



WHITE PAPER

# **Maximizing Oracle Database 12c Performance with Fusion ioMemory PCIe Application Accelerators and Oracle Force Full Database Caching Mode**

## Table of Contents

<b>Introduction</b>	<b>3</b>
<b>Summary of Findings</b>	<b>3</b>
<b>About the Fusion ioMemory Platform</b>	<b>5</b>
<b>About Oracle Force Full Database Caching Mode</b>	<b>6</b>
<b>Testing Methodology</b>	<b>7</b>
<b>Test Results</b>	<b>7</b>
DSS Test Results	7
OLTP Test Results	8
<b>Observations</b>	<b>9</b>
<b>Conclusion</b>	<b>10</b>
<b>Appendix A: Test Configuration</b>	<b>12</b>
Server Specs	12
Operating System	12
Test Software	13
Oracle Database Software	13
Oracle ASM Configuration	13
Oracle Database Configuration	14
HammerDB Configuration	14

## Introduction

This paper examines how a new feature in Oracle 12c – Force Full Database Caching Mode – impacts performance on Decision Support System (DSS) and On-Line Transaction Processing (OLTP) workloads. It also compares the performance of running in-memory to running on-memory using the Fusion ioMemory™ platform. The intended audience of this paper is Oracle database administrators and system architects, but the level of technical detail in this paper is appropriate for all audiences.

Force Full Database Caching Mode leverages the Oracle database buffer cache to accelerate repeat reads of data. The initial read of each block loads that block into the cache, and subsequent reads are serviced from the cache. While this process is conceptually true for all relational databases, the experienced user knows all databases attempt to keep memory clean by flushing blocks back to storage, which requires those blocks to be later re-read into the cache. Databases also keep the cache neat and efficient by instructing long sequential reads to skip the cache and send the blocks to a private user space such as the Oracle PGA. Force Full Database Caching Mode changes these behaviors of the standard database cache: the standard LRU algorithm is replaced, big tables are candidates for caching, and full table scans are serviced by cached data.

Writes are not expected to be accelerated by in-memory caching solutions. Checkpoints, redo logging, redo archiving, and backups are not accelerated by the new Force Full Database Caching Mode. Write acceleration is addressed through storage modernization, such as replacing legacy disk with the Fusion ioMemory platform.

Many database servers lack sufficient memory to implement Force Full Database Caching Mode, particularly the very data warehouse and Decision Support Systems targeted by this new feature. Expanding memory to the required level may be cost-prohibitive or simply not possible, for reasons discussed later in this paper. Customers in such a predicament can achieve equal or better performance by running their Oracle databases on Fusion ioMemory rather than in-memory. The Fusion ioMemory platform combines enterprise class flash memory and software into a highly reliable and high performance primary storage solution. The Fusion ioMemory platform enables all database workloads to run at memory speeds. See the description of the Fusion ioMemory platform in this paper.

## Summary of Findings

Our testing has led to the following conclusions:

- All database workloads can be significantly accelerated by simply moving the data from legacy storage to Fusion ioMemory storage.
- Decision Support System databases should be stored on Fusion ioMemory storage devices, and the new Oracle 12c Force Full Database Caching Mode should be enabled if possible. The combined effect of flash storage and the new Oracle cache mode improved performance of our test database by 4,397%.
- On-Line Transaction Processing databases should be stored on Fusion ioMemory storage devices, but the new Oracle 12c Force Full Database Caching Mode need not be enabled, as it provided no additional performance gains. A database running on Fusion ioMemory storage without Force Full Database Caching was over 12x faster than the same database on legacy storage with Force Full Database Caching.

Force Full Database Caching Mode improved the performance of both OLTP and DSS workloads when the database was hosted on an array of legacy hard disk storage. Mixed results were reported for Force Full Database Caching Mode when the same two databases were hosted on a Fusion ioMemory PCIe Application Accelerator. The table below summarizes the change in performance realized by enabling Force Full Database Caching Mode:

	OLTP Workload	DSS Workload
HDD Array	35% more transactions per minute	2381% faster report run times
SX350	1% fewer transactions per minute	751% faster report run times

As a result of in-memory caching the performance gap narrowed between storage classes. For the OLTP workload, the relative performance of a single flash drive compared to a disk array dropped from 18:1 to 13:1. For the DSS workload, the relative performance dropped from 6:1 to 2:1. Details are provided later in this paper.

