When upgrading database servers, an essential IT strategy for many enterprises, businesses are looking for powerful hardware that offers caching and flash-based solutions to help maximize performance. Adding flash technology directly into servers can be an attractive, high-performance option for both caching and storing data, but the number of required drives for sufficient capacity and fault tolerance for each individual workload may be cost-prohibitive. Dell Fluid Cache for SAN offers an attractive alternative. This innovative server-side read/write caching solution leverages high-speed NVMe PCIe SSDs in validated servers and low-latency private cache network technology to form a cache pool. When connected to a Dell Compellent SAN, this solution provides a highly available data caching solution that can dramatically improve application performance.

The new Dell PowerEdge FX2s chassis populated with PowerEdge FC630 servers packs significant compute power into only 2U of rack space. When equipped with Dell Fluid Cache for SAN, this solution provides outstanding OLTP database performance.

In our hands-on tests in the Principled Technologies labs, we compared the OLTP database performance of this solution to a legacy solution using three-year-old HP ProLiant DL380 Gen8 servers with a competing caching solution. We configured both Dell Fluid Cache for SAN and the legacy servers with competing cache solution with an Oracle Real Application Cluster (RAC). For the Dell Fluid Cache for SAN solution, we first configured it with two nodes to compare to the legacy servers with competing cache solution, then with four nodes to demonstrate the peak performance capabilities of four
FC630 servers running both RAC and Dell Fluid Cache for SAN. The Dell Fluid Cache for SAN solution delivered 3.7 times the Oracle database transactions per minute of the competing solution with two nodes, and 4 times the performance with four nodes, making it an excellent choice over the legacy servers with competing cache solution.

WHAT IS DELL FLUID CACHE FOR SAN?

Dell designed Fluid Cache for SAN to accelerate application IO, and boost transactional performance for both read and write operations. Fluid Cache for SAN utilizes Dell PowerEdge Express Flash NVMe PCIe SSDs in validated servers to create a high performance cache pool. All cache pool servers each contain a Mellanox Connect X-3 card and connect into a validated Dell Networking switch for the low-latency private cache network. All of the servers are then connected to a Dell Compellent SAN.

The solution offers high availability by replicating blocks of data between cache nodes connected to the high-speed RDMA network, so your data remains safe in cache even in the unlikely event of a node or PCIe SSD failure in the cache pool. Fluid Cache for SAN provides a single management interface and other capabilities such as cache-aware snapshots, replication, and compression using Compellent Enterprise Manager. Dell Fluid Cache for SAN also allows you room to grow, with the ability to scale up to eight nodes per cache pool, scale up cache to 1.6 TB per server, and attach multiple cache pools to one Compellent SAN to meet the future demands of your business.

Figure 1 shows an example environment highlighting how Fluid Cache for SAN communicates between server and storage. Three validated nodes are required to run Dell Fluid Cache for SAN software. Two of these three nodes are required to have a minimum of one Dell PowerEdge Express Flash NVMe PCIe SSD each, and the network for the private cache network is based on the low-latency RDMA protocol. The RDMA network handles the cache pool communication between the nodes. Administrators use the Dell Compellent Enterprise manager software to create and manage the Fluid Cache cluster through automatic cache server discovery.
OVERVIEW

We first tested our database workload with two HP ProLiant DL380 Gen8 servers running Oracle 12c database Enterprise Edition in an Oracle RAC configuration running on Red Hat Enterprise Linux (RHEL) 6.4. This solution used a competing cache solution. We also configured a new Oracle RAC environment on the Dell PowerEdge FX2s and FC630 servers upgraded to RHEL 6.5, and added Fluid Cache to the picture to test the performance improvements. Testing the Dell solution, we first configured the four Dell PowerEdge FC630 servers with two nodes in an Oracle RAC with all four running Fluid Cache for SAN. We compared this configuration to the legacy server with competing cache solution. We then added the third and fourth cache nodes to our Oracle RAC to demonstrate the overall performance of all FX2s nodes running the database. (For system configuration information, see Appendix A and for step-by-step test details, see Appendix B.)
WHAT WE FOUND

Figure 2 shows the normalized test results, as Oracle licensing prohibits disclosure of the actual results. The Dell Fluid Cache for SAN solution provided 3.7 times the number of database transactions per minute (TPM) in a two-node Oracle RAC and four times the number of TPM in a four-node RAC that the competing cache solution provided.

Figure 2: The Dell PowerEdge FX2s with Dell Fluid Cache for SAN achieved up to four times the transactions per minute of the two-node legacy HP server with the competing cache solution.

CONFIGURATION AND TEST DETAILS

Competing cache solution

For the competing solution, we configured two HP ProLiant DL380 Gen8 servers with Red Hat Enterprise Linux 6.4. We then configured a two-node Oracle 12c database Standard Edition RAC cluster on the HP ProLiant DL380 Gen8 servers with a 460GB database leveraging Oracle Automatic Storage Management. For the cluster management and ASM traffic, we attached the first port of a dual-port 10Gb card in each DL380 server to a Dell Networking S4810 switch. For the client traffic, we attached one of each server’s onboard 1Gb NICs to a Dell Networking 6248 switch. We then configured the servers with the competing cache solution, utilizing a PCIe-SSD card and cache management software, as well as mapped the storage volumes on the Dell Compellent SC8000 storage (see Figure 3.)
Dell Fluid Cache for SAN solution

We configured two Dell PowerEdge FC630 servers in an Oracle RAC. We then installed the Dell Fluid Cache software and configured a dual-port 40Gb NIC in each FC630 for Fluid Cache traffic. The two ports were split, connecting to two different Dell Networking S4810 switch for redundancy. To complete our Fluid Cache cluster and add cache devices, we configured each of the four Dell PowerEdge FC630 servers with one 800GB Dell PowerEdge Express Flash NVMe PCIe SSD (see Figure 4). For our last scenario, we configured Oracle RAC on the remaining two nodes and added them to the existing cluster.

We then installed the Fluid Cache software and configured the management and Fluid Cache networking on all four servers. We then configured a Fluid Cache cluster via Dell Compellent Enterprise Manager. To enable Fluid Cache, we unmapped the volumes on the Dell Compellent SC8000 from the HP ProLiant DL380 Gen8 servers and competing cache solution, and remapped them to the Dell PowerEdge FC630 servers, enabling Fluid Cache with default settings.
Storage

For our storage, we configured two Dell Compellent SC8000 controllers connected to two Compellent SC220 disk enclosures, filled with rotational HDDs (48 x 146GB 15K drives). We configured a separate server to run the Dell Compellent Enterprise Manager Suite. For our storage LUNs, we configured two 500GB LUNs for backups, one 21GB LUN for the cluster registry volume, four 200GB LUNs for ASM data, and four 30GB LUNs for ASM logs. We connected each server to Compellent via Fibre Channel using a Brocade 300 Fabric switch. We then mapped each volume to two Dell FC630 servers.
ABOUT THE DELL POWEREDGE FX2S ENCLOSURE

The shared infrastructure approach of the Dell PowerEdge FX2s enclosure increases flexibility and can help you make the most of your data center space while reducing the cables and switches needed to run your servers. The Dell PowerEdge FX2s enclosure has a standard 2U footprint and features a modular design that can hold different combinations of compute and storage nodes to meet your specific goals. The PowerEdge FX2s fits four half-width or eight quarter-width compute nodes to increase the compute density in your rack and optimize the space in your data center. You can deploy the FX2s solution like a traditional rack-mounted server while gaining the benefits and features that more expensive dense blade solutions provide. Important features of the FX2s enclosure include:

- Up to eight low-profile PCIe expansion slots
- Two pass-through or optional networking FN I/O Aggregator modules
- Embedded network adapters within the server nodes
- Offers both chassis-based management through the Chassis Management Controller and rack-based management through Integrated Dell Remote Access (iDRAC) with Lifecycle Controller on each compute node

The Dell PowerEdge FX2s enclosure fits a number of server and storage options, including the PowerEdge FC430, FC630, and FC830 servers, and PowerEdge FD332 storage – all Intel-powered. The Dell PowerEdge FX2s solution we tested included four half-width Dell PowerEdge FC630 server sleds powered by Intel Xeon processors E5-2650 v3.

