

Accelerating Oracle Databases and Reducing Storage Complexity and Costs. Virident FlashMAX SCM as Primary Storage.

Solution Guide

Revision 1.0, January 2012





Table of Contents

Overview	2
Putting Entire Database on Flash	2
Native PCIe: Unleashing Performance of Flash	3
Performance Headroom for the Unexpected.....	4
Capacity and Density	4
Data Integrity.....	4
High Availability	4
Reference Server Configuration	5
Software Configuration Tips	7
Measuring Performance in Oracle	10
Summary.....	12

Overview

Virident FlashMAX™ Storage Class Memory (SCM) provides simple and powerful storage solution for applications using Oracle database. Using FlashMAX one can place an entire Oracle database on the high-performance flash memory attached directly to PCIe. Resulting high performance and quality of service allow quicker application deployments with enough headroom for future growth or for unexpected workload spikes.

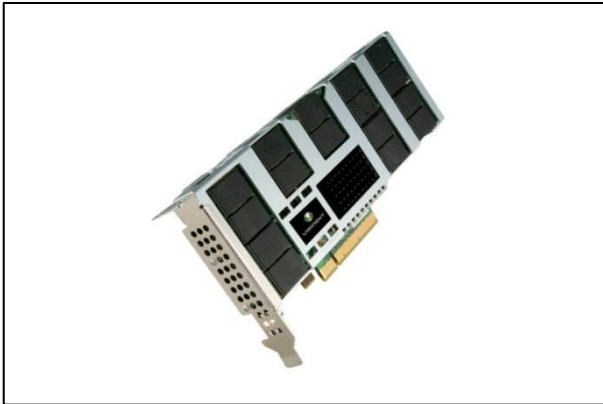


Figure 1. Virident FlashMAX card

The simple and elegant “all flash” storage architecture reduces both capital and operational costs by eliminating the need for traditional storage elements, such as SAN, RAID, or JBODs and by eliminating the complexity of multiple storage or caching tiers.

In this document we discuss advantages that Virident FlashMAX Storage Class Memory offers when used as primary storage for Oracle databases. We also provide configuration recommendations that should help with fully realizing those advantages.

Putting Entire Database on Flash

Traditional approach to tackling storage I/O performance challenges includes optimizing SQL code, adding spindles, adding DRAM, adding caching devices, then repeating all of the above. The new approach is having all I/O intensive data stored on flash based storage.

Having several Terabytes of high performance flash memory directly attached to PCIe enables a revolutionary approach to database storage architecture. The entire database can be placed on FlashMAX PCIe devices installed in the database server. There is no need for SAN, RAID controllers, HBAs, JBOD enclosures, or HDDs. This approach not only offers the highest performance, but also reduces complexity, lowers administration costs, and improves overall reliability.

Native PCIe: Unleashing Performance of Flash

FlashMAX has highly parallel flash architecture, direct PCIe connectivity and short I/O path with no legacy storage layers. In addition, Virident vFAS (Flash management with Adaptive Scheduler) software layer implements sophisticated I/O scheduling mechanism optimized for flash memory. This allows minimizing I/O latencies while delivering the highest IOPS rates. A single FlashMAX M1400 card delivers 325K random 4KB read IOPS at 0.8ms latency or 220K IOPS at 0.2ms latency (see figure 4).

Figure 1 shows I/O performance of an Oracle 11g2 database with various numbers of cards in a single four-processor system. Being able to achieve over 1 million Oracle IOPS in a single 4-processor server enables new applications that previously were not possible due to prohibitive costs of SAN infrastructure with thousands of HDDs required to deliver comparable performance.

