Accelerating Oracle Transactional Databases with SanDisk® Flash Storage

Author: Prasad Venkatachar

SanDisk Data Propulsion Labs

August 2015

SanDisk®
a Western Digital brand

Western Digital Technologies, Inc.
951 SanDisk Drive, Milpitas, CA 95035

www.SanDisk.com

Table of Contents

Executive Summary

Oracle Database 12c brings an interesting set of new features to the table, such as pluggable databases for cloud environments and In-Memory option for accelerating analytical and transactional workloads. These enhancements can help DBAs meet performance objectives, but only if the data is supplied to the Oracle database at the speed at which the Oracle engine requires. Solid state disks can meet these data access requirements, thus accelerating applications for cloud, transaction, and data warehouse workloads. This white paper provides complete insight into the transactional workload performance advantages of using SanDisk flash storage for Oracle Database 12c. It also highlights different approaches and corresponding benefits for implementing Oracle Database on flash storage.

Historically, short-stroking a spinning hard disk drive was a commonly used technique for increasing storage performance and accelerating database applications. However, today the practice of short stroking drives is passé. It adds a considerable amount of work for storage admins, with a significant reduction in storage capacity, and results in only minimal performance improvements. The emergence of flash storage has provided database administrators (DBAs) new ways to leverage databases storage such as Oracle. Using flash storage shifts the DBA's efforts from troubleshooting database storage issues to optimizing stored procedures and packages for additional database improvements.

Solution Objectives

The testing reported in this white paper has the following objectives:

- Highlight the benefits of using SanDisk solid state drives for Oracle database 12c transactional workloads
- Provide Oracle 12c performance comparison points, using flash disks vs. short-stroked, high-performance spinning drives
- Provide various options for deploying flash storage to an Oracle 12c database
- Evaluate the peak performance of a single-instance Oracle 12c database on a Lenovo x3650 server using SanDisk Optimus drives

Solution Overview

The solution discussed in this white paper demonstrates the deployment of a single-instance Oracle 12c database on a Lenovo x3650 server. The database storage is initially configured using JBODs with short-stroked, high-performance hard disk drives and is later switched to SanDisk Optimus solid state drives. The SanDisk flash implementation for Oracle database 12c was tested with the following configuration:

- Oracle Database Smart Flash Cache
- Oracle redo log files on flash storage
- Oracle partial schema on flash storage
- Oracle full schema on flash storage

The solution uses the following deployment configuration:

- Oracle 12c Database Server Enterprise Edition
- Lenovo x3650 server with SanDisk Optimus drives using Oracle Linux 6.5
- Oracle ASM using ASM libraries with external redundancy for ASM disk groups
- A single instance Oracle 12c database loaded with a 500GB transactional schema
- On-line transaction processing (OLTP) workload generated by HammerDB

