SPECsfs2014 within Emerald

Don Capps
What is *not* in this presentation

- How to setup AC power
- How to setup the Power meter
- How to install and setup Windows or Unix systems.
- How to setup and properly configure DNS (Windows or Unix)
- How to setup the switches and NICs in your lab.
- How to install, and properly configure a Windows Active directory domain.
- How to properly setup a lab to handle high stress workloads.
Introduction to SPEC SFS2014

- Video: Application level benchmarking with SPECsfs2014
  https://youtu.be/4wfeM1q0zHA
- Video: Using SPEC SFS with the SNIA Emerald Program for EPA Energy Star Data Center Storage Program
  https://youtu.be/7gDgcDYatvM
Setting up the environment (toc)

- Block diagram of the benchmark architecture
- What is sFlow®
- Setting up sFlow on switches
- Setting up sFlow collection
- Installing client operating systems
- Configuring client nodes
  - Installing Python (if needed)
  - Installing SPECsfs2014 software
  - Disable firewalls (Iptables and selinux)
  - Setting up ssh with keys, not passwords.
  - Client OS tunes
  - Two NICs is good idea. One for client access/control and the other for data movement.
  - How much RAM for the workloads.
Setting up the environment (toc)

- Ensure that DNS is properly configured
- Setting up storage server
  - Tuning storage server
  - Balancing load across spindles and NICs
  - Allocating enough space for the workloads.
- Configuring SPECsfs2014
  - CLIENT_MOUNTPOINTS= USER=
  - BENCHMARK= PASSWORD=
  - LOAD= INCR_LOAD=
  - NUM_RUNS = WARMUP_TIME=
Setting up the environment (toc)

- Running SPECsfs2014
  - Starting it running
  - Monitoring its progress
  - Examining the results

- Finding the optimal peak performance for each workload
Setting up the environment (toc)

- Merging the sFlow data with the power meter results
- Tag2014
- Filling out the TDR
If things go wrong (toc)

- Examining the SFS2014 logs
- Client process logs
- Summary logs
- Client performance logs
- Console output
Diagram of the benchmark architecture

- **Environmental Meter** (Clock synchronized with I/O load driver)
- **Power Meter** (Clock synchronized with I/O load driver)
- **I/O Load Driver System(s)**
- **I/O Throughput Data collector** (sFlow collection from IP Switch) NAS Only
- **PUT**
What is sFlow

- sFlow is the leading, multi-vendor, standard for monitoring high-speed switched and routed networks.
- sFlow technology is built into network equipment and gives complete visibility into network activity, enabling effective monitoring, management and control of network resources.
- sFlow is available from most leading network equipment vendors, including: Alcatel-Lucent, Allied Telesis, Arista, Brocade, Cisco, Dell, D-Link, Enterasys, Extreme, Fortinet, Hewlett-Packard, Hitachi, Huawei, IBM, Juniper Networks, NEC and ZTE. For a complete list of products supporting sFlow, see sFlow.org.
Setting up sFlow on the switch
Setting up sFlow on the switch

Quanta LB6M 10Gbe switch:  Login to prompt

```
[cappas@centos7 ~]$ ssh -l admin LB6M
admin@lb6m's password:
(LB6M) >
```
Setting up sFlow on the switch

Elevate privilege level

```
$ ssh -l admin LB6M
admin@lb6m's password:
(LB6M) >enable
Password:
(LB6M) #
```
Setting up Sflow on the switch

Switch to configuration menu
Setting up sFlow on the switch

Set the IP address for the collector

```
(LB6M) #
(LB6M) #
(LB6M) #
(LB6M) #
(LB6M) #
(LB6M) #
(LB6M) #configure

(LB6M) (Config)#sflow receiver 1 ?

ip       Configure IP Address of the Receiver.
maxdatagram Configure Maximum Datagram Size.
owner    Configure Owner String of the Receiver.
port     Configure Receiver Port.

(LB6M) (Config)#sflow receiver 1 ip ?

<ip>   Enter Ipv4/ipv6 Address.

(LB6M) (Config)#sflow receiver 1 ip 10.0.0.231 ?

<cr>   Press enter to execute the command.
```
Setting up sFlow on the switch

Select ports to monitor on the switch

```
(LB6M) (Config)#
(LB6M) (Config)#
(LB6M) (Config)#
(LB6M) (Config)#
(LB6M) (Config)#
(LB6M) (Config)#
(LB6M) (Config)#
(LB6M) (Config)#
(LB6M) (Config)#
(LB6M) (Config)#
(LB6M) (Config)#
(LB6M) (Config)#
(LB6M) (Config)#
(LB6M) (Config)#
(LB6M) (Config)#
(LB6M) (Config)#
(LB6M) (Config)#
(LB6M) (Config)#
(LB6M) (Config)#
(LB6M) (Config)#
(LB6M) (Config)#
(LB6M) (Config)#
(LB6M) (Config)#
(LB6M) (Config)#
(LB6M) (Config)#
(LB6M) (Config)#
(LB6M) (Config)#
(LB6M) (Config)#
```

(Interface 0/1-0/28)
Setting up sFlow on the switch

Set the polling index to use

```
voice
Configure Voice VLAN Parameters.

(LB6M) (Interface 0/1-0/28)#port ?

lacpmode
Enable/Disable the port's LACP mode.

lacptime
Configure the port's LACP timeout.

(LB6M) (Interface 0/1-0/28)#sflow ?

poller
Configure poller options on interface.

sampler
Configure sampler options on interface.

(LB6M) (Interface 0/1-0/28)#sflow poller <index>

interval
Enter Receiver Index <1-8>.

Configure poll interval.

(LB6M) (Interface 0/1-0/28)#sflow poller 1

<cr>
Press enter to execute the command.

(LB6M) (Interface 0/1-0/28)#sflow poller 1
```

