

# EMC Symmetrix DMX-4 Enterprise Flash Drives with Microsoft SQL Server Databases

*Applied Technology*

---

**Abstract**

This white paper examines many of the implementation details for Microsoft SQL Server databases on EMC<sup>®</sup> Symmetrix<sup>®</sup> enterprise Flash drives. A comparison is made with existing drive and allocation methodologies used by administrators to obtain optimal performance from traditional disk drive technologies.

January 2009

---

---

Copyright © 2009 EMC Corporation. All rights reserved.

EMC believes the information in this publication is accurate as of its publication date. The information is subject to change without notice.

THE INFORMATION IN THIS PUBLICATION IS PROVIDED “AS IS.” EMC CORPORATION MAKES NO REPRESENTATIONS OR WARRANTIES OF ANY KIND WITH RESPECT TO THE INFORMATION IN THIS PUBLICATION, AND SPECIFICALLY DISCLAIMS IMPLIED WARRANTIES OF MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE.

Use, copying, and distribution of any EMC software described in this publication requires an applicable software license.

For the most up-to-date listing of EMC product names, see EMC Corporation Trademarks on [EMC.com](http://EMC.com)

All other trademarks used herein are the property of their respective owners.

Part Number h6018

---

## Table of Contents

<b>Executive summary .....</b>	<b>4</b>
<b>Introduction .....</b>	<b>4</b>
Audience .....	4
<b>Technology overview .....</b>	<b>5</b>
Symmetrix DMX-4 .....	5
Symmetrix DMX enterprise Flash drives .....	5
Microsoft SQL Server 2008.....	5
<b>EFD vs. HDD .....</b>	<b>6</b>
Current methods for database storage design .....	7
Database workloads best suited for enterprise Flash drives .....	8
Microsoft SQL Server log files .....	9
Microsoft SQL Server tempdb files .....	9
Host I/O operations versus physical I/O operations.....	9
<b>SQL Server 2008 OLTP databases and Flash drives .....</b>	<b>11</b>
Test configuration .....	11
Server configuration .....	11
Enterprise Flash drive configuration.....	11
Traditional hard disk drive configuration .....	16
<b>Conclusion .....</b>	<b>20</b>
<b>Appendix: Flash drives and Information Lifecycle Management strategy .....</b>	<b>21</b>

---

## Executive summary

EMC has enhanced the latest release of Symmetrix® Enginuity™ version 5773 to integrate enterprise-class Flash drives (EFDs) into Symmetrix DMX-4 storage arrays. EMC® Symmetrix is the first and only enterprise array with support for this emerging generation of drive technology. With this capability, EMC creates a new “Tier 0” ultra-performance storage tier that removes the performance limitations previously imposed by magnetic disk drives. EFDs also provide substantial total cost of ownership (TCO) advantages over traditional disk drives, by virtue of lower power consumption, weight, and heat dissipation. By combining EFDs optimized with EMC technology and advanced Symmetrix functionality, organizations now have new options previously unavailable from any enterprise storage vendor.

EFDs dramatically increase performance for read latency sensitive applications. EFDs, also known as solid state drives (SSD), contain no moving parts and appear as standard Fibre Channel drives to existing Symmetrix management tools, allowing administrators to manage Tier 0 without special processes or custom tools. Tier 0 enterprise Flash storage is ideally suited for applications with high transaction rates and those requiring the fastest possible retrieval and storage of data, such as currency exchange and electronic trading systems, or real-time data acquisition and processing. A Symmetrix DMX-4 with Flash drives can deliver single-millisecond application response times and significantly more input/output operations per second (IOPS) than traditional Fibre Channel hard disk drives (HDD). Additionally, because there are no mechanical components, Flash drives consume up to 98% less energy per I/O than traditional hard disk drives.

## Introduction

This white paper examines deployments of SQL Server 2008 using enterprise Flash drives. It shows that Flash drives deliver vastly increased performance to the database application when compared to traditional Fibre Channel drives, both in transaction rates per minute as well as transaction response time. In the tested configuration, each Symmetrix DMX-4 EFD device provided an I/O rate of 2,900 IOPS with an average response time of less than 5 ms. Testing for the EFD configuration was limited by CPU utilization. In contrast, the HDD configuration provided 250 IOPS at a latency of 17 ms. This represents an I/O improvement of 1,140%, based on a comparison of four enterprise Flash drives to 34 traditional Fibre Channel drives.

Testing was conducted jointly by Microsoft SQL Server and EMC Symmetrix Partner Engineering within the Microsoft SQL Server Customer Advisory Team labs in Redmond (<http://www.sqlcat.com>).

## Audience

This white paper is intended for Microsoft SQL Server database administrators, storage architects, customers, and EMC field personnel who want to understand the implementation of enterprise Flash drives in Microsoft SQL Server environments to improve the performance of business applications and assist in reducing overall TCO.