For more information about the Intel-powered Dell PowerEdge FX2s solution, visit www.dell.com/us/business/p/poweredge-fx/pd.

ABOUT OUR TESTING

For our workload, we ran HammerDB v2.16 with a TPC-C-like workload. We ran each test with 101 users for a 30-minute ramp-up time and a 60-minute test duration. We ran three tests with no Fluid Cache and three tests with Fluid Cache enabled on the ASM Data LUNs. The 3.7 times and 4.0 times advantages are reported from the median run of each set of three.
CONCLUSION

Dell Fluid Cache for SAN provides the key benefits of speed and high availability in server and array SSD caching. In our testing using Dell Fluid Cache for SAN and PowerEdge Express Flash NVMe PCIe SSDs on the new Dell PowerEdge FX2s architecture, the Dell solution with two RAC nodes delivered 3.7 times as many transactions per minute as the two legacy HP ProLiant DL380 Gen8 servers with the competing cache solution. When we configured the Dell PowerEdge FX2s with four Dell PowerEdge FC630 servers, the Dell solution delivered 4 times as many transactions per minutes than the legacy HP ProLiant DL380 Gen8 servers and competing cache solution did. With this kind of boost in performance, upgrading your old infrastructure to the new Dell FX2s with PowerEdge FC630 servers and Dell Fluid Cache for SAN is a sound investment. Increasing performance, the FX2s with Dell Fluid Cache for SAN gives your current customers excellent performance now, while giving you a great platform for future growth.
## APPENDIX A – SERVER CONFIGURATION INFORMATION

Figures 5 and 6 provide detailed configuration information for the test enclosure and server modules.

### Power supplies

<table>
<thead>
<tr>
<th>System</th>
<th>Dell PowerEdge FX2s server enclosure</th>
</tr>
</thead>
<tbody>
<tr>
<td>Number of power supplies</td>
<td>2</td>
</tr>
<tr>
<td>Vendor and model number</td>
<td></td>
</tr>
<tr>
<td>Wattage of each (W)</td>
<td>1,600</td>
</tr>
</tbody>
</table>

### Midplane

<table>
<thead>
<tr>
<th>System</th>
<th>Dell PowerEdge FX2s server enclosure</th>
</tr>
</thead>
<tbody>
<tr>
<td>Chassis midplane</td>
<td>1.0</td>
</tr>
</tbody>
</table>

### Chassis firmware

<table>
<thead>
<tr>
<th>System</th>
<th>Dell PowerEdge FX2s server enclosure</th>
</tr>
</thead>
<tbody>
<tr>
<td>Chassis Management Controller</td>
<td>1.10.A00.201410160007</td>
</tr>
<tr>
<td>Chassis Infrastructure</td>
<td>1.10.A00.201410066</td>
</tr>
<tr>
<td>IOM</td>
<td>11.0.0.0</td>
</tr>
<tr>
<td>Dell PowerEdge IOA FN 410S 10 GigaE</td>
<td>9-6(0-15)</td>
</tr>
</tbody>
</table>

### I/O modules

<table>
<thead>
<tr>
<th>System</th>
<th>Dell PowerEdge FX2s server enclosure</th>
</tr>
</thead>
<tbody>
<tr>
<td>Switch</td>
<td>Dell PowerEdge IOA FN2210S 10 Gb</td>
</tr>
<tr>
<td>Occupied bay</td>
<td>IOM Slot A1, IOM Slot A2</td>
</tr>
</tbody>
</table>

### Management modules

<table>
<thead>
<tr>
<th>System</th>
<th>Dell PowerEdge FX2s server enclosure</th>
</tr>
</thead>
<tbody>
<tr>
<td>Chassis Management Controller slot 1</td>
<td>Chassis Management Controller Hardware X05</td>
</tr>
</tbody>
</table>

**Figure 5:** Configuration details for the test enclosure.

### General

<table>
<thead>
<tr>
<th>System</th>
<th>Dell PowerEdge FC630 server module</th>
</tr>
</thead>
<tbody>
<tr>
<td>Number of processor packages</td>
<td>2</td>
</tr>
<tr>
<td>Number of cores per processor</td>
<td>10</td>
</tr>
<tr>
<td>Number of hardware threads per core</td>
<td>2</td>
</tr>
</tbody>
</table>

### CPU

<table>
<thead>
<tr>
<th>System</th>
<th>Dell PowerEdge FC630 server module</th>
</tr>
</thead>
<tbody>
<tr>
<td>Vendor</td>
<td>Intel</td>
</tr>
<tr>
<td>Name</td>
<td>Xeon</td>
</tr>
<tr>
<td>Model number</td>
<td>E5-2650 v3</td>
</tr>
<tr>
<td>Socket type</td>
<td>FCLGA2011-3</td>
</tr>
<tr>
<td>Core frequency (GHz)</td>
<td>2.30</td>
</tr>
<tr>
<td>Bus frequency (GHz)</td>
<td>4.80</td>
</tr>
<tr>
<td>L1 cache</td>
<td>10 x 32 KB</td>
</tr>
<tr>
<td>L2 cache</td>
<td>10 x 256 KB</td>
</tr>
<tr>
<td>L3 cache</td>
<td>25 MB</td>
</tr>
</tbody>
</table>

### Platform

<table>
<thead>
<tr>
<th>System</th>
<th>Dell PowerEdge FC630 server module</th>
</tr>
</thead>
<tbody>
<tr>
<td>Vendor and model number</td>
<td>Dell PowerEdge FC630</td>
</tr>
<tr>
<td>BIOS name and version</td>
<td>1.0.3</td>
</tr>
<tr>
<td>BIOS settings</td>
<td>Default</td>
</tr>
</tbody>
</table>
Oracle RAC performance: Dell PowerEdge FX2s with Fluid Cache for SAN vs. competing cache solution

Figure 6: Configuration details for the test server modules.

<table>
<thead>
<tr>
<th>System</th>
<th>Dell PowerEdge FC630 server module</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Memory module(s)</strong></td>
<td></td>
</tr>
<tr>
<td>Total RAM in system (GB)</td>
<td>64</td>
</tr>
<tr>
<td>Vendor and model number</td>
<td>Hynix HMA41GR7MF8N-TF</td>
</tr>
<tr>
<td>Type</td>
<td>PC4-2133P</td>
</tr>
<tr>
<td>Speed (MHz)</td>
<td>2133</td>
</tr>
<tr>
<td>Speed running in the system (MHz)</td>
<td>2133</td>
</tr>
<tr>
<td>Size (GB)</td>
<td>8</td>
</tr>
<tr>
<td>Number of RAM module(s)</td>
<td>8</td>
</tr>
<tr>
<td>Chip organization</td>
<td>Double-sided</td>
</tr>
<tr>
<td>Rank</td>
<td>Dual</td>
</tr>
<tr>
<td><strong>PCIe Extender</strong></td>
<td></td>
</tr>
<tr>
<td>Vendor</td>
<td>Dell</td>
</tr>
<tr>
<td>Firmware version</td>
<td>1</td>
</tr>
<tr>
<td><strong>Hard drive</strong></td>
<td></td>
</tr>
<tr>
<td>Vendor and model number</td>
<td>Dell MZ-WEI8000</td>
</tr>
<tr>
<td>Number of disks in system</td>
<td>1</td>
</tr>
<tr>
<td>Size (GB)</td>
<td>800</td>
</tr>
<tr>
<td>Type</td>
<td>PCIe SSD</td>
</tr>
<tr>
<td><strong>Ethernet adapters</strong></td>
<td></td>
</tr>
<tr>
<td>Vendor and model number</td>
<td>Broadcom NetXtreme II 10 Gb Ethernet BCM57810</td>
</tr>
<tr>
<td>Type</td>
<td>Integrated</td>
</tr>
<tr>
<td>Vendor and model number</td>
<td>Mellanox 40G ConnectX-3</td>
</tr>
<tr>
<td>Type</td>
<td>Discrete</td>
</tr>
<tr>
<td><strong>USB ports</strong></td>
<td></td>
</tr>
<tr>
<td>Number</td>
<td>2</td>
</tr>
<tr>
<td>Type</td>
<td>2.0</td>
</tr>
<tr>
<td><strong>Firmware</strong></td>
<td></td>
</tr>
<tr>
<td>iDRAC8 Enterprise</td>
<td>2.10.10.10 (16)</td>
</tr>
<tr>
<td>Lifecycle Controller</td>
<td>2.10.10.10</td>
</tr>
<tr>
<td>Backplane</td>
<td>2.09</td>
</tr>
<tr>
<td>Broadcom NetXtreme II 10 Gb Ethernet BCM57810</td>
<td>7.10.12</td>
</tr>
</tbody>
</table>
Figure 7 provides detailed information for the test storage.