---

SNIA Emerald™ Training ~ February–March 2018

www.sniaemerald.com
Setting up sFlow on the switch

Set the polling rate to 5 seconds

```
(LB6M) (Interface 0/1-0/28)#
(LB6M) (Interface 0/1-0/28)#
(LB6M) (Interface 0/1-0/28)#
(LB6M) (Interface 0/1-0/28)#
(LB6M) (Interface 0/1-0/28)#
(LB6M) (Interface 0/1-0/28)#
(LB6M) (Interface 0/1-0/28)#
(LB6M) (Interface 0/1-0/28)#
(LB6M) (Interface 0/1-0/28)#
(LB6M) (Interface 0/1-0/28)#
(LB6M) (Interface 0/1-0/28)#
(LB6M) (Interface 0/1-0/28)#
(LB6M) (Interface 0/1-0/28)#
(LB6M) (Interface 0/1-0/28)#
(LB6M) (Interface 0/1-0/28)#
(LB6M) (Interface 0/1-0/28)#
(LB6M) (Interface 0/1-0/28)#
(LB6M) (Interface 0/1-0/28)#
(LB6M) (Interface 0/1-0/28)#
(LB6M) (Interface 0/1-0/28)#
(LB6M) (Interface 0/1-0/28)#
(LB6M) (Interface 0/1-0/28)#
(LB6M) (Interface 0/1-0/28)#
(LB6M) (Interface 0/1-0/28)#
(LB6M) (Interface 0/1-0/28)#
(LB6M) (Interface 0/1-0/28)#
```

#sflow poller interval 5 ?
Setting up sFlow collection tools
Setting up sFlow collection

- Download sflowtool from inmon.com
- Download sflowTrend from inmon.com
Download sflowtool

sflowtool is available as source code. A pre-compiled version of sflowtool is available for Windows.

sflowtool scripts

Scripts using sflowtool to analyze traffic. These scripts are easily modified to perform customized traffic studies. Included are:

- ipTopTalkers: Minute by minute top talking IP sources.
- sflowRRDLoad/sflowRRDChart.cgi: Use rrdf tool to log and chart sFlow counters.
Running sflowtool

- sflowtool -4 -L > c:\tmp\sflowdata.txt
## Running sflowtool

Data that is in the sflowtool file

<table>
<thead>
<tr>
<th>SwitchIP</th>
<th>Port</th>
<th>InBytes</th>
<th>OutBytes</th>
</tr>
</thead>
<tbody>
<tr>
<td>CNTR,10.0.0.248,2018-03-04 19:10:31</td>
<td>24,6</td>
<td>4294967295</td>
<td>1,3,153494289,570659,2838,1467,0,0,0,1122830633,919463,169,228,0,0,1</td>
</tr>
<tr>
<td>CNTR,10.0.0.248,2018-03-04 19:10:36</td>
<td>24,6</td>
<td>4294967295</td>
<td>1,3,164427808,650122,2860,1472,0,0,0,1296109924,1056224,170,228,0,0,1</td>
</tr>
<tr>
<td>CNTR,10.0.0.248,2018-03-04 19:10:41</td>
<td>24,6</td>
<td>4294967295</td>
<td>1,3,172979004,712033,2893,1478,0,0,0,1430569898,1162449,170,228,0,0,1</td>
</tr>
<tr>
<td>CNTR,10.0.0.248,2018-03-04 19:10:46</td>
<td>24,6</td>
<td>4294967295</td>
<td>1,3,184035482,792030,2898,1481,0,0,0,1604336944,1299946,170,228,0,0,1</td>
</tr>
<tr>
<td>CNTR,10.0.0.248,2018-03-04 19:10:51</td>
<td>24,6</td>
<td>4294967295</td>
<td>1,3,192577062,854056,2900,1484,0,0,0,1739303201,1406648,171,228,0,0,1</td>
</tr>
<tr>
<td>CNTR,10.0.0.248,2018-03-04 19:10:56</td>
<td>24,6</td>
<td>4294967295</td>
<td>1,3,20344482,933274,2906,1487,0,0,0,1912086457,1543110,171,228,0,0,1</td>
</tr>
<tr>
<td>CNTR,10.0.0.248,2018-03-04 19:11:01</td>
<td>24,6</td>
<td>4294967295</td>
<td>1,3,212235634,997451,2908,1489,0,0,0,2052267079,1653613,171,228,0,0,1</td>
</tr>
<tr>
<td>CNTR,10.0.0.248,2018-03-04 19:11:06</td>
<td>24,6</td>
<td>4294967295</td>
<td>1,3,220564553,1057904,2914,1490,0,0,0,2183741866,1757467,172,228,0,0,1</td>
</tr>
<tr>
<td>CNTR,10.0.0.248,2018-03-04 19:11:12</td>
<td>24,6</td>
<td>4294967295</td>
<td>1,3,231717183,1138865,2942,1493,0,0,0,2359889118,1896604,172,228,0,0,1</td>
</tr>
<tr>
<td>CNTR,10.0.0.248,2018-03-04 19:11:17</td>
<td>24,6</td>
<td>4294967295</td>
<td>1,3,240374506,1201242,2960,1495,0,0,0,2494988054,2003832,172,228,0,0,1</td>
</tr>
<tr>
<td>CNTR,10.0.0.248,2018-03-04 19:11:22</td>
<td>24,6</td>
<td>4294967295</td>
<td>1,3,251079469,1278233,2965,1498,0,0,0,2661588115,2136189,173,228,0,0,1</td>
</tr>
<tr>
<td>CNTR,10.0.0.248,2018-03-04 19:11:27</td>
<td>24,6</td>
<td>4294967295</td>
<td>1,3,259898164,1341837,2968,1499,0,0,0,2799269345,2245425,173,228,0,0,1</td>
</tr>
<tr>
<td>CNTR,10.0.0.248,2018-03-04 19:11:32</td>
<td>24,6</td>
<td>4294967295</td>
<td>1,3,270594864,1419493,2970,1502,0,0,0,2968310559,2379005,173,228,0,0,1</td>
</tr>
</tbody>
</table>
Download sflowtrend (optional)