---

## Technology overview

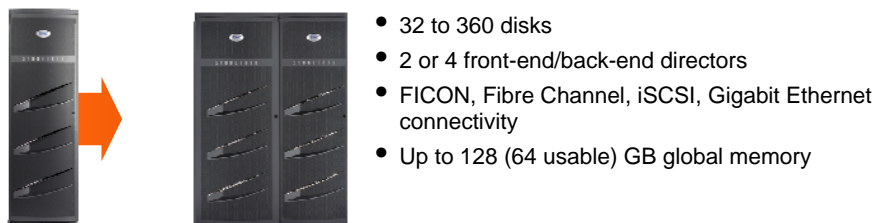
### **Symmetrix DMX-4**

The new Symmetrix DMX-4 system is the next generation in the Symmetrix DMX™ series and extends EMC's leadership in the high-end enterprise storage market. The DMX-4 delivers immediate support for the latest generation of disk drive technologies: Enterprise Flash drives and 4 Gb/s Fibre Channel drives for high performance and SATA II for high capacity. Symmetrix DMX-4 is the first and only high-end storage system that can support both of these latest generations of disk drive technologies. DMX-4 with the latest Enginuity release 5773 has been optimized for maximum performance and tiered storage functional flexibility.



*\*Combinations may be limited or restricted based on configuration*

**Figure 1. Symmetrix DMX-4: World's largest high-end storage array**



*\*Combinations may be limited or restricted based on configuration*

**Figure 2. Symmetrix DMX-4 950: Entry point for DMX-4 technology**

### **Symmetrix DMX enterprise Flash drives**

With the introduction of the Symmetrix DMX-4 running Enginuity 5773, EMC now supports enterprise-class Flash drives. The enterprise-class EMC Flash drives are constructed with nonvolatile semiconductor NAND Flash memory and are packaged in a standard 3.5-inch disk drive form factor used in existing Symmetrix DMX-4 drive array enclosures. These drives are especially well suited for low-latency applications that require consistently low read/write response times. These drives exhibit as much as 30 times improvement in IOPS over traditional HDD technology.

EFDs benefit from the advanced capabilities that Symmetrix provides, including local and remote replication, cache partitioning, and priority controls.

### **Microsoft SQL Server 2008**

SQL Server 2008 provides a scalable, high-performance database engine for mission-critical applications that require the highest levels of availability and security, while reducing TCO through enhanced enterprise-class manageability.

SQL Server 2008 delivers on Microsoft's Data Platform vision by helping an organization manage any data, any place, any time. You have the ability to store data from structured, semi-structured, and unstructured documents, such as images and rich media, directly within the database. SQL Server 2008 delivers a rich set of integrated services that enable you to do more with your data such as query, search, synchronize, report, and analyze.

SQL Server 2008 provides the highest levels of security, reliability, and scalability for your business-critical applications. To take advantage of new opportunities in today's fast-moving business world, companies need the ability to create and deploy data-driven solutions quickly. SQL Server 2008 reduces the time and cost of management and development of applications.

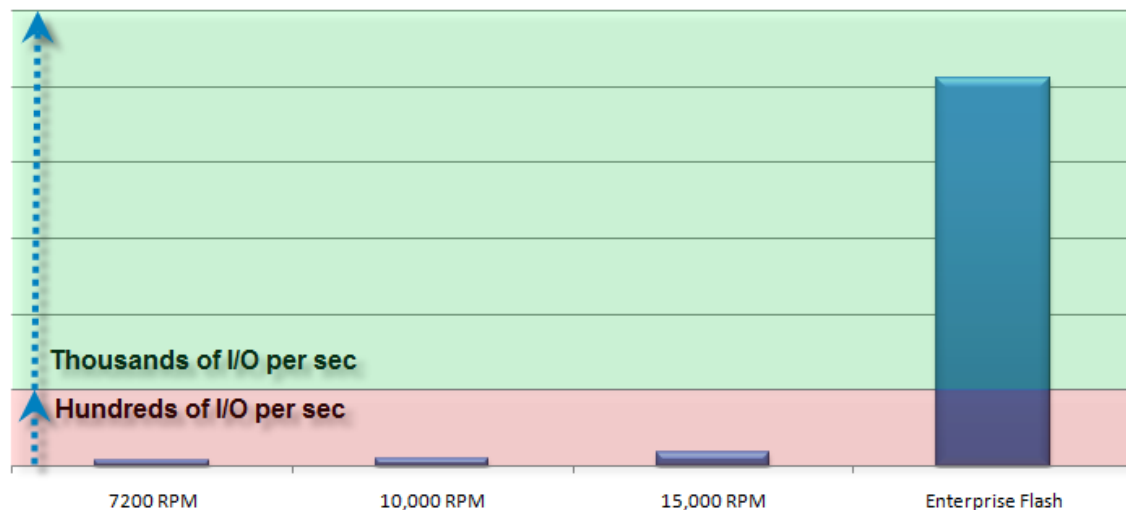
A complete set of features provided by each edition of SQL Server 2008 can be found at the Microsoft Developer Network site (<http://msdn.microsoft.com/en-us/library/cc645993.aspx>). In addition, more information about SQL Server in general can be found on the MSDN site (<http://www.microsoft.com/sqlserver/2008/en/us/default.aspx>).