Figure 1: Test environment layout

Component Details

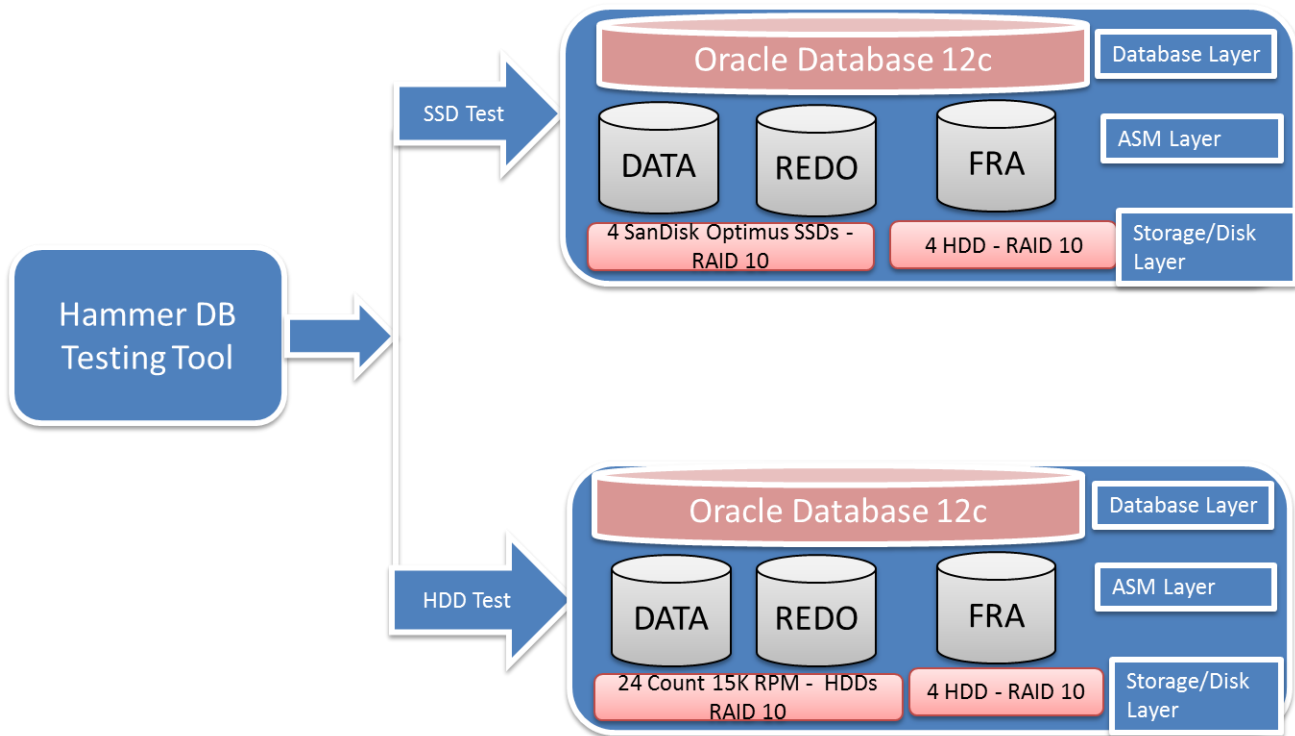| Component	Version
Server and Client	(2) IBM x3650 Intel Xeon E5-2690 2-socket, 20 cores, 64GB RAM
Storage SSD Configuration: HDD Configuration:	4 SanDisk Optimus SSDs, 800GB each, RAID 10 24 SAS 15K RPM HDDs, 300GB each (short-stroked to 50%, or 150GB), RAID 10
Operating System	Oracle Linux 6.5
Database	Oracle Database 12c (12.1.0.2)

Figure 2: Test Execution

Solution Components

SanDisk Optimus SAS SSD

SanDisk is a leader in flash storage solutions. SanDisk's portfolio of solid-state drives (SSDs) supports the megatrends in the industry that are driving new application deployments, including cloud, big data/analytics, mobility, and social media. Because many of these new applications store their data in Oracle databases, Oracle NoSQL and these applications must deliver optimized performance to support timely business results.

SanDisk Optimus SAS SSDs address a wide range of enterprise storage and server applications. Optimus SAS SSDs support high performance for applications and databases while providing consistent quality of service (QoS). Their wide range of storage capacities, from 200GB to 4TB (Optimus MAX™ SSD), enables more flexibility for IT deployments. It is worth noting that the Mean Time Between Failures (MTBF) for the SanDisk Optimus SAS SSDs is very high—2.5 Million Hours MTBF—with a warranty period of five years. Optimus SAS SSDs are protected by SanDisk's Guardian™ Technology platform, which provides FlashGuard™, DataGuard™, and EverGuard™ capabilities that increase the durability, recoverability, and prevention of data loss and corruption.

Specifically,

- FlashGuard technology reliably extracts significantly more usable life from MLC flash than would otherwise be provided by the standard flash specifications

- DataGuard technology provides full data-path protection, ensuring that data will be safe throughout the entire data path, and provides the ability to recover data from failed pages and NAND blocks

- EverGuard technology prevents the loss of user data during unexpected power interruptions

For additional details, refer to the Guardian Technology data sheet on http://www.sandisk.com/.

Oracle Database 12c

Oracle Database 12c, released in June, 2013, has introduced many new features, including expanded support for cloud-based workloads, support for storage tiering in data warehouse environments using segment level heat maps, and other enhancements for performance, security, and data protection. (Details can be found on the Oracle website at www.oracle.com.) These new features provide a broad range of new capabilities for the worldwide database marketplace leveraging Oracle database products.

For example, Oracle Multitenant introduces a new architecture that supports a multitenant container database that holds many pluggable databases. This enables better support for cloud workloads in Oracle Database 12c, so customers can consolidate multiple databases into a single-container database for easier management and IT efficiency. This approach reduces server sprawl (particularly for virtual servers, or VMs, that support databases), a chief concern of first-generation adopters of virtualization in database environments. The enhancements to Oracle Database 12c build on the benefits of modern solid state disk technology, enabling customers to achieve a better consolidation ratio than possible with legacy hard disk storage.

This paper is focused mainly on SanDisk Optimus solid storage device vs. short-stroked, spinning disk performance in Oracle Database 12c transaction workloads. The rest of this paper provides more details about the specific testing that was done to demonstrate the advantages of using SSDs in Oracle database deployments.