This remainder of this section summarizes the results of testing OLTP and DSS workloads on two classes of storage (legacy disk array and a single Fusion ioMemory SX350-3200) with and without the new Force Full Database Caching Mode enabled. First, the DSS results are summarized and then the OLTP results.

The DSS workload was accelerated 4,397% on average, compared to a baseline configuration, by moving the database from a legacy disk array onto a Fusion ioMemory storage device and then enabling the Oracle 12c new feature Force Full Database Caching Mode. For DSS databases already hosted on a single Fusion ioMemory storage device, our testing found that enabling Force Full Database Caching Mode could improve performance by an average of 751%. When the database was left on the legacy disk array, enabling the new caching mode improved performance by 2,381%, and moving this cache-enabled database from disk to the Fusion ioMemory platform nearly doubled performance to the 4,397% figure noted earlier.

Our battery of OLTP testing on the hard disk array completed fewer operations per minute (i.e., performance was slower) with the new Oracle 12c Force Full Database Caching Mode enabled, compared with the same workload on a Fusion ioMemory storage device *without* the new caching feature enabled. That is, storing the database on flash memory yielded higher performance than running in-memory. Legacy hard disk arrays are indeed a bottleneck for OLTP workloads, and so leveraging the new Oracle 12c Force Full Database Caching Mode improved OLTP performance considerably in our tests. However, even when the new caching feature was enabled, the OLTP performance on the legacy hard disk array was 1,259% slower than the same workload on a single Fusion ioMemory storage device without the new caching feature.

OLTP workloads are often bound by CPU or latching mechanisms, with additional I/O issues such as high latency on single block reads, and slow writes to the on-line redo logs, archive logs, and all datafiles during checkpoint operations. Force Full Database Caching addresses the issue of latency on random reads but does not address any of the write operations, which could account for 30-50% of transactions. The Fusion ioMemory platform eliminates all of these I/O bottlenecks and takes the OLTP system to its logical peak performance. For these

reasons consumers often see no performance gains running the OLTP workload in more expensive DRAM caches compared to more economical Fusion ioMemory flash storage.

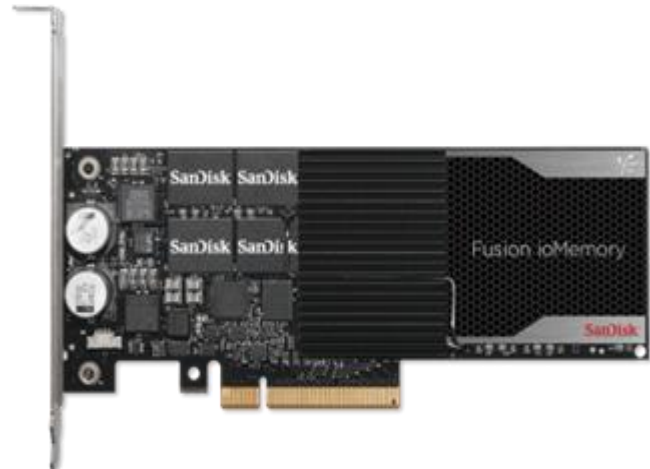
## About the Fusion ioMemory Platform

The Fusion ioMemory platform combines Fusion ioMemory VSL™ (Virtual Storage Layer) software with Fusion ioMemory hardware to take enterprise applications and databases to the next level. The Fusion ioMemory platform provides consistent microsecond latency access for mixed workloads, multiple gigabytes per second access, and hundreds of thousands of IOPS from a single product. The sophisticated Fusion ioMemory architecture allows for nearly symmetrical read and write performance with best-in-class low queue depth performance, making the Fusion ioMemory platform ideal across a wide variety of real world, high-performance enterprise environments.

