Accelerating Dell Oracle 11g R2 Database Solutions Using PCIe Solid State Storage

A Dell Technical White Paper

Dell │ Database Solutions Engineering

Zafar Mahmood

August 2010

Executive Summary

The performance gap between server processing and the storage technology has been widening. Dell's latest 11G server product line is being shipped with multi-core Nehalem EP processors which provide exceptional processing power. On the other hand, storage technology traditionally has lagged behind and database performance bottleneck is increasingly shifting towards the storage. Although the storage interconnect technology has been evolving over the years in all categories like FC, iSCSI, SAS and SATA, the backend disk rotational speed has topped at 15K due to physical and mechanical limitations. Solid State Storage (SSS) promises to narrow this gap between server and storage performance. There are many different types of Solid State Storage (SSS) devices available in the market today. The SSS can either be made available to the servers via the traditional FC, SAS, iSCSI interfaces or placed directly into the server PCIe slot without requiring traditional storage network interfaces. Since PCIe based SSS device sit directly in the server PCIe bus infrastructure, it promises the best SSS performance available to the database customers running applications which require sub-second query response times. With the addition of the Fusion-io's family of products, Dell™ now even better enables customers with the speed of solid state storage technology for improved response time for I/O intensive database applications. In this white paper we will discuss various use cases of Dell's Fusion-io PCIe-SSS based solution with Oracle 11gR2 single node and Real Application Cluster databases.

THIS WHITE PAPER IS FOR INFORMATIONAL PURPOSES ONLY, AND MAY CONTAIN TYPOGRAPHICAL ERRORS AND TECHNICAL INACCURACIES. THE CONTENT IS PROVIDED AS IS, WITHOUT EXPRESS OR IMPLIED WARRANTIES OF ANY KIND.

August 2010

Contents

Tables

Figures

Introduction

Solid State Storage (SSS) is a data storage device that uses solid state memory to store persistent data (No electro-mechanical parts). There are two types of Solid State Storage devices available in the market today.

DRAM SSS

DRAM SSS is based on volatile memory such as DRAM. These are characterized by ultrafast data access and latency in the range of only tens of microseconds and are used primarily to accelerate applications which are held back in performance by the traditional hard disk drives and even flash-based solid state storage. Following are some of the characteristics of DRAM SSS:

- Lowest latency media
- More expensive per GB
- SSS based on volatile memory
- Incorporates internal battery and backup storage systems
- Extremely fast read/writes
- Negligible seek time, noiseless (no moving parts)
- Higher power consumption than flash SSS and hard disks (both operating and when turned off)
- No limitations on number of write cycles

NAND Flash SSS

Today NAND Flash SSS is the more prevalent form of the solid state storage since it is more cost effective and does not require backup batteries to preserve data. In addition, non-volatility allows flash SSS to retain memory even during sudden power outages, ensuring data persistence. Following are some of the characteristics of NAND Flash SSS:

- Cost effective
- Non-volatile, high density, low power
- Slower than DRAM, but at least 10x faster than hard disk drives
- Negligible seek time, noiseless (no moving parts)
- Extremely fast reads (slower writes as compared to DRAM SSS)
- Low power consumption
- Limited write (erase) cycles—flash-memory cells will often wear out after 1,000 to 10,000 write cycles for MLC, and up to 100,000 write cycles for SLC (have to use techniques such as wear leveling)
- Slower write speeds (specially the large sequential)

These are available in various form factors and interfaces such as FC, SAS and SATA, and some which sit directly in the PCIe bus such as Fusion-io SSS modules. The Solid State Storage which utilizes traditional storage interface presents SSS storage as another fast hard disk drive. True performance is restricted due to the following bottlenecks:

- Array's external FC, SAS, iSCSI or SATA interface is bridged to the PCIe bus via the HBA's and traditional storage drivers spoofing SSS as another hard disk drive.

- Disk hardware only capable of reading/writing one location at a time; SSS can be reading/writing many places at once. So, spoofing SSS as a fast hard disk drive does not fully take advantage of its capabilities.
- Hard disk drives hold multiple IO requests in a queue to process while SSS can process requests in parallel requiring specialized drivers to take full advantage of the capabilities of SSS.
- Operating System's IO subsystem has layers that add latency and limits parallelism since it was designed specifically for rotational media. There is need for specialized drivers to obtain optimal performance out of SSS devices. The optimized IO subsystem would ideally bypass SCSI, ATA layers to reduce CPU overhead, bypass request queue layers, and eliminate disk oriented optimizations.

