An Oracle White Paper
September 2014

Optimizing Oracle Database Performance on Oracle Linux with Flash

Introduction

This document explains how to increase Oracle database performance and improve user response time by deploying Oracle’s Flash Accelerator hardware with Oracle’s Database Smart Flash Cache feature. Oracle’s Database Smart Flash Cache feature is supported only on Oracle Linux and Oracle Solaris operating systems beginning with Oracle 11gR2.

The benefits outlined here apply to all Oracle Flash products on Oracle Linux or Oracle Solaris. For this particular study Oracle’s PCIe based in-server Sun Flash Accelerator F40 PCIe Card and Oracle Linux with Unbreakable Enterprise Kernel were used. The findings apply to both the Sun Flash Accelerator F40 and F80 PCIe Cards.

With very high IOPS performance, low latency and a low CPU burden, the Sun Flash Accelerator PCIe Card maximizes transactional I/O performance for applications like online transaction processing, data warehousing and data mining when coupled with a database and operating system having support for the technology. By delivering the I/O performance of hundreds of traditional hard disk drives at a fraction of their latency and power, the Sun Flash Accelerator PCIe card improves business responsiveness and user satisfaction with faster application performance, greater productivity and server efficiency.

Oracle Linux with Unbreakable Enterprise Kernel was chosen for this study because the Unbreakable Enterprise Kernel is modern, includes the latest upstream hardware support relevant to data center operations, and includes features that can benefit SSD performance such as SSD Detect.

Figure 1 Sun Flash Accelerator PCIe Card

Advantages of Using Flash-based Caching / Storage with an Oracle Database and Oracle Linux

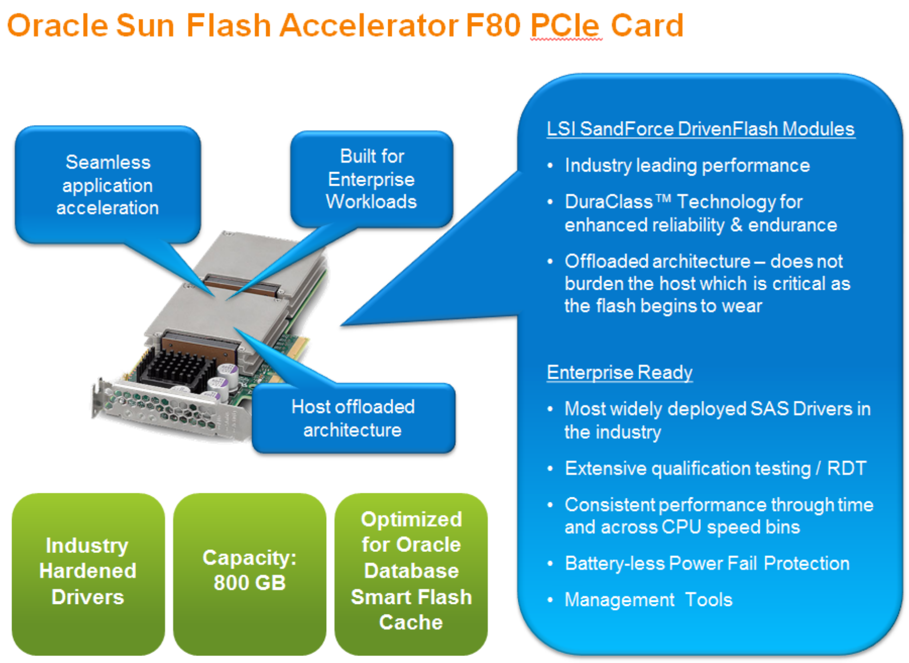
Online transaction processing (OLTP) and Data Warehousing & Analytics (DW) are typical types of applications that use an Oracle database. OLTP and DW applications have demanding requirements for fast response times and high throughput which makes it difficult for database administrators (DBAs) to maintain and scale their infrastructure as the number of users grows and the amount of data increases. During the application life cycle, performance bottlenecks might originate in one or more areas, such as the network, the processor, and the storage devices. Correcting a bottleneck in one area might cause another bottleneck to appear in another area.

Flash-based storage provides a new tool for DBAs to solve their performance issues. For example, 5 ms is the typical response time for a small data read from a hard disk drive. Flash-based devices like the Oracle Sun Flash Accelerator PCIe Card can complete the same read, on average, in 50 µs to 300 µs, an improvement of several orders-of-magnitude in response time.

Flash-based storage provides performance that falls between the performance levels of hard disk drives and DDR3 memory. Initial implementations of flash-based drives (SSDs, or solid state drives) were intended to replace a hard disk drive in direct-attach storage or RAID subsystems. Mounting SSDs on a PCIe card is a recent innovation that alleviates throughput constraints that are also caused by the storage interface allowing it to be used as a cache device and not just persistent storage. Oracle’s Sun Flash Accelerator PCIe Cards offer up to 800 GB capacity with over 155,000 random input/output operations (IOPs) and 2.1 GB/s bandwidth performance with a single low-profile PCIe card.

Overview of Oracle’s Sun Flash Accelerator PCIe Card

Oracle’s Sun Flash Accelerator PCIe Card offers high performance with low latency and a low CPU burden. Oracle’s Sun Flash Accelerator PCIe Card maximizes transactional I/O performance for Oracle databases and for other applications that require high-performance computing. Oracle’s Sun Flash Accelerator PCIe Card performs consistently across reads and writes regardless of workload by using industry-standard and widely deployed Oracle SAS software for easier system integration and management and a faster time to market.



Oracle Sun Flash Accelerator F80 PCIe Card

Seamless application acceleration

Built for Enterprise Workloads

Host offloaded architecture

Industry Hardened Drivers

Capacity: 800 GB

Optimized for Oracle Database Smart Flash Cache

LSI SandForce DrivenFlash Modules

- Industry leading performance
- DuraClass™ Technology for enhanced reliability & endurance
- Offloaded architecture – does not burden the host which is critical as the flash begins to wear

Enterprise Ready

- Most widely deployed SAS Drivers in the industry
- Extensive qualification testing / RDT
- Consistent performance through time and across CPU speed bins
- Battery-less Power Fail Protection
- Management Tools

Figure 2 High level technology overview and benefits of the PCIe card

Configuring Oracle Linux and the Oracle Database for Optimum I/O Performance

You can achieve significant database I/O improvement by implementing Oracle Linux with Unbreakable Enterprise Kernel (UEK) and following some simple guidelines in setting up the kernel and the Oracle Database. This section will describe the configuration steps for Oracle Linux with Unbreakable Enterprise Kernel and the Oracle Database to use and optimize the Oracle's Sun Flash Accelerator PCIe Card with the Oracle Database Smart Flash Cache feature.