The test results in this paper are based on a single Fusion ioMemory SX350-3200 storage device, which has a raw usable capacity of 3.2 TB using SanDisk NAND flash. The SX350 line includes products ranging in capacity from 1.25 TB to 6.4 TB. Additional products are available in a mezzanine form factor ideal for blade servers.

The Fusion ioMemory SX350-3200 storage device uses a low profile PCIe 2.0 x8 slot, making it compatible with nearly all enterprise-class servers. Most servers support multiple devices, somewhere between four and twelve depending on the server. Each Fusion ioMemory SX350-3200 storage device keeps data center costs to a minimum by consuming less than 25 watts of power. The product's endurance rating is 11 petabytes written.

Performance of the SX350-3200 during our Oracle OLTP testing averaged 209,500 8K random read IOPS with sub-millisecond latency, and 2780 MB/s sequential reads during Oracle testing with a DSS workload.



Fusion ioMemory PCIe Application Accelerators are unique in their ability to sustain writes as well as or better than reads. Most Oracle databases perform more read operations than write operations, but writes can still be a bottleneck for Oracle. Consider that many OLTP and Operational Data Stores have a workload consisting of 40% writes. These writes include inserts, updates, and deletes to row data and indexes, and corresponding Undo, Redo, and Archive Log writes generated by these operations. Even Decision Support Systems and analytics databases may experience slowness on checkpoints and logging. When selecting a storage product it is imperative to consider the database's dependency on write operations, not just read acceleration.

The SX350 line of Fusion ioMemory PCIe Application Accelerators featured in this paper was introduced in April 2015. Refer to the video at <https://www.youtube.com/watch?t=50&v=qweog75HTL8> or the data sheet at

<http://www.sandisk.com/assets/data-sheets/fusion-iomemory-sx350-pcie-application-accelerators-datasheet.pdf>.

## About Oracle Force Full Database Caching Mode

Oracle Force Full Database Caching Mode is a free feature in Oracle 12c Enterprise Edition patchset 12.1.0.2 and higher. It changes Oracle's caching algorithm to favor caching the whole database in-memory. When Force Full Database Caching Mode is enabled, the instance will check to see if there is sufficient space in the database block buffer cache's DEFAULT pool for the entire database, and if doing so would be beneficial. If both conditions are true, then the instance automatically caches the full database. Full caching includes the contents of all datafiles – even LOBs flagged as NOCACHE.

Without this feature Oracle will never attempt to cache the full database in-memory, regardless of the amount of memory allocated. This is because the default algorithms assume that loading more data into memory requires evicting other data from memory. Regular, random block access loads blocks into memory and places colder blocks on the list to be evicted. Sequential scans typically skip the SGA and go directly to the PGA: by default Oracle only caches reads on full table scans when the object size is less than 2% of the DB\_CACHE\_SIZE. The new Force Full Database Caching Mode changes all of these behaviors.

To enable this new feature, start the database in MOUNT mode, run an ALTER DATABASE statement, and then open the database. For example, the following statements could be issued in a SQL\*Plus window to enable Force Full Database Caching Mode:

```
CONNECT / AS SYSDBA
SHUTDOWN IMMEDIATE;
STARTUP MOUNT;
ALTER DATABASE FORCE FULL DATABASE CACHING;
ALTER DATABASE OPEN;
```

To disable the feature, use the same commands as shown above replacing FORCE with NO FORCE.

To see if the new feature is enabled or disabled, you can query the view V\$DATABASE as shown below. A value of YES means the feature is enabled.

```
SQL> SELECT FORCE_FULL_DB_CACHING FROM V$DATABASE;

FOR
---
```

```
YES
```

Oracle RAC and Oracle Multitenant are both supported with Force Full Database Caching Mode. The feature applies globally, however, which means it cannot be turned on or off at the individual PDB level or instance level. Once Force Full Database Caching Mode has been enabled, all instances and PDBs will silently take advantage of it.