## EFD vs. HDD

Database performance has long been constrained by the I/O capability of HDDs and the performance of the HDD has been limited by intrinsic mechanical delays of head seek and rotational latency. EFDs, however, have no moving parts and therefore no seek or rotational latency delays, which dramatically improves their ability to sustain very high numbers of IOPS with very low overall response times.

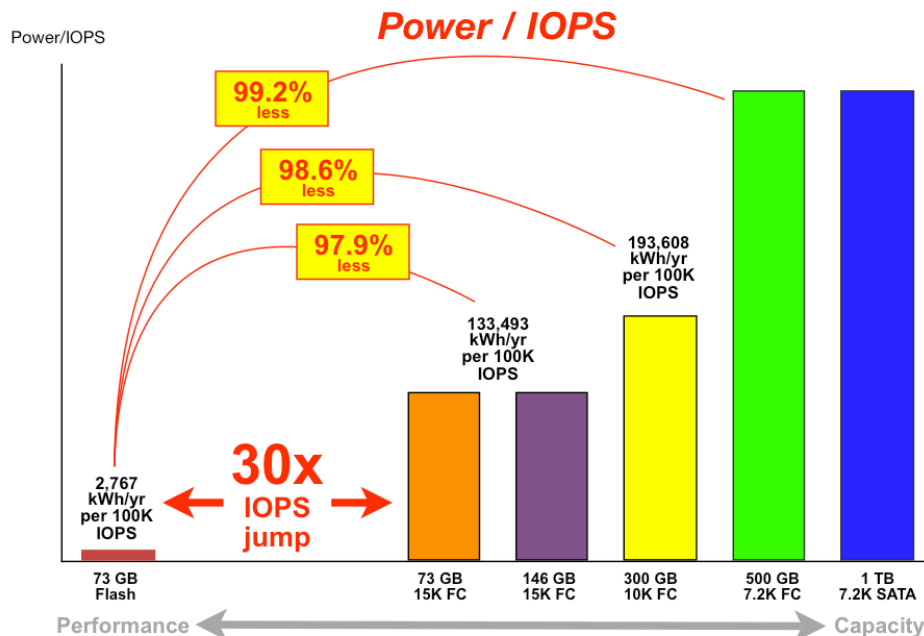
Figure 3 shows the relative I/O rates that can be sustained by traditional HDD based on average seek and latency times as compared to EFD technology for purely 8 KB random read workloads, which is typical of SQL Server OLTP environments. Over the past 25 years, the rotational speeds of HDDs have improved only from 3,600 rpm to 15,000 rpm, yielding only a 4x improvement in IOPS. EFD technology represents a significant leap in performance and can, given the right workload characteristics, sustain over 30 times the IOPS of traditional HDD technology.

### Enterprise Flash Drives versus Traditional Hard Disk Drives



**Figure 3. Relative random 8 KB Read IO/s of various drive technologies**

In addition to the performance improvements available for EFDs, other important criteria are also positively affected. Symmetrix DMX EFDs consume less power than traditional HDDs. Figure 4 shows the relative power consumption of EFDs and traditional spinning disk drives. Coupled with the ability to consolidate multiple drives worth of I/O capacity to a smaller number of EFDs, this can represent a substantial power saving option to customer deployments.



**Figure 4. Power consumption: HDD vs. EFD**

In data centers where space, or physical footprint, is a growing concern, the ability to consolidate also represents a valuable attribute. While the magnitude of the consolidation may be limited on a customer-by-customer basis, the ability to reduce both the overall footprint required, as well as floor weighting, are characteristics that add positive value to any TCO calculations.

### ***Current methods for database storage design***

When designing storage configurations for databases, administrators have typically dealt with two primary elements:

- Storage capacity requirements (how large the database itself is)
- I/O throughput (the number of I/Os per second required to deliver required response times to the host/application). This is referred to as I/O capability below, and is typically expressed in one of two attributes
  - The total number of I/O operations executed in a given interval
  - The total throughput expressed in MB or GB per second

With traditional spinning hard disk drive technology, available storage capacity (the size of the HDD) has continued to increase while the I/O capability provided by the drive has remained relatively static. The primary limitation of traditional disk drive technology has been the mechanical nature of the read/write heads.