Configure Oracle's Sun Flash Accelerator PCIe Card as a File System

The Oracle's Sun Flash Accelerator PCIe Card is a block device optimized for block sizing and alignment that work best with most applications (consistent with that of Oracle databases). This section explains actions you can take to tune Oracle's Sun Flash Accelerator PCIe Card for maximum performance in an Oracle Linux environment with Unbreakable Enterprise Kernel. The following steps configure Oracle's Sun Flash Accelerator PCIe Card as one (1) file system that one database can use. Other options would be to create multiple aligned partitions on Oracle's Sun Flash Accelerator PCIe Card and allocate these partitions to other databases residing on the server for their own Oracle Database Smart Flash Cache.

- Align the Oracle Sun Flash Accelerator PCIe Card on a 1M boundary in Oracle Linux. The following example shows how to calculate the boundary alignment, using Oracle's Sun Flash Accelerator F40 PCIe Card specifications:

```
fdisk -lu /dev/sda
Disk /dev/sda: 400.0 GB, 399999762432 bytes
255 heads, 63 sectors/track, 48630 cylinders, total 781249536
sectors
```

To align Oracle's Sun Flash Accelerator Card, the "`sfdisk`" command can be executed to create the aligned partition. The following "`sfdisk`" command that will align Oracle's Sun Flash Accelerator F40 Card on a 1M boundary:

```
echo "2048,," | sfdisk -uS /dev/sda
```

Verify that the partition was created starting at sector 2048 by running "`fdisk -lu /dev/sda`" or running "sfdisk –l –uS /dev/sda" after creating the aligned partition.

For creating an EXT-4 file system:

```
mkfs –t ext4 /dev/sda1  or mkfs.ext4 /dev/sda1
```

When mounting the new EXT-4 device:

```
mount -t ext4 -o noatime,nodiratime,max_batch_time=0,nobarrier
/dev/sda1 /mountpoint
```

The DEADLINE I/O scheduler is enabled by default in Oracle's Unbreakable Enterprise Kernel.

To verify that DEADLINE is enabled, issue the following statement as root:

```
cat /sys/block/sda/queue/scheduler
noop anticipatory [deadline] cfq
```

- In addition to changing the I/O scheduler, use the

  "`noatime,nodiratime,max_batch_time=0,nobarrier`" file system mount options in the `/etc/fstab` file. These options eliminate system overhead when performing IO. These options also enable faster access to the files, plus the benefit of less wear on Oracle's Sun Flash Accelerator PCIe Cards. This example shows how the `/etc/fstab` entry invokes these options:

```
/dev/sda1 /MP ext4 defaults,noatime,nodiratime,max_batch_time=0,nobarrier 1 2
```

- **Performance of Sun Flash Accelerator PCIe Card can benefit by increasing Queue Depth (QD)** from the default of 128 to 512 or higher, depending on the load. Since the latency of Sun Flash Accelerator PCIe Card is so small, more I/O operations can be run in parallel on the PCIe Card. In order to modify queue depth, the "`nr_requests`" parameter will need to be modified to the same or larger value of the new queue depth value. Here are examples of modifying both the "`nr_requests`" and "`queue_depth`" parameters for `/dev/sda` which was used in our tests:

```
echo "512" > /sys/block/sda/queue/nr_requests
echo "512" > /sys/block/sda/device/queue_depth
```

To ensure these settings persist across reboots, place these commands in the `/etc/rc.local` file.

Configure Oracle ASM Using Multiple Oracle's Sun Flash Accelerator PCIe Cards for Mirroring or for Increased Smart Flash Cache Capacity

To provide expanded capacity for the Database Smart Flash Cache multiple Oracle Sun Flash Accelerator PCIe Cards can be deployed using volume management software. Multiple Oracle Sun Flash Accelerator PCIe Cards can be mirrored, but in the following example, two cards were installed using Oracle ASM to expand the cache area capacity.

A benefit of using an ASM diskgroup for the database smart flash cache is that multiple databases on the server will be able to share this diskgroup to create multiple database smart flash caches for the different databases.

To configure Oracle ASM over two Oracle Sun Flash Accelerator PCIe Cards:

1. Oracle Grid release 11gR2 will need to be installed and configured to use ASM. For reference on installing 11gR2 Grid, go to:

   http://www.oracle.com/technetwork/database/enterprise-edition/downloads/index.html

2. Review the section: *Configure Oracle's Sun Flash Accelerator PCIe Card to use as a Filesystem* in this paper to configure both Oracle Sun Flash Accelerator PCIe Cards. When using ASM, the only steps that need to be performed in this referenced section are aligned on a 1M boundary and changing the Oracle Linux I/O scheduler to "`deadline`".

   Oracle Linux provides an Oracle ASM tool to create the ASM disks and the diskgroup.

3. The following Oracle ASM commands will create one ASM disk for each Oracle Sun Flash Accelerator PCIe Card. These commands must be run as root:

   ```
   /usr/bin/oracleasm createdisk D1 /dev/sda1
   /usr/bin/oracleasm createdisk D2 /dev/sdb1
   ```

4. Use these commands to create the diskgroup, specifying external redundancy from SQLPlus from the ASM instance:

   ```
   SQL> create diskgroup FLASH disk 'ORCL:D1', 'ORCL:D2' external
   redundancy;
   ```

Another option for creating an ASM diskgroup is to use ASMCA which uses a graphical user interface to create the diskgroup.