**Note:** This feature requires sufficient memory. If you enable Force Full Database Caching Mode on a database whose cache is too small to hold the complete data set (or the collection of RAC instances is too small), then Oracle will silently disable the feature. Oracle RAC users may spread the memory requirements over all instances, but be aware the aggregate memory required is higher due to Oracle's propensity to cache duplicate blocks across instances.

When Force Full Database Caching Mode is enabled Oracle attempts to cache the entire database in the buffer cache's DEFAULT pool. Oracle strives to keep all data in this cache. Thus, you only need to configure the DEFAULT pool and not the KEEP pool or RECYCLE pool. Likewise, the nK caches need not be configured except in the usual case of supporting Oracle Transportable Tablespaces.

**Note:** Some customers prefer to set the default database block size to 4KB and then configure a secondary cache with a 16KB block size for indexes. For this paper we did not test the nK caches.

## Testing Methodology

To assess the impact of the Oracle 12c Force Full Database Caching Mode feature, four performance baselines were established using two storage configurations and two different workloads. The two storage configurations were a RAID 0 array of hard disk drives and a single Fusion ioMemory storage device. The two workloads were On-Line Transaction Processing (OLTP) and Decision Support System (DSS) instrumented using the HammerDB software.

After the four baselines were established we enabled Force Full Database Caching Mode and repeated both workloads on both storage configurations. Each combination of a storage configuration and a database workload was tested twelve times to rule out anomalies during testing. The test results used to create the charts in this paper are based on average scores. We did not attempt to produce artificial maximum performance: the operating system and database were tuned using only common practices: no hacks or unconventional techniques were leveraged, and ACID compliance was maintained throughout testing.

Complete details of the test environment are provided in the Appendix of this paper.