In this white paper we will concentrate on Fusion-io NAND flash SSS. Fusion-io SSS modules present a server-centric view of the Solid State Storage, while FC, SAS, and SATA interface SSS present the storage-array-centric view. In the server-centric approach, SSS may be utilized as either a second level caching deice below RAM, or as actual storage media for non clustered database solutions since shared storage database solutions such as Oracle 11g RAC require the storage to be available to all cluster nodes as shared. Here is a comparison of associated overheads for the server-centric or PCIe bus oriented and storage centric SSS solutions:

	Server Centric SSS	Storage Centric SSS
Latency	PCIe Bus overhead only	PCIe Bus overhead + PCIe Bus to HBA + HBA to SAN switch + SAN switch to storage controller Bus + storage controller processing stack + storage controller bus to backend storage controller adapter + storage controller adapter to SSS
IOPS	No inherent limitation	No inherent limitation
Bandwidth	Single PCIe bus performance	Storage controller/Interface performance

As seen from the above table, latency is one of the main performance factors at which PCIe based SSS excels. Low latency results and reduced response time have been the most important reasons for SSS adoption for database solutions. The reduced response times basically results in lowering the time that a database application spends waiting for storage to return query results, resulting in application acceleration. So, it is a perfect fit for high transaction database architectures where low latency is sought.

Write Performance Considerations

Flash SSS by design provide the maximum performance for workloads which are either read only or mostly reads. The reason is that although it is able to read data at page level in a random access pattern, the write operations are performed at the block level each of which is comprised of 64x4K pages. The write performance is dependent upon the availability of the free programmable data blocks. Initially, an SSS device has all data blocks available (all bits set to 1) to be programmed or written. Once a page within the block has some bits cleared to '0' or programmed, however, an update requires the whole block be erased and programmed with few exceptions, and this is an expensive operation. For this reason, NAND Flash SSS may not perform optimally for application workloads comprising of mostly small random write operations. The performance of a NAND flash device reaches steady state level after sustained usage, and any performance characterization effort should take that anomaly into consideration.

Fusion-io PCIe-SSS

Fusion-io's family of products are based on a new solid state technology that increases bandwidth and application performance, reduces latency, and simplifies IT infrastructure. These storage devices integrate with servers at the system bus and kernel level, creating a new flash memory tier that result in database application acceleration, improved response times, and better efficiency. They also reduce storage latencies and eliminate I/O bottlenecks by delivering the performance of thousands of disk drives in a single server, as we will see in the next sections.

NAND Flash: Multi level Cell (MLC) Vs Single Level Cell (SLC)

There are two types of NAND Flash based PCIe storage modules available: 160GB IO Drive (SLC) and 640GB IO Duo Drive (MLC). MLC NAND SSS devices can store more data per cell since each cell operates with 4 voltage levels (2 bits/cell) compared to SLC which operates with only two voltage levels (1 bit/cell). MLC also has more pages per block, resulting in higher storage capacities. The drawback is that MLC chips have lower write/erase cycles as compared to SLC resulting in lower endurance. With improving wear leveling algorithms which basically convert physical page to logical LBA, the writes are spread across varying cells each time, resulting in improved endurance. For enterprise datacenters, NAND Flash based on SLC technology is typically used since they provide higher write endurance and performance. However, depending on the application workload, in many cases MLC based NAND SSS can be more cost effective while providing the required level of performance and endurance for certain database workloads. The workloads which are either read only or mostly reads and require larger storage capacities equivalent to the hard disk drives can take advantage of the MLC SSS devices as Fusion IO Duo Drive (MLC).

Configuration and Setup

In this white paper we used Oracle® 11gR2 RAC and single node configurations with Dell™ EqualLogic™ 6010XV storage in conjunction with Dell Fusion-io PCIe SSS devices to build an optimal configuration best suited for applications requiring low latency data access. In this section we will go over the setup and configuration steps to build such an infrastructure.

1. Perform a basic install of the Oracle Enterprise Linux operating system

2. Install the Oracle Validated RPM which can be found on the Oracle Enterprise Linux CD's. Installing the Oracle validated RPM will result in an Operating System install which contains all the prerequisite RPM's necessary to install and configure an Oracle 11gR2 RAC or single node database.

At the minimum, the following RPM's are required for a successful Oracle 11gR2 setup on OEL 5.5:

```
Name                                              Version             Release
------------------------------------------------  ------------------  ---------------------------
compat-db                                                4.2.52                5.1
compat-gcc-34                                           3.4.6                 4
compat-gcc-34-c++                                       3.4.6                 4
```

compat-libstdc++-33	3.2.3	61
compat-libstdc++-33	3.2.3	61
elfutils-libelf-devel	0.137	3.el5
gcc-c++	4.1.2	48.el5
gdb	7.0.1	23.el5_5.1
libXp	1.0.0	8.1.el5
libaio-devel	0.3.106	5
libaio-devel	0.3.106	5
libstdc++-devel	4.1.2	48.el5
sysstat	7.0.2	3.el5
unixODBC	2.2.11	7.1
unixODBC	2.2.11	7.1
unixODBC-devel	2.2.11	7.1
unixODBC-devel	2.2.11	7.1
elfutils-libelf-devel-static	0.137	3.el5