<table>
<thead>
<tr>
<th>Storage array</th>
<th>Dell Compellent SC8000</th>
</tr>
</thead>
<tbody>
<tr>
<td>Number of active storage controllers</td>
<td>2</td>
</tr>
<tr>
<td>Firmware revision</td>
<td>6.5.2.12</td>
</tr>
<tr>
<td>Storage controller model</td>
<td>CT-SC8000</td>
</tr>
<tr>
<td><strong>Tray 1</strong></td>
<td></td>
</tr>
<tr>
<td>Number of disks</td>
<td>24</td>
</tr>
<tr>
<td>Disk vendor and model number</td>
<td>Dell ST9146853SS</td>
</tr>
<tr>
<td>Disk size (GB)</td>
<td>146</td>
</tr>
<tr>
<td>Disk buffer size (MB)</td>
<td>64</td>
</tr>
<tr>
<td>Disk RPM</td>
<td>15,000</td>
</tr>
<tr>
<td>Disk type</td>
<td>SAS</td>
</tr>
<tr>
<td><strong>Tray 2</strong></td>
<td></td>
</tr>
<tr>
<td>Number of disks</td>
<td>24</td>
</tr>
<tr>
<td>Disk vendor and model number</td>
<td>Dell ST9146853SS</td>
</tr>
<tr>
<td>Disk size (GB)</td>
<td>146</td>
</tr>
<tr>
<td>Disk buffer size (MB)</td>
<td>64</td>
</tr>
<tr>
<td>Disk RPM</td>
<td>15,000</td>
</tr>
<tr>
<td>Disk type</td>
<td>SAS</td>
</tr>
</tbody>
</table>

Figure 7: Detailed information for the test storage.
APPENDIX B – DETAILED TEST PROCEDURE

We used the following steps to configure each server and our Oracle RAC environment. For our competitor comparison, we installed a PCIe-SSD card in each HP server. We then installed and configured the competing cache solution using their best practices for an Oracle RAC configuration with RHEL 6.4.

Installing Red Hat Enterprise Linux 6.5
1. Insert the Red Hat Enterprise Linux 6.5 DVD into the server, and boot to it.
2. Select Install or upgrade an existing system.
3. If you are unsure of the fidelity of the installation disk, select OK to test the installation media; otherwise, select Skip.
4. In the opening splash screen, select Next.
5. Choose the language you wish to use, and click Next.
6. Select the keyboard layout, and click Next.
7. Select Basic Storage Devices, and click Next.
8. Select Fresh Installation, and click Next.
9. Insert the hostname, and select Configure Network.
10. In the Network Connections menu, configure network connections.
11. After configuring the network connections, click Close.
12. Select the nearest city in your time zone, and click Next.
13. Enter the root password, and click Next.
14. Select Use All Space, and click Next.
15. Select the appropriate Data Store Device and select where the Bootloader will go.
16. Remove the home partition, modify the Swap file to 20GB, and set the root partition to 100GB.
17. When the installation prompts you to confirm that you are writing changes to the disk, select Write changes to disk.
18. Select Basic Server, and click Next.
19. When the installation completes, select Reboot to restart the server.

Configuring RHEL 6.5 for Oracle RAC
1. Log onto the server as root.
2. Download and install the latest Mellanox driver from Dell.
3. Once the Mellanox driver is installed, assign an IP in the 192.168.6.0 (ASM subnet) to one port and 192.168.7.0 (Fluid Cache subnet) to the other.
4. Configure NTP on all servers.
5. Ensure that all hosts, virtual IPs, and then desired cluster SCAN name is resolvable via the host file or DNS.
6. Disable the firewall:
   `chkconfig iptables off`
   `chkconfig ip6tables off`
7. Disable SELinux:
   `vi /etc/selinux/config`
   `SELINUX=disabled`
8. Ensure that the following packages are installed and updated to the listed versions or later:
   - binutils-2.20.51.0.2-5.11.el6 (x86_64)
9. Using yum, install the following prerequisite packages for Oracle Database:
   yum install elfutils-libelf-devel
   yum install xhost
   yum install unixODBC
   yum install unixODBC-devel
   yum install xorg-x11-utils

10. Create Oracle users and groups by running these shell commands:
    groupadd -g 1001 oinstall
    groupadd -g 1002 dba
    groupadd -g 1003 asmadmin
    groupadd -g 1004 asmdba
    useradd -m -u 1002 -g oinstall -G dba,asmadmin,asmdba oracle
    useradd -m -u 1003 -g oinstall -G dba,asmadmin,asmdba grid

11. Add the following lines to the .profile file for the oracle user:
    export TMP=/tmp
    export TMPDIR=$TMP
    export ORACLE_HOSTNAME=hostname
    export ORACLE_UNQNAME=ORCL
    export ORACLE_BASE=/u01/app/oracle
    export GRID_HOME=/u01/app/12.1.0/grid
    export DB_HOME=$ORACLE_BASE/product/12.1.0/dbhome_1
    export ORACLE_HOME=$DB_HOME
    export ORACLE_SID=oracle_instance
    export ORACLE_TERM=xterm
    export BASE_PATH=/usr/sbin:$PATH
A Principled Technologies test report

Oracle RAC performance: Dell PowerEdge FX2s with Fluid Cache for SAN vs. competing cache solution

12. Add the following lines to the .profile file for the oracle user:

```bash
export TMP=/tmp
export TMPDIR=$TMP
export ORACLE_HOSTNAME=hostname
export ORACLE_BASE=/u01/app/grid
export GRID_HOME=/u01/app/12.1.0/grid
export ORACLE_HOME=$GRID_HOME
export ORACLE_SID=+ASM1
export ORACLE_TERM=xterm
export BASE_PATH=/usr/sbin:$PATH
export PATH=$ORACLE_HOME/bin:$BASE_PATH
export LD_LIBRARY_PATH=$ORACLE_HOME/lib:/lib:/usr/lib
export CLASSPATH=$ORACLE_HOME/JRE:$ORACLE_HOME/jlib:$ORACLE_HOME/rdbms/jlib
```

13. Create the following directories, and assign the following permissions.

```bash
mkdir -p /u01/app/oracle
mkdir /u01/app/grid
chown -R oracle:oinstall /u01/app
chmod -R g+w /u01/app
```

14. Create passwords for the oracle and grid accounts with passwd.

15. Append the following to /etc/security/limits.conf:

```bash
oracle - nofile 65536
oracle - nproc 16384
oracle - stack 32768
oracle - memlock 152043520
grid - nofile 65536
grid - nproc 16384
grid - stack 32768
* soft memlock unlimited
* hard memlock unlimited
```