To be able to service a read or write request, the read/write head is required to be located over the physical area where the data is or will reside – this physical activity of relocating the read/write head is referred to as a disk head seek. Disk head seeking can be one of the most time-consuming operations to service a read or write request. In general disk seek times are in the order of single-digit millisecond times. As traditional HDDs attempt to deal with increased read/write requests that are distributed across the entire storage provided by the drive, the seek time becomes a major contributor to read or write latency.

A secondary factor impacting performance is the rotational latency of the HDD itself. Once the read/write head has been relocated to the required cylinder to process the read or write operation, the actual location of the data is required to be directly under the read/write head. Typically, this rotational latency effect is significantly smaller than the latency introduced by disk head seeking. Faster rotating speeds can help

---

mitigate this rotational effect. EFDs do not suffer from rotational latencies as they have no moving components, so seek times can be assumed to be zero.

For both EFD and HDD to service a write or read request, data needs to be transferred to or from the physical drive itself. This data transfer operation is similar for EFD and HDD devices, and is an electronic operation that transfers the actual data contents within the storage array. One factor that causes longer transfer operations is the size of the data within the I/O operation itself. Small I/O operations, for example read requests for 8 KB of data, are processed much faster than larger I/O requests such as a read request for 256 KB of data. It is typical to consider smaller I/O operations in terms of the total IOPS, and larger operations in terms of throughput or MB/s. SQL Server will typically generate random 8 KB read requests for OLTP style activity, and larger 256 KB sequential read requests for operations such as table scans.

In an effort to reduce the effect of disk seek time, storage and database administrators have designed around the physical limitations of traditional HDD devices. One methodology implemented by many storage administrators has been to utilize only a small amount of storage capacity on the disk, thus reducing seek times significantly. This is often referred to as “short-stroking” disk drives. This technique attempts to mitigate the impact of disk head seek times by physically limiting the range that the disk head must move over to service I/O requests. In many deployments the approach of short-stoking drives is unavoidable because I/O sizing for enterprise class database application is best approached by sizing on the number of disks required to sustain the number of expected I/Os per second (IOPS) with reasonable response times.

Short-stroking a disk drive adversely affects the usable amount of storage available to servers and applications. As the range of the disk drive head is limited, so is the storage capacity. Thus a 146 GB HDD may be short-stroked by half to increase I/O performance, but only yield half of the storage capacity (73 GB). While the short-stroking strategy can provide performance improvements for application response times, it provides a very high TCO.

Limiting storage allocation in an effort to improve performance by utilizing a short-stroking methodology requires that more physical HDDs be allocated to the workload to provide the required storage capacity. In almost all cases, the storage allocation, or the size of the database, will remain as a constant and as the storage provided by each physical disk drive is artificially limited, more drives are required. Drives configured in this manner continue to consume their full requirement of power, cooling, floor space, and weighting.

This disparity of storage capacity and I/O capability and the overwhelming demand for performance improvements has led to the introduction of the industry’s first enterprise-class Flash drives in EMC Symmetrix DMX-4 disk arrays. Companies no longer have to purchase large numbers of the fastest Fibre Channel disk drives and only utilize a small portion of their capacity to satisfy the IOPS performance requirements of workloads with very demanding I/O patterns (for example, traditional Online Transaction Processing database systems).

Relational databases are often at the core of business applications and increasing their performance, while keeping storage power consumption and footprint to a minimum, reduces TCO and helps in dealing with data centers constraints. The deployment of Tier 0 EFDs together with slower tiers of storage devices such as traditional, spinning Fibre Channel and SATA drives enables customers to structure the application data layout where each tier of storage meets the I/O demands of the application data it hosts.

## ***Database workloads best suited for enterprise Flash drives***

It is important to understand that any I/O request from the host is serviced by the Symmetrix DMX from its global cache. Under normal circumstances, a write request is always written to cache and incurs no delay due to physical disk access (internal memory transfers will be required to complete the request from cache, but are measured in micro-seconds). The fully protected write operation will typically be destaged to the physical drive at a later time and may benefit from other optimization mechanisms such as write coalescing. In the case of a read request, if the requested data is in the global cache either because of recent read or write of the same data, or due to sequential prefetch, the request is immediately serviced without physical disk I/O. A read serviced from cache without causing disk access is called a read hit. If the requested data is not in the global cache, the Symmetrix DMX must retrieve it from disk; this is referred to

---

as a read miss. A read miss will incur increased I/O response time since the data must be retrieved from the physical drives.

As workloads with high Symmetrix DMX cache read-hit rates are already serviced at memory access speed, deploying them on enterprise Flash drive technology may not show a significant benefit. Workloads that will benefit most from Flash drives can be described at a high level as those with the following:

- Low Symmetrix DMX cache read-hit rates
- Random I/O patterns
- Small I/O requests of up to 16 KB
- Requiring high transaction throughput

Database and application managers can easily identify mission-critical applications, which if made much faster, would directly affect an increase in business revenue and productivity. In a similar way the storage managers can point to these same applications, since for performance planning reasons they use large number of short-stroked drives. When such applications are identified, EFDs can provide a number of very important benefits.