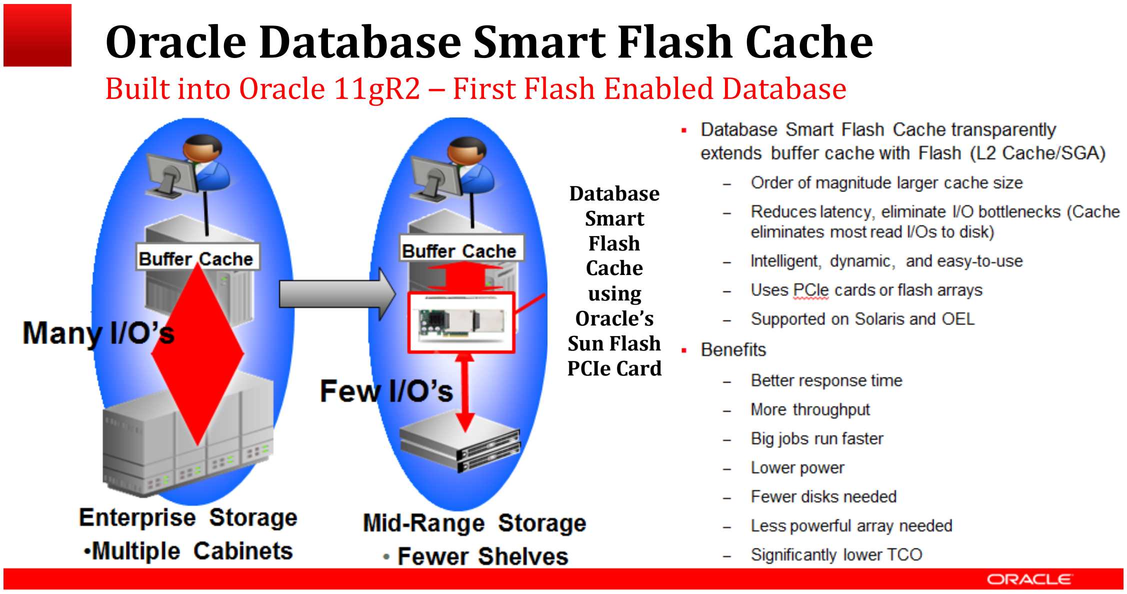
Configuring the Oracle Database to Use Database Smart Flash Cache

This section describes the changes needed to enable and configure an Oracle 11g R2 database with the Database Smart Flash Cache feature.

Oracle 11g Release 2 Database Smart Flash Cache

With Oracle Database 11g Release 2 Enterprise Edition, Oracle introduced Database Smart Flash Cache. This feature allows customers to be able to use flash devices, such as the Sun PCIe Flash Card, to increase the effective size of the Oracle database buffer cache without adding more main memory. For transaction-based workloads, Oracle database blocks are normally loaded into a dedicated shared memory area in main memory called the system global area (SGA). The Smart Flash Cache feature allows you to expand the database buffer cache beyond the SGA in main memory to a second-level cache on flash memory device(s).

This document demonstrates how you can use the Oracle Sun Flash Accelerator PCIe Card with the Smart Flash Cache feature to increase performance of an Oracle database. Figure 3 shows a system without and with Smart Flash Cache. This unique feature takes full advantage of flash by treating it as a transparent extension of the buffer cache (Level 2 cache). With this implementation, instead of the application having to wait for data from slow hard disk drives for their I/O needs, frequently accessed data is cached and sent to flash thus improving the I/O service times.



Oracle Database Smart Flash Cache

Built into Oracle 11gR2 – First Flash Enabled Database

Buffer Cache

Many I/O's

Enterprise Storage

•Multiple Cabinets

Buffer Cache

Few I/O's

Mid-Range Storage

• Fewer Shelves

Database Smart Flash Cache using Oracle's Sun Flash PCIe Card

- Database Smart Flash Cache transparently extends buffer cache with Flash (L2 Cache/SGA)
  - Order of magnitude larger cache size
  - Reduces latency, eliminate I/O bottlenecks (Cache eliminates most read I/Os to disk)
  - Intelligent, dynamic, and easy-to-use
  - Uses PCIe cards or flash arrays
  - Supported on Solaris and OEL
- Benefits
  - Better response time
  - More throughput
  - Big jobs run faster
  - Lower power
  - Fewer disks needed
  - Less powerful array needed
  - Significantly lower TCO

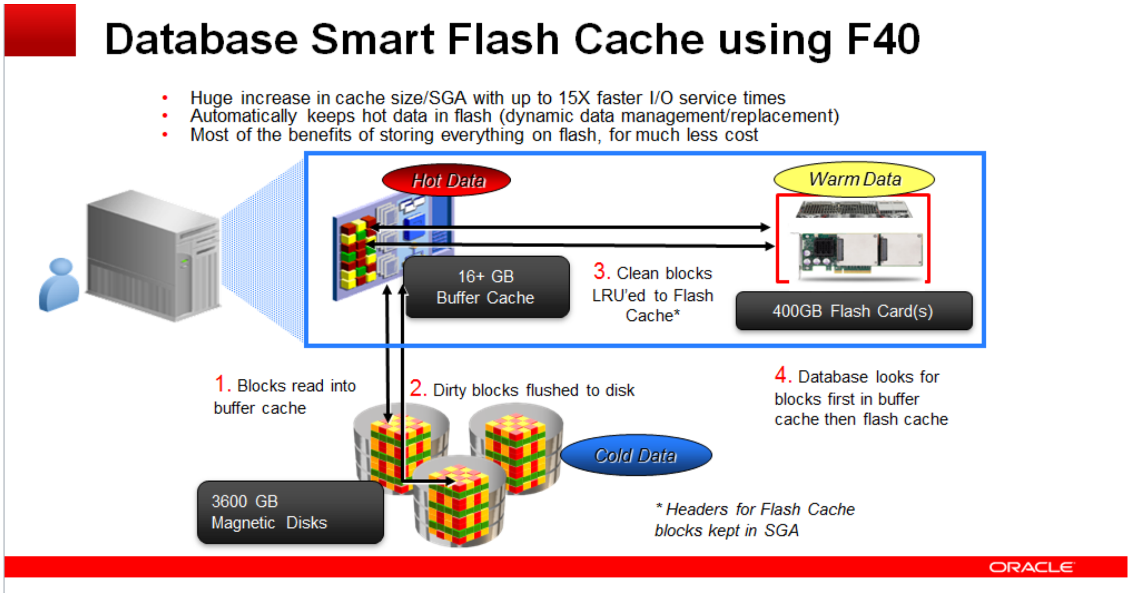
ORACLE

Figure 3 Smart Flash Cache acts as a transparent extension of buffer cache

Both OLTP and DW environments can benefit from using Oracle's Database Smart Flash Cache with Oracle's Sun Flash Accelerator PCIe Card to achieve maximum performance in new or existing Oracle database deployments with I/O intensive workloads and memory constraints. Benefits can include both improved transaction throughput and application response times. The following types of Oracle database environments can potentially make effective use of the Oracle's Database Smart Flash Cache feature and the Oracle Sun Flash Accelerator PCIe Card:

- Workloads with repeated short transactions in which many users access the same data
- Storage systems that exhibit intensive disk read activity, high latency and I/O bottlenecks

- Systems under heavy main memory pressure, which prevents more memory being allocated to the SGA buffer cache



Database Smart Flash Cache using F40

- Huge increase in cache size/SGA with up to 15X faster I/O service times
- Automatically keeps hot data in flash (dynamic data management/replacement)
- Most of the benefits of storing everything on flash, for much less cost

