/cpu/cpu[0-9]* | awk '{print $NF}') ; do echo performance > $a/cpufreq/scaling_governor ; done*

Note: The setting above does not persist after a reboot; hence, the command needs to be executed when the server comes back online. To avoid running the command after a reboot, place the command in the /etc/rc.local file.

- iSCSI Data Network

  Nimble recommends using 10GbE iSCSI for all databases.

2 separate subnets for redundancy

2 x 10GbE iSCSI NICs

Use jumbo frames (MTU 9000) for iSCSI networks

Example of MTU setting for eth1:

DEVICE=eth1

HWADDR=00:25:B5:00:00:BE

TYPE=Ethernet

UUID=31bf296f-5d6a-4caf-8858-88887e883edc

ONBOOT=yes

NM_CONTROLLED=no

BOOTPROTO=static

IPADDR=172.18.127.134

NETMASK=255.255.255.0

MTU=9000

To change MTU on an already running interface:

[root@bigdata1 ~]# *ifconfig eth1 mtu 9000*

- /etc/sysctl.conf

net.core.wmem_max = 16780000

net.core.rmem_max = 16780000

net.ipv4.tcp_rmem = 10240 87380 16780000

net.ipv4.tcp_wmem = 10240 87380 16780000

Run sysctl –p command after editing the /etc/sysctl.conf file.

- max_sectors_kb

  Change max_sectors_kb on all volumes to 1024 (default 512).

To change max_sectors_kb to 1024 for a single volume:

[root@bigdata1 ~]# *echo 1024 > /sys/block/sd?/queue/max_sectors_kb*

To change all volumes:

```
multipath -ll | grep sd | awk -F":" '{print $4}' | awk '{print $2}' | while read LUN
do
  echo 1024 > /sys/block/${LUN}/queue/max_sectors_kb
done
```

Note: To make this change persistent after reboot, add the commands in /etc/rc.local file.

- VM dirty writeback and expire

  Change VM dirty writeback and expire to 100 (default 500 and 3000, respectively)

To change VM dirty writeback and expire:

[root@bigdata1 ~]# *echo 100 > /proc/sys/vm/dirty_writeback_centisecs*
[root@bigdata1 ~]# *echo 100 > /proc/sys/vm/dirty_expire_centisecs*

Note: To make this change persistent after reboot, add the commands in /etc/rc.local file.

Recommended Nimble Volumes for Splunk Indexer

Table 1:

Nimble Volume Role	Recommended Number of Volumes per Indexer	Recommended Number of Indexer CPU Cores per Array	Nimble Storage Caching Policy	Volume Block Size (Nimble Storage)
EXT4 - Splunk raw data and Index files	4 – Indexer with 8 cores or less 6 – Indexer with 12 cores 8 – Indexer with more than 16 cores	96 to 128, depending on workload	Yes - Normal	4KB

EXT4 File System

When creating an EXT file system on a logical volume, the ***stride*** and ***stripe-width*** options must be used. These two parameters minimize the IO unalignment on the Nimble array.

For example:

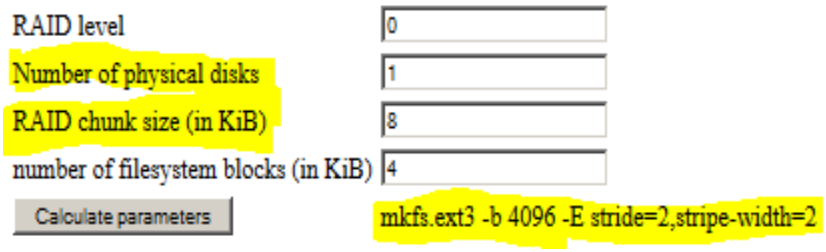
stride=2,stripe-width=16 (for Nimble performance policy 8KB block size with 8 volumes)

stride=4,stripe-width=32 (for Nimble performance policy 16KB block size with 8 volumes)

stride=8,stripe-width=64 (for Nimble performance policy 32KB block size with 8 volumes)

Note: The stripe-width value depends on the number of volumes and the stride size. The calculator can be found here: http://busybox.net/~aldot/mkfs_stride.html.