16. We modified the system’s kernel parameters by appending the following to /etc/sysctl.conf:

```bash
fs.file-max = 6815744
kernel.sem = 250 32000 100 128
kernel.shmmax = 4096
kernel.shmall = 1073741824
kernel.shmmni = 439804651104
net.core.rmem_default = 262144
net.core.rmem_max = 20971520
net.core.wmem_default = 20971520
net.core.wmem_max = 20971520
fs.aio-max-nr = 1048576
net.ipv4.ip_local_port_range = 9000 65500
vm.nr_hugepages = 26214
vm.hugetlb_shm_group = 1001
```

17. Create the following file in /etc/udev/ruled.d/, replacing the ID_SERIAL to match the LUN IDs in Compellent Enterprise Manager.
Oracle RAC performance: Dell PowerEdge FX2s with Fluid Cache for SAN vs. competing cache solution

Installing Oracle Grid and RAC
1. Log in to the server as the grid account.
2. Set the X Window DISPLAY variable as appropriate for your configuration.
3. Copy the extracted download files for the Grid install to /grid.
4. Run the Grid Infrastructure GUI installer.
   ./grid/runInstaller
5. On the Download Software Updates screen, select Skip software updates, and click Next.
6. On the Select Installation Option screen, select Install and Configure Grid Infrastructure for a Cluster, and click Next.
7. On the Select Cluster Type screen, select Configure a Standard cluster, and click Next.
8. On the Select Installation Type screen, select Advanced Installation and click Next.
9. On the Select Product Languages screen, keep the default (English), and click Next.
10. On the Grid Plug and Play Information screen, deselect Configure GNS, enter a Cluster Name and a SCAN Name, and click Next.
12. On the Add Cluster Node Information pop-up screen, enter the hostname for node two for the Public hostname and a different hostname Virtual Hostnames, and click OK.
13. Click SSH Connectivity.
14. Enter the OS password for the grid user account, click Setup, then click Next.
15. On the Specify Network Interface Usage screen, set em1 to Public, p2p1 to ASM & Private, and the rest to Do Not Use, and click Next.
16. On the Grid Infrastructure Management Repository Option Screen, select No, and click Next.
17. On the Storage Option Information screen, select Use Oracle Flex ASM for storage, and click Next.
18. On the Create ASM Disk Group screen, click Change Discovery Path.
19. On the Change Discovery Path pop-up screen, enter /dev/oracleasm/, and click OK.
20. Change the Disk Group Name to CRV.
21. Select disk CRV1, and click Next.
22. On the Specify ASM Password, select Use same password for these accounts, and enter the ASMSNMP password (twice), and click Next.
23. On the Failure Isolation Support screen, select Do not use Intelligent Platform Management Interface (IPMI), and click Next.
24. On the Privileged Operating System Groups screen, keep the defaults, and click Next.
25. On the specify Installation location screen, enter /u01/app/grid for Oracle Base, /u01/app/12.1.0/grid for Software Location, and Click Next.
26. On the Create Inventory screen, enter /u01/app/oraInventory for the Inventory Directory, and click Next.
27. On the Root script execution configuration screen, click Next.
28. On the Perform Prerequisite Checks screen, click Fix and Check Again (to install the missing cvuqdisk package).
29. On the Execute Fixup Scripts pop-up screen, follow the instructions for running the fix-up script on each node. Click OK when done.
30. Click Next and after the check is finished.
31. On the Summary screen, click Install.
32. When the Execute Configuration scripts pop-up screen appears, follow the instructions for running the scripts, and click OK when finished.
33. Click Close to end the installation.

Creating Oracle ASM disk groups for the database
1. Log into the system as the grid user.
2. Set the X Window DISPLAY variable as appropriate for your configuration.
3. Start the ASM configuration assistant, asmca.
4. On the Disk Groups tab, click Create.
5. On the Create Disk Group pop-up screen, enter DATA for the Disk Group Name.
6. Select External (None) for Redundancy.
7. Select /dev/oracleasm/ for the Disk Path.
8. Select the four Data disks.
9. Click Show Advanced Options, and set the ASM and Database compatibilities to 12.1.0.0.
10. Click OK to create the DATA disk group, and click OK on the completed-task pop-up screen
11. Repeat steps 4-10 to create disk groups for log and backup.
12. Click Exit to close the ASM configuration assistant.

**Installing the software for Oracle Database 12c**

1. Log into the system as the oracle user.
2. Set the X Window DISPLAY variable as appropriate for your configuration.
3. Copy the extracted download files for the Database install to /database.
4. Run the Database GUI installer.
   ```
   ./database/runInstaller
   ```
5. On the Configure Security Updates screen, unselect I wish to receive security updates via My Oracle Support, and click Next.
6. On the warning pop-up screen, click Yes.
7. On the Download Software Updates screen, select Skip software updates, and click Next.
8. On the Select Installation Option screen, select Install database software only, and click Next.
9. On the Grid Installation Options screen, select Oracle Real Application Clusters database installation, select the nodes in the cluster, and click SSH Connectivity.
10. Enter the OS password for the grid user account, click Setup, then click Next.
11. On the Select Product Languages screen, keep the default (e.g., English) and click Next.
13. On the Specify Installation Location screen, enter /u01/app/oracle for Oracle Base and /u01/app/oracle/product/12.1.0/dbhome_1 for Software Location and click Next.
15. On the Perform Prerequisite Checks screen, click Next.
17. When the Execute Configurations scripts screen appears, follow the instructions and click OK.
18. On the final screen, click Close to end the installation.

**Creating the database**

We used the following steps to create our Oracle RAC database. For specific spfile configurations, see Appendix D.

1. Log into the system as the oracle user.
2. Set the X Window DISPLAY variable as appropriate for your configuration.
3. Start the database configuration assistant, dbca.
4. On the Database Operations screen, select Create a Database and click Next.
5. On the Creation Mode screen, select Advanced Mode, and click Next.
6. On the Database Templates screen, select Oracle Real Application Clusters (RAC) database for the Database Type, select Admin-Managed for the Configuration Type, and select General Purpose or Transaction Processing.
7. Click Next.
8. On the Database Identification screen, enter orcl for Global Database Name and SID Prefix, and click Next.
9. On the Database Placement screen, add both RAC nodes, and click Next.
10. On the Management Options screen, keep the defaults and click Next.
11. On the Database Credentials screen, select Use the Same Administrative Password for All Accounts, enter the password twice, and click Next.
12. On the Storage Locations screen, select ASM for Storage Type, select Use Common Location for All Database Files, enter +DATA for Database Files Locations.
13. Under Recovery Related Files, select ASM for the Storage Type, enter +BACKUP for the Fast Recovery Area, and enter 950GB for the Fast Recovery Size.
14. On the Initialization Parameters screen, leave the defaults, and click Next.
15. On the Creation Options screen, select Create Database, and click Next.
16. On the Pre Requisite Checks screen, allow the prerequisite checker to complete, resolve any issues, and click Next.
17. On the Summary pop-up screen, click OK.
18. Click Exit on the final pop-up screen to close the configuration assistant.
19. Once the database installed, we created two redo log groups. Each contained a single 10GB file on the +LOG ASM group. All default groups were removed.
20. On the database, we created a tablespace named TPCC, set it to bigfile at 600GB, and enabled autoextend at 1GB intervals.

**Generate HammerDB data**

**About HammerDB**
HammerDB is an open-source benchmark tool that tests the database performance of many leading databases, including Oracle Database, Microsoft® SQL Server®, PostgreSQL®, MySQL™, and more. The benchmark includes two built-in workloads derived from industry-standard benchmarks: a transactional (TPC-C-like) workload and a data warehouse (TPC-H-like) workload. For this study, we used the transactional workload. Our tests were not official TPC results and are not comparable in any manner. For more information about HammerDB, visit hammerora.sourceforge.net

We generated the data using one Oracle Enterprise Linux client with Oracle Database 12c Client and HammerDB installed.