Hot Data

Warm Data

16+ GB Buffer Cache

3. Clean blocks LRU'ed to Flash Cache*

400GB Flash Card(s)

1. Blocks read into buffer cache

2. Dirty blocks flushed to disk

4. Database looks for blocks first in buffer cache then flash cache

Cold Data

3600 GB Magnetic Disks

** Headers for Flash Cache blocks kept in SGA*

ORACLE

Figure 4 Accelerate database performance intelligently and efficiently with Smart Flash Cache

By adding a Sun Flash Accelerator PCIe Card (or several cards) to your server, you can increase your buffer cache size to be several orders larger than your existing memory cache so more data comes directly from the solid-state Flash Cache instead of disk. While more cache is always better, depending on the workload, Flash Cache size of 4x – 10x that of the database SGA size is recommended. This will allow you to offload most of your disk I/O to much faster flash. The first I/O from the disk is also stored in the flash buffer cache. All subsequent reads from that particular row are then done on flash. This is clean (read) cache as any dirty block (writes) are flushed to disk. This provides the necessary data protection as any changes are already written to disk so no RAID or mirroring is required.

When the database application requests data I/O, it first looks in the buffer pool. If not found, the search is then performed in the Smart Flash Cache buffer. If again the data is not found, the search then goes to the disk storage. By replacing some of your disk I/O with solid-state flash IO, not only is performance and response times greatly improved, you realize much better IOPS/$, IOPS/GB, IOPS/Watt and server utilization efficiency.

Setting up the Database Smart Flash Cache is very easy and requires just a few steps to define and aggregate its size and point to its location, as per below:

- Aggregate flash cards/modules to pool:
- No mirroring needed - it is a cache!
- Set two init.ora parameters:

- Path to flash devices

  db_flash_cache_file = </filesystem > or <+FLASH/filename>
- Size of flash device

  db_flash_cache_size = <flash pool size>

While Smart Flash Cache is dynamic and very efficient by automatically migrating and evicting data as needed, the DBA has the option to pin objects/hot data into the Smart Flash Cache with the KEEP command. Normally the DBA would pin an object into the KEEP buffer pool which resides in memory. By pinning the object in the Smart Flash Cache, real memory requirements are reduced, but performance is increased since more selected objects/hot data can now be accessed directly from much faster Oracle Flash device instead of much slower disk. Syntax for pinning an object in Smart Flash Cache is:

```
alter table|index object_name storage (flash_cache keep);
```

NOTE: Currently, the Oracle Database Smart Flash Cache feature is available only on Linux and Solaris® environments.

There are multiple database patches to enable Oracle Database Smart Flash Cache. The patch for the early releases of 11gR2 is 8974084:META BUG FOR FLASH CACHE 11.2PL BUGS TO BACKPORT TO 11.2.0.1 OEL. The patch for the latest release of the 11gR2 release (11.2.0.3) that was used for these tests is 12949806: Flash Cache Check is Against Enterprise Release. After installing the patch and bouncing the database, the DB_FLASH_CACHE_FILE and DB_FLASH_CACHE_SIZE parameters were set in the database to enable this feature.

The following database settings were used in these tests when invoking Oracle Database Smart Flash Cache:

```
SQL> alter system set
db_flash_cache_file='/osfc/oradata/osfc/flash.dbf' scope=spfile;

SQL> alter system set db_flash_cache_size=175g scope=spfile;

SQL> show parameter flash
NAME                 TYPE                    VALUE
------------------------------------------------------------------
db_flash_cache_file   string         /osfc/oradata/osfc/flash.dbf
db_flash_cache_size   big integer   175G
```

These commands enabled Oracle Database Smart Flash Cache inside the database using file system `/osfc` and allocated 175 GB to this cache. The database must be bounced before it can use this cache.

To implement Oracle Database Smart Flash Cache using multiple Oracle Sun Flash Accelerator PCIe Cards, we installed and implemented ASM to create a diskgroup over multiple Oracle Sun Flash Accelerator PCIe Cards to increase the capacity of the Database Smart Flash Cache. For details, see the section in this guide titled Configure Oracle ASM Using Multiple PCIe Cards for Mirroring or for Increased Smart Flash Cache Capacity.

The following database settings were used in the benchmarks when using multiple Oracle Sun Flash Accelerator PCIe Cards and the Database Smart Flash Cache:

```
SQL> alter system set db_flash_cache_file= '+FLASH/flash.dbf'
scope=spfile;

SQL> alter system set db_flash_cache_size=250g scope=spfile;

SQL> show parameter flash

NAME                           TYPE                VALUE
-------------------------------------------------------------------
db_flash_cache_file            string              +FLASH/flash.dbf
db_flash_cache_size            big integer  250G.
```

Database Settings

The following database settings were used for all benchmarks:

- All tablespaces for the database were installed on one LUN (except the data and indexes used by Quest Benchmark Factory)
- Tablespace for Quest Benchmark Factory was created on a separate LUN
- A 175 GB Oracle Database Smart Flash Cache was allocated on one Oracle Sun Flash Accelerator PCIe Card
- SGA=5g
- filesystemio_options=SETALL
- disk_async_io=TRUE
- 4GB redo logs

Benchmark Results

Before adding the Oracle Sun Flash Accelerator PCIe Card and the Oracle Database Smart Flash cache to the test configuration, Oracle Linux with Unbreakable Enterprise Kernel and the Oracle database were configured using techniques described in the above sections. OLTP benchmarks were executed to establish a baseline before and after implementing the Oracle Database Smart Flash Cache feature. Quest's Benchmark Factory was used for these benchmarks and each transactions per second (TPS) and average response time results were documented.

Baseline Results

The results from the benchmark before enabling Oracle Database Smart Flash Cache are:

Average TPS:	58.31
Average response (in seconds):	1.242

Results with Database Smart Flash Cache Enabled

The results from the benchmark after enabling the Oracle Database Smart Flash Cache feature with one Flash F40 Card:

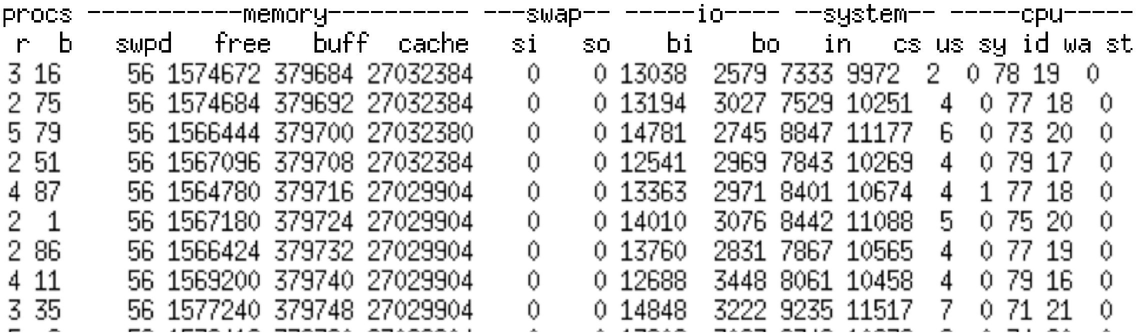
Average TPS:	71.23
Average response time (seconds):	0.543

Observations

During these benchmarks, IOSTAT, VMSTAT, and Oracle Automatic Workload Repository (AWR) reports were generated.