3. Next, create the required Oracle users and groups to host both the Grid Infrastructure home for an Oracle Real Applications Clusters deployment and the Oracle home for the Oracle software binaries.
4. For an Oracle 11gR2 deployment, which now comes with its own Cluster Time Synchronization Service (CTSS) disable the OS NTP services.

```
service ntpd stop

chkconfig ntpd off

mv /etc/ntp.conf /etc/ntp.conf.orig

rm /var/run/ntpd.pid
```

5. Make sure that both the Grid Infrastructure user and Oracle users have sufficient resource limits defined to be able to spawn Oracle processes.
   i. Edit /etc/security/limits.conf with the following settings:

```
oracle    soft    nofile     131072
oracle    hard    nofile     131072
oracle    soft    nproc      131072
oracle    hard    nproc      131072
oracle    soft    core       unlimited
oracle    hard    core       unlimited
oracle    soft    memlock    50000000
oracle    hard    memlock    50000000
grid      soft    nproc      2047
grid      hard    nproc      16384
grid      soft    nofile     1024
grid      hard    nofile     65536
```

   ii. On each node, add or edit the following line in the /etc/pam.d/login file, if it does not already exist:

```
session    required     pam_limits.so
```

iii. Edit the /etc/profile file and add the following entry:

```
if [ $USER = "oracle" ] || [ $USER = "grid" ]; then
                  if [ $SHELL = "/bin/ksh" ]; then
                        ulimit -p 16384
                        ulimit -n 65536
                  else
                        ulimit -u 16384 -n 65536
                  fi
            umask 022
fi
```

6. Oracle 11gR2 can be configured with its new Grid Naming service (GNS) feature. Although not a requirement but it is very useful when it comes to name resolution for numerous RAC service names with underlying virtual IP addresses such as SCAN listeners. It would be a good idea at this point to configure GNS dependencies. For detailed information on this topic, please refere to Dell Tested and Validated configurations documentation available at www.dell.com/oracle. For the purpose of this white paper we used GNS and configured the domain name server to resolve the GNS VIP address which will be used for our configuration later.
7. Configure the shared storage for RAC using Oracle ASM Library driver:

   We configured the following disks for 11gR2 RAC database:

```
[root@zfusionr810 ]# service oracleasm listdisks
```

Diskgroup	**Usage**
DATA1	Oracle datafiles for seed database
DATA2	OLTP read intensive workload datafiles and indexes
OCRVOTE	Oracle Grid Infrastructure files(OCR, Vote)
ORAHOME	Oracle ASM Cluster File System volume for Oracle Home

8. Install Oracle Grid Infrastructure Home and verify that all the GI resources are up and running by issuing the following command:

```
[root@zfusionr810 ~]# /opt/app/11.2.0/grid/bin/crsctl stat res –t
```

```
NAME                           TARGET  STATE         SERVER          STATE_DETAILS
--------------------------------------------------------------------------------
Local Resources
--------------------------------------------------------------------------------
ora.LISTENER.lsnr
                               ONLINE  ONLINE        zfusionr710
                                ONLINE  ONLINE       zfusionr810
ora.OCRVOTE.dg
                               ONLINE  ONLINE        zfusionr710
                               ONLINE  ONLINE        zfusionr810
ora.asm
                               ONLINE  ONLINE        zfusionr710      Started
                               ONLINE  ONLINE        zfusionr810      Started
ora.eons
                               ONLINE  ONLINE        zfusionr710
                               ONLINE  ONLINE        zfusionr810
ora.gsd
                               OFFLINE OFFLINE       zfusionr710
                                OFFLINE OFFLINE      zfusionr810
ora.net1.network
                               ONLINE  ONLINE        zfusionr710
                               ONLINE  ONLINE        zfusionr810
ora.ons
                               ONLINE  ONLINE        zfusionr710
                               ONLINE  ONLINE        zfusionr810
ora.registry.acfs
                               ONLINE  ONLINE        zfusionr710
                               ONLINE  ONLINE        zfusionr810
-------------------------------------------------------------------
Cluster Resources
-------------------------------------------------------------------
ora.LISTENER_SCAN1.lsnr        1     ONLINE  ONLINE     zfusionr710
ora.LISTENER_SCAN2.lsnr        1     ONLINE  ONLINE     zfusionr810
ora.LISTENER_SCAN3.lsnr        1     ONLINE  ONLINE     zfusionr810
ora.gns                        1     ONLINE  ONLINE     zfusionr810
ora.gns.vip                    1     ONLINE  ONLINE     zfusionr810
ora.oc4j                       1     OFFLINE OFFLINE
ora.scan1.vip                  1     ONLINE  ONLINE     zfusionr710
ora.scan2.vip                  1     ONLINE  ONLINE     zfusionr810
ora.scan3.vip                  1     ONLINE  ONLINE     zfusionr810
ora.zfusionr710.vip            1     ONLINE  ONLINE     zfusionr710
ora.zfusionr810.vip            1     ONLINE  ONLINE     zfusionr810
```

9. Configuring ASM Cluster File System for Oracle Home

Using the ASM Configuration Assistant ASMCA, configure a volume and file system to host the shared Oracle Home if using RAC. This is optional but it makes patch installation simpler and also provides the capability of snapshots of the volume which can be used to revert to rollback the undesirable effects of software changes.