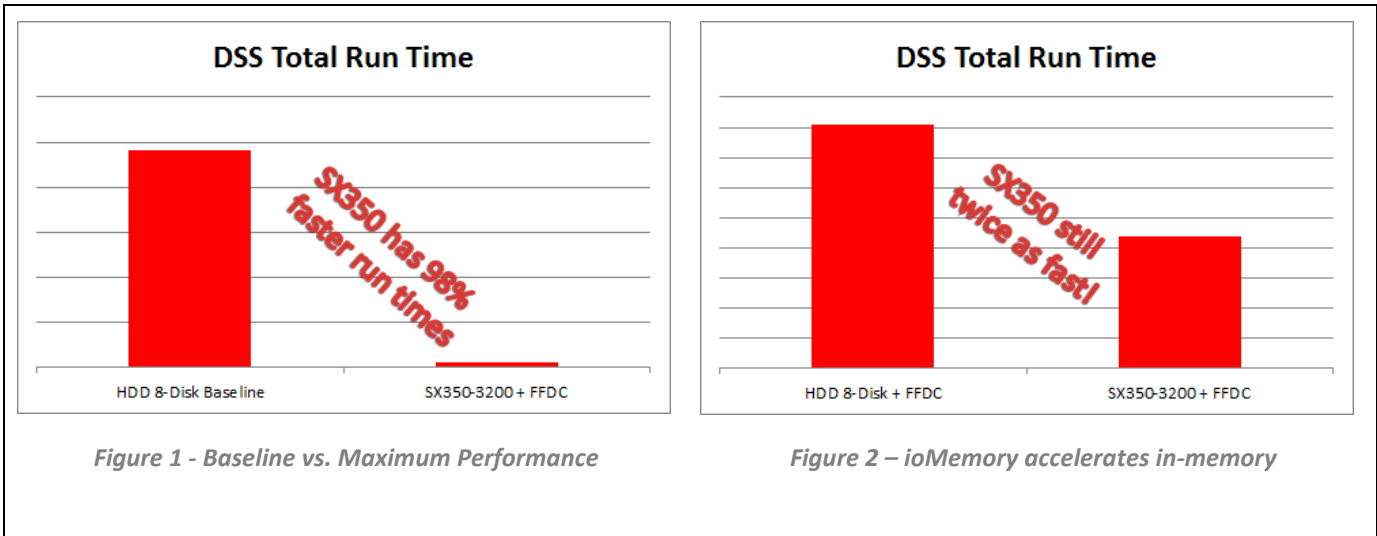
## Test Results

### DSS Test Results

For Decision Support System (DSS) testing, the impact of the Oracle 12c new feature Force Full Database Caching Mode was very clear and pronounced. Enabling this feature on our test database stored on a legacy hard disk array cut run times by 95%. For the same database stored on the Fusion ioMemory platform, run times were cut by 86% simply by enabling the new caching feature. Combining the two had the greatest impact: moving the database from the disk array to the Fusion ioMemory platform and simultaneously enabling Force Full Database Caching improved performance by almost 98%.



The following two charts summarize our test results for the Decision Support System workload. The chart on the left compares the performance of a standard hard drive array without additional caching versus a single Fusion ioMemory PCIe Application Accelerator with Force Full Database Caching Mode enabled, and shows the total run time decreased by 98%. The chart on the right compares the two storage platforms when Force Full Database Caching is enabled in both cases, and shows the single SX350 accelerates Oracle by nearly double that of in-memory processing.



Perhaps the most important point to take away from the above charts is the ability of Fusion ioMemory storage devices to accelerate a database running entirely in-memory.

### OLTP Test Results

For OLTP testing, a baseline was established using DB\_CACHE\_SIZE=280G and a schema size of 275 GB. After the baseline was established the OLTP tests were repeated on the same schema (restored from backup) with Force Full Database Caching Mode enabled. All scores were divided by the lowest score and then multiplied by 100 to create a relative index, and then posted to the table below. The table shows a modest 35% improvement in the performance of Oracle 12c on a legacy disk array by enabling Force Full Database Caching Mode, but the performance still pales in comparison to that of a single flash storage device with or without Force Full Database Caching Mode. Maximum performance of the OLTP workload was achieved by using flash storage without Force Full Database Caching Mode. One could interpret these results to mean it is simply more efficient to fetch data from flash storage than to lookup blocks in a very large SGA.



Configuration	OLTP Workload Relative Score	Percent Faster than Baseline on HDD Array	Improvement over FFDC on Disk Array
Baseline on HDD Array	100	0	0
*FFDC on HDD Array	135	35%	1X
Baseline on SX350	1,837	1737%	13.61X
FFDC on SX350	1,819	1719%	13.47X

\* FFDC indicates Force Full Database Caching was enabled for this test. Baseline tests did not use Force Full Database Caching.

### Observations

The impact of Force Full Database Caching was more pronounced on workloads involving table scans than random reads. By default, single block random reads are fetched into the Oracle buffer cache while full table scans bypass the SGA and the data is fetched directly into the user’s private memory space (Oracle PGA). Data in the Oracle buffer cache is reusable and sharable by all users, but data in the PGA space is not addressable by other users. Once Force Full Database Caching has been enabled, Oracle loads data from all tables, including “big” tables, into the buffer cache and service full table scans using this cached data.

During DSS testing – with the database stored on the Fusion ioMemory platform and Force Full Database Caching Mode enabled – it was observed that read bandwidth utilization dropped from 2800 MB/s to 200 MB/s after the first few queries. The I/O utilization remained at this low level for the duration of each test. Similarly, the hard disk utilization dropped from hundreds of MB/s to below 10 MB/s after the cache had been warmed up. CPU utilization remained high, and in some cases it increased slightly.

Latency of small block random reads is critical during OLTP testing. The Fusion ioMemory platform’s average latency was 61X better than that of the hard disk array. This facilitated the Fusion ioMemory platform completing 13X as many transactions per unit of time.

The Oracle database block buffer cache must be large enough to cache the entire database. When the cache is too small, Oracle reverts to legacy behavior, even though Force Full Database Caching Mode is enabled. Many servers today can be populated with sufficient quantities of DRAM to enable full caching. Some database servers, however, simply cannot hold enough DRAM to satisfy this requirement: consider a typical 10 TB database running on a server that supports at most 8 TB of DRAM.