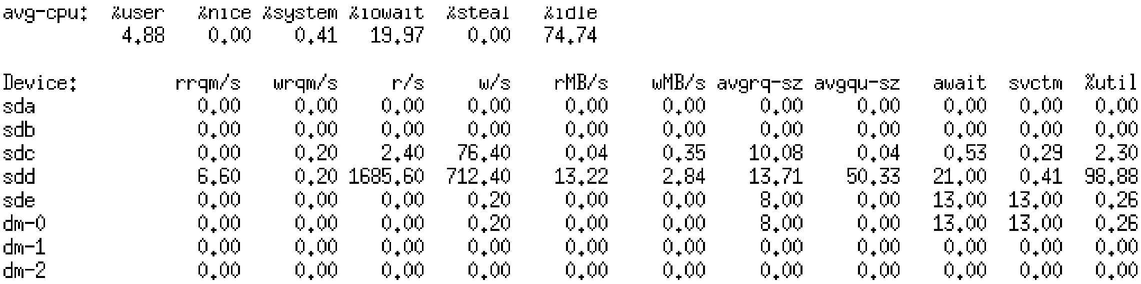
Following are the results from the baseline benchmark with hard disk drives:

VMSTAT with Hard Drives:



```
procs -----------memory---------- ---swap-- -----io---- --system-- -----cpu-----
 r  b   swpd   free   buff  cache   si   so    bi    bo   in   cs us sy id wa st
 3 16     56 1574672 379684 27032384    0    0 13038  2579 7333 9972  2  0 78 19  0
 2 75     56 1574684 379692 27032384    0    0 13194  3027 7529 10251  4  0 77 18  0
 5 79     56 1566444 379700 27032380    0    0 14781  2745 8847 11177  6  0 73 20  0
 2 51     56 1567096 379708 27032384    0    0 12541  2969 7843 10269  4  0 79 17  0
 4 87     56 1564780 379716 27029904    0    0 13363  2971 8401 10674  4  1 77 18  0
 2  1     56 1567180 379724 27029904    0    0 14010  3076 8442 11088  5  0 75 20  0
 2 86     56 1566424 379732 27029904    0    0 13760  2831 7867 10565  4  0 77 19  0
 4 11     56 1569200 379740 27029904    0    0 12688  3448 8061 10458  4  0 79 16  0
 3 35     56 1577240 379748 27029904    0    0 14848  3222 9235 11517  7  0 71 21  0
```

IOSTAT with Hard Drives:



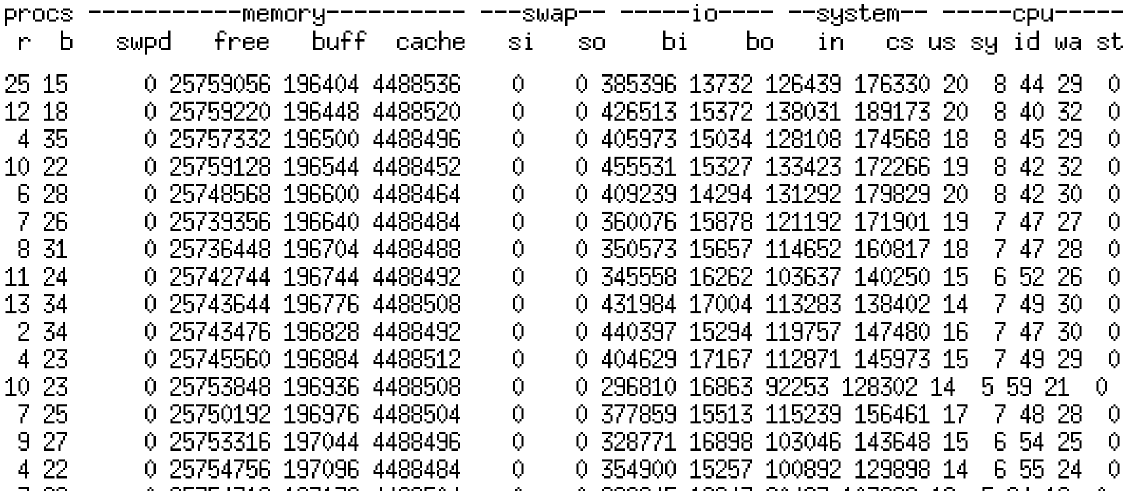
```
avg-cpu:  %user   %nice %system %iowait  %steal   %idle
           4.88    0.00    0.41   19.97    0.00   74.74

Device:         rrqm/s   wrqm/s     r/s     w/s    rMB/s    wMB/s avgrq-sz avgqu-sz   await  svctm  %util
sda               0.00     0.00    0.00    0.00     0.00     0.00     0.00     0.00    0.00   0.00   0.00
sdb               0.00     0.00    0.00    0.00     0.00     0.00     0.00     0.00    0.00   0.00   0.00
sdc               0.00     0.20    2.40   76.40     0.04     0.35    10.08     0.04    0.53   0.29   2.30
sdd               6.60     0.20 1685.60  712.40    13.22     2.84    13.71    50.33   21.00   0.41  98.88
sde               0.00     0.00    0.00    0.20     0.00     0.00     8.00     0.00   13.00  13.00   0.26
dm-0              0.00     0.00    0.00    0.20     0.00     0.00     8.00     0.00   13.00  13.00   0.26
dm-1              0.00     0.00    0.00    0.00     0.00     0.00     0.00     0.00    0.00   0.00   0.00
dm-2              0.00     0.00    0.00    0.00     0.00     0.00     0.00     0.00    0.00   0.00   0.00
```

AWR Top 5 Timed Foreground Events with Hard Drives:

Event	Waits	Time(s)	Avg wait (ms)	% DB Time	Wait Class
db file sequential read	1,889,017	119,533	63	97.1	User I/O
db file parallel read	8,249	1,597	194	1.3	User I/O
DB CPU		1,508		1.23	
read by other session	15,544	796	51	0.65	User I/O
log file sync	57,122	30	0.525	0.02	Commit