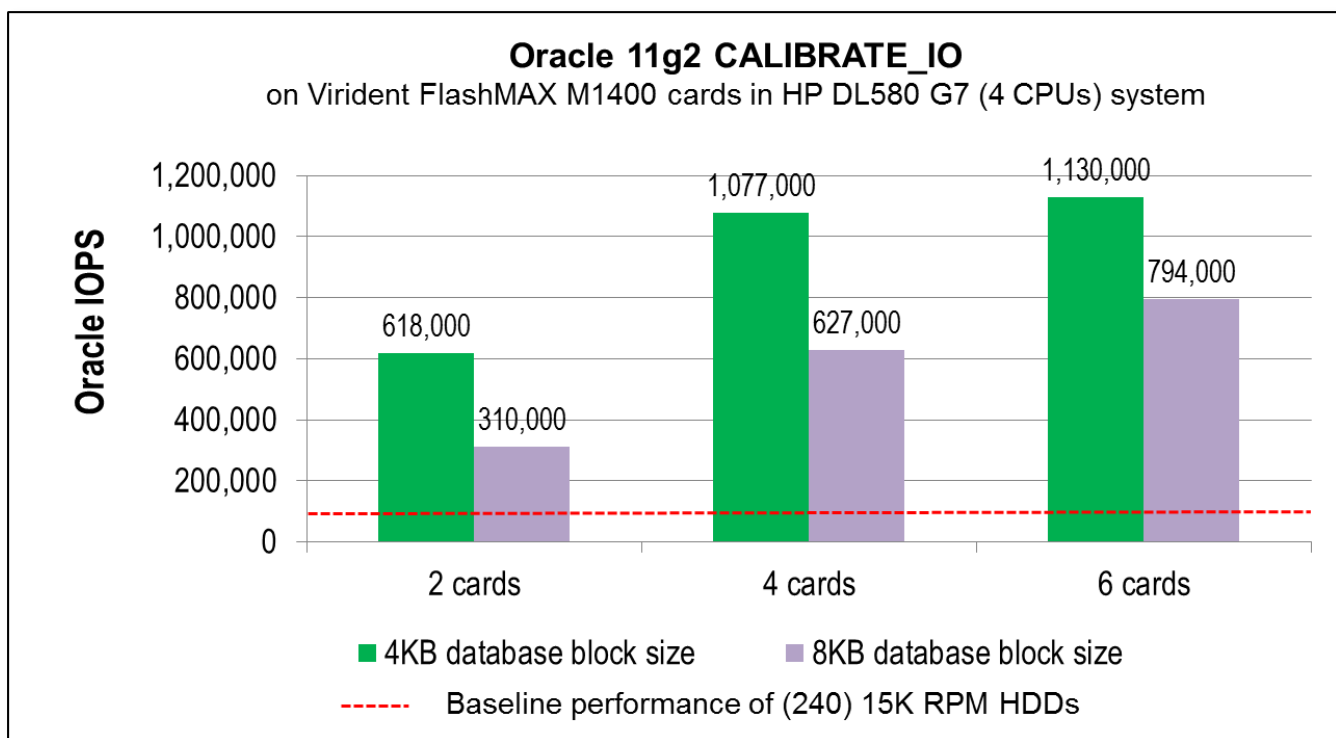


Figure 2. Results of running DBMS_RESOURCE_MANAGER.CALIBRATE_IO procedure in Oracle 11g2 with Oracle ASM in Normal Redundancy mode on Virident FlashMAX M1400 cards in HP DL580 G7 (four Intel Xeon E7-4850 CPUs) system. Reported latency was 0ms (<1ms).

Performance Headroom for the Unexpected

High storage I/O performance with FlashMAX not only accelerates applications, but it also makes maintenance and failure recovery tasks, such as DB checks, DB backups, and replays run much faster with minimal or no impact on the application performance.

Placing all I/O intensive data on FlashMAX devices provides sufficient I/O performance headroom to account for the unexpected: workload spikes, failures, or new I/O hungry queries. This approach frees up development and operations resources that otherwise would be used for troubleshooting and optimizing storage and application performance.

Capacity and Density

Virident FlashMAX devices built with compute class MLC flash memory offer high capacities that can accommodate many large databases. The M1400 model offers 1400GB of user capacity in low-profile half-length PCIe form factor. Using systems available today from major server OEMs it is possible to achieve up to 9.8TB (7 PCIe slots) in 2U or up to 15.4TB (11 PCIe slots) in 4U systems.

Data Integrity

Error checking and correction (ECC) is implemented in the flash controller hardware and relies on extra bits stored with each block of user data. ECC allows detecting and correcting most single- and multi-bit errors within a block. In addition, data blocks are proactively relocated (refreshed) when the number of error bits in a block reaches certain threshold.

Power loss protection is ensured by using onboard capacitors. Any committed data will be written to the flash media even if power is interrupted.

High Availability

Flash-Aware RAID

Flash-aware RAID logic within the vFAS software stores one block of parity information for every 7 blocks of user data, creating a stripe with parity similar to RAID-5. Each block in the stripe is stored on a separate flash chip. In case a block of data is unreadable, the vFAS software recovers the data using the parity information and rewrites the data to a new physical location, restoring data redundancy and protection. Errors that can be recovered using RAID parity include multi-bit errors that exceed ECC correction capability, entire flash block failures, entire flash die failures, and entire flash chip failures. Unlike regular RAID-5 implementations, the RAID logic in the vFAS software is tightly integrated with flash memory management and provides higher level of protection than regular RAID-5 with minimal impact on performance. The flash aware RAID implementation is fully automatic and does not require any configuration by user.