Make sure that the ACFS volume and diskgroup is in online state on all nodes before proceeding to install Oracle software binaries as shown below:

```
ora.ACFSORAHOME.dg
                                       ONLINE  ONLINE     zfusionr710
                                       ONLINE  ONLINE     zfusionr810
ora.acfsorahome.acfsorahome.acfs
                                       ONLINE  ONLINE     zfusionr710
                                       ONLINE  ONLINE     zfusionr810
```

10. Install Oracle database software in the Oracle Home
11. Install the patch 8974084 to enable Flash Cache for Database in ORACLE HOME using OPatch. The patch is available on Oracle metalink support website.
12. At this point create the seed Oracle database. For the purposes of this white paper, we configured the database with 10GB of memory (memory_target=10GB). After DB creation, one can configure the smart flash cache for Oracle database as discussed in the next section.

Installing The Fusion-io driver

The kernel development and header packages and rpm-build are required to build the Fusion-io drivers from the source. For detailed information about building and installing the Fusion-io drivers, please refer to the documentation available at www.fusionio.com.

After driver installation, check the status of the Fusion-io devices as follows:

```
[root@zfusionr810 fusion]# fio-status
```

The resulting information will display the detailed status of the Fusion IoDrive modules including the device names, driver versions and firmware versions. The 'fio-status' utility also shows a health indicator that starts at 100 and counts down to 0. It is a best practice to regularly check the health status of the NAND Flash device to make sure health thresholds are not crossed after long term write intensive usage.

Table 1. Tested Hardware and Software Configurations

Servers	Backend Database Storage	PCIe SSS	Maximum number of SSS Cards supported	Operating System	Oracle Database Version
R710 R810 M610x	Dell Equallogic PS6010XV	640GB ioDrive duo (MLC) 160GB ioDrive (SLC)	2	Oracle Enterprise Linux 5.5 RedHat Enterprise Linux 5.5 (Flash Cache not supported)	Oracle 11gR2 10.2.0.1 with patch 8974084

PCIe SSS Storage Use cases

The Fusion IO flash devices may be used to accelerate an Oracle database query performance according to the following use cases:

1. Placing all Oracle database files onto the PCIe Flash storage for non-clustered databases using mirrored Fusion-io Duo SSS cards (Dual 160GB Fusion-io Drive modules or two Fusion-io ioDrive Duo modules required based on database size requirements).

2. Hybrid configuration: Placing the IO intensive Oracle objects onto PCIe Flash storage for non-clustered databases using dual mirrored 160GB Fusion-io Drive SLC or 640Gb Fusion-io ioDrive MLC modules.

3. Using a single Fusion-io ioDrive 160GB SLC or 640GB Duo card for flash cache for non-clustered databases.

4. Using a single Fusion-io ioDrive 160GB SLC card per Oracle RAC node for flash cache in a clustered database environment.

Among the above mentioned use cases, we will discuss the performance impact of placing the IO intensive database objects onto the Fusion IO 160GB SLC modules for single node databases (use case 2) as well as the flash cache option for a RAC database (use case 4). Use case 1 and 3 can be covered indirectly by exploring the use cases 2 and 4.

NOTE: Software RAID utility ‘mdadm’ available on Linux operating system may be used to configure RAID1 for use cases 1 and 2. For detailed instructions on how to configure software RAID on Fusion-io devices, please refer to the IoDrive Product Family User Guide for Linux which is also available online at www.fusionio.com.

Use Case 2

As shown in the figure below, the Hybrid SSS storage solution results in significant Query response time reduction while preserving the original storage configuration and with minimal interruption in service. In this case only the indexes and the temporary tablespaces of a typical OLTP workload were moved to the Fusion ioDrive modules. To make a more analytic decision, one can also move the additional hot objects to the Fusion-io ioDrive modules based on AWR report pointing to the database objects showing higher than normal IO operations per second and latency. The candidate objects should not only exhibit high IO rates and latency but also the size, which should easily fit into the available storage space of the Fusion-io module being used to obtain optimal cost to performance ratio.