Lesser numbers of EFDs can replace many short-stroked drives due to their ability to provide very high transaction rates (IOPS), while utilizing their entire storage space. This reduction in the required number of physical disks can increase power savings by not having to keep many spinning disks active. This reduction in the number of physical drives also leads to a reduction in cooling and floor space required in the data center as well as the associated weight of the additional traditional disk drives.

### ***Microsoft SQL Server log files***

Microsoft SQL Server transaction log file activity is mostly sequential write, thus the primary benefit of high performance operations of EFD will not be immediately available for logs placed on EFD. This effect results from Symmetrix DMX cache being used to receive all write operations and providing acknowledgement to the host for write operation completion. However, workloads with very high transaction rates may result in high, sustained write to the transaction log volume. Such sustained write workloads will necessitate back-end (physical storage) write operations as cache allocations may ultimately become limited. Being able to support very fast write operations to the actual storage device may be a consideration for placing SQL Server transaction logs on EFD storage. In addition, having transaction log files placed on these devices will help support other I/O operations against the transaction logs (i.e., backup/restore and transactional replication).

### ***Microsoft SQL Server tempdb files***

Tempdb is a database residing in every instance of Microsoft SQL Server, serving as a global resource for temporary objects and various operations performed by the engine. In some scenarios workloads may benefit by placing tempdb files on EFDs; however, this will be dependent on the workload as well as storage configuration.

Given the behavior of the cache resources on the DMX array more cache hits are likely since the data written to tempdb is generally read back within a short time period. Validation to determine the practical benefit for a particular workload would be necessary before choosing to deploy tempdb on EFD. For the workload used in this testing, tempdb activity was minimal.

### ***Host I/O operations versus physical I/O operations***

Advanced storage arrays implement various parity protection mechanisms (RAID levels) to ensure protection against physical drive failures. Depending on the type of protection implemented, additional disk I/O operations may be required to service the RAID level selected. In general, any performance penalty will occur for write operations, as parity information needs to be recalculated when data is updated.

The result of parity calculations within the array is not directly visible to the host generating the write operation, although it may be necessary to understand and quantify the amount of additional activity when planning for I/O capacity.

---

$$PhysicalIO = ((Total\_IO * read\_IO\%) - read\_Hit) + ((Total\_IO * write\_IO\%) * RaidFactor)$$

Where the components are defined as:

- *PhysicalIO* is the actual back-end I/O against the physical spindles
- *Total\_IO* is the host generated I/O workload
- *read\_IO%* is the percentage of the *Total\_IO* that is a read workload
- *Read\_Hit* is the amount of read workload serviced from the array cache
- *write\_IO%* is the percentage of the *Total\_IO* that is a write workload
- *RaidFactor* is the additional I/O activity required for write operations to cater for parity calculations (2 for RAID 1, 4 for RAID 5, 6 for RAID 6)

Using the provided calculation, it is possible to calculate the required physical I/O requirements for a given, known workload. Or to calculate the I/O capability required for an expected workload. In the following example, a host workload of 11,700 IOPS is used, with a read workload of 80% and a write workload of 20%. It is assumed that there is no read hit benefit in this calculation, and thus represents the worst-case calculation. The planned protection scheme is RAID 5.

$$PhysicalIO = ((11,700 * 80\%) - 0) + ((11,700 * 20\%) * 4) = 18,720$$

The resulting RAID adjusted workload, which must be serviced by the disk media, is 18,720 IOPS, compared to 11,700 IOPS as anticipated from the host. Thus an incremental 7,020 disk operations are required to cater for RAID overhead based on the write activity. For the same workload, based on a RAID 1 configuration, the total Physical I/O would be 14,040 IOPS, or 2,340 additional IOPS for RAID 1 parity calculations.

It is important to understand and cater for the additional workload when designing physical disk layouts within storage arrays. If not appropriately catering for the RAID factor, it is possible to underconfigure the required physical spindles required for a given host workload. This will result in greater latency numbers for the given workload and thus limit scalability.

---

## SQL Server 2008 OLTP databases and Flash drives

SQL Server Online Transaction Processing (OLTP) environments demonstrate the type of I/O workload and profile that, in a very active environment, can significantly benefit from EFDs. In an effort to simulate the style of workload, an industry standard OLTP workload was used against a configuration built on EMC Symmetrix DMX-4 EFD devices. This same workload was then deployed on traditional spinning HDDs with the number of drives sized to attain the same high level of I/O workload at reasonable latencies.

To form the basis of the comparison, a single set of four EFDs of 146 GB capacity was configured in a RAID 5 (3+1) configuration within a DMX-4 2500 array. RAID 5 is recommended for EFD deployments as this provides the parity protection against any single drive failure, with the best cost of ownership design. In this configuration, the parity overhead is 25% of the configured drives, as compared to a 100% overhead for RAID 1 configurations.