Solution Testing

The solution test used HammerDB to execute a transactional workload on an Oracle 12c single-instance database. The Lenovo server setup configuration and database setup were kept consistent for both HDD and SSDs tests. HammerDB was configured with 100 users and tests were executed for 60 minutes duration. The transaction output metrics of Transactions per Minute (TPM) and New Orders per Minute were captured from the workload response perspective. To evaluate database performance, an Automatic Workload Repository (AWR) report was generated for every test execution to identify the top wait events. The system performance metrics were also captured, to identify any system bottlenecks. These three performance metrics of workload response, database performance, and system throughput were utilized to see the overall performance advantages provided by SanDisk Optimus SSD's over a large quantity of high performance short-stroked spinning hard disk drives.

Baseline Testing

The performance metrics from the hard disks served as the baseline for comparison. During the Oracle transaction workload test, average user CPU utilization was measured at about 15 percent, and CPU I/O wait time averaged around 50 percent. Using Oracle AWR reports helped to identify I/O issues from the Oracle database perspective.

AWR reports showed 14.4 million DB File Sequential Read wait events, or single-block data requests that needed to be fetched from disk drives to process the transactions. The short-stroked drives served DB File Sequential Read Requests at 13.61 ms latency, contributing 55.1percent of the database time for the test. The next most frequent wait event was Log File Sync. It used 34.9 percent of the database time, although the actual work done by Oracle during test duration was only 5 percent of the database time. This is a clear indication that Oracle

waited for data to be supplied from storage (the HDDs). Figure 3 shows the top 10 wait events from the AWR report.

Wait Classes by Total Wait Time

Wait Class	Waits	Total Wait Time (sec)	Avg Wait (ms)	% DB time	Avg Active Sessions
User I/O	14,404,027	196,059	13.61	55.1	54.0
Commit	8,272,975	138,131	16.70	38.8	38.1
System I/O	2,474,156	21,987	8.89	6.2	6.1
DB CPU		17,457		4.9	4.8
Configuration	2,141	2,202	1028.64	.6	0.6
Concurrency	1,171,967	1,719	1.47	.5	0.5
Application	19,647	966	49.17	.3	0.3
Other	68,609	334	4.86	.1	0.1
Network	9,060,098	12	0.00	.0	0.0

Figure 3: HDD test Oracle AWR performance baseline

The section below provides different flash implementation approaches to address these wait events and increase transaction throughput.

Flash Implementation Approach 1: Oracle Smart Flash Cache

To take advantage of flash storage benefits for database performance, Oracle introduced the Database Smart Flash Cache feature starting with Oracle 11g Release 2. This feature transparently extends the Oracle database buffer cache from main memory to a second level cache on flash memory. (Details of this feature and its implementation are discussed in the SanDisk white paper "Oracle Smart Flash Cache using SanDisk Flash".) With this featured enabled, tests showed a 136 percent improvement in transactions (see Figure 4 below) from the baseline performance. This performance improvement is due to single-block data requests being serviced from flash storage: metric `DB flash cache single block physical read` indicates the Flash Cache serviced 19.9 million I/O requests with 0.28 millisecond latency, as shown in Figure 4. Oracle served 6 million additional single-block requests from the hard disk storage with an average wait time of 13.06 ms. Comparing the performance of the two storage mediums, the flash storage latency (0.28 ms) was 48x better than the hard disk latency (13.61 ms). Latency reduction for single-block requests with Flash Cache improved CPU utilization to 6.7 percent during the test, compared to 4.9 percent with the baseline testing.

Top 10 Foreground Events by Total Wait Time

Event	Waits	Total Wait Time (sec)	Wait Avg(ms)	% DB time	Wait Class
log file sync	11,236,170	238.5K	21.23	67.5	Commit
db file sequential read	6,026,655	78.7K	13.06	22.3	User I/O
DB CPU		25.2K		7.1	
db flash cache single block physical read	19,933,363	5558.7	0.28	1.6	User I/O
library cache: mutex X	2,350,834	4100.9	1.74	1.2	Concurrency
enq: TX - row lock contention	39,292	1246.7	31.73	.4	Application
log file switch (private strand flush incomplete)	1,771	253.2	142.98	.1	Configuration
db file scattered read	11,058	183.7	16.61	.1	User I/O
cursor: pin S	96,647	105.8	1.09	.0	Concurrency
write complete waits: flash cache	494	83.6	169.28	.0	Configuration