2. Download and install HammerDB from hammerora.sourceforge.net/download.html
3. Edit the tnsnames file located in the Oracle home network admin folder to target the virtual IP and SID for the first node and database.
4. Repeat the edit on the second client, targeting the virtual IP and SID for the second node and database.
5. Launch the HammerDB UI.
6. In the HammerDB UI, click Options→Benchmark, and check Oracle and TPC-C. Click OK.
7. Click OK again to confirm the benchmark choice.
9. Double-click Options to open the Build Options menu.
10. For Oracle Service Name, type the SID of the database: orcl.
11. Change the system user password to match the password set during the database installation.
12. Enter TPCC for the default tablespace.
13. Set the number of warehouses to 5,000, and the users to 24.
14. Leave the rest of the fields as default.

15. To start the database generation, double-click Build.

**Configuring Dell Fluid Cache for SAN**

We used the following steps to configure and enable Fluid Cache for our caching tests. Before configuring Fluid Cache, we assigned a static IP in a new subnet on the 40Gb ports to be used for the Fluid Cache network. We also configured the four Dell PowerEdge FC630 servers with the Dell Fluid Cache software.

**Installing Dell Fluid Cache for SAN**

1. Prior to installing Fluid Cache, add the following lines to /etc/multipath.conf on each server:
   
   ```
   blacklist {
     devnode "^fldc[0-9]*"
     devnode "^nvme[a-z]*"
     devnode "^rssd[a-z]"
   }
   ```

2. Copy and extract the Fluid Cache software on each server.
3. In the extracted directory, run the rpm –i command on the Fluid Cache RPM to install.
4. Navigate to the /opt/dell/fluidcache/bin directory, and run ./hcn_setup.py.
5. Press Enter to begin the configuration.
6. Enter the desired subnet for the management interface (i.e. 192.168.1.0). This was our 1Gb management and client subnet.
7. Enter the desired subnet mask for the connection.
8. Enter Y to confirm the settings.
9. Enter N to skip configuring another network.
10. Enter the desired subnet for the cache network (i.e. 192.168.2.0). This was our 40Gb network for Fluid Cache.
11. Enter the desired subnet mask for the connection.
12. Enter Y to confirm the settings.
13. Enter Y to start the Fluid Cache software.

**Configuring the Fluid Cache cluster**

1. On a separate machine running Dell Compellent Enterprise Manager connected to the client network, log in to the EM.
2. At the top on the page, click Configure Fluid Cache Cluster.
3. Enter the management IP of one of the three R630 servers, and enter the root user and password.
4. Click Next.
5. Ensure that all servers are discovered and selected, and click Next.
6. Type a name for the new cluster and select a license file for the new cluster.
7. Click Next.
8. On the Select Devices page, select all three 800GB Dell PowerEdge Express Flash NVMe PCIe SSDs, and click Next.
9. Select the desired Storage Center for the Fluid Cache Cluster that contains the Oracle LUNs, and click Finish.

To enable Fluid Cache on the Data LUNs, unmap the four LUNs from the Oracle RAC nodes. Then remap the LUNs to the Oracle nodes, checking the box to enable Fluid Cache before click finish. In this testing, we used the default Fluid Cache settings for each data LUN.
Running the test

HammerDB settings
We configured HammerDB to run with the AWR snapshot script for a 30-minute warmup time and a 60 minute run. To ensure we had sufficient load on the environment, we set the user count 101 (one user is used as a management user). We also selected the options to Show Output and Log Output to Temp. To receive real-time data during the run, we configured the Transaction Counter to monitor TPM performance. Once the run completed, we gathered the NOPM output used this number for our comparison. Due to the nature of multi-node RAC and HammerDB output gathering, we had to edit the run script to gather NOPM and TPM results from both servers and combine them. The driver script we used is located in Appendix E with the customization in bold.

Run workflow
We used the following steps during each run iteration. After generating the database, we used RMAN to create a backup of each. These backups were stored in our Fast Recovery area on +BACKUP.

1. Restore the database using RMAN.
2. Reboot the server.
3. Once the two nodes are backup, allow them to sit idle for 20 minutes to ensure that Oracle RAC startup processes have completed.
4. Ensure that all settings in the HammerDB UI for each client are configured properly.
5. Start the Transaction Counter, and click Create Users
6. Start server stats gathering on each RAC node. We used sar to output system statistics to a bin file for 95 minutes using the following script:
   ```
   nohup sar -o /tmp/sar_$ (hostname -s).bin 15 380 > /dev/null 2>&1 &
   ```
7. On the HammerDB UI, hit the green → at the top of the page to begin the test.
8. Once the test completes, gather the sar bin files from the RAC nodes and the HammerDB output log on each client.
APPENDIX C – ORACLE SPFILE

orcl1.__data_transfer_cache_size=0
orcl2.__data_transfer_cache_size=0
orcl3.__data_transfer_cache_size=0
orcl4.__data_transfer_cache_size=0
orcl2.__db_cache_size=13757317120
orcl3.__db_cache_size=15636365312
orcl1.__db_cache_size=34225520640
orcl4.__db_cache_size=33688649728
orcl1.__java_pool_size=469762048
orcl2.__java_pool_size=469762048
orcl3.__java_pool_size=469762048
orcl4.__java_pool_size=469762048
orcl1.__large_pool_size=1140850688
orcl2.__large_pool_size=1140850688
orcl3.__large_pool_size=1140850688
orcl4.__large_pool_size=1140850688
orcl1.__oracle_base='/u01/app/oracle'#ORACLE_BASE set from environment
orcl2.__oracle_base='/u01/app/oracle'#ORACLE_BASE set from environment
orcl3.__oracle_base='/u01/app/oracle'#ORACLE_BASE set from environment
orcl4.__oracle_base='/u01/app/oracle'#ORACLE_BASE set from environment
orcl1.__pga_aggregate_target=6777995264
orcl2.__pga_aggregate_target=6777995264
orcl3.__pga_aggregate_target=6777995264
orcl4.__pga_aggregate_target=6777995264
orcl1.__sga_target=20266876928
orcl2.__sga_target=20266876928
orcl3.__sga_target=20266876928
orcl4.__sga_target=20266876928
orcl1.__shared_io_pool_size=536870912
orcl4.__shared_io_pool_size=536870912
orcl2.__shared_io_pool_size=536870912
orcl3.__shared_io_pool_size=536870912
orcl2.__shared_pool_size=4227858432
orcl3.__shared_pool_size=2415919104
orcl1.__shared_pool_size=4429185024
orcl4.__shared_pool_size=4429185024
orcl1.__streams_pool_size=0
orcl2.__streams_pool_size=0
orcl3.__streams_pool_size=0
orcl4.__streams_pool_size=0
*.audit_file_dest='/u01/app/oracle/admin/orcl/adump'
*.audit_trail='NONE'
*.cluster_database=true
*.compatible='12.1.0.0.0'
*.control_files='+DATA/ORCL/CONTROLFILE/current.261.865534663', '+BACKUP/ORCL/CONTROLFILE/current.256.865534663' #Restore Controlfile
*.db_16k_cache_size=8589934592
*.db_block_size=8192
*.db_cache_size=34359738368
*.db_create_file_dest='+DATA'
*.db_domain=''
*.db_name='orcl'
*.db_recovery_file_dest='+BACKUP'
*.db_recovery_file_dest_size=950g
*.diagnostic_dest='/u01/app/oracle'
*.dispatchers='(PROTOCOL=TCP) (SERVICE=orclXDB)' 
*.fast_start_mttr_target=60
*.filesystemio_options='setall'
*.log_checkpoints_to_alert=TRUE
*.open_cursors=2000
*.pga_aggregate_target=6442450944
*.processes=1000
*.query_rewrite_enabled='false'
*.recyclebin='off'
*.remote_login_passwordfile='exclusive'
*.sessions=1500
*.shared_pool_size=4294967296
orcl4.thread=4
orcl3.thread=3
orcl2.thread=2
orcl1.thread=1
*.trace_enabled=FALSE
*.undo_retention=1
orcl4.undo_tablespace='UNDOTBS2'
orcl1.undo_tablespace='UNDOTBS1'
orcl3.undo_tablespace='UNDOTBS3'
orcl2.undo_tablespace='UNDOTBS4'
*.use_large_pages='only'
APPENDIX D – HAMMERDB DRIVER SCRIPT

Below is the driver script we used in HammerDB. Customizations are in bold.