Instant Self-Healing

A physical flash block that returns an error gets retired and will not be used for storing data. Such retired blocks are immediately replaced with blocks from the reserved capacity pool and full redundancy is automatically restored. This self-healing capability ensures high availability of data. It is completely transparent to the OS and to the Oracle database and does not require any user intervention.

Data Mirroring in Oracle ASM (Automatic Storage Manager)

Oracle ASM provides effective means for ensuring data redundancy and performance scaling across multiple FlashMAX devices. Virident recommends using two or more FlashMAX devices configured for Normal Redundancy or High Redundancy using Oracle ASM.

Redundancy across Servers with Oracle Data Guard

Oracle Data Guard allows keeping one or more synchronized standby database replicas. Synchronization is performed by sending redo data over TCP/IP network from the primary database to the standby database(s). More details about implementing Oracle Data Guard can be obtained in the whitepaper at <http://www.oracle.com/technetwork/database/features/availability/twp-dataguard-11gr2-1-131981.pdf>

Reference Server Configuration

The following table presents a reference configuration that can be used for building an Oracle database server with FlashMAX based storage.

FlashMAX cards	One or more M1400 (1400GB) for higher capacity configurations One or more M1000 (1000GB) for smaller capacity configurations
PCIe slots	One PCIe slot per FlashMAX card. PCIe x8 recommended. Low-profile, half-length.
CPU	2 or more, Intel Westmere (56xx, 75xx, E7-series) or later
DRAM	32GB or more
Boot Device	(2) HDD in RAID-1 using onboard storage controller
OS	OEL 5, RHEL 5, CentOS 5, SLES11 SP1 (other Linux versions as supported by Oracle)
Oracle Software	Database 11g2, Grid Infrastructure 11g2 (for ASM support)



Software Configuration Tips

Configuring Oracle ASM

FlashMAX volumes are seen by the OS as standard block devices and can be configured in Oracle ASM similar to regular disks. In Linux before FlashMAX devices can be used with Oracle ASM they need to be configured using one of the following options:

- 1) using UDEV rule (recommended)
- 2) using ASMLIB

Follow the steps below to set up a UDEV rule that will automatically configure permissions for FlashMAX devices on every boot, so that they become accessible by Oracle ASM:

- 1) Create a file named:
 /etc/udev/rules.d/01-vgc-oracle.rules if using RHEL6/OEL6/CentOS6
 or
 /etc/udev/rules.d/99-vgc-oracle.rules if using SLES11, RHEL5/OEL5/CentOS5
- 2) Add the following line to the file above and save it (change owner name from "grid" to the one you use for administering Oracle ASM if different):
 KERNEL=="vgc??", OWNER="grid", GROUP="dba", MODE=660
- 3) Run the following command to apply the rule:
 # udevtrigger if using RHEL5/OEL5/CentOS5
 or
 # udevadm trigger if using SLES11, RHEL6/OEL6/CentOS6
- 4) Double-check that permissions are correct by running:
 # ls -l /dev/vgc??
 brw-rw---- 1 grid dba 252, 0 Dec 4 15:03 /dev/vgca0
 brw-rw---- 1 grid dba 252, 16 Dec 4 14:59 /dev/vgcb0
- 5) In Oracle ASM use the following Disk Discovery Path: /dev/vgc??
- 6) If the permissions are set correctly, but you still cannot discover the drives in ASM, you may need to erase the drives. This may happen if the drives were previously used by ASM and some metadata is still on the drives. To erase the drives run the following commands:
 # vgc-config -r -d /dev/vgca
 # vgc-config -r -d /dev/vgcb
 ...
 # service vgcd reload

Partitions and File Systems

Using Oracle ASM is a preferred method of configuring FlashMAX devices for an Oracle database. However, if needed, FlashMAX devices can also be partitioned and used with a file system.

If you need to create more than one partition on a FlashMAX device, it is important to make sure the start sector of each partition is multiple of 8 (aligned on 4KB). In Linux to set the starting sector numbers run fdisk with '-u' parameter:

```
# fdisk -u /dev/vgcX0
```

FlashMAX can be used with any file system including XFS and EXT2/3/4. However, XFS has better mechanisms for handling shared access to the same file by multiple processes and provides higher concurrency and performance. We recommend creating XFS with sector size parameter set to 4KB.

```
# mkfs.xfs -s size=4096 /dev/vgcX0
```

Placing Online Redo Log Files

FlashMAX can be used for storing both Oracle data files and online redo log files on the same device. Oracle log writer process (LGWR) generates small block (can be as small as 512B) sequential write I/Os to online redo log files. These I/Os are synchronous and sensitive to latencies.