Figure 4: Oracle AWR performance Top 10 wait events – *Approach 1* testing
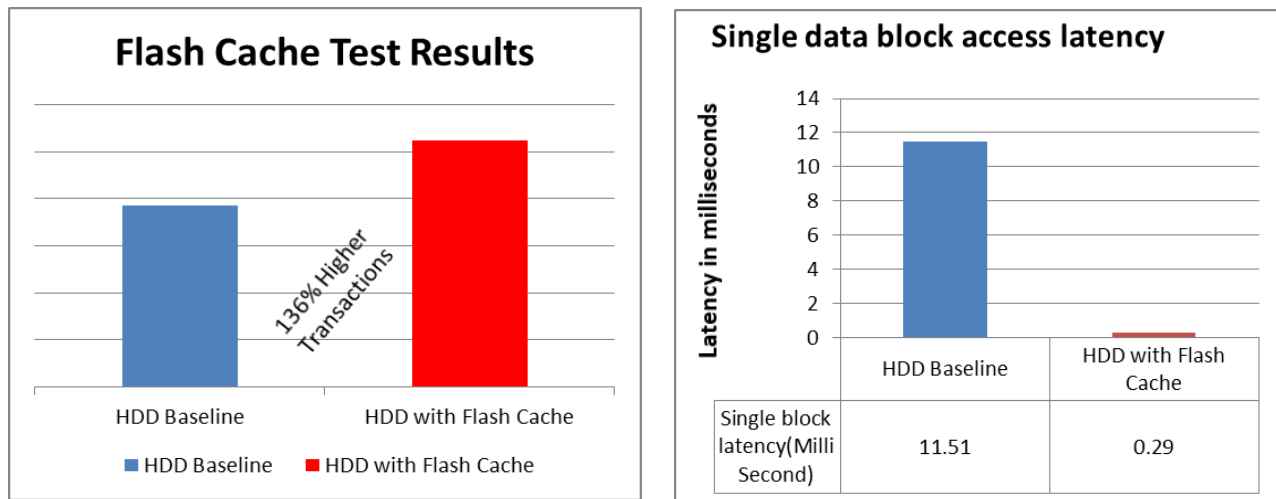

Figure 5: Oracle 12c Flash Cache performance and DB file sequential read latency comparison

Flash Implementation Approach 2: Oracle Redo Log Files on Flash Storage

As noted earlier the Oracle Database Smart Flash Cache (Flash implementation approach 1) mitigated the top wait event of single-block reads to a great extent and increased transactional throughput. The second top priority wait event from AWR reporting is the Logfile Sync Event. This wait event becomes a major bottleneck in OLTP environments, resulting in significant reduction in transaction throughput. Logfile sync events are triggered when a user session commits the transaction, and the contents of the log buffer must be written to the redo log file to confirm that the transaction is committed and fully secured. Figure 6 gives a high-level view of the Oracle database commit operation.
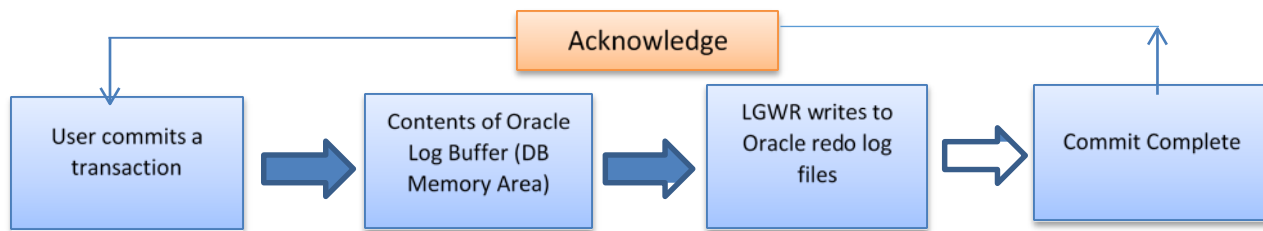
Figure 6: Oracle commit operation and redo log file sync event

The majority of this wait event depends on write I/O operations from the LGWR to redo log files. The speed of the I/O subsystem is major factor in getting commit acknowledgement back to the user process. For high-transactional workload scenarios, placing redo log files on flash storage helps provide the required log file acceleration. In flash implementation approach 2, log files were migrated from hard disk storage to the ASM REDO disk group that was provisioned using SanDisk Optimus SSDs. The sector size of the log file was changed from a default of 512 bytes to 4KB. It was confirmed during testing that increasing the sector size from 512 bytes to 4KB provided additional improvement in transaction throughput: the HammerDB transaction workload test results showed a 63 percent improvement. Latency for the logfile sync wait event was reduced from the baseline of 17.51 milliseconds to 0.27 milliseconds, and DB CPU utilization increased from the baseline of 4.9 percent to 7.2 percent.