sFlowTrend™ is a free, graphical network and server monitoring tool. sFlowTrend makes use of the popular sFlow® standard to generate real-time displays of the top users and applications making use of network bandwidth. sFlowTrend also uses the extensions to the sFlow standard for monitoring physical and virtual server performance, to link network, server, and application performance and provide an end-end view of networked system performance. sFlowTrend accepts sFlow data from at most five switches or hosts and stores one hour of data in memory. sFlowTrend-Pro does not limit the number of switches or hosts that can be monitored and stores historical data to disk.

- Quickly understand who is using the network and what they are doing.
- Enforce corporate acceptable network use policies.
- Rapidly identify the cause of any problems or abnormal traffic.
- Monitor critical host performance parameters (e.g., CPU and memory utilization).
- Understand trends in usage and accurately target upgrades.
- Generate management reports on current and historical performance.
Running sflowtrend (It is nice to see the big picture)

The main dashboard view
Running sflowtrend (handy reports)

One of the reports that can be generated:

**Top servers**

This report shows top servers and their associated ports.
Running sflowtrend (handy report)

Port utilization view
Installing the Client OS.
Installing client operating system

Boot install screen
Installing client operating system

Select preferred language
Installing client operating system

Select software: NFS client, Development tools
Installing client operating system

Setup the network interface
Installing client operating system

Select where to install the system

[Image of installation interface]
Installing client operating system

Setup root and user names and passwords
Installing client operating system

Login to start client configuration
Installing client operating system (ssh)

Setup ssh keys
Installing client operating system (ssh)

Test ssh login to self (and all other client nodes after distributing the ssh keys)

```
capps@Centos7m4:~
File Edit View Search Terminal Help
[capps@Centos7m4 .ssh]$ ssh centos7m4
The authenticity of host 'centos7m4 (10.0.0.169)' can't be established.
Are you sure you want to continue connecting (yes/no)? yes
Warning: Permanently added 'centos7m4,10.0.0.169' (ECDSA) to the list of known hosts.
Last login: Wed Feb 28 11:53:51 2018
[capps@Centos7m4 ~]$  
```
Installing required software
Installing Python 2.6 not 3.x (If needed)

Go get Python, if you need it. Not needed in this example
Installing SPECsfs2014
Do this on all client nodes
Installing SPECsfs2014

Mount the DVD, or filesystem that contains the SFS2014 kit

```
[root@Centos7m4 /]# mount 10.0.0.120:/tmp /mnt/distro
[root@Centos7m4 /]#
```
Installing SPECsfs2014

Copy to location where one wishes to run

```
capps@Centos7m4:~
File Edit View Search Terminal Help
[root@Centos7m4 ~]# mount 10.0.0.120:/tmp /mnt/distro
[root@Centos7m4 ~]# exit
logout
[capps@Centos7m4 ~]$ cd
[capps@Centos7m4 ~]$ cp -R /mnt/distro/SPECsfs2014_SP2 .~
```
Installing SPECsfs2014

Validate the SPECsfs2014_SP2 directory exists

![Image showing ls command output]

```
capps@Centos7m4:~ $ ls
Desktop    Downloads  Pictures  SPECsfs2014_SP2  Videos
Documents  Music      Public    Templates
```
Installing SPECsfs2014

Validate the contents of the kit look like this

```
[capps@Centos7m4 ~]$ ls
Desktop  Downloads  Pictures  SPECsfs2014_SP2  Videos
Documents  Music  Public  Templates
[capps@Centos7m4 ~]$ cd SPECsfs2014_SP2
[capps@Centos7m4 SPECsfs2014_SP2]$ ls
benchmarks.xml  Map_share_script  sfs_ext_mon
binaries        mbuild            sfs_ext_mon.cmd
bin.sh          netmist           SfsManager
copyright.txt   NOTICE            sfs.rc
docs            psdm               SPEC_LICENSE.txt
Example_run_script.sh  rechangelog.txt  SpecReport
future_direction  README.md       submission_template.xml
Makefile         redistributeable_sources  win32lib
makefile.in      sfs2014result.css
```
Configuring the client
Client configuration (ssh keys)

The script from the SFS2014 User’s guide that sets up the ssh keys

See SFS2014 User’s guide appendix “B”

Appendix B – Setting up password-less SSH

Here is a sample script that can be used to set up password-less SSH on Linux clients.

```bash
# Define the hosts to be involved in the trust here
# DO NOT include the host you are running, it is added by default

hosts="s2 s3 s4 s5 s6"

echo ""
echo ""
echo "This script will generate SSH keys for the specified machines,"
echo " and set up password-less authentication between them."
echo " You will be prompted for passwords several times during this process."
echo ""
```

SNIA Emerald™ Training ~ February—March 2018  www.sniaemerald.com  46
Client configuration (ssh keys)

Cont appendix “B”