Real Application Clusters (RAC) allows spreading the cache requirement across all instances, with some extra memory required for duplicate blocks cached in multiple instances. RAC was not evaluated in this paper.

Administrators often leave large amounts of DRAM unallocated, for potential use by a file system page cache to augment Oracle’s buffer cache. With Oracle 12c’s Force Full Database Caching Mode, all data can be cached within the Oracle buffer cache, so there’s no need to augment it with a file system page cache. It makes sense, therefore, to move that DRAM into the Oracle buffer cache.

If the SGA\_TARGET or MEMORY\_TARGET database initialization parameter is set, but DB\_CACHE\_SIZE is not set, then be aware Oracle may silently reallocate memory away from the cache to other memory spaces. If the cache size falls below the data size, then Oracle will be unable to cache the entire data set in memory and will silently disable the feature.

Direct I/O is recommended. Because a file system page cache is unnecessary for the reasons noted above, you should enable direct I/O to bypass the file system page cache and read/write directly between storage and the Oracle buffer cache. Oracle ASM uses direct I/O. For traditional file systems, instruct Oracle to use direct I/O by setting the FILESYSTEMIO\_OPTIONS database initialization parameter to either DIRECTIO or SETALL.

It may not be financially worthwhile to use the new caching feature, compared to simply running the database with a limited amount of DRAM and stored entirely on the Fusion ioMemory platform. The cost of the additional DRAM is one consideration. Another consideration is upgrading to a larger server with more processors to accommodate the additional DRAM chips. And finally, one must consider the impact of the additional processor on Oracle core-based licenses.

For planning purposes, consider setting the DB\_CACHE\_SIZE database parameter larger than the sum of all segments, less the free space. You must also consider database growth when setting parameter DB\_CACHE\_SIZE, because once the amount of data exceeds the size of the cache, Oracle will silently disable Force Full Database Caching Mode and revert to legacy behaviors. For example, if you calculate that your database uses 250 GB will grow 15% per year, then consider sizing the cache to 300 GB. This estimate does not include Oracle's various other memory spaces, such as the shared pool, large pool, etc.

## Conclusion

Decision Support System (DSS) performance was maximized by using a Fusion ioMemory PCIe Application Accelerator as primary storage, in conjunction with the new Oracle 12c feature Force Full Database Caching Mode. The workload completed in just 2% of the baseline time, on average, thus allowing 44 times more reports to complete in the same amount of time compared to the baseline configuration – a legacy hard disk array without the new Oracle 12c feature Force Full Database Caching Mode.

On-Line Transaction Processing (OLTP) performance was maximized by using only the Fusion ioMemory PCIe Application Accelerator as primary storage, and no additional benefits were realized by enabling the new Oracle 12c feature Force Full Database Caching Mode. In short, forcing the I/O associated with an OLTP workload to the Fusion ioMemory platform was more efficient than implementing a large Oracle SGA.

There may be cases where it is not physically possible to implement Force Full Database Caching Mode, such as whenever the database size equals or exceeds the server's DRAM capacity. There may also be times when it is not economically feasible to implement Force Full Database Caching Mode, such as when the expansion of DRAM requires moving to a larger server with higher CPU core counts that in turn drive up the Oracle licensing fees. In such cases, the Oracle database can still be greatly accelerated by using just the Fusion ioMemory PCIe Application Accelerator as primary storage, with no additional caching software.



## Appendix A: Test Configuration

### Server Specs

The database server is 2-socket system. Physically, it is a 2U rack mount server. There are two 14-core processors with Hyper-Threading enabled, for a total of 28 cores and 56 threads. The processor model is Intel® Xeon® CPU E5-2697 v3 (Grantley) @ 2.60 GHz.

The server has 768 GB RAM. 600 GB is allocated to HugePages for use by the Oracle SGAs of all databases on the server. 100 GB is logically allocated to the Oracle PGA for all databases on the server. The remaining 68 GB supports the operating system, drivers, and the VSL (Virtual Storage Layer) software. Memory does not need to be kept in reserve for a filesystem page cache, because Oracle ASM is used with direct I/O – all reads go directly into either the Oracle SGA or PGA.