Figure 1. Use Case 2 (Hybrid Configuration)

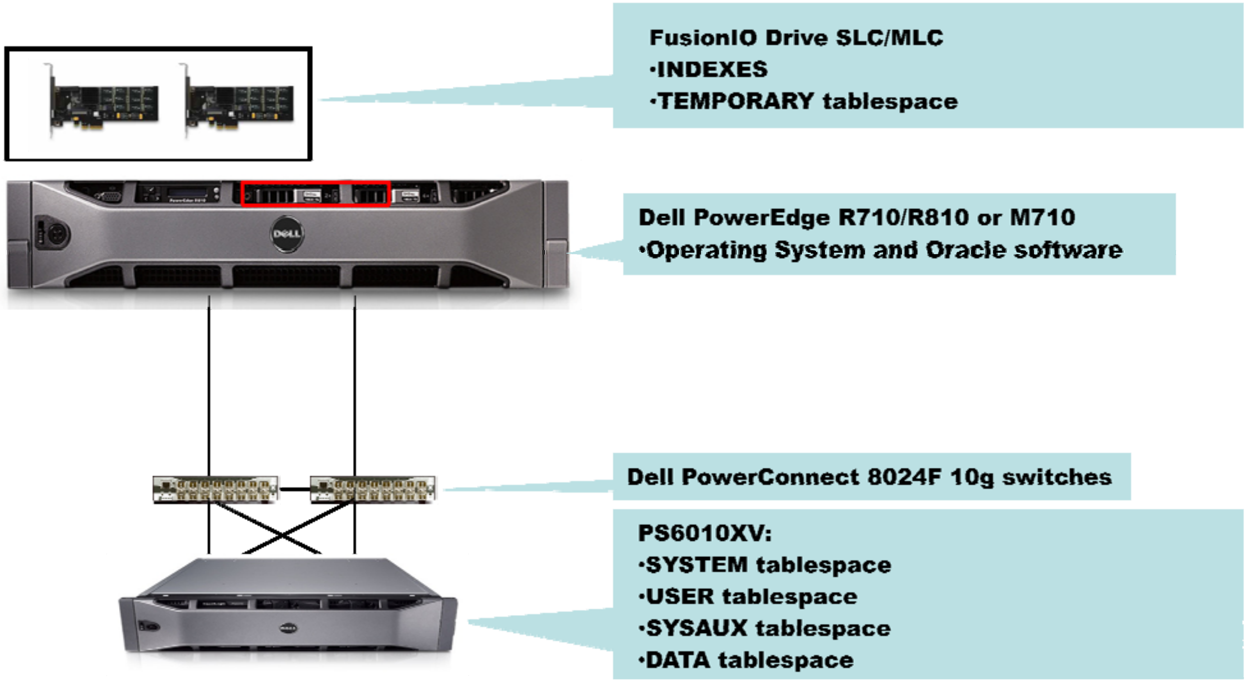
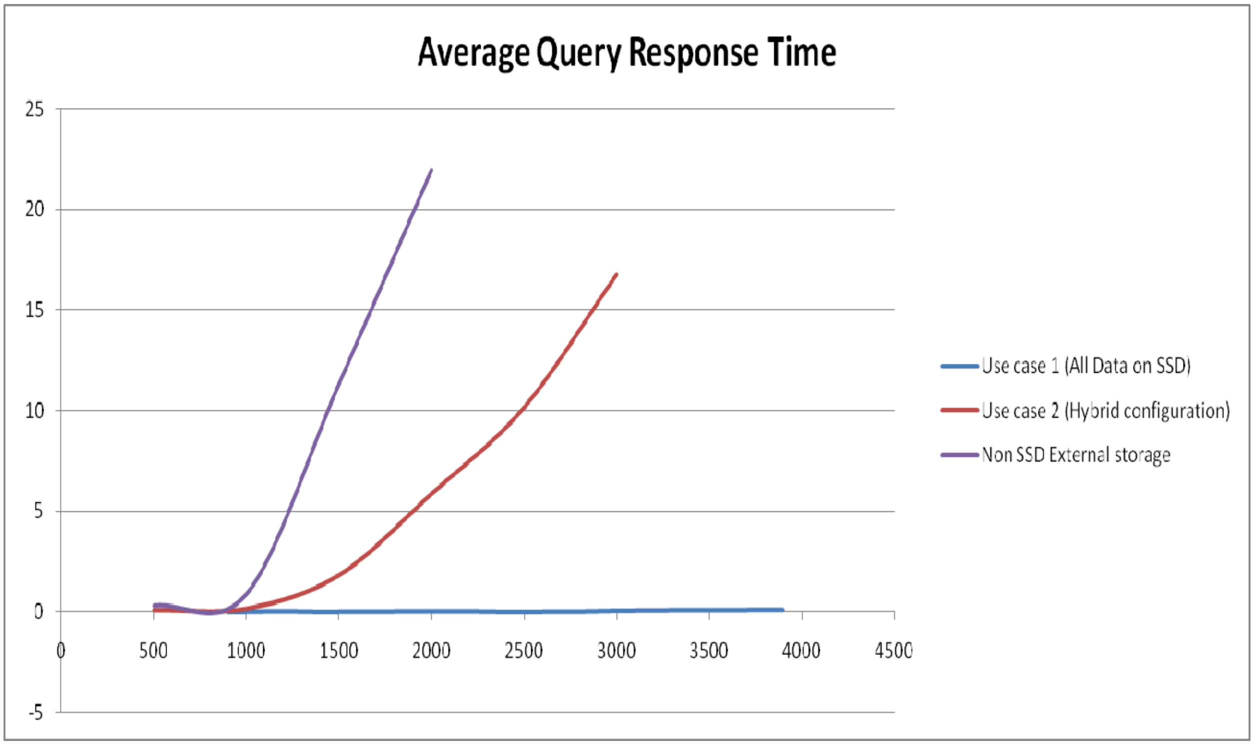