Traditionally, online redo logs were placed on dedicated HDDs as sequential nature of the redo log I/Os make HDD performance sufficient. However, the only way to achieve acceptable write latencies with HDDs while ensuring data integrity is using a RAID controller with write caching and a battery. This makes the HDD-based redo log storage complicated and susceptible to hardware and human errors.

Highly parallel flash architecture of FlashMAX, optimization of sequential write streams, and power protected SRAM write buffer minimize interference between online redo log file I/O and other types of I/O and allow co-locating different Oracle file types on the same FlashMAX devices.

Oracle Database Block Size

When Oracle data is stored on FlashMAX devices, reducing Oracle database block size from the default value of 8192 bytes (8KB) to 4096 bytes (4KB) can provide substantial performance benefits in many applications. With HDDs, reading/writing 4KB takes essentially the same amount of time as 8KB as most of the time is spent on moving heads. In contrast, FlashMAX can perform 2x the amount of IOPS with 4KB block size compared to 8KB block size, or the same amount of IOPS at lower latencies.

Figure 2 compares how average I/O latencies depend on the amount of IOPS for 4KB and 8KB block size in case of 20% write/80% read random workload. While 8KB performance saturates at 75 000 IOPS, the 4KB performance extends to 160,000 IOPS. Alternatively, if we generate fixed amount of IOPS, for example 60

000, the 8KB workload has 0.7ms latencies, while the 4KB workload has <0.2ms latencies.

To change the database block size, set the following initialization parameter: DB_BLOCK_SIZE=4096

You can set this parameter in several different ways:

- 1) By adding it to init *ORACLE_SID*.ora file (or changing if the parameter already exists)
- 2) By setting the parameter in SPFILE:

```
SQL> ALTER SYSTEM SET DB_BLOCK_SIZE=4096 SCOPE=SPFILE;
SQL> SHUTDOWN IMMEDIATE
SQL> STARTUP
```
- 3) By setting it on *Initialization Parameters -> Sizing* tab of the Database Configuration Assistant

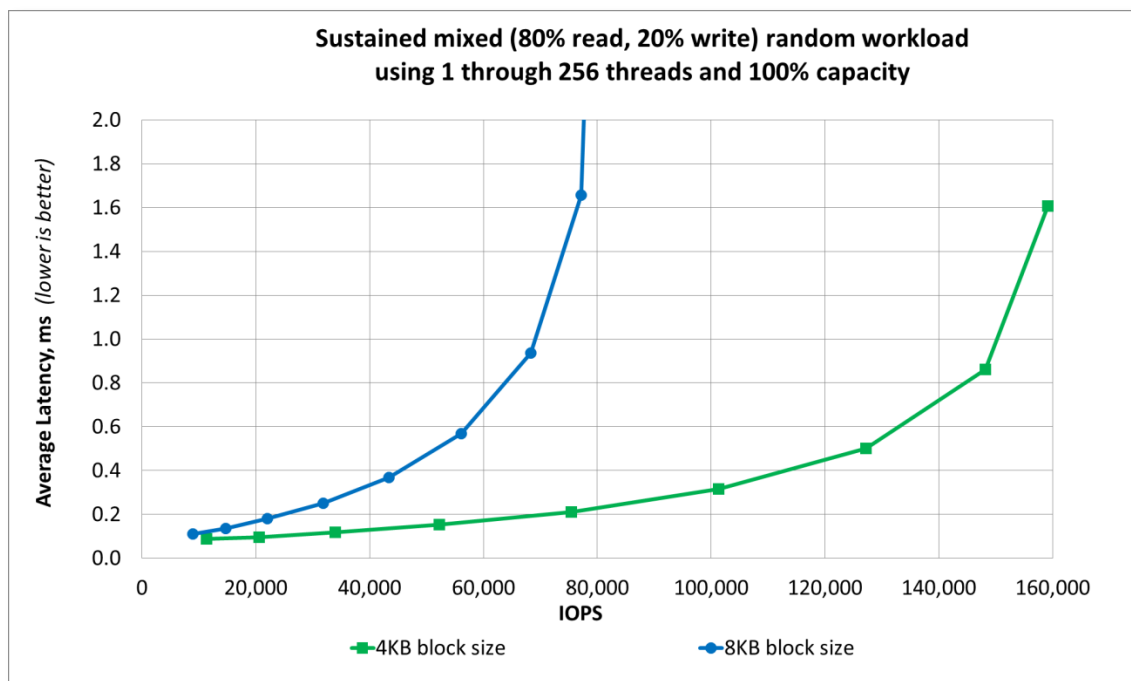


Figure 3. FlashMAX M1400 performance: comparing 4KB and 8KB block sizes. Each data point represents a particular number of the workload threads changing from 1 (left) to 256 (right).