Top 10 Foreground Events by Total Wait Time

Event	Waits	Total Wait Time (sec)	Wait Avg(ms)	% DB time	Wait Class
db file sequential read	14,343,567	253K	17.64	70.4	User I/O
log file switch (checkpoint incomplete)	1,960	51.3K	26175.84	14.3	Configuration
DB CPU		25.8K		7.2	
free buffer waits	5,074	22.1K	4356.45	6.2	Configuration
log file sync	13,476,872	3581.2	0.27	1.0	Commit
enq: TX - row lock contention	38,968	2407.5	61.78	.7	Application
db flash cache single block physical read	3,779,024	875.1	0.23	.2	User I/O
db file scattered read	10,996	548.5	49.88	.2	User I/O
read by other session	24,230	462.5	19.09	.1	User I/O
enq: FB - contention	55	261.8	4760.43	.1	Other

Figure 7: Oracle AWR performance Top 10 wait events – *Approach 2* testing

Figure 8 shows the transactions throughput and redo log file access latency comparison of HDD baseline to redo log files on flash storage.
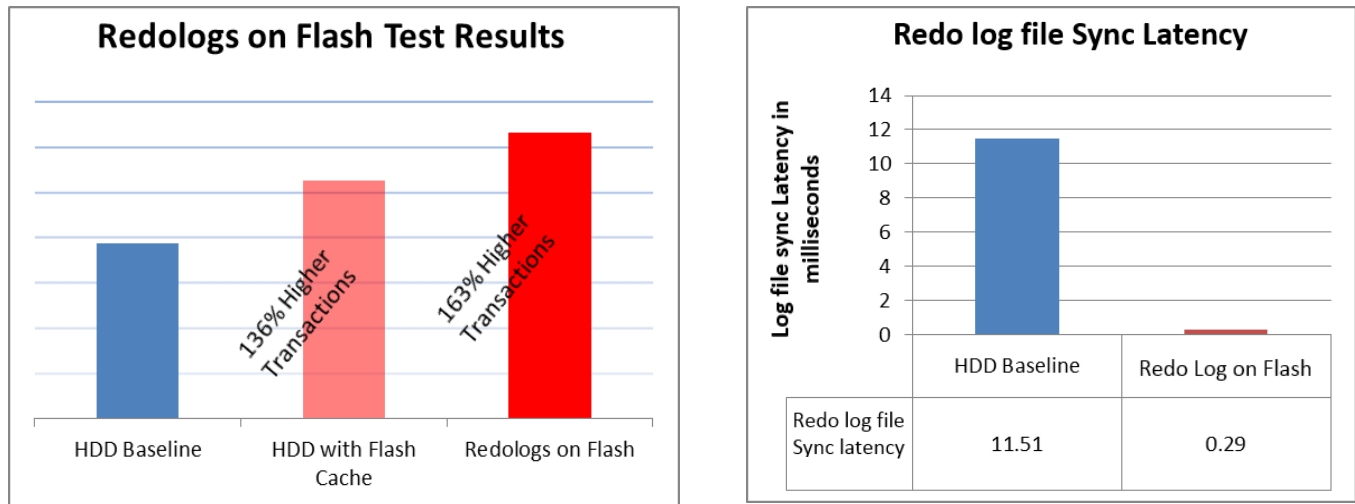
9

Figure 8: Redo log files on flash performance and Redo log file sync latency comparison

Flash Implementation Approach 3: Partial Transactional Schema on Flash Storage

Flash implementation approaches 1 and 2 yielded higher transactional throughput by minimizing I/O wait events such as single-block read and log file sync. A third approach is to identify candidate database objects that could take advantage of flash storage to increase overall transactional throughput of the database. Such frequently accessed tables and indexes can be identified in the AWR report. Once the hot objects were identified, they were moved to flash storage, with the rest of the database objects retained on spinning disk.

To implement approach 3, two new tablespaces named SSDDATA and SSDINDX were created on the flash storage ASM diskgroup. Hot tables were moved to SSDDATA, and frequently-used indexes were rebuilt on the SSDINDX tablespace. With these changes, the HammerDB workload test execution resulted in a massive 302 percent improvement in transaction count. DB CPU utilization increased up to 11.6 percent and system CPU utilization up to 40 percent, based on iostat reporting. The increased DB CPU and system CPU utilization indicates that the Oracle database engine was driving more work and waiting less for data to be supplied from storage.