As discussed in the “Host I/O operations versus physical I/O operations” section on page 9, the parity configuration can generate additional I/O requirements at the physical drive level, depending on the RAID selection. In the configured environment, all write operations will require additional back-end disk operations to service the parity recalculations required by the RAID 5 protection scheme. In a configuration where writes are generally small random operations, as is the case in the OLTP workload, each write operation for a RAID 5 device will result in four back-end disk operations.

---

EMC Engenuity microcode includes enhanced functionality to optimize large sequential write streams on RAID 5 configurations, resulting in mitigation of the additional I/O overhead. In random write workloads, RAID 5 optimized writes are usually not possible.

---

### ***Test configuration***

For the purposes of characterizing the baseline performance of the tested storage array configurations, a single testbed was utilized to drive the workload. The following sections detail the server and storage platform designs used for all testing conducted.

#### **Server configuration**

- The Dell R900 server had the following:
  - 16 x 2.4 GHz processors
  - 64 GB memory
  - 4 x 4 Gb Emulex LightPulse A8003A
- The operating system was Windows Server 2008 Enterprise Edition (x64).
- The Microsoft SQL Server 2008 (x64) edition used was Enterprise Edition.

The Emulex HBAs were directly connected to Fibre Channel controllers within the DMX-4 array in a Fibre Channel Arbitrated Loop (FC-AL) configuration. No SAN switches were used in the tested configuration.

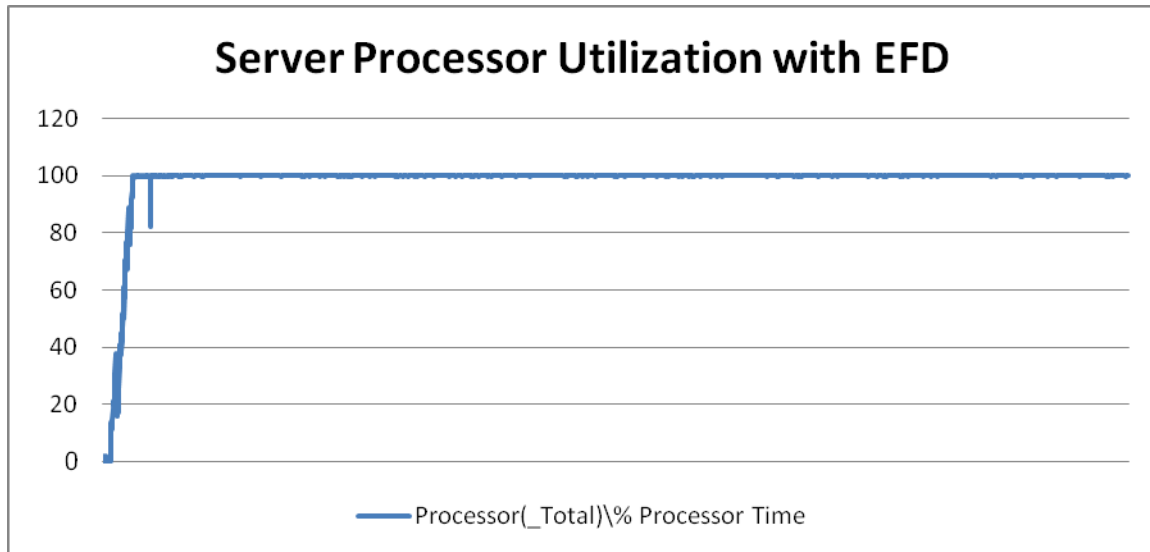
#### **Enterprise Flash drive configuration**

Initially, an EFD configuration comprised of four 146 GB physical drives was used to service the OLTP workload. The four EFDs were configured to form a single RAID 5 (3+1) set. From this storage, a single Logical Unit (LUN) was presented to the host to contain all the SQL Server database files for both the test database and tempdb database. An additional LUN was constructed from the EFD storage, and presented to the host to contain the SQL Server transaction log.

The EFD storage provided by the single RAID 5 set was implemented as 13 individual hypervolumes of 30 GB size. The first 11 hypervolumes were then concatenated to present a single 330 GB metavolume for the data files. This volume was then partitioned to create a single NTFS volume, and is referenced as drive E: in subsequent sections. The remaining two hypervolumes were also concatenated to present a single 60 GB metavolume for the transaction log. This volume was then partitioned to create a single NTFS volume, and is referenced as drive L: in subsequent sections. The SQL Server database was comprised of six physical data files located on drive E: and one transaction log file located on drive L:. The tempdb database files

were also placed on the E: drive. It should be noted that for this particular workload tempdb is not heavily utilized though some SQL Server workloads may realize substantial benefit by placing tempdb database files on EFDs.