# Get current user
user=`who -m | awk {'print $1'}`
echo "Trust will be configured for user $user"
echo ""
echo "If this is not correct, stop and login as the appropriate user"
echo -n "(RETURN to continue, CTRL-C to exit) "

read continue

# Configure keys on current host
cd $HOME
ssh-keygen -t rsa
cat .ssh/id_rsa.pub >> .ssh/authorized_keys
chmod 700 .ssh
chmod 600 .ssh/*
Client configuration (ssh keys)

Cont appendix “B”

```bash
for host in $hosts
do
    ssh $user@$host 'ssh-keygen -t rsa'
    ssh $user@$host 'cat .ssh/id_rsa.pub' | cat - >> ~/.ssh/authorized_keys
done

for host in $hosts
do
    scp .ssh/authorized_keys $host:/.ssh
    ssh $user@$host 'chmod 700 .ssh ; chmod 600 .ssh/*'
done

exit
```
Configuring client operating system

Apply the recommended client tunes from the SFS2014 User’s guide

- **Apply the tunes:**
  
  ```
  if your NIC type == 10 GigE
      echo "300000" > /proc/sys/net/core/netdev_max_backlog
  fi
  
echo "131071" > /proc/sys/net/core/rmem_default
  echo "131071" > /proc/sys/net/core/rmem_max
  echo "131071" > /proc/sys/net/core/wmem_default
  echo "131071" > /proc/sys/net/core/wmem_max
  echo "4096 87380 8388608" > /proc/sys/net/ipv4/tcp_rmem
  echo "4096 87380 8388608" > /proc/sys/net/ipv4/tcp_wmem
  echo "128" > /proc/sys/sunrpc/tcp_slot_table_entries
  echo "65536" > /proc/sys/net/core/somaxconn
  echo "5" > /proc/sys/net/ipv4/tcp_fin_timeout
  ```
Listing firewalls

Check to see if you have any firewalls running

```bash
[capps@Centos7m4 SPECsfs2014_SP2]$ systemctl list-unit-files | fgrep firewalld
firewalld.service enabled
```

Disabling firewalls

Disable any/all firewalls on every client

```
root@Centos7m4:~
File   Edit  View  Search  Terminal  Help

[root@Centos7m4 ~]# systemctl disable firewalld.service
rm /etc/systemd/system/dbus-org.fedoraproject.FirewallD1.service
rm /etc/systemd/system/basic.target.wants/firewalld.service
[root@Centos7m4 ~]#
```
Checking the amount of RAM

Check to make sure the clients have enough RAM for the workloads

» Client memory requirements per business metric:

DATABASE $=\ 55$ Mbytes per LOAD increment  
SWBUILD $=\ 400$ Mbytes per LOAD increment  
VDA $=\ 10$ Mbytes per LOAD increment  
VDI $=\ 8$ Mbytes per LOAD increment
Checking DNS works
Ensure DNS works !!!!!

This is the most common problem that folks have. You MUST have a working DNS

Tests:

- Ssh from the prime to every client node. Ensure the password-less ssh works.
- nslookup every client and ensure that its name gave you an IP that will nslookup back to that name.
Setting up the Storage server
Setting up the storage server

Ensure that the storage server has sufficient space, and that the accesses are balanced across all of the filesystems and NICs

- Setting up storage server
  - Tuning storage server (Best practices from the vendor)
  - Balancing load across spindles and NICs (Use all data paths) By using the CLIENT_MOUNTPOINTS and the mounts done by each client.
  - Configure enough space for the workloads. (Reminder, it’s a bummer to run out)
Configuring SPECsfs2014
Configuring SPECsfs2014

• Make 4 copies of the sfs_rc/configuration file, from here:
and name them:
  • sfs_rc_database
  • sfs_rc_vdi
  • sfs_rc_vda
  • sfs_rc_swbuild
Configuring SPECsfs2014

Sfs_rc variables that you need to set in each sfs_rc file

- CLIENT_MOUNTPOINTS= clientname:/mountpoint  clientname:/mountpoint
  or clientname:\\servername\sharename for Windows.
- USER=  The valid user name to use for login and running the tests.
  Unix: Username
  Windows: Domain\Username
- BENCHMARK= SWBUILD | VDA | VDI | DATABASE ( have this match the sfs_rc_name )
- PASSWORD= (only needed for Windows)
- LOAD=  Load value. from1 to big. Where big does not produce INVALID results.
- INCR_LOAD= ( set this the same as you did for LOAD )
- NUM_RUNS = 10 ( Is the minimum set of load points)
- WARMUP_TIME= ( Optional. It will default to 300 seconds )
Getting ready to run SFS2014

Ensure that all of the clients have the filesystems mounted before starting

- Mount the test area from the storage server on each client node.
- You must mount the storage on every client node before you start the test. I highly recommend you put the mount in /etc/fstab so you won’t have to type this over if you reboot.
Getting ready to run SPECsfs2014 (mount)

Example of mounting the filesystem

```
[root@Centos7m5 ~]# cd /mnt
[root@Centos7m5 mnt]# mkdir test
[root@Centos7m5 mnt]# mount 10.0.0.216:/Public /mnt/test
[root@Centos7m5 mnt]# ...
```
Free space on the storage server

You need this much space * LOAD value in the sfs_rc file

You need this much storage space:

- DATABASE = 24 Gigabytes per DATABASE
- SWBUILD = 5 Gigabytes per BUILD
- VDI = 12 Gigabytes per DESKTOP
- VDA = 24 Gigabytes per STREAM
Ready to run SPECsfs2014 (check free space)

Example of checking free space before starting test

```
[root@Centos7m5 test]# cd /mnt/test
[root@Centos7m5 test]# df .
Filesystem 1k-blocks Used Available Use% Mounted on
10.0.0.216:/Public/test 11578263616 9614677408 2563586208 78% /mnt/test
[root@Centos7m5 test]#
```
Start the sFlow, power, env, collection and GO.
Start collection tools

- Start the `sflowtool` collection now. Save its output file in a safe location. Example: `sflowtool –L -4 > c:\temp\sflowdata.txt`

- Start the Power meter. Save its output in a safe place.