Table 2 and Figure 2 show the performance comparison between various configurations utilizing Fusion ioDrive modules. The non SSS external storage was configured which exhibited serious IO bottleneck beyond 1000 user load which was easily removed by Moving Indexes and temporary segments to Fusion ioDrive. The hybrid configuration is the most cost effective configuration and also exhibits significant reduction in the Average Query Response time compared to a database residing on traditional storage. The SSS only configuration exhibited around 60 times reductions in Average Query Response time compared to the non SSS external storage configuration. We can also observe that the user load could be scaled beyond 1000 users for both hybrid and SSS only cases with an acceptable Average Query Response Time.

Table 2. Average Query Response Time Comparison

User count	**Traditional Storage:** **Resp. time (Sec)**	**Hybrid Configuration: Response Time (Sec)**	**SSS only Configuration:** **Resp. Time (Sec)**
1000	.905	.156 (5.8 fold improvement Vs traditional storage)	.015 (60 fold improvement Vs traditional storage)

Figure 2. Average Query Response Time Comparison



NOTE: The results we have provided are intended only for the purpose of comparison between two environments consisting of specific configurations in a lab environment. The results do not portray the maximum capabilities of any system, database software, or storage.

Use Case 4: Oracle Flash Cache

For use case 4, an Oracle RAC database was configured consisting of two Dell PowerEdge™ servers, each populated with one Fusion-io ioDrive 160GB SLC module which were used as flash cache destination. For this case study, the flash cache feature was enabled to study the performance impact of using the Fusion-io SSS modules as a second tier Oracle buffer cache. In an Oracle RAC configuration with flash cache, the RAC nodes keep a pointer to database blocks which are retired from the main buffer cache to the secondary flash cache location. Thus cache fusion seamlessly works across cluster nodes in case a remote node requires a database block from a remote node's SGA. The FucionIO SSS device practically functions as an extension of Oracle buffer cache.

In the following sections, we will study the setup, configuration and performance analysis of Oracle Flash cache impacting an Oracle 11gR2 RAC database.

Configuring Database Flash Cache

One should consider using the Flash Cache feature for Oracle database under the following conditions:

- The database is running on the Enterprise Linux operating system.
- The Buffer Pool Advisory section of your Automatic Workload Repository (AWR) report or STATSPACK report indicates that increasing the size of the buffer cache would be beneficial.

- From the Oracle AWR report, db file sequential read is a top wait event.
- The CPU resources are not a bottleneck.

One should keep in mind that the device configured for Flash Cache may not be shared among multiple Oracle instances.

Sizing the Flash Cache:

As a general rule, size the flash cache to be between 2 times and 10 times the size of the buffer cache. For the purposes of this white paper, with our memory_target of 10GB, we will be configuring the Flash Cache on Fusion-io SLC 160GB device since any flash cache multiplier less than two would not provide any benefit. In case automatic shared memory management is being used, the flash cache should be between 2 times and 10 times the size of SGA_TARGET as recommended by Oracle. Using 80% of the size of SGA_TARGET instead of the full size should be adequate for this calculation.

Tuning Memory for the Flash Cache

For each database block moved from the buffer cache to the flash cache, a small amount of metadata about the block is kept in the buffer cache. For a single instance database, the metadata consumes approximately 100 bytes. For an Oracle Real Application Clusters (RAC) database, it is closer to 200 bytes. One must therefore take this extra memory requirement into account when adding the flash cache.

- If you are managing memory manually, increase the size of the buffer cache by an amount approximately equal to the number of database blocks that fit into the flash cache multiplied by 100 (or 200 for Oracle RAC).
- If you are using automatic memory management, increase the size of MEMORY_TARGET using the algorithm described above. You may first have to increase the size of MEMORY_MAX_TARGET.
- If you are using automatic shared memory management, increase the size of SGA_TARGET.