#!/usr/local/bin/tclsh8.6
if [catch {package require Oratcl} ] { error "Failed to load Oratcl - Oracle OCI Library Error" }
#AWR SNAPSHOT DRIVER SCRIPT##############################################
#THIS SCRIPT TO BE RUN WITH VIRTUAL USER OUTPUT ENABLED
#EDITABLE OPTIONS###################
set total_iterations 10000000 ;# Number of transactions before logging off
set RAISEERROR "false" ;# Exit script on Oracle error (true or false)
set KEYANDTHINK "false" ;# Time for user thinking and keying (true or false)
set CHECKPOINT "true" ;# Perform Oracle checkpoint when complete (true or false)
set rampup 30;  # Rampup time in minutes before first snapshot is taken
set duration 60;  # Duration in minutes before second AWR snapshot is taken
set mode "Local" ;# HammerDB operational mode
set timesten "false" ;# Database is TimesTen
set systemconnect system/Password1@ORCL ;# Oracle connect string for system user
set connect tpcc/tpcc@ORCL ;# Oracle connect string for tpcc user
#EDITABLE OPTIONS###################
#CHECK THREAD STATUS
proc chk_thread {} {
    set chk [package provide Thread]
    if {[string length $chk]} {
        return "TRUE"
    } else {
        return "FALSE"
    }
}
#STANDARD SQL
proc standsql { curn sql } {
    set ftch ""
    if {[catch {orasql $curn $sql} message]} {
        error "SQL statement failed: $sql : $message"
    } else {
        orafetch $curn -datavariable output
        while {
            oramsg $curn == 0
        } {
            lappend ftch $output
            orafetch $curn -datavariable output
        }
        return $ftch
    }
}
#Default NLS
proc SetNLS { lda } {
    set curn_nls [oraopen $lda ]
    set nls(1) "alter session set NLS_LANGUAGE = AMERICAN"
    set nls(2) "alter session set NLS_TERRITORY = AMERICA"
    for { set i 1 } { $i <= 2 } { incr i } {
        if {[catch {orasql $curn_nls $nls($i)} message ]} {
            puts "$message $nls($i)"
        }
    }
    return $curn_nls
}
puts [ oramsg $curn_nls all ]
}
}
oraclose $curn_nls

if { [ chk_thread ] eq "FALSE" } {
error "AWR Snapshot Script must be run in Thread Enabled Interpreter"
}
set mythread [thread::id]
set allthreads [split [thread::names]]
set totalvirtualusers [expr [llength $allthreads] - 1]
set myposition [expr $totalvirtualusers - [lsearch -exact $allthreads $mythread]]
if { ![catch {set timeout [tsv::get application timeout]}] } {
if { $timeout eq 0 } {
set totalvirtualusers [ expr $totalvirtualusers - 1 ]
set myposition [ expr $myposition - 1 ]
}
}
if { [ string toupper $timesten ] eq "TRUE"} {
set timesten 1
set systemconnect $connect
} else {
set timesten 0
}
switch $myposition {
1 {
if { $mode eq "Local" || $mode eq "Master" } {
set lda [oralogon $systemconnect]
if { !$timesten } { SetNLS $lda }
set lda1 [oralogon $connect]
if { !$timesten } { SetNLS $lda1 }
oraautocom $lda on
oraautocom $lda1 on
set curn1 [oraopen $lda ]
set curn2 [oraopen $lda1 ]
if { $timesten } {
puts "For TimesTen use external ttStats utility for performance reports"
sql1 "select (xact_commits + xact_rollbacks) from sys.monitor"
} else {
sql1 "BEGIN dbms_workload_repository.create_snapshot(); END;"
oraparse $curn1 $sql1
}
set ramptime 0
puts "Beginning rampup time of $rampup minutes"
set rampup [ expr $rampup*60000 ]
while {($ramptime != $rampup) } {
if { [ tsv::get application abort ] } { break } else { after 60000 }
set ramptime [ expr $ramptime+6000 ]
if { ![ expr ($ramptime % 60000) ] } {
puts "Rampup [ expr $ramptime / 60000 ] minutes complete ..."
}
if { [ tsv::get application abort ] } { break }
if { $timesten } {
  puts "Rampup complete, Taking start Transaction Count."
set start_trans [ standsql $curn2 $sql1 ]
} else {
  puts "Rampup complete, Taking start AWR snapshot."
if {[catch {oraplexec $curn1 $sql1} message]} {
    error "Failed to create snapshot : $message"
} set sql2 "SELECT INSTANCE_NUMBER, INSTANCE_NAME, DB_NAME, DBID, SNAP_ID, TO_CHAR(END_INTERVAL_TIME,'DD MON YYYY HH24:MI') FROM (SELECT DI.INSTANCE_NUMBER, DI.INSTANCE_NAME, DI.DB_NAME, DI.DBID, DS.SNAP_ID, DS.END_INTERVAL_TIME FROM DBA_HIST_SNAPSHOT DS, DBA_HIST_DATABASE_INSTANCE DI WHERE DS.DBID=DI.DBID AND DS.INSTANCE_NUMBER=DI.INSTANCE_NUMBER AND DS.STARTUP_TIME=DI.STARTUP_TIME ORDER BY DS.SNAP_ID DESC) WHERE ROWNUM=1"
if {[catch {orasql $curn1 $sql2} message]} {
  error "SQL statement failed: $sql2 : $message"
} else {
  orafetch $curn1 -datavariable firstsnap
  split $firstsnap " "
}
set sql4 "select sum(d_next_o_id) from district"
set start_nopm [ standsql $curn2 $sql4 ]
puts "Timing test period of $duration in minutes"
set testtime 0
set durmin $duration
set duration [ expr $duration*60000 ]
while {$testtime != $duration} {
  if { [ tsv::get application abort ] } { break } else { after 6000 }
  set testtime [ expr $testtime+6000 ]
  if { ![ expr ($testtime % 60000) ] } {
    puts -nonewline "[ expr $testtime / 60000 ] ...,"
  }
}
if { [ tsv::get application abort ] } { break }
if { $timesten } {
  puts "Test complete, Taking end Transaction Count."
set end_trans [ standsql $curn2 $sql1 ]
set end_nopm [ standsql $curn2 $sql4 ]
set tpm [ expr {($end_trans - $start_trans)/$durmin} ]
set nopm [ expr {($end_nopm - $start_nopm)/$durmin} ]
puts "$totalvirtualusers Virtual Users configured"
puts "TEST RESULT : System achieved $tpm TimesTen TPM at $nopm NOPM"
} else {
  puts "Test complete, Taking end AWR snapshot."
  oraparse $curn1 $sql1
  if {[catch {oraplexec $curn1 $sql1} message]} {
    error "Failed to create snapshot : $message"
  } if {[catch {orasql $curn1 $sql2} message]} {

error "SQL statement failed: $sql2 : $message"
} else {
  orafetch $curn1 -datavariable endsnap
  split $endsnap " "
  puts "Test complete: view report from SNAPID [ lindex $firstsnap 4 ] to [ lindex $endsnap 4 ]!"