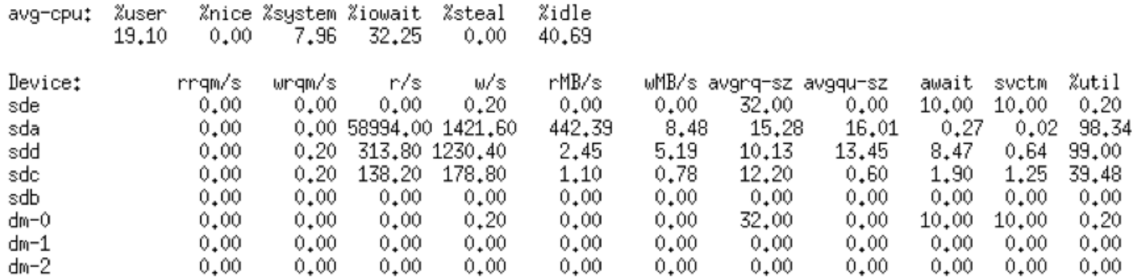
The following results are from the benchmark using the Oracle Database Smart Flash Cache feature with the Oracle Sun Flash Accelerator PCIe Card:

VMSTAT with Flash Accelerator Card:



```
procs -----------memory---------- ---swap-- -----io---- --system-- -----cpu-----
 r  b   swpd   free   buff  cache   si   so    bi    bo   in   cs us sy id wa st

25 15      0 25759056 196404 4488536    0    0 385396 13732 126439 176330 20  8 44 29  0
12 18      0 25759220 196448 4488520    0    0 426513 15372 138031 189173 20  8 40 32  0
 4 35      0 25757332 196500 4488496    0    0 405973 15034 128108 174568 18  8 45 29  0
10 22      0 25759128 196544 4488452    0    0 455531 15327 133423 172266 19  8 42 32  0
 6 28      0 25748568 196600 4488464    0    0 409239 14294 131292 179829 20  8 42 30  0
 7 26      0 25739356 196640 4488484    0    0 360076 15878 121192 171901 19  7 47 27  0
 8 31      0 25736448 196704 4488488    0    0 350573 15657 114652 160817 18  7 47 28  0
11 24      0 25742744 196744 4488492    0    0 345558 16262 103637 140250 15  6 52 26  0
13 34      0 25743644 196776 4488508    0    0 431984 17004 113283 138402 14  7 49 30  0
 2 34      0 25743476 196828 4488492    0    0 440397 15294 119757 147480 16  7 47 30  0
 4 23      0 25745560 196884 4488512    0    0 404629 17167 112871 145973 15  7 49 29  0
10 23      0 25753848 196936 4488508    0    0 296810 16863 92253 128302 14  5 59 21  0
 7 25      0 25750192 196976 4488504    0    0 377859 15513 115239 156461 17  7 48 28  0
 9 27      0 25753316 197044 4488496    0    0 328771 16898 103046 143648 15  6 54 25  0
 4 22      0 25754756 197096 4488484    0    0 354900 15257 100892 129898 14  6 55 24  0
```

IOSTAT with Flash Accelerator Card:



```
avg-cpu:  %user   %nice %system %iowait  %steal   %idle
          19.10    0.00    7.96   32.25    0.00   40.69

Device:         rrqm/s   wrqm/s     r/s     w/s    rMB/s    wMB/s avgrq-sz avgqu-sz   await  svctm  %util
sde               0.00     0.00    0.00    0.20     0.00     0.00    32.00     0.00   10.00  10.00   0.20
sda               0.00     0.00 58994.00 1421.60   442.39     8.48    15.28    16.01    0.27   0.02  98.34
sdd               0.00     0.20  313.80 1230.40     2.45     5.19    10.13    13.45    8.47   0.64  99.00
sdc               0.00     0.20  138.20  178.80     1.10     0.78    12.20     0.60    1.90   1.25  39.48
sdb               0.00     0.00    0.00    0.00     0.00     0.00     0.00     0.00    0.00   0.00   0.00
dm-0              0.00     0.00    0.00    0.20     0.00     0.00    32.00     0.00   10.00  10.00   0.20
dm-1              0.00     0.00    0.00    0.00     0.00     0.00     0.00     0.00    0.00   0.00   0.00
dm-2              0.00     0.00    0.00    0.00     0.00     0.00     0.00     0.00    0.00   0.00   0.00
```

AWR Top 5 Timed Foreground Events with Flash Accelerator Card:

Event	Waits	Time(s)	Avg wait (ms)	% DB Time	Wait Class
db flash cache single block phys read	21,467,471	7,556	0.352	43.14	User I/O
read by other session	14,492,743	5,082	0.351	29.01	User I/O
db file sequential read	214,400	2,877	13	16.42	User I/O
DB CPU		2,534		14.47	
latch: cache buffers chains	343,567	32	0.931	0.18	Concurrency

Comparison of Benchmark Results

Several important statistics exists that are important to evaluate between the two benchmarks:

VMSTAT

Block column statistic, which indicates if system resources are waiting on I/O.

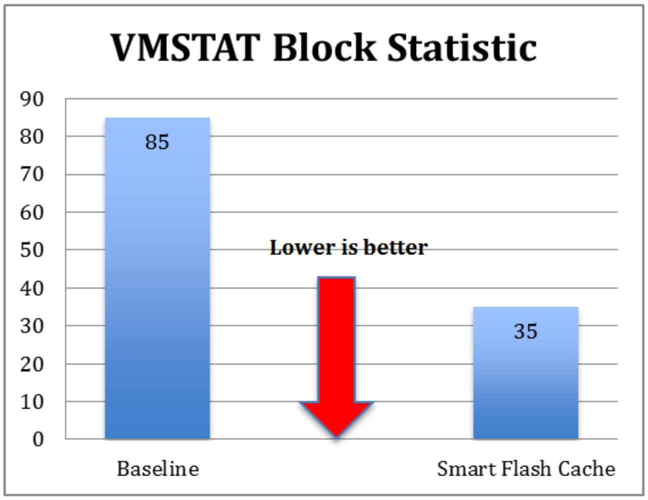


Figure 5 Kernel wait threads on I/O reduced from 85 to < 35 on averag

IOSTAT

Columns “r/s” (reads per second) and “await” (average wait time).