For example, if there is one Nimble volume with 8KB block size performance policy, then it should look like this:

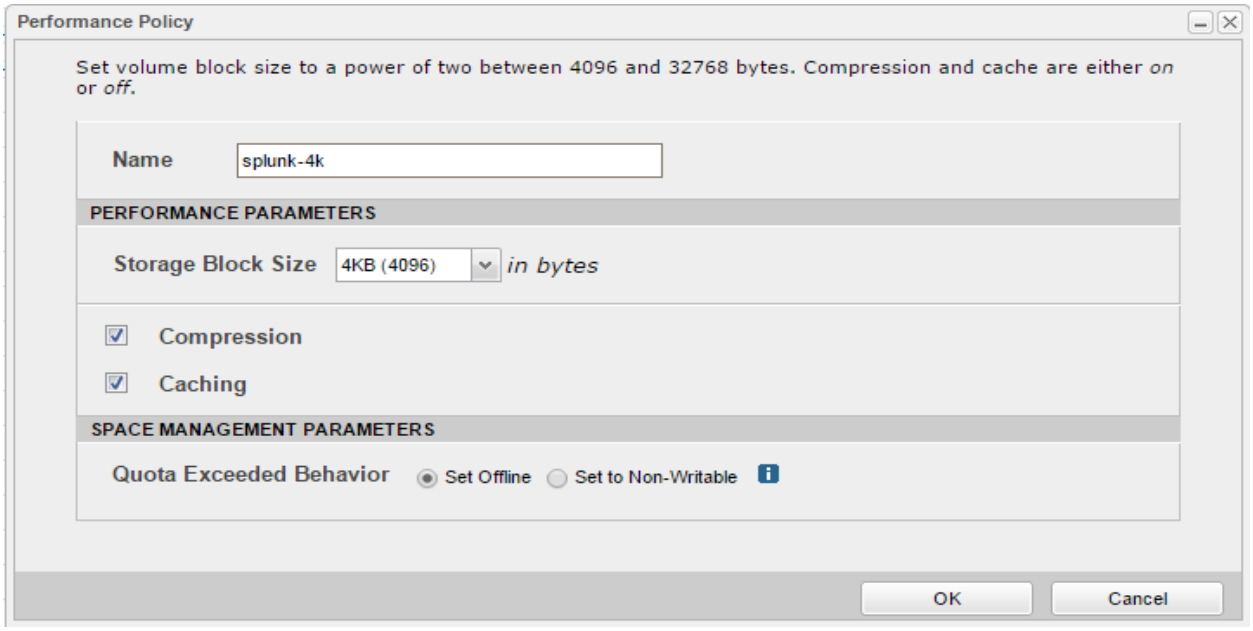


Nimble Recommended Disk Layout for Splunk Indexer

- Number of volumes per Indexer (refer to Table 1 above)
- Use whole disk partition
- Create 1 LVM volume group for all indexes

Creating Nimble Performance Policies

On the Nimble Management GUI, click on “Manage/Performance Policies” and click on the “New Performance Policy” button. Enter the appropriate settings then click “OK”.



Creating the LVM Volume Group

Example Setup with 4 Volumes:

Create Volume Group

[root@mktg04 ~]# *vgcreate vg01 /dev/mapper/<vol[1-4]>*

Create Logical Volume

[root@mktg04 ~]# *lvcreate -l <# of extents> -i 4 -I 4096 -n vol1 vg01*

Create EXT4 file system

[root@mktg04 ~]# *mkfs.ext4 /dev/vg01vol1 -b 4096 -E stride=8,stripe-width=32*

Mount options in /etc/fstab file

/dev/vg01/vol1 /$SPLUNK_DB/<index> ext4 _netdev,noatime,nodiratime,discard,barrier=0 0 0

Nimble Validation and Benchmark

Hardware

- 6 x Cisco UCS C200M3 as Indexers
- 16 CPU cores per indexer
- 12GB RAM per indexer
- 4 Nimble volumes per indexer
- 1 Nimble Storage
- 10GbE iSCSI infrastructure

Benchmark

- SplunkIT test tool
- 50GB data set per indexer

Benchmark Results

SplunkIT Tool								
Rack Server	CPU	Memory (GB)	# Volumes	Index Size (GB)	Avg KBps	Avg EPS	Avg TFE	Avg TTS
1	16	12	4	50	21,323.56	70,521.30	0.38	14.05
2	16	12	4	50	21,060.95	70,449.66	0.35	14.10
3	16	12	4	50	21,856.51	72,268.91	0.35	14.15
4	16	12	4	50	21,323.13	70,472.11	0.36	14.19
5	16	12	4	50	21,589.06	71,323.79	0.37	14.15
6	16	12	4	50	21,861.08	72,299.68	0.37	13.84
Aggregate Results								
Sum					129,014.29	427,335.45	N/A	N/A
Average/Server					21,502.38	71,222.58	0.36	14.08

Result Interpretation

- **Throughput (KBps)** – The amount of input data read per second
- **Events Per Second (EPS)** – The amount of events indexed per second
- **Time to First Event (TFE)** – The time taken to return first event from search (in seconds)
- **Time To Search (TTS)** – The time taken to return all events from search (in seconds)

Nimble Storage, Inc.

211 River Oaks Parkway, San Jose, CA 95134

Tel: 877-364-6253 | www.nimblestorage.com | info@nimblestorage.com