  set sql3 "select round((sum(tps)*60)) as TPM from (select e.stat_name, (e.value - b.value) / (select avg( extract( day from (e1.end_interval_time - bl.end_interval_time) )*24*60*60+ extract( hour from (e1.end_interval_time - bl.end_interval_time) )*60*60+ extract( minute from (e1.end_interval_time - bl.end_interval_time) )*60+ extract( second from (e1.end_interval_time - bl.end_interval_time))) ) from dba_hist_snapshot b, dba_hist_snapshot e where b.snap_id = [ lindex $firstsnap 4 ] and e.snap_id = [ lindex $endsnap 4 ] and b.dbid = [lindex $firstsnap 3] and e.dbid = [lindex $endsnap 3] and b.instance_number = 1 and e.instance_number = 1 and b.stat_id = e.stat_id and b.stat_name in ('user commits','user rollbacks') and e.stat_name in ('user commits','user rollbacks') order by 1 asc)"
  set tpm1 [ standsql $curn1 $sql3 ]
  puts "Instance 1 TPM: $tpm1"

  set sql3 "select round((sum(tps)*60)) as TPM from (select e.stat_name, (e.value - b.value) / (select avg( extract( day from (e1.end_interval_time - bl.end_interval_time) )*24*60*60+ extract( hour from (e1.end_interval_time - bl.end_interval_time) )*60*60+ extract( minute from (e1.end_interval_time - bl.end_interval_time) )*60+ extract( second from (e1.end_interval_time - bl.end_interval_time))) ) from dba_hist_snapshot b, dba_hist_snapshot e where b.snap_id = [ lindex $firstsnap 4 ] and e.snap_id = [ lindex $endsnap 4 ] and b.dbid = [lindex $firstsnap 3] and e.dbid = [lindex $endsnap 3] and b.instance_number = 2 and e.instance_number = 2 and b.stat_id = e.stat_id and b.stat_name in ('user commits','user rollbacks') and e.stat_name in ('user commits','user rollbacks') order by 1 asc)"
  set tpm2 [ standsql $curn1 $sql3 ]
  puts "Instance 2 TPM: $tpm2"

  set sql3 "select round((sum(tps)*60)) as TPM from (select e.stat_name, (e.value - b.value) / (select avg( extract( day from (e1.end_interval_time - bl.end_interval_time) )*24*60*60+ extract( hour from (e1.end_interval_time - bl.end_interval_time) )*60*60+ extract( minute from (e1.end_interval_time - bl.end_interval_time) )*60+ extract( second from (e1.end_interval_time - bl.end_interval_time))) ) from dba_hist_snapshot b, dba_hist_snapshot e where b.snap_id = [ lindex $firstsnap 4 ] and e.snap_id = [ lindex $endsnap 4 ] and b.dbid = [lindex $firstsnap 3] and e.dbid = [lindex $endsnap 3] and b.instance_number = 3 and e.instance_number = 3 and b.stat_id = e.stat_id and b.stat_name in ('user commits','user rollbacks') and e.stat_name in ('user commits','user rollbacks') order by 1 asc)"
  set tpm3 [ standsql $curn1 $sql3 ]
  puts "Instance 3 TPM: $tpm3"

  set sql3 "select round((sum(tps)*60)) as TPM from (select e.stat_name, (e.value - b.value) / (select avg( extract( day from (e1.end_interval_time - bl.end_interval_time) )*24*60*60+ extract( hour from (e1.end_interval_time - bl.end_interval_time) )*60*60+ extract( minute from (e1.end_interval_time - bl.end_interval_time) )*60+ extract( second from (e1.end_interval_time - bl.end_interval_time))) ) from dba_hist_snapshot b, dba_hist_snapshot e where b.snap_id = [ lindex $firstsnap 4 ] and e.snap_id = [ lindex $endsnap 4 ] and b.dbid = [lindex $firstsnap 3] and e.dbid = [lindex $endsnap 3] and b.instance_number = 4 and e.instance_number = 4 and b.stat_id = e.stat_id and b.stat_name in ('user commits','user rollbacks') and e.stat_name in ('user commits','user rollbacks') order by 1 asc)"
  set tpm4 [ standsql $curn1 $sql3 ]
  puts "Instance 4 TPM: $tpm4"
1. startup_time and b1.end_interval_time < e1.end_interval_time) as tps from 
   dba_hist_sysstat b, dba_hist_sysstat e where b.snap_id = [ lindex $firstsnap 4 ]
   and e.snap_id = [ lindex $endsnap 4 ] and b.dbid = [lindex $firstsnap 3] and 
   e.dbid = [lindex $endsnap 3] and b.instance_number = 3 and e.instance_number = 3
   and b.stat_id = e.stat_id and b.stat_name in ('user commits','user rollbacks')
   and e.stat_name in ('user commits','user rollbacks') order by 1 asc)
set tpm3 ['standsql $curn1 $sql3]
puts "Instance 3 TPM: $tpm3"
set sql3 "select round((sum(tps)*60)) as TPM from (select e.stat_name, (e.value -
   b.value) / (select avg( extract( day from (e1.end_interval_time-
   b1.end_interval_time) )*24*60*60+ extract( hour from (e1.end_interval_time-
   b1.end_interval_time) )*60*60+ extract( minute from (e1.end_interval_time-
   b1.end_interval_time) )*60+ extract( second from (e1.end_interval_time-
   b1.end_interval_time) ) ) from dba_hist_snapshot b1, dba_hist_snapshot e1 where
   b1.snap_id = [ lindex $firstsnap 4 ] and e1.snap_id = [ lindex $endsnap 4 ] and
   b1.dbid = [lindex $firstsnap 3] and e1.dbid = [lindex $endsnap 3] and
   b1.instance_number = 4 and e1.instance_number = 4 and b1.startup_time =
   e1.end_interval_time and bl.end_interval_time < e1.end_interval_time) as tps from
   dba_hist_sysstat b, dba_hist_sysstat e where b.snap_id = [ lindex $firstsnap 4 ]
   and e.snap_id = [ lindex $endsnap 4 ] and b.dbid = [lindex $firstsnap 3] and 
   e.dbid = [lindex $endsnap 3] and b.instance_number = 4 and e.instance_number = 4
   and b.stat_id = e.stat_id and b.stat_name in ('user commits','user rollbacks')
   and e.stat_name in ('user commits','user rollbacks') order by 1 asc)"
set tpm4 ['standsql $curn1 $sql3]
puts "Instance 4 TPM: $tpm4"
set tpm [ expr $tpm1 + $tpm2 + $tpm3 + $tpm4]
set end_nopm [ standsql $curn2 $sql4 ]
set nopm [ expr {($end_nopm - $start_nopm)/$durmin} ]
puts "$totalvirtualusers Virtual Users configured"
puts "TEST RESULT : System achieved $tpm Oracle TPM at $nopm NOPM"
}

}
puts "Operating in Slave Mode, No Snapshots taken..."

} else {

default {

# RANDOM NUMBER
proc RandomNumber {m M} {return [expr {int($m+rand())*($M+1-$m))}]
#NURand function
proc NURand { iConst x y C } {return [ expr {(((RandomNumber 0 $iConst) | [RandomNumber $x $y]) + $C) % ($y - $x + 1)) + $x }]
# RANDOM NAME
proc randname { num } {
array set namearr { 0 BAR 1 OUGHT 2 ABLE 3 PRI 4 PRES 5 ESE 6 ANTI 7 CALLY 8 ATION 9 EING }
set name [ concat $namearr([ expr {( $num / 100 ) % 10 }])$namearr([ expr {( $num / 10 ) % 10 }])$namearr([ expr {( $num / 1 ) % 10 }]) ]
return $name
}
# TIMESTAMP
proc gettimestamp { } {
set tstamp [ clock format [ clock seconds ] -format %Y%m%d%H%M%S ]
return $tstamp
}
# KEYING TIME
proc keytime { keying } {
after [ expr {$keying * 1000} ]
return
}
# THINK TIME
proc thinktime { thinking } {
set thinkingtime [ expr {abs(round(log(rand()) * $thinking))} ]
after [ expr {$thinkingtime * 1000} ]
return
}
# NEW ORDER
proc neword { curno_no w_id_input RAISEERROR } {
# 2.4.1.2 select district id randomly from home warehouse where d_w_id = d_id
set no_d_id [ RandomNumber 1 10 ]
# 2.4.1.2 Customer id randomly selected where c_d_id = d_id and c_w_id = w_id
set no_c_id [ RandomNumber 1 3000 ]
# 2.4.1.3 Items in the order randomly selected from 5 to 15
set ol_cnt [ RandomNumber 5 15 ]
# 2.4.1.6 order entry date O_ENTRY_D generated by SUT
set date [ gettimestamp ]
orabind $curno_no :no_w_id $w_id_input :no_d_id $no_d_id :no_c_id $no_c_id :no_o_ol_cnt $ol_cnt :no_d_tax {} :no_w_tax {} :no_d_next_o_id {0} :timestamp $date
if { [catch { orarec $curno_no } message] }
if { [ RAISEERROR ] }
error "New Order : $message [ oramsg $curno_no all ]"
else {