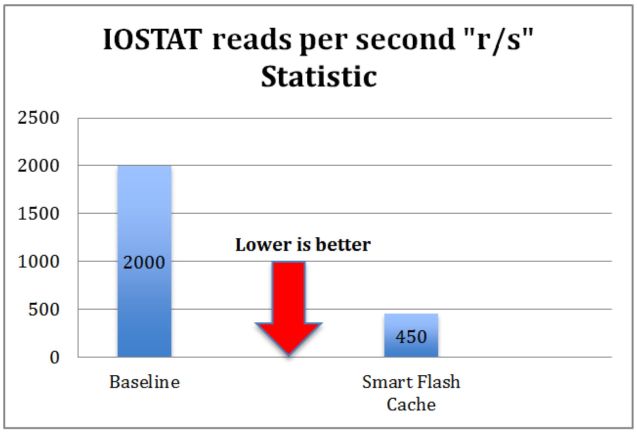


Figure 6 Reads per second drastically reduced as physical reads come from Smart Flash Cache instead of hard disk

IOSTAT

Average Wait Time

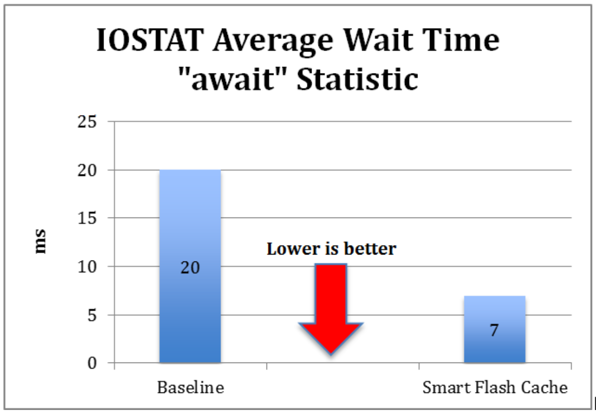
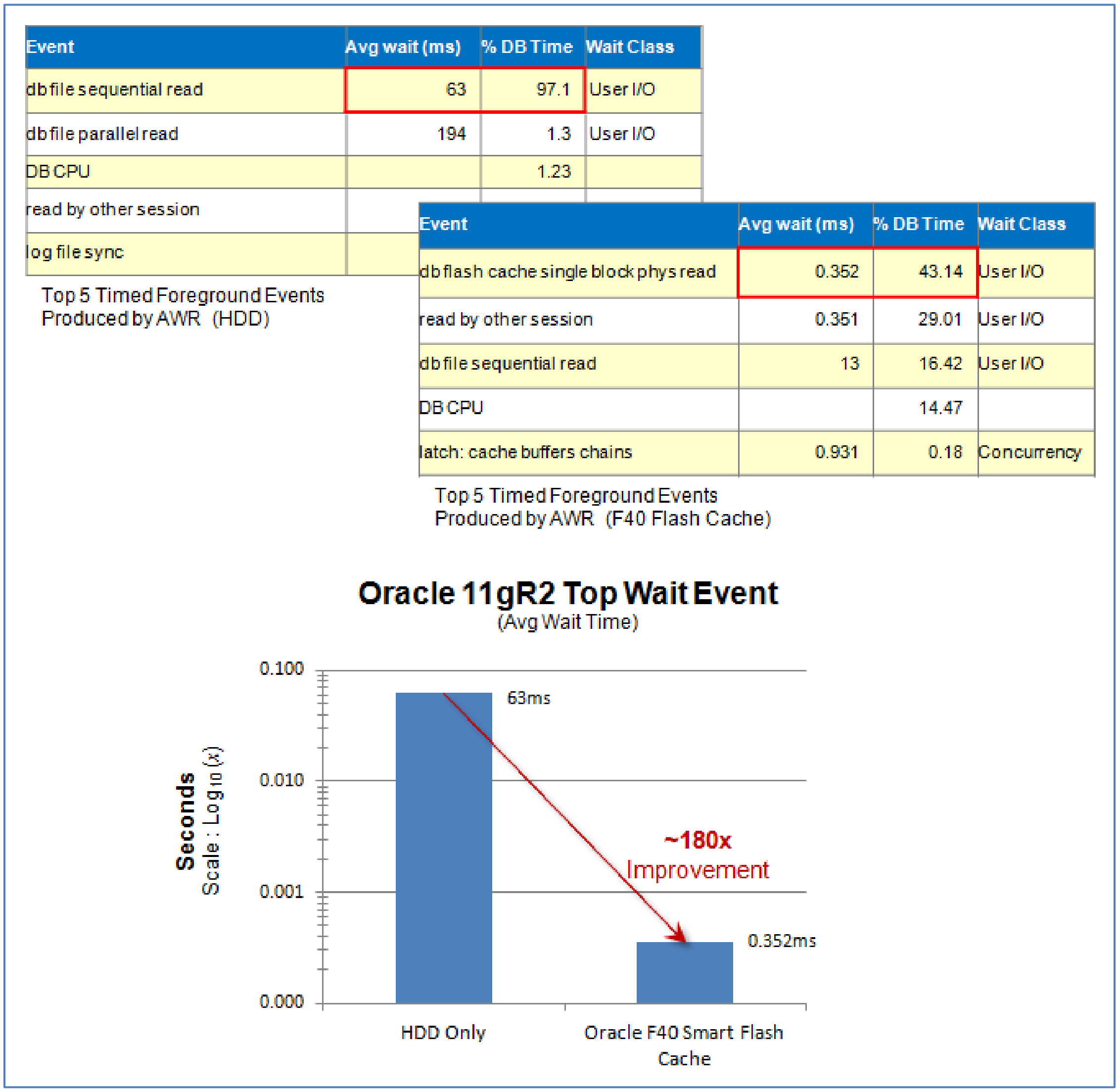


Figure 7 Average wait time with disk drops from 20ms to 7ms

Automatic Workload Repository (AWR)

Reviewing the AWR top five wait events will help one determine the effectiveness of adding PCIe flash. If the “db file sequential reads” are a large percentage of total wait times or averaging greater than 5 ms, that’s an indication of storage related I/O issues; this is an area where Smart Flash Cache and the Oracle Sun Flash Accelerator PCIe Card can help alleviate these bottlenecks and improve response times.

Referring to Figure 8 the “db file sequential read” event was the top wait event during the baseline benchmark with an average of 63ms for each wait. This same event went from number 1 in the baseline to the 3rd ranking wait event with database smart flash cache enabled. The “db file sequential reads” event wait time dropped to an average of 13ms per wait, almost 5x improvement in wait times.