Storage consists of ten 1 TB hard disk drives and two Fusion ioMemory SX350-3200 devices. The Fusion ioMemory storage devices use VSL software version 4.2.1 and firmware version 8.9.1. Two of the hard disks are used for a mirrored boot drive, and the other eight disks are configured as a RAID 0 group managed by Oracle ASM for database files. Only one Fusion ioMemory storage device is used for database storage and for comparison to the array of disks. The second Fusion ioMemory storage device is used for scratch space and backups: storing backups on flash allows the database to be re-baselined very quickly between each test. All storage devices (hard disks and flash) were formatted with a sector size of 512 bytes. Partitions were created using the Linux “parted” utility. Ownership and permissions were set on the partitions using the Linux udev rules facility.

HugePages were configured to 600 GB by editing the `/dev/shm` device in `/etc/fstab`, and by setting the appropriate parameters in `/etc/sysctl.conf`. By default, on Linux the device `/dev/shm` is sized to only one-half of DRAM (384 GB on our test server) and HugePages is disabled.

### Operating System

The operating system is Red Hat Enterprise Linux (RHEL) 6.6 with kernel 2.6.32-504.12.2.el6.x86\_64. The Java version was upgraded to 1.8.0\_45., and the following customizations were made to the OS environment:

- Kernel tuning parameters were adjusted based on Oracle documentation and best practices.
- The `/etc/grub.conf` file was updated to disable c-states and Transparent HugePages (THP).
- The `/etc/sysconfig/cpuspeed` file was edited to set the p-state to `GOVERNOR=PERFORMANCE`.
- The `/dev/shm` device was increased from its default size of ½ RAM to 605 GB.
- Linux HugePages were enabled by increasing the size from 0 to 307,200 pages (600 GB) in `/etc/sysctl.conf`.

To ensure the HugePages fit into the shared memory space without issue, we made the shared memory device slightly larger – 605 GB – by editing `/etc/fstab` and adjusting the `/dev/shm` device. By default, on Linux the device `/dev/shm` is sized to only one-half of DRAM (384 GB on our test server) and HugePages is disabled.

## Test Software

The following test software was installed on the server:

- HammerDB version 2.16 for Linux (64-bit). This software was used to create the test schemas and to execute all tests. The schemas and data follow the general conventions of TPC-C and TPC-H, but the test procedures do not. For example, all tests were run directly on the database server rather than using test terminals.
- Flexible IO Tester (fio) version 2.2.7 was used to test each storage device prior to installing Oracle, just to make sure all devices were performing acceptably.
- i7z was used to test the CPUs for c-state issues, after both the BIOS and operating system were configured to disable c-states. The i7z utility reported all processors operating at c-state 0 as desired.

## Oracle Database Software

The RDBMS is Oracle Database Server Enterprise Edition version 12.1.0.2 with Grid Infrastructure (ASM). The software was obtained from My Oracle Support using patchset ID number 17694377. Only .zip files 1-4 were downloaded and installed. Oracle automatically installs many optional (separately licensed) software products, but only the options listed below were used in our testing:

- Oracle Enterprise Manager Diagnostic and Tuning Packs, required for generating AWR reports
- Oracle Partitioning: Only table TPCC.ORDER\_LINES was partitioned. No other tables or indexes in any application schemas were partitioned.

## Oracle ASM Configuration

Oracle Automatic Storage Management (ASM) is used as the file system and logical volume manager. The server has one ASM instance named +ASM, and two ASM disk groups named GRID and DATA, as described below. All disk groups use default settings for sector size, allocation unit size, etc.