Table 3. Flash Cache Initialization Parameters

db_flash_cache_file	Specifies the path and file name for the file to contain the flash cache, in either the operating system file system or an Oracle Automatic Storage Management disk group. If the file does not exist, the database creates it during startup. The file must reside on a flash disk device. If you configure the flash cache on a disk drive (spindle), performance may suffer. The following is an example of a valid value for db_flash_cache_file: /dev/fioa
db_flash_cache_size	Specifies the size of the flash cache. Must be less than or equal to the physical memory size of the flash disk device. Expressed as *n*G, indicating the number of gigabytes (GB). For example, to specify a 16 GB flash cache, set db_flash_cache_size to 16G.

NOTE: Oracle smart flash cache with 11g R2 requires that the flash cache destination on both nodes be identical (/dev/fioa). Make sure to populate the Fusion-io flash devices on the same PCIe slots on all cluster nodes.

NOTE: DB_FLASH_CACHE_SIZE specifies the size of the Database Smart Flash Cache. This parameter may only be specified at instance startup. One can dynamically disable flash cache by changing this parameter to 0 after the database is started. Dynamic resizing of DB_FLASH_CACHE_SIZE or re-enabling flash cache to a different size is not supported.

Enabling Flash Cache:
To enable flash cache, perform the following steps in order which involve grating the read and write permission on the Fusion-io device to the Oracle database user account and setting up the flash cache parameters:

```
chown oracle.oinstall /dev/fioa (Where ‘fioa’ is the Fusion-io SSS device)

chmod 775 /dev/fioa
```

One can modify /etc/rc.local file and these entries so that the device ownership and permissions are setup correctly upon system restart.

Next modify the database parameters as followed:

```
SQL> alter system set db_flash_cache_file='/dev/fioa' scope=spfile;
```

System altered.

```
SQL> alter system set db_flash_cache_size=140G scope=spfile;
```

Next restart the cluster database for the changes to take effect:

```
[oracle@zfusionr810 dbs]$ srvctl stop database -d fusiondb

[oracle@zfusionr810 dbs]$ srvctl start database -d fusiondb
```

Verifying flash cache status in Oracle 11gR2 RAC
The following query can be used to obtain information about the various Oracle buffer cache and flash cache areas.

```
SQL> SELECT

             name,

            value

FROM

             v$sysstat

WHERE

             name IN
             ('physical read flash cache hits',
   'physical reads',
   'consistent gets',
   'db block gets',
   'flash cache inserts');
```

Initially, the flash cache area does not have any database blocks as shown below:

```
NAME                                                VALUE
-------------------------------------------     ------------------ ----------
db block gets                                       3904
consistent gets                                     102232
physical reads                                      8637
physical read flash cache hits                      0
flash cache inserts                                 0
```

After running workload, the flash cache hits start accumulating as shown below:

Node1:

```
NAME                                                        VALUE
-------------------------------------------------           ------------- ----------
db block gets                                               3674966
consistent gets                                             103091755
physical reads                                               4918152
physical read flash cache hits                              3169260
flash cache inserts                                         1937640
```

Node2:

```
NAME                                                        VALUE
```

```
----------------------------------------------------          ------------ ----------

db block gets                                                  2637154

consistent gets                                                 95254235

physical reads                                                 4690577

physical read flash cache hits                                 3039198

flash cache inserts                                            1837669
```

For flash cache size of 120GB per node we get the following IO statistics for the Oracle buffer cache:

```
NAME                                                           VALUE

------------------------------------------------          ---------------- ----------

db block gets                                                  3674966

consistent gets                                                103091755

physical reads                                                 4918152

physical read flash cache hits                                 3169260

flash cache inserts                                            1937640
```

One can observe from the above results that now most of the IO activity is happening in the buffer cache and the flash cache while physical IO is only a small percentage (4.2%) of the overall IO activity. One can also monitor which DB objects have been cached in the flash cache by running the following query:

```
select
   owner||'.'||object_name
from
   v$bh, dba_objects
where
 v$bh.status like ('flash%');
```

Analyzing flash cache performance
To analyze the performance impact of the Oracle Flash cache on a database running a workload, one needs to carefully analyze the Oracle Workload repository data. To capture the AWR report, one needs to take snapshots before and after running the workload and analyze the impact on the buffer cache advisory and top wait events. Expect for the CPU time to increase and the User IO wait events to drop from the list of the top five wait events as a result of enabling the Flash cache. Following is the procedure to take AWR snapshots:

- Take a snapshot of database activity before running the workload:

```
SQL> exec DBMS_WORKLOAD_REPOSITORY.CREATE_SNAPSHOT;
```

- Take another snapshot after completing workload:

```
SQL> exec DBMS_WORKLOAD_REPOSITORY.CREATE_SNAPSHOT;
```

- Generate an AWR report between the interval:

```
SQL> @?/rdbms/admin/awrrpt.sql
```

- Generate a snapshot before running load without flash cache
- Generate snapshot after running load without flash cache
- Generate snapshot before running load with flash cache
- Generate snapshot after running load with flash cache

One can then compare the AWR reports before and after enabling the Flash Cache feature to analyze the impact on performance.