- Start the environmental monitor and save its output in a safe location.
Example script to run the 4 mandatory workloads. You need to edit this to tailor it for your environment.

```
# Example script to run the 4 SPECsfs2014 workloads required by Emerald.
# Run the SWBUILD workload
# Edit this to tailor it for your configuration
# Cleanup space for use by the next workload.
```

```python
python SfsManager -r sfs_rc_swbuild -s swbuild
```
Running SPECsfs2014 (cont)

# Example:
# cd /mnt/workdir
# for i in *
# do
#   rm -rf $i &
# done
# wait
#
# Run the VDA workload
python SfsManager -r sfs_rc_vda -s vda
#
# Edit this to tailor it for your configuration
# Cleanup space for use by the next workload.
#
Running SPECsfs2014 (cont)

# Example:
# cd /mnt/workdir
# for i in *
# do
# rm -rf $i &
# done
# wait
#

#########################################################################
# Run the VDI workload
#########################################################################
python SfsManager -r sfs_rc_vdi -s vdi
#
# Edit this to tailor it for your configuration
# Cleanup space for use by the next workload.
#
Running SPECsfs2014 (cont)

# Example:
# cd /mnt/workdir
# for i in *
# do
#  rm -rf $i &
# done
# wait

# Run the DATABASE workload

python SfsManager -r sfs_rc_database -s database

# Cleanup space for use in later runs
#
# Example:
# cd /mnt/workdir
Running SPECsfs2014 (cont)

```bash
# for i in *
# do
#  rm -rf $i &
# done
# wait
```

`# for i in *` for i in *
do
do rm -rf $i &
done
done wait
Running SPECsfs2014 (console output)
Running SPECsfs2014 (console output)

```plaintext
Each process file size = 16 kbytes
Client data set size = 4296 MiBytes
Total starting data set size = 8593 MiBytes
Total initial file space = 8593 MiBytes
Total max file space = 9375 MiBytes

Starting tests: Wed Feb 28 12:53:19 2018

Launching 10 processes.
Starting test client: 0 Host: Centos7m4.home.org Workload: SWBUILD Location: /mnt/test
Starting test client: 1 Host: Centos7m4.home.org Workload: SWBUILD Location: /mnt/test
Starting test client: 2 Host: Centos7m4.home.org Workload: SWBUILD Location: /mnt/test
Starting test client: 3 Host: Centos7m4.home.org Workload: SWBUILD Location: /mnt/test
Starting test client: 4 Host: Centos7m4.home.org Workload: SWBUILD Location: /mnt/test
Starting test client: 5 Host: Centos7m5.home.org Workload: SWBUILD Location: /mnt/test
Starting test client: 6 Host: Centos7m5.home.org Workload: SWBUILD Location: /mnt/test
Starting test client: 7 Host: Centos7m5.home.org Workload: SWBUILD Location: /mnt/test
Starting test client: 8 Host: Centos7m5.home.org Workload: SWBUILD Location: /mnt/test
Starting test client: 9 Host: Centos7m5.home.org Workload: SWBUILD Location: /mnt/test

Wed Feb 28 12:53:28 2018 Primo's GO 'start comm' average message latency 4 ms
Waiting to finish initialization. Wed Feb 28 12:53:28 2018
Wed Feb 28 12:53:30 2018 Starting INIT phase
Wed Feb 28 12:53:30 2018 Init Heartbeat __/\_/\__ from client 5
```
Running SPECsfs2014 (console output)

The heartbeat messages tell you that it’s running. These arrive every minute.

```
capps@Centos7m4:~/SPECsfs2014_SP2

Testing begins: Wed Feb 28 12:57:08 2018
Wed Feb 28 12:57:08 2018 Prime's GO message latency plus delay for potential network jitter 9 ms
Wed Feb 28 12:57:09 2018 Actual average warmup GO latency: 16 ms
Waiting for tests to finish. Wed Feb 28 12:57:09 2018
Wed Feb 28 12:57:14 2018 Starting WARM phase
Wed Feb 28 12:57:42 2018 Warm-up 10 percent complete from client 9
Wed Feb 28 12:58:12 2018 Warm-up 20 percent complete from client 9
Wed Feb 28 12:58:42 2018 Warm-up 30 percent complete from client 9
Wed Feb 28 12:58:44 2018 Warm Heartbeat Client 9: 168.078 Ops/sec
Wed Feb 28 12:59:12 2018 Warm-up 40 percent complete from client 9
Wed Feb 28 12:59:42 2018 Warm-up 50 percent complete from client 9
Wed Feb 28 12:59:44 2018 Warm Heartbeat Client 9: 123.175 Ops/sec
Wed Feb 28 13:00:12 2018 Warm-up 60 percent complete from client 9
Wed Feb 28 13:00:42 2018 Warm-up 70 percent complete from client 9
Wed Feb 28 13:00:44 2018 Warm Heartbeat Client 9: 39.634 Ops/sec
Wed Feb 28 13:01:18 2018 Warm-up 80 percent complete from client 9
Wed Feb 28 13:01:42 2018 Warm-up 90 percent complete from client 9
Wed Feb 28 13:01:44 2018 Warm Heartbeat Client 9: 93.589 Ops/sec
Wed Feb 28 13:02:12 2018 Warm-up 100 percent complete from client 9
Wed Feb 28 13:02:15 2018 Starting RUN phase
Wed Feb 28 13:02:44 2018 Run Heartbeat Client 9: 86.215 Ops/sec
Wed Feb 28 13:02:44 2018 Run 10 percent complete from client 9
```
Running SPECsfs2014 (console output)