- The GRID disk group is created during installation of the Oracle Grid Infrastructure software. In non-RAC environments it provides a simple place holder for the ASM instance. It holds only the ASM instance's parameter file and password file, but in RAC environments it also holds CRS metadata. Our GRID disk group uses a 1 GB partition from all eight hard disks and the one flash device, and it is configured with High Redundancy.
- The DATA disk group uses the remaining capacity from each storage device and is configured with External Redundancy. The purpose of DATA is to hold all database files, including data files, redo logs, temp, etc. To start, the DATA disk group used all eight hard disk drives and none of the flash memory. In the second round of testing, one flash device was added and the hard disks removed, so that ASM moved the database from the old to the new storage. ASM automatically moves extents to new devices as they are added and removes extents from the old devices as they are ejected.

## Oracle Database Configuration

Two databases named DB1 and DB2 were created using the Oracle Database Configuration Assistant (DBCA). Both databases share one ASM disk group named DATA. The first database (DB1) supports OLTP workloads: it uses an 8KB block size and does not use parallelism except during maintenance operations. The second database (DB2) supports DSS workloads: it uses a 32KB block size and parallel query processes.

All database files for both databases are stored on the ASM disk group "DATA". The control files and online redo logs are multiplexed, so there are two copies of each of these files written in parallel by Oracle. (This is necessary to protect the database, as it is stored on an ASM disk group without redundancy or underlying RAID protection.) The on-line redo logs are 5 GB per member, 2 members per group, and 5 groups per database.

All tablespaces are Bigfile Tablespaces and use the database's default block size. All system datafiles and tempfiles were created with an initial size of 1 GB with AUTOEXTEND ON NEXT 1G. Application tablespace datafiles are documented in the section on HammerDB Configuration.

## HammerDB Configuration

Tablespaces for the HammerDB schemas were created in SQL\*Plus using the default names suggested by HammerDB, which are TPCCTAB and TPCHTAB. Both tablespaces were created as a Bigfile Tablespaces of size 500 GB with attributes AUTOEXTEND ON NEXT 100G MAXSIZE 1T. Tablespace TPCCTAB was created in database DB1, and tablespace TPCHTAB was created in database DB2.

Schemas were created and populated using HammerDB. The schemas generally follow the guidelines of the TPC-C and TPC-H specification; however, no attempt was made to perform actual TPC benchmarking. The TPC-C-like schema was used for OLTP testing and only implemented in database DB1, and the TPC-H-Like schema was used for DSS testing and only implemented in database DB2.

The scale, or size, of each schema is noted below:

- The scale of the TPC-C-Like schema is 5,000. This equates to approximately 275 GB of data and 130 GB of indexes, for a total schema size of 405 GB. This schema grows during testing, and so the scale 5,000 was appropriate for the SGA size. Table ORDER\_LINE is partitioned by hash, but no other tables are partitioned, as this is the default behavior when partitioning is selected in the HammerDB application.
- The scale of the TPC-H-Like schema is 300. The largest table LINEITEM starts with 1.8 billion rows, which consumes 234 GB of space, not counting indexes. The total schema size is just under 400 GB, with 340 GB of data and 50 GB of indexes. Oracle Partitioning is not used, as it is not available in the HammerDB application for TPC-H-like schemas. Users are free to manually partition tables and indexes, but this was not done in our testing.

©2017 Western Digital Corporation or its affiliates. All rights reserved. Western Digital, SanDisk, the SanDisk logo, Fusion ioMemory, Fusion ioSphere, SanDisk ION Accelerator, VSL and ioDrive are registered trademarks or trademarks of Western Digital Corporation or its affiliates in the US and/or other countries. All other marks are the property of their respective owners. Mellanox and ConnectX are registered trademarks of Mellanox Technologies, Ltd. IBM is a trademark of International Business Machines Corporation, registered in many jurisdictions worldwide. One MB is equal to one million bytes, one GB is equal to one billion bytes, one TB equals 1,000GB and one PB equals 1,000TB when referring to HDD/SSD capacity. Accessible capacity will vary from the stated capacity due to formatting and partitioning of the HDD/SSD drive, the computer's operating system, and other factors.

Western Digital Technologies, Inc., is the seller of record and licensee in the Americas of SanDisk® products.

DPL2015-02 EN 20170628