Top 10 Foreground Events by Total Wait Time

Event	Waits	Total Wait Time (sec)	Wait Avg(ms)	% DB time	Wait Class
db file sequential read	34,743,844	200.8K	5.78	69.3	User I/O
DB CPU		52.4K		18.1	
log file switch (checkpoint incomplete)	3,035	21.4K	7039.19	7.4	Configuration
log file sync	24,907,581	7263.1	0.29	2.5	Commit
free buffer waits	1,439	3671.9	2551.67	1.3	Configuration
enq: TX - row lock contention	59,326	2401.1	40.47	.8	Application
read by other session	69,711	1517.7	21.77	.5	User I/O
buffer busy waits	56,183	312.4	5.56	.1	Concurrency
library cache: mutex X	155,829	299.9	1.92	.1	Concurrency
cursor: pin S	217,999	243.1	1.11	.1	Concurrency

Figure 9: Oracle AWR performance Top 10 wait events – *Approach 3* testing

Figure 10 shows the transaction count comparison chart of approaches 1, 2, and 3:
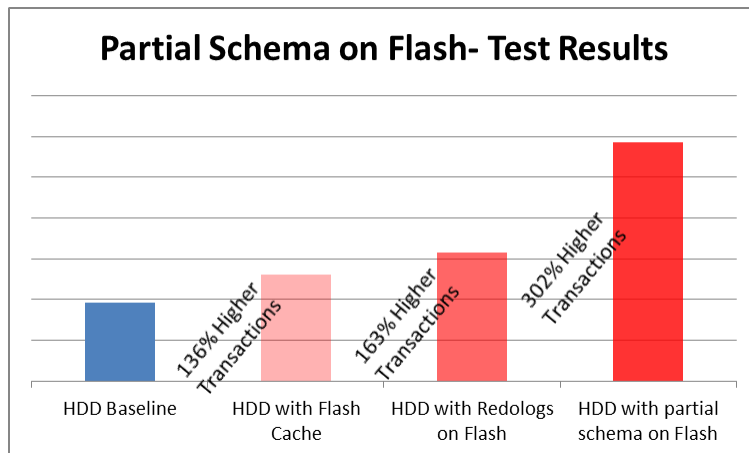

Figure 10: Storage performance comparison - partial schema on flash

Flash Implementation Approach 4: Full Transactional schema on Flash Storage

Flash implementation approaches 1 through 3 increased transactional throughput and greatly reduced Oracle I/O wait events. The AWR reports and system metrics, however, suggest that CPU utilization was not fully optimized to drive peak transactional throughput from the server, and that some CPU cycles were spent waiting for I/O. The slow I/O occurred because some data was still being read from or written to the legacy hard disks drives, even after moving frequently accessed data to flash storage.

Approach 4 involves placing the entire transactional schema on flash storage to drive more transactions and fully take advantage of system resources. This configuration setup involves placing the redo log and data files on the ASM disk groups created on Optimus SSD drives. The Oracle Database Smart Flash Cache was disabled for this test. The HammerDB benchmark workload was executed, and test results showed a significant 538 percent (5.3X) increase in transaction count. The average transactions per any given minute during the full 60-minute test run showed an amazing 6-digit transactions count. AWR reported that DB CPU utilization increased from 4 percent on the HDD baseline to 35.7 percent. This approach serves as an excellent configuration choice for customer-facing applications needing high-transaction throughput, with minimal system and database latency.

Top 10 Foreground Events by Total Wait Time

Event	Waits	Total Wait Time (sec)	Wait Avg(ms)	% DB time	Wait Class
DB CPU		115.3K		36.1	
db file sequential read	80,900,010	93.1K	1.15	29.2	User I/O
log file sync	42,783,790	90.4K	2.11	28.3	Commit
library cache: mutex X	1,460,527	4174	2.86	1.3	Concurrency
cursor: pin S	1,929,027	3286.2	1.70	1.0	Concurrency
buffer busy waits	249,163	1992.4	8.00	.6	Concurrency
latch: enqueue hash chains	444,775	1658.9	3.73	.5	Other
latch: In memory undo latch	1,459,564	757	0.52	.2	Concurrency
enq: TX - row lock contention	78,964	313.4	3.97	.1	Application
log file switch (private strand flush incomplete)	5,251	221.4	42.17	.1	Configuration