| Band 1 | 20us:56349 | 40us:19799 | 60us:1765 | 80us:431 | 100us:62 |
| Band 2 | 200us:10 | 400us:88684 | 600us:43643 | 800us:25863 | 1ms:12711 |
| Band 3 | 2ms:17970 | 4ms:19643 | 6ms:2595 | 8ms:1160 | 10ms:712 |
| Band 4 | 12ms:749 | 14ms:313 | 16ms:199 | 18ms:161 | 20ms:141 |
| Band 5 | 40ms:638 | 60ms:410 | 80ms:338 | 100ms:0 | 100ms:0 |
| Band 6 | 200ms:318 | 400ms:164 | 600ms:129 | 800ms:78 | 1s:39 |

Overall average latency: 2.021 Milli-seconds
Overall SPEC SFS2014 SP2: 1060.032 Ops/sec
Overall Read throughput: ~ 7119.807 Kbytes/sec
Overall Write throughput: ~ 1829.166 Kbytes/sec
Overall throughput: ~ 8948.973 Kbytes/sec
Public Finger Print: 24267
Running SPECsfs2014 (console output)

Each load point will create one of these. You need 10 load points per workload type

```
Wed Feb 28 13:07:14 2018 Prime receiving results from child 9
Shutting down clients, and communications layer...

<table>
<thead>
<tr>
<th>Overall average latency</th>
<th>2.021 Milli-seconds</th>
</tr>
</thead>
<tbody>
<tr>
<td>Overall SPEC SFS2014 SP2</td>
<td>1880.832 Ops/sec</td>
</tr>
<tr>
<td>Overall Read throughput</td>
<td>~ 7119.897 Kbytes/sec</td>
</tr>
<tr>
<td>Overall Write throughput</td>
<td>~ 1829.166 Kbytes/sec</td>
</tr>
<tr>
<td>Overall throughput</td>
<td>~ 8940.973 Kbytes/sec</td>
</tr>
<tr>
<td>Public Finger Print</td>
<td>24267</td>
</tr>
</tbody>
</table>

Band 1: 26us:56349 48us:19799 60us:1765 80us:431 100us:82
Band 2: 266us:10 408us:88564 604us:43643 899us:25863 1ms:12711
Band 3: 2ms:17970 4ms:19643 6ms:2595 8ms:1160 10ms:712
Band 4: 12ms:749 14ms:313 16ms:199 18ms:161 20ms:141
Band 5: 46ms:636 60ms:418 80ms:338 100ms:0
Band 6: 260ms:318 488ms:164 688ms:129 800ms:78 1s:39
Band 7: 2s:29 4s:4 6s:0 8s:9 10s:0
Band 8: 26s:6 40s:0 60s:0 80s:0 120s:0
Band 9: 1264s:6
```

netmist completed successfully, summarizing.
[capps@Centos7m4 SPECsfs2014_SP2]$
Steps in the test procedure

- **Per-Workload Calibration**
  - Find maximum valid load for each workload on PUT (SPEC SFS 2014 metrics)

- **Measurement**
  - Run each workload as ten evenly-spaced load points up to the maximum valid load point
  - Collect environmental, power, and sFlow data

- **Data Reduction**
  - Derive efficiency metrics for each workload using data reduction methods

- **SNIA Emerald Metrics**
  - Find “sweet-spot” for all four workloads
Collecting the results

- The SFS2014 results are in $TOP/results on the Prime.
  sfssumm.[swbuild,vda,vdi,database].txt
  sfslog.[swbuild,vda,vdi,database].txt
  Save these in some safe place.
- Save the sflowtool output file in this same location.
- Save the power monitor results in this same location.
Example results
Example results (sFlow and Power)

- [http://www.demartek.com/private/SNIA_Emerald/Test4_files.zip](http://www.demartek.com/private/SNIA_Emerald/Test4_files.zip)
- sFlow raw data:
  ```
  CNTR,10.0.0.248, 2018-03-12 13:14:26
  IP addr   Timestamp
  7,6,4294967295,1,3,66153,389,247,1,0,0,0, 598533882572,1274548005,30285,11610,44,0,1
  Bytes_in   Bytes_out
  ```