Enabling Direct I/O and Asynchronous I/O

It is important for most of the storage I/O to go directly to FlashMAX devices bypassing Linux page buffer. The Linux page buffer cannot sustain the high I/O rates that FlashMAX provides and is likely to become a bottleneck. Also using asynchronous I/O provides better I/O concurrency and performance. To enable both direct I/O and asynchronous I/O set the following Oracle initialization parameter:

```
FILESYSTEMIO_OPTIONS=SETALL
```

You can set this parameter in several different ways:

- 1) By adding it to init *ORACLE_SID*.ora file (or changing if the parameter already exists)
- 2) By setting the parameter in SPFILE:
SQL> ALTER SYSTEM SET FILESYSTEMIO_OPTIONS=SETALL SCOPE=SPFILE;
SQL> SHUTDOWN IMMEDIATE
SQL> STARTUP
- 3) By setting it in Database Configuration Assistant in *Initialization Parameters (step 9)* -> *All Initialization Parameters...* -> *Show Advanced Parameters*

Measuring Performance in Oracle

The easiest way to assess performance of Oracle storage subsystem is using Oracle's standard PL/SQL procedure called CALIBRATE_IO. This procedure uses real Oracle database processes accessing the actual blocks in the database files. It can be used with any storage configuration including Oracle ASM and including various redundancy modes.

Syntax: DBMS_RESOURCE_MANAGER.CALIBRATE_IO (<DISKS>, <MAX_LATENCY>, iops, mbps, lat);

The CALIBRATE_IO procedure has two input parameters:

DISKS: This parameter affects the increment and the maximum number of outstanding I/Os used during the test. In case of HDDs, user is supposed to set this parameter equal to the number of physical spindles. However, due to much higher concurrency of FlashMAX, we recommend setting it to 8 per card. For example, if you have 3 cards installed, set this parameter to 24. Setting this parameter lower may produce lower IOPS result. Setting this parameter higher may produce higher latency result.

MAX_LATENCY: This parameter sets the limit for acceptable latency in *milliseconds*. The procedure will stop increasing the amount of outstanding I/Os when latencies become higher than this parameter. We recommend setting this parameter to the minimum allowed value of 10 milliseconds.

The CALIBRATE_IO procedure has three output values:

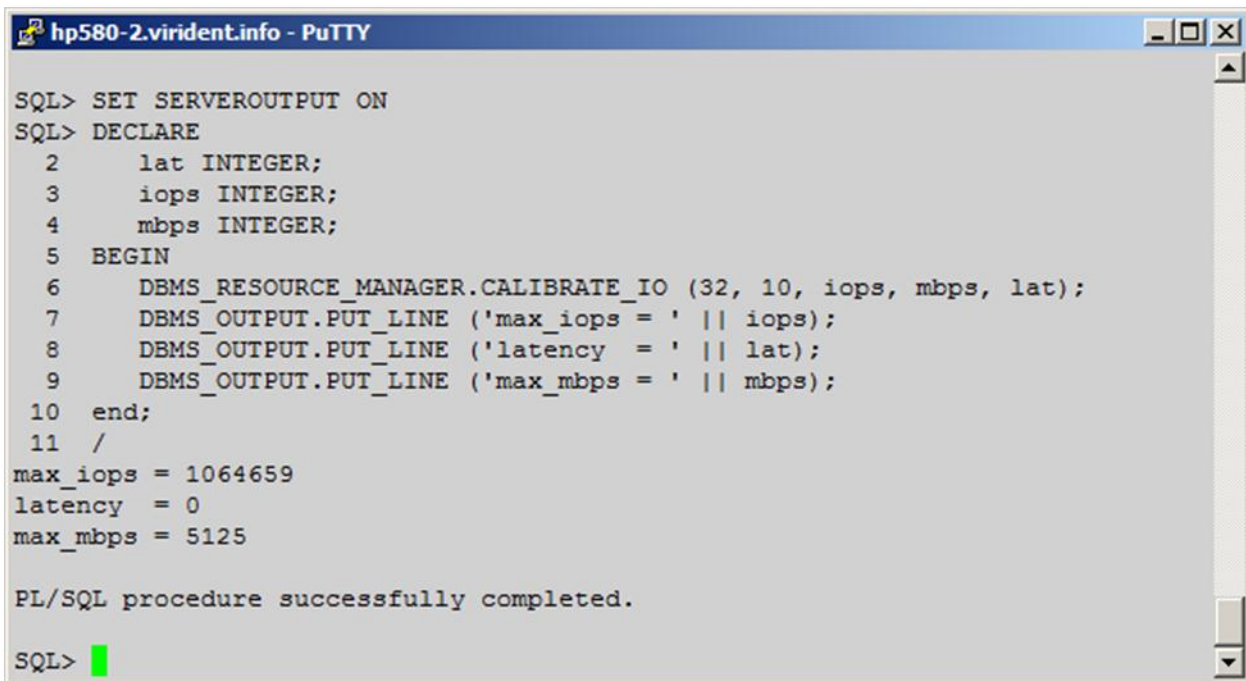
iops: This value returns the best measured number of I/Os per second achieved with latencies lower than MAX_LATENCY.