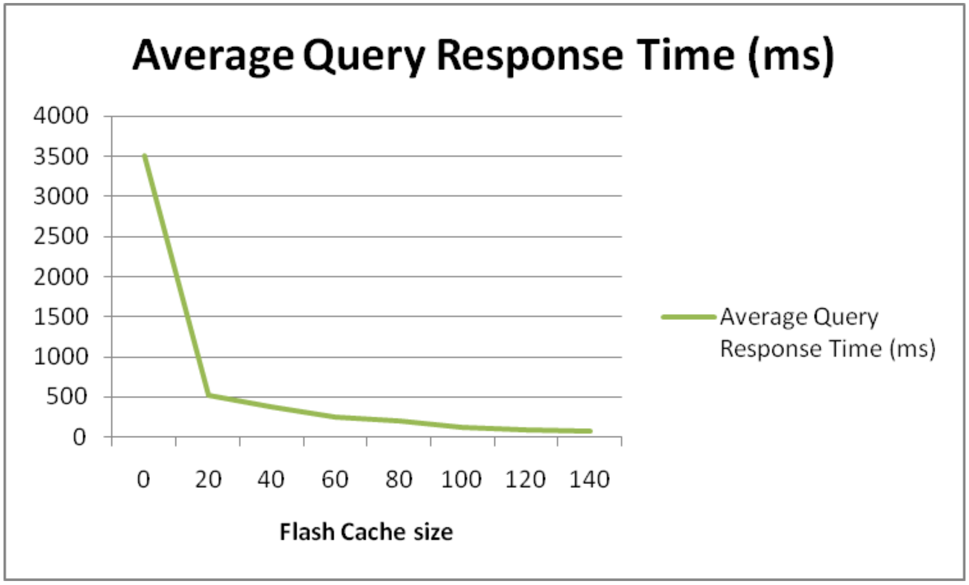
The impact of flash cache was studied for various sizes of db_flash_cache_size parameter and plotted against the average query response time of a typical OLTP workload as shown the following Table 3 and Figure 3. The database nodes were stressed with high userload in order to make the shared storage tier a bottleneck resulting in an average query response time reaching beyond 3 seconds. One can observe that for a 300GB database with memory target of 10GB per node, allocating 140GB of flash cache per node resulted in speedup by a factor of 45 in terms of average query response time.

Table 4. Average Query Response Time Improvement With Various Flash cache Sizes

Flash Cache size (GB)	Average Query Response Time (ms)
0	3500
20	515
40	376
60	256
80	198
100	120
120	95
140	77

As seen from Table 4 and Figure 3, the original storage configuration was exhibiting a serious IO bottleneck under high stress resulting in an Average Query Response Time reaching unacceptable levels. One might remedy the situation by adding additional spindles to the storage tier. Another option would be to add Fusion-io SSS modules as Flash Cache destination. As evident from the above table, the addition of Flash cache lowered the Average Query response Time by a factor of 45 with minimal disruption to the backend storage tier.

Figure 3. Average Query Response Time Improvement With Various Flash Cache Sizes



NOTE: The results we have provided are intended only for the purpose of comparison between two environments consisting of specific configurations in a lab environment. The results do not portray the maximum capabilities of any system, database software, or storage.

Summary

In this white paper, we analyzed various use cases of Dell's Fusion-io solid state family of products for single node as well as clustered database solutions. Fusion-io solid state products reside directly on the PCI express bus and result in acceleration of database applications which requires very low latency and high IO rates. The Fusion-io family of products integrate with servers at the system bus and kernel level, creating a new flash memory tier which can be either utilized to store database objects which require high IO rates as well as low latency or it can be used as a second level database buffer cache as we discussed in this white paper. The introduction of PCI express-based flash storage in the database server not only results in reduced latency but also removes IO bottlenecks which may be present in the original storage architecture with minimal downtime. These solid state storage modules are capable of servicing tens of thousands more small random reads IO operations per second as compared to traditional rotational disk drives while maintaining very low latency.

References

Solid State Storage Architectures
http://www.snia.org/education/tutorials/2010/spring/solid/JamonBowen_Solid_State_Storage_Architectures.pdf

Oracle 11*g* Database New Features
http://download.oracle.com/docs/cd/E11882_01/server.112/e10881/chapter1.htm

Using the Oracle 11GR2 database flash cache
http://guyharrison.squarespace.com/blog/2009/11/24/using-the-oracle-11gr2-database-flash-cache.html

Oracle® Database Administrator's Guide 11*g* Release 2 (11.2)
http://download.oracle.com/docs/cd/E11882_01/server.112/e10595/memory005.htm#ADMIN13391