Figure 11: Oracle AWR performance Top 10 wait events – *Approach 4* testing

Figure 12 shows the transaction comparison chart for various test configurations. Having the full transactional schema on SSD delivered the highest transaction throughput.
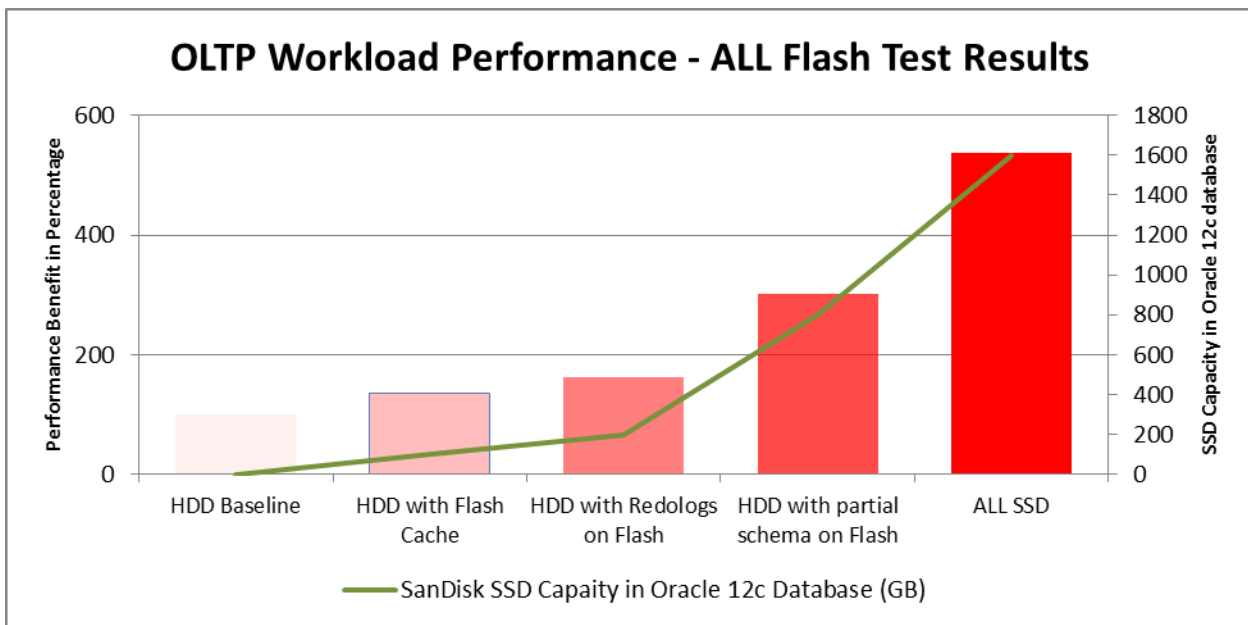

Figure 12: Performance comparison - flash storage implementation

To demonstrate the CPU utilization benefits with this approach, Figure 13 provides a comparison of the HDD baseline test versus the all SSD tests. The HDD baseline tests show user CPU utilization at just under 15 percent of the system's 20 cores (40 hyper-threads) and I/O wait at around 50 percent. By contrast, when the full schema was on SanDisk SSDs, the user CPU utilization averaged 75 percent and I/O waits dropped below 11

percent. These CPU utilization metrics highlight the significant benefit of using SSD storage for Oracle database: it enables Oracle to deliver higher transaction throughput by supplying the data at higher speeds, minimizing the I/O waits. HDD storage kept Oracle Database in wait mode, while SSD storage allowed the Oracle engine to switch from wait mode to work mode.
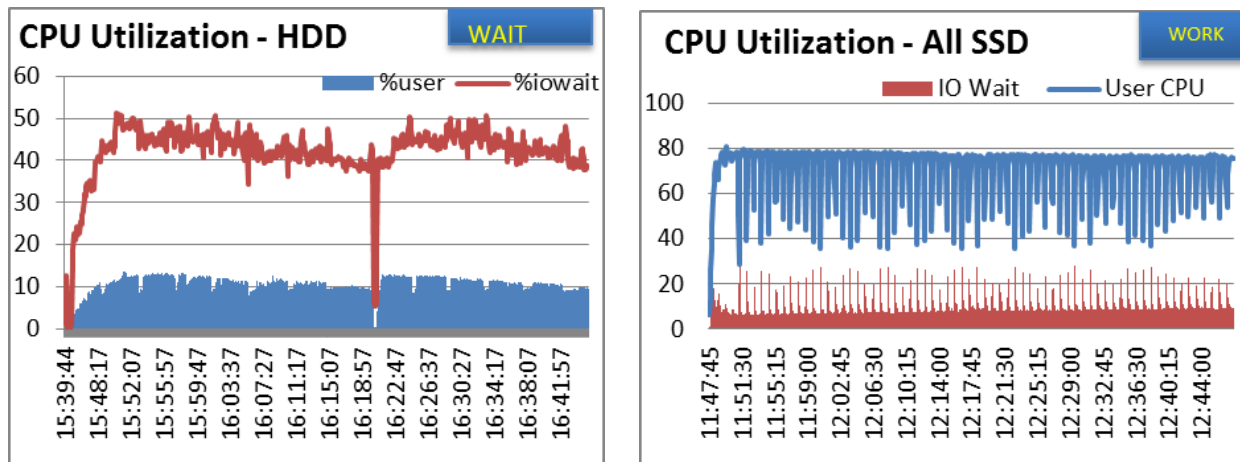