Event	Avg wait (ms)	% DB Time	Wait Class
db file sequential read	63	97.1	User I/O
db file parallel read	194	1.3	User I/O
DB CPU		1.23	
read by other session			
log file sync			

Top 5 Timed Foreground Events Produced by AWR (HDD)

Event	Avg wait (ms)	% DB Time	Wait Class
db flash cache single block phys read	0.352	43.14	User I/O
read by other session	0.351	29.01	User I/O
db file sequential read	13	16.42	User I/O
DB CPU		14.47	
latch: cache buffers chains	0.931	0.18	Concurrency

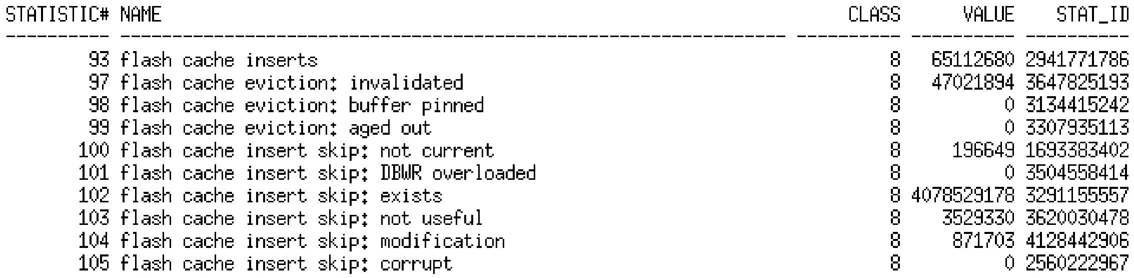
Top 5 Timed Foreground Events Produced by AWR (F40 Flash Cache)

Figure 8 Test Results—HDD vs. Oracle F40 Smart Flash Cache

The new “db flash cache single block physical read” wait event became #1 in the top 5 wait events. The average wait time to read a single data block was 35µs (microseconds) compared to 63ms (milliseconds) without Smart Flash Cache enabled, or nearly 180x improvement in wait time.

Smart Flash Cache Statistics

```
select * from v$sysstat where name like 'flash cache%';
```



```
STATISTIC# NAME                                                                  CLASS      VALUE    STAT_ID
---------- ---------------------------------------------------------------- ---------- ---------- ----------
        93 flash cache inserts                                                       8   65112680 2941771786
        97 flash cache eviction: invalidated                                         8   47021894 3647825193
        98 flash cache eviction: buffer pinned                                       8          0 3134415242
        99 flash cache eviction: aged out                                            8          0 3307935113
       100 flash cache insert skip: not current                                      8     196649 1693383402
       101 flash cache insert skip: DBWR overloaded                                  8          0 3504558414
       102 flash cache insert skip: exists                                           8 4078529178 3291155557
       103 flash cache insert skip: not useful                                       8    3529330 3620030478
       104 flash cache insert skip: modification                                     8     871703 4128442906
       105 flash cache insert skip: corrupt                                          8          0 2560222967
```

Descriptions for Smart Flash Cache statistics:

flash cache eviction: aged out	Flash cache buffer is aged out of the flash cache
flash cache eviction: buffer pinned	Flash cache buffer is invalidated due to object or range reuse, and so on. The flash cache buffer was in use at the time of eviction.
flash cache eviction: invalidated	Flash cache buffer is invalidated due to object or range reuse, and so on. The flash cache buffer was not in use at the time of eviction.
flash cache insert skip: corrupt	In-memory buffer was skipped for insertion into flash cache because the buffer was corrupted
flash cache insert skip: DBWR overloaded	In-memory buffer was skipped for insertion into flash cache because DBWR was busy writing other buffers
flash cache insert skip: exists	In-memory buffer was skipped for insertion into flash cache because it was already in the flash cache
flash cache insert skip: modification	In-memory buffer was skipped for insertion into flash cache because it was being modified
flash cache insert skip: not current	In-memory buffer was skipped for insertion into flash cache because it was not current
flash cache insert skip: not useful	In-memory buffer was skipped for insertion into flash cache because the type of buffer was not useful to keep
flash cache inserts	Total number of in-memory buffers inserted into flash cache

From the Smart Flash Cache statistics taken from the benchmarks, the two outstanding statistics that were important to describe and useful to evaluate if the Smart Flash Cache is working effectively:

- **Flash Cache inserts:** Number of blocks written to Smart Flash Cache
- **Flash cache insert skip: exists:** If a block is read from the Smart Flash Cache into the buffer cache, it then remains in the Smart Flash Cache. There is no need to write the block back into the Smart Flash Cache.

Conclusion

Flash accelerates applications, increases productivity and improves business responsiveness. Based upon the benchmarks that were executed for this best practices guide using an Oracle Sun Flash Accelerator PCIe Card and Oracle's Database Smart Flash Cache feature running on Oracle Linux with Unbreakable Enterprise Kernel, large performance gains were realized. Whether running an Oracle Database or other I/O intensive applications, similar performance gains and improved response times can be realized in the enterprise using the configuration presented in this guide for workloads that are:

- Disk bound
- Mostly read or read-only
- I/O bound by large number of disk IOPS.

As a side benefit, by implementing Oracle's Database Smart Flash Cache feature along with the Oracle Sun Flash Accelerator PCIe Card, not only will this combination reduce the hard disk IOPS for reads resulting in large performance gains, this reduction in IOPS for reads will result in the capability to perform physical writes with less latency.

These are significant benefits for customers running large databases. Oracle's Database Smart Flash Cache feature with the Oracle Sun Flash Accelerator PCIe Card and Oracle Linux with Unbreakable Enterprise Kernel or Oracle Solaris provide a platform that can scale and perform to the demanding needs of growing enterprises.

Resources

- Oracle's Sun Flash Accelerator F40 PCIe Card
- Oracle Database 11g
- Oracle Linux with Unbreakable Enterprise Kernel

Optimizing Oracle Database Performance on
Oracle Linux with Flash
September 2014
Contributing Authors: Rick Stehno—Database Strategist–Seagate Technologies, Phillip Goerl—Oracle, John Szlendak—Oracle

Oracle Corporation
World Headquarters
500 Oracle Parkway
Redwood Shores, CA 94065
U.S.A.
Worldwide Inquiries:
Phone: +1.650.506.7000
Fax: +1.650.506.7200
oracle.com

This document is provided for information purposes only, and the contents hereof are subject to change without notice. This document is not warranted to be error-free, nor subject to any other warranties or conditions, whether expressed orally or implied in law, including implied warranties and conditions of merchantability or fitness for a particular purpose. We specifically disclaim any liability with respect to this document, and no contractual obligations are formed either directly or indirectly by this document. This document may not be reproduced or transmitted in any form or by any means, electronic or mechanical, for any purpose, without our prior written permission.