mbps: This value returns the best measured bandwidth in MB/s using large blocks. Note that the mbps value is not the iops value multiplied by block size. A separate test using larger block sizes is used for measuring the mbps value.

lat: This value returns the actual latencies in milliseconds measured during the iops test. On FlashMAX the iops value is expected to be zero in most cases, as the actual latencies are < 1ms.

Below is a PL/SQL script that you can run in SQL*Plus to execute the CALIBRATE_IO procedure. It takes several minutes to complete. Make sure to modify the DISKS parameter corresponding to the number of FlashMAX cards being used (8 per card).

```
SET SERVEROUTPUT ON
DECLARE
  lat INTEGER;
  iops INTEGER;
  mbps INTEGER;
BEGIN
  DBMS_RESOURCE_MANAGER.CALIBRATE_IO (8, 10, iops, mbps, lat);
  DBMS_OUTPUT.PUT_LINE ('max_iops = ' || iops);
  DBMS_OUTPUT.PUT_LINE ('latency = ' || lat);
  DBMS_OUTPUT.PUT_LINE ('max_mbps = ' || mbps);
end;
/
```



```
hp580-2.virident.info - PuTTY
SQL> SET SERVEROUTPUT ON
SQL> DECLARE
  2   lat INTEGER;
  3   iops INTEGER;
  4   mbps INTEGER;
  5 BEGIN
  6   DBMS_RESOURCE_MANAGER.CALIBRATE_IO (32, 10, iops, mbps, lat);
  7   DBMS_OUTPUT.PUT_LINE ('max_iops = ' || iops);
  8   DBMS_OUTPUT.PUT_LINE ('latency = ' || lat);
  9   DBMS_OUTPUT.PUT_LINE ('max_mbps = ' || mbps);
 10 end;
 11 /
max_iops = 1064659
latency = 0
max_mbps = 5125

PL/SQL procedure successfully completed.

SQL> █
```

Figure 4. Results of running CALIBRATE_IO on a database with (4) M1400 cards in ASM Normal Redundancy mode with 4KB database block size.



Summary

Placing Oracle database primary data on Virident FlashMAX Storage Class Memory devices makes it possible to have simple storage architecture with large performance headroom. This allows focusing resources on more productive tasks and cutting down application development and operational costs.

About Virident

Virident Systems, Inc. is the leading provider of high performance PCIe based solid-state storage solutions for scale-out datacenters and enterprises, allowing storage to finally catch up with CPU evolution. Virident has leveraged its deep expertise in various types of Flash technologies to deliver products based on the "Storage Class Memory" architecture, which combines memory-like performance with disk-drive capacity and persistence. Virident is driving the solid-state revolution in Storage with innovative hardware and software solutions, extending the benefits of PCIe solid-state storage beyond a server.

Virident, Inc., Virident, virident.com, the Virident logo, and FlashMAX are trademarks or registered trademarks of Virident Systems, Inc., in the United States, other countries, or both. If these and other Virident trademarked terms are marked on their first occurrence in this information with a trademark symbol (or), these symbols indicate U.S. registered or common law trademarks owned by Virident at the time this information was published. Such trademarks may also be registered or common law trademarks in other countries.