Figure 13: CPU utilization comparison chart – all HDD vs. all SSD

Figure 14 below shows the activity distribution for the test period. The figure was produced from Oracle Enterprise Manager and represents a 4 ½ minute window while the HammerDB benchmark was executing.
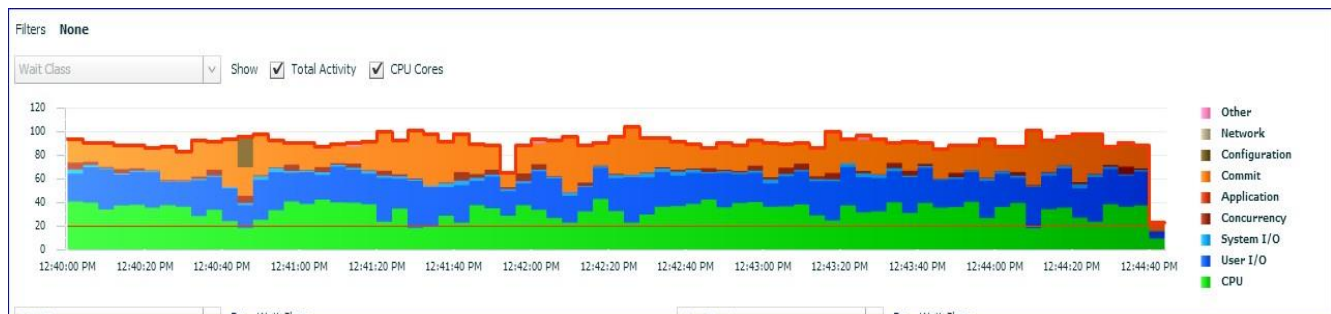

Figure 14: Oracle Enterprise Manager Performance report during ALL SSD test

The above figures show the system CPUs were carrying out database transactions, data requests were fulfilled with user I/O operations, and transactions were completed with commit operations. No other bottlenecks in system I/O, database configuration issues, or network issues were seen in the graph. The results of this report coincide with those of the AWR performance report shown in Figure 12.

Cost Benefits Analysis

The table below summarizes the cost and performance benefits of using the SanDisk Optimus SSD solution for Oracle OLTP workloads. Only the storage costs are considered for highlighting the SanDisk SSD benefits: server infrastructure and Oracle Database configuration setup remained constant during the test as database storage was switched from legacy hard disk drives to modern SanDisk SSDs. The table below shows that the *SanDisk SSDs delivered 5X the performance benefit for Oracle OLTP workloads, at 63 percent lower cost than short-stroked, high-performance spinning disk drives*. Because it is important to address both reliability and high

performance and follow Oracle best practices for OLTP workload, RAID 10 configuration was used. It is possible to use just three SanDisk SSDs with a RAID 5 configuration to further reduce storage costs while maintaining performance.

Drives	SanDisk Optimus SSD	HDD
Total Number of Disks/Drives	4	24
RAID for OLTP Workload	RAID 10	RAID 10
Drive Cost	$ 1567 x 4 = $ 6268	$ 419 X 24 = $ 10056
Enclosure	---	$ 4495
Total	$6268	$ 14551
Cost Benefit Percentage	63%	0%
OLTP Performance Benefit	1X transaction count	5X transaction count
SQL IOPS during OLTP Test	32000 IOPS	5500 IOPS

Figure 15: Storage cost benefit analysis

Conclusion

This white paper demonstrated the various SanDisk flash implementation approaches and related performance benefits for Oracle Database 12c, single-instance, on a Lenovo X3650 server. In our performance benchmarking tests, we demonstrated how four SanDisk Optimus SSDs outperformed 24 short-stroked, high-performance 15K RPM hard disk drives. Using part of a single Optimus SSD with Oracle Flash Cache, delivered a 136 percent performance benefit increase and up to 538 percent benefit at 63 percent lower cost in the case of full transactional schema on the all SanDisk SSD configuration. These test results will help customers evaluate the advantages of a SanDisk flash implementation, starting with Oracle Database 12c Flash Cache and extending to placing the entire application schema on SanDisk flash storage.

This white paper highlights how significant storage bottlenecks can be avoided for Oracle transaction workloads, helping the database administrator focus on other important tasks. The SanDisk Optimus SSDs are protected by the Guardian Technology platform, which reduces the flash wear and protects data as required by Oracle transactional systems. With significant advantages in performance, cost, and reliability (with the Guardian Technology platform), customers can safely transition from traditional short-stroked, spinning disks to SanDisk flash storage such as Optimus SSDs.