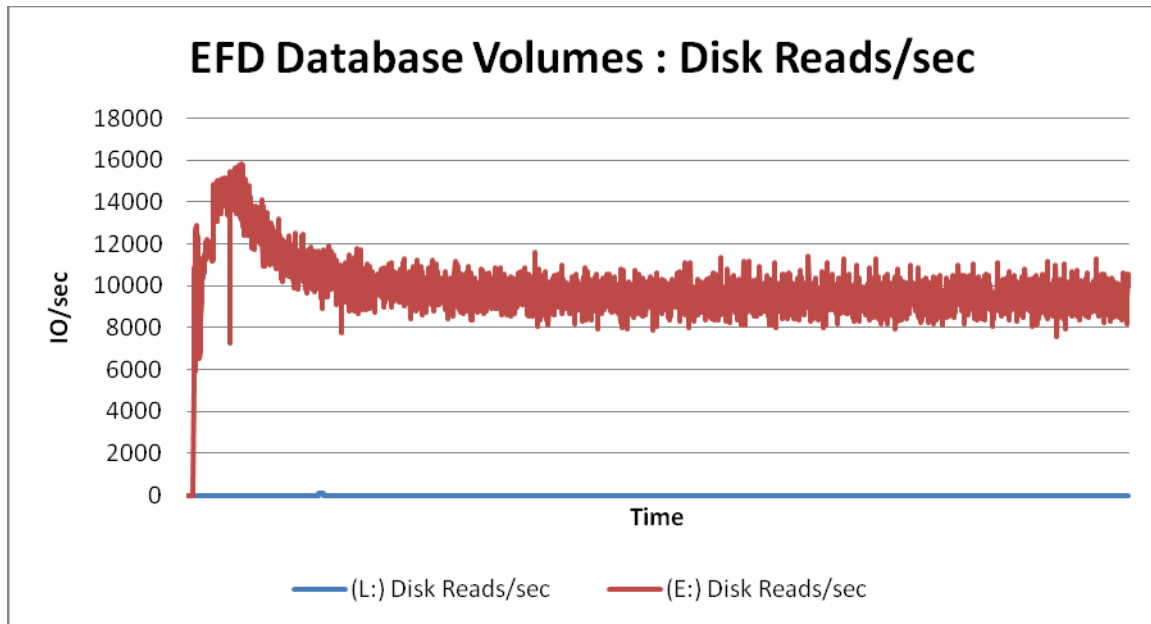
To create a baseline workload for the EFD configuration, the workload was increased during consecutive test runs, with a plan to determine when the EFDs were exhibiting increased latency. However, it became obvious that the limiting factor was CPU utilization of the server configuration, as shown in Figure 5.



**Figure 5. CPU utilization of the server with EFD**

As it was not possible to increase the number of CPUs beyond 16, the workload generated at the maximum rate detailed in Figure 5 was used for the EFD baseline. At this level, the read workloads can be seen in Figure 6. All read and write workloads were reported from Windows Performance Monitor (perfmon) captures of the Logical Drive counters.

The read workload consisted primarily of 8 KB read sizes. At initialization of the workload, the SQL Server buffer pool was empty, resulting in a significantly greater number of read requests. The initial burst of workload can be seen as well as the normalization of the workload as a steady state was achieved.