- Power meter raw data

<table>
<thead>
<tr>
<th>Store No.</th>
<th>Date</th>
<th>Time</th>
<th>Millisecond</th>
<th>WT1:U-1</th>
<th>WT1:I-1</th>
<th>WT1:P-1</th>
<th>WT1:S-1</th>
<th>WT1:Q-1</th>
<th>WT1:PF-1</th>
<th>WT1:Phi-U</th>
<th>WT1:Freq-U</th>
<th>WT1:Freq-I</th>
<th>WT1:Uthd</th>
<th>WT1:ithd-U</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>2/27/2018</td>
<td>16:18:33</td>
<td>353</td>
<td>230.29</td>
<td>2.4751</td>
<td>5.35E+02</td>
<td>5.70E+02</td>
<td>1.97E+02</td>
<td>0.9387</td>
<td>-20.17</td>
<td>60.002</td>
<td>60.003</td>
<td>230.29</td>
<td>0.439</td>
</tr>
<tr>
<td>2</td>
<td>2/27/2018</td>
<td>16:18:38</td>
<td>353</td>
<td>230.29</td>
<td>2.4622</td>
<td>5.32E+02</td>
<td>5.67E+02</td>
<td>1.96E+02</td>
<td>0.9383</td>
<td>-20.23</td>
<td>60.002</td>
<td>60.002</td>
<td>230.29</td>
<td>0.422</td>
</tr>
<tr>
<td>3</td>
<td>2/27/2018</td>
<td>16:18:43</td>
<td>353</td>
<td>230.29</td>
<td>2.4735</td>
<td>5.35E+02</td>
<td>5.70E+02</td>
<td>1.97E+02</td>
<td>0.9385</td>
<td>-20.19</td>
<td>60.002</td>
<td>60.002</td>
<td>230.29</td>
<td>0.431</td>
</tr>
<tr>
<td>4</td>
<td>2/27/2018</td>
<td>16:18:48</td>
<td>354</td>
<td>230.3</td>
<td>2.4497</td>
<td>5.29E+02</td>
<td>5.64E+02</td>
<td>1.96E+02</td>
<td>0.9375</td>
<td>-20.36</td>
<td>60.003</td>
<td>60.002</td>
<td>230.3</td>
<td>0.442</td>
</tr>
</tbody>
</table>
### Example results (SPEC sfs2014 results)

#### Example of the sfssum.vda.txt contents

*Used for calibration. Not included in Emerald results*

<table>
<thead>
<tr>
<th>Business</th>
<th>Requested</th>
<th>Achieved</th>
<th>Avg Lat</th>
<th>Total Read</th>
<th>Write Run</th>
<th># Cl</th>
<th>Avg File</th>
<th>Cl Data</th>
<th>Start Data</th>
<th>Init File</th>
<th>Max File</th>
<th>Workload</th>
<th>Valid</th>
</tr>
</thead>
<tbody>
<tr>
<td>Metric</td>
<td>Op Rate</td>
<td>Op Rate</td>
<td>(ms)</td>
<td>KBps</td>
<td>KBps</td>
<td>KBps</td>
<td>Sec Cl</td>
<td>Proc Size KB</td>
<td>Set MiB</td>
<td>Set MiB</td>
<td>Set MiB</td>
<td>Space MiB</td>
<td>Name</td>
</tr>
<tr>
<td>9</td>
<td>90.00</td>
<td>90.071</td>
<td>0.798</td>
<td>41599.242</td>
<td>3647.205</td>
<td>37952.037</td>
<td>300</td>
<td>1 18 1048576</td>
<td>202752</td>
<td>202752</td>
<td>202752</td>
<td>221184</td>
<td>VDA</td>
</tr>
<tr>
<td>18</td>
<td>180.00</td>
<td>180.116</td>
<td>2.875</td>
<td>83026.161</td>
<td>6817.453</td>
<td>76208.708</td>
<td>300</td>
<td>1 36 1048576</td>
<td>405504</td>
<td>405504</td>
<td>405504</td>
<td>442368</td>
<td>VDA</td>
</tr>
<tr>
<td>27</td>
<td>270.00</td>
<td>270.191</td>
<td>4.763</td>
<td>124814.636</td>
<td>10843.302</td>
<td>113971.334</td>
<td>300</td>
<td>1 54 1048576</td>
<td>608256</td>
<td>608256</td>
<td>608256</td>
<td>663552</td>
<td>VDA</td>
</tr>
<tr>
<td>36</td>
<td>360.00</td>
<td>360.239</td>
<td>4.554</td>
<td>164445.594</td>
<td>13859.482</td>
<td>150586.112</td>
<td>300</td>
<td>1 72 1048576</td>
<td>811008</td>
<td>811008</td>
<td>811008</td>
<td>884736</td>
<td>VDA</td>
</tr>
<tr>
<td>45</td>
<td>450.00</td>
<td>450.304</td>
<td>7.286</td>
<td>208721.185</td>
<td>17872.621</td>
<td>190848.564</td>
<td>300</td>
<td>1 90 1048576</td>
<td>1013760</td>
<td>1013760</td>
<td>1013760</td>
<td>1105920</td>
<td>VDA</td>
</tr>
<tr>
<td>54</td>
<td>540.00</td>
<td>540.359</td>
<td>8.675</td>
<td>249397.387</td>
<td>21335.173</td>
<td>228062.214</td>
<td>300</td>
<td>1 108 1048576</td>
<td>1216512</td>
<td>1216512</td>
<td>1216512</td>
<td>1327104</td>
<td>VDA</td>
</tr>
<tr>
<td>63</td>
<td>630.00</td>
<td>630.426</td>
<td>10.091</td>
<td>291766.929</td>
<td>24809.548</td>
<td>266957.381</td>
<td>300</td>
<td>1 126 1048576</td>
<td>1419264</td>
<td>1419264</td>
<td>1419264</td>
<td>1548288</td>
<td>VDA</td>
</tr>
<tr>
<td>72</td>
<td>720.00</td>
<td>720.483</td>
<td>12.211</td>
<td>333282.071</td>
<td>29162.061</td>
<td>304120.009</td>
<td>300</td>
<td>1 144 1048576</td>
<td>1622016</td>
<td>1622016</td>
<td>1622016</td>
<td>1769472</td>
<td>VDA</td>
</tr>
<tr>
<td>81</td>
<td>810.00</td>
<td>810.540</td>
<td>11.029</td>
<td>373242.740</td>
<td>31460.608</td>
<td>341782.132</td>
<td>300</td>
<td>1 162 1048576</td>
<td>1824768</td>
<td>1824768</td>
<td>1824768</td>
<td>1990656</td>
<td>VDA</td>
</tr>
<tr>
<td>90</td>
<td>900.00</td>
<td>900.606</td>
<td>13.486</td>
<td>416206.961</td>
<td>35670.615</td>
<td>380536.346</td>
<td>300</td>
<td>1 180 1048576</td>
<td>2027520</td>
<td>2027520</td>
<td>2027520</td>
<td>2211840</td>
<td>VDA</td>
</tr>
</tbody>
</table>
Use sflowtool -4 –L to capture the sFlow data from the switch.