}
proc payment { curn_py p_w_id w_id_input RAISEERROR } {
#2.5.1.1 The home warehouse id remains the same for each terminal
#2.5.1.1 select district id randomly from home warehouse where d_w_id = d_id
set p_d_id [ RandomNumber 1 10 ]
#2.5.1.2 customer selected 60% of time by name and 40% of time by number
set x [ RandomNumber 1 100 ]
set y [ RandomNumber 1 100 ]
if { $x <= 85 } {
    set p_c_d_id $p_d_id
    set p_c_w_id $p_w_id
} else {
    #use a remote warehouse
    set p_c_d_id [ RandomNumber 1 10 ]
    set p_c_w_id [ RandomNumber 1 $w_id_input ]
    while { ($p_c_w_id == $p_w_id) && ($w_id_input != 1) } {
        set p_c_w_id [ RandomNumber 1 $w_id_input ]
    }
}
set nrnd [ NURand 255 0 999 123 ]
set name [ randname $nrnd ]
set p_c_id [ RandomNumber 1 3000 ]
if { $y <= 60 } {
    #use customer name
    #C_LAST is generated
    set byname 1
} else {
    #use customer number
    set byname 0
    set name {}
}
#2.5.1.3 random amount from 1 to 5000
set p_h_amount [ RandomNumber 1 5000 ]
#2.5.1.4 date selected from SUT
set h_date [ gettimestamp ]
#2.5.2.1 Payment Transaction
#change following to correct values
orabind $curn_py :p_w_id $p_w_id :p_d_id $p_d_id :p_c_w_id $p_c_w_id :p_c_d_id
$p_c_d_id :p_c_id $p_c_id :byname $byname :p_h_amount $p_h_amount :p_c_last $name
:p_w_street_1 {} :p_w_street_2 {} :p_w_city {} :p_w_state {} :p_w_zip {}
:p_c_first {} :p_c_middle {} :p_c_street_1 {} :p_c_street_2 {} :p_c_city {}
:p_c_state {} :p_c_zip {} :p_c_phone {} :p_c_since {} :p_c_credit (0)
:p_c_credit_lim {} :p_c_discount {} :p_c_balance (0) :p_c_data {} :timestamp
$h_date
if {[ catch {oraexec $curn_py} message]} {
    if { $RAISEERROR } {
} else {
orafetch $curn_no -datavariable output
; 
}
}
#PAYMENT
}
error "Payment : $message [ oramsg $curn_py all ]"
            ) else {

            ;
        } else {
        orafetch $curn_py -datavariable output
        ;
    }
}

#ORDER_STATUS
proc ostat { curn_os w_id RAISEERROR } {
#2.5.1.1 select district id randomly from home warehouse where d_w_id = d_id
set d_id [ RandomNumber 1 10 ]
set nrnd [ NURand 255 0 999 123 ]
set name [ randname $nrnd ]
set c_id [ RandomNumber 1 3000 ]
set y [ RandomNumber 1 100 ]
if { $y <= 60 } {
    set byname 1
} else {
    set byname 0
}
set name {}
orabind $curn_os :os_w_id $w_id :os_d_id $d_id :os_c_id $c_id :byname $byname
    :os_c_last $name :os_c_first {} :os_c_middle {} :os_c_balance {0} :os_o_id {}
    :os_entdate {} :os_o_carrier_id {}
if {[catch {oraexec $curn_os} message]} {
    if { $RAISEERROR } {
        error "Order Status : $message [ oramsg $curn_os all ]"
    } else {

    }
} else {
        orafetch $curn_os -datavariable output
        ;
    }
}

#DELIVERY
proc delivery { curn_dl w_id RAISEERROR } {
    set carrier_id [ RandomNumber 1 10 ]
    set date [ gettimestamp ]
orabind $curn_dl :d_w_id $w_id :d_o_carrier_id $carrier_id :timestamp $date
    if {[catch {oraexec $curn_dl} message]} {
        if { $RAISEERROR } {
            error "Delivery : $message [ oramsg $curn_dl all ]"
        } else {

        }
    } else {
        orafetch $curn_dl -datavariable output
        ;
    }
}

#STOCK LEVEL
proc slev { curn_sl w_id stock_level_d_id RAISEERROR } {

A Principled Technologies test report

Oracle RAC performance: Dell PowerEdge FX2s with Fluid Cache for
SAN vs. competing cache solution
Oracle RAC performance: Dell PowerEdge FX2s with Fluid Cache for SAN vs. competing cache solution
Oracle RAC performance: Dell PowerEdge FX2s with Fluid Cache for SAN vs. competing cache solution

{ elseif ($choice <= 23) {
  if { $KEYANDTHINK } { keytime 2 }
  ostat $curn_os $w_id $RAISEERROR
  if { $KEYANDTHINK } { thinktime 5 }

  oraclose $curn_no
  oraclose $curn_py
  oraclose $curn_dl
  oraclose $curn_sl
  oraclose $curn_os
  oralogoff $lda
  } }
ABOUT PRINCIPLED TECHNOLOGIES

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.

Principled Technologies is a registered trademark of Principled Technologies, Inc. All other product names are the trademarks of their respective owners.

Disclaimer of Warranties; Limitation of Liability:
PRINCIPLED TECHNOLOGIES, INC. HAS MADE REASONABLE EFFORTS TO ENSURE THE ACCURACY AND VALIDITY OF ITS TESTING, HOWEVER, PRINCIPLED TECHNOLOGIES, INC. SPECIFICALLY DISCLAIMS ANY WARRANTY, EXPRESSED OR IMPLIED, RELATING TO THE TEST RESULTS AND ANALYSIS, THEIR ACCURACY, COMPLETENESS OR QUALITY, INCLUDING ANY IMPLIED WARRANTY OF FITNESS FOR ANY PARTICULAR PURPOSE. ALL PERSONS OR ENTITIES RELYING ON THE RESULTS OF ANY TESTING DO SO AT THEIR OWN RISK, AND AGREE THAT PRINCIPLED TECHNOLOGIES, INC., ITS EMPLOYEES AND ITS SUBCONTRACTORS SHALL HAVE NO LIABILITY WHATSOEVER FROM ANY CLAIM OF LOSS OR DAMAGE ON ACCOUNT OF ANY ALLEGED ERROR OR DEFECT IN ANY TESTING PROCEDURE OR RESULT.

IN NO EVENT SHALL PRINCIPLED TECHNOLOGIES, INC. BE LIABLE FOR INDIRECT, SPECIAL, INCIDENTAL, OR CONSEQUENTIAL DAMAGES IN CONNECTION WITH ITS TESTING, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGES. IN NO EVENT SHALL PRINCIPLED TECHNOLOGIES, INC.’S LIABILITY, INCLUDING FOR DIRECT DAMAGES, EXCEED THE AMOUNTS PAID IN CONNECTION WITH PRINCIPLED TECHNOLOGIES, INC.’S TESTING. CUSTOMER’S SOLE AND EXCLUSIVE REMEDIES ARE AS SET FORTH HEREIN.