Use tag2014 to merge the power and sFlow data into a time correlated data set of the RUN/measurement phase.

https://github.com/powernap/tag2014/releases/tag/v1.5

Author Nick Principe at IXsystems

Use Excel to produce the MiBytes/sec/Watt efficiency values for the peak load point RUN phase of each workload.

Patrick Stanko presents spreadsheet walkthrough.
Data reduction

C:\tmp>
C:\tmp>c:"program files (x86)"\python36-32\python tag2014.py
Must specify an input file
USAGE: tag2014.py {-a|-c} [-f ts_col ... ] [-m] [-r] [-n] -i in_file -l sfslog -o out_file [-t time_shift]
   -a : Analyzer data (CSV data produced by Unisphere Analyzer)
   -c : CSV data
   -f ts_col : field(s) that contains timestamp information
   -m : restrict output, include WARMUP data
   -r : restrict output, include RUN data
   -n : restrict output, include RUN_TAIL data
   -i in_file : input data file
   -l sfslog : sfslog file
   -o out_file : output file, omit for STDOUT
   -t time_shift : shift the time in the data file by time_shift seconds
## Data reduction. Example SWBUILD

<table>
<thead>
<tr>
<th>Run #</th>
<th>sFlow MiB/sec</th>
<th>Avg Power</th>
<th>MiB/sec/Watt</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>3000</td>
<td>1500</td>
<td>2.00</td>
</tr>
<tr>
<td>2</td>
<td>5000</td>
<td>1502</td>
<td>3.32</td>
</tr>
<tr>
<td>3</td>
<td>7000</td>
<td>1504</td>
<td>4.65</td>
</tr>
<tr>
<td>4</td>
<td>9000</td>
<td>1506</td>
<td>5.97</td>
</tr>
<tr>
<td>5</td>
<td>11000</td>
<td>1508</td>
<td>7.29</td>
</tr>
<tr>
<td>6</td>
<td>13000</td>
<td>1510</td>
<td>8.61</td>
</tr>
<tr>
<td>7</td>
<td>15000</td>
<td>1512</td>
<td>9.92</td>
</tr>
<tr>
<td>8</td>
<td>17000</td>
<td>1514</td>
<td>11.22</td>
</tr>
<tr>
<td>9</td>
<td>19000</td>
<td>1516</td>
<td>12.53</td>
</tr>
<tr>
<td>10</td>
<td>21000</td>
<td>1518</td>
<td>13.83</td>
</tr>
</tbody>
</table>
Data reduction. Example VDA
## Data reduction

<table>
<thead>
<tr>
<th>Workloads</th>
<th>SWBUILD</th>
<th>VDI</th>
<th>VDA</th>
<th>DATABASE</th>
</tr>
</thead>
<tbody>
<tr>
<td>Maximum Efficiency</td>
<td>13.83 MiB/sec/Watt</td>
<td>18.75 MiB/sec/Watt</td>
<td>20.20 MiB/sec/Watt</td>
<td>17.33 MiB/sec/Watt</td>
</tr>
</tbody>
</table>
BFF and range submitted

After one has found the peak MiB/sec/Watt from each of the SFS2014 workloads. Find the best foot forward by increasing and decreasing drive count and run suite again. This may take several iterations.

Required by the EPA’s Energy Star program.
For the low and high range points use either of the below

- Efficiency results with -40% drive count and + 15% drive count.
- Storage device count points where Perf/W value is 15% < the BFF point
BFF and range submitted

Reference previous V2.0.2 Emerald Training slides

“Product Family, Best Foot Forward, Test Points and Qualification Ranges” see starting with slide #30
If things go wrong

- Filenames in the SFS2014 distribution are all lowercase or all upper case. This happens on many Unix systems. One can either fight their way through the RockRidge extensions at mount time, or just load the DVD in a Windows system, and copy the contents to a common shared location.

- Netmist load generator stores per process logs on each client in `/tmp/netmist_C*.log` (or wherever, if specified) These contain details of causes for failures.

- The sfslog.[vda,vdi,swbuild,database].txt files contain a high level summary, with a pointer to which client process failed. (so you know what log to examine in `/tmp`
Common failures

- DNS not setup properly. Both hostname to IP address and IP address to hostname need to work for all clients.
- Ensure that 127.0.0.1 hostname is *not* in /etc/hosts
- Password challenges during startup. Failure to setup the ssh keys so that challenges are not needed.
- Hangs at the beginning. Forgot to disable firewalls.
- ENOSPC. Failure to configure sufficient disk space.
- Runs marked INVALID. Load is exceeding server capabilities.
Windows failure to startup. You must have a dedicated Prime that does not present load.

You must have credentials that give the benchmark permission to access each other via WMI, and permission to access the storage. These are Active Directory configuration issues. Talk to the IT person that maintains the Active directory for your lab.

Set User in the sfs_rc file to: DOMAIN\accountname and the Password to the account’s password.

If all else fails, there is a SPECsfs2014 User’s guide here: https://www.spec.org/sfs2014/index.html#userguide
Common failures

Contact the SPEC Storage subcommittee support at:

sfs2014support@spec.org
Q & A session