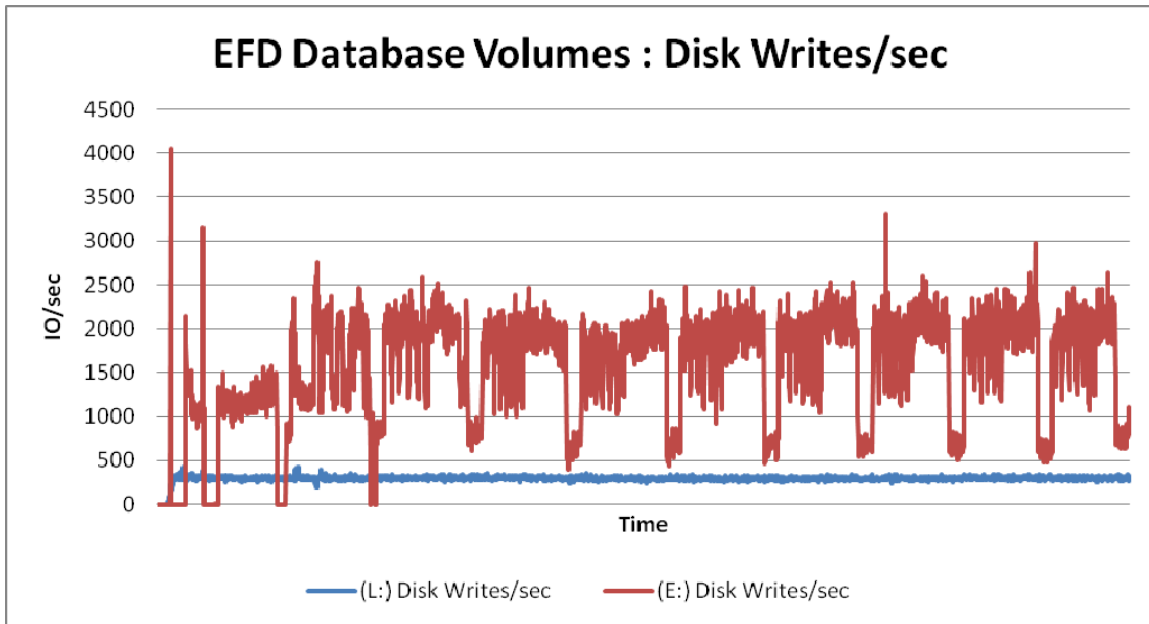


**Figure 6. Host Read operations/sec for data and log volumes with EFD**

As the OLTP workload also resulted in data changes, write workload was created. Microsoft SQL Server writes dirty pages to disk via a process referred to as checkpoint. The frequency of this operation is most commonly determined by the amount of transaction log generated. During a checkpoint process SQL Server scans the buffer pool for dirty pages and flushes these to disk. Checkpoint operations are generally random in nature; however, a checkpoint will attempt to find and flush adjacent pages in a single I/O request.

In addition to checkpoint operations SQL Server also has a lazy writer background process that flushes dirty pages to disk in an attempt to keep an adequate amount of pages on the free list. Whether or not the lazy write is active in this process depends on the need to replenish free pages. In the test case there was little to no lazy writer activity but frequent checkpoint operations due to the high volume of transactions in the workload.

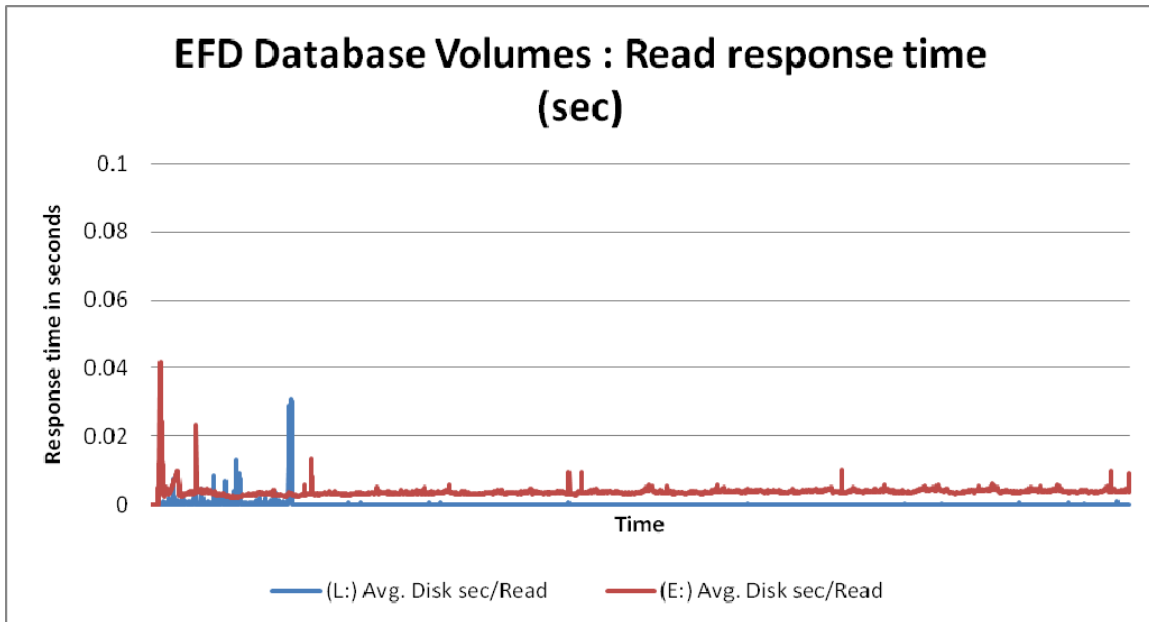
Writes to the SQL Server transaction log file are sequential in nature and done in a synchronous fashion. The log manager populates log buffers and flushes these to disk when a transaction commit is issued. Figure 7 details the overall write workload from the SQL Server environment for both data files (drive E:) and transaction log (drive L:). It should be noted that write workloads to Symmetrix DMX volumes will always be satisfied by system cache initially, irrespective of the nature of the physical drives (EFD or HDD). Constant write workloads can benefit from EFD configurations as such drives will be able to destage data faster than rotating media.



**Figure 7. Host write operations for data and log volumes with EFD**

Beyond the total read and write workloads themselves, the latency at the given workload is equally important. While maintaining the high levels of read and write workload, the EFD configuration continued to maintain extremely low latency. Figure 8 shows read response times for both the data files (drive E:) and the transaction log (drive L:). At the given workload, the average read response time for the data reads was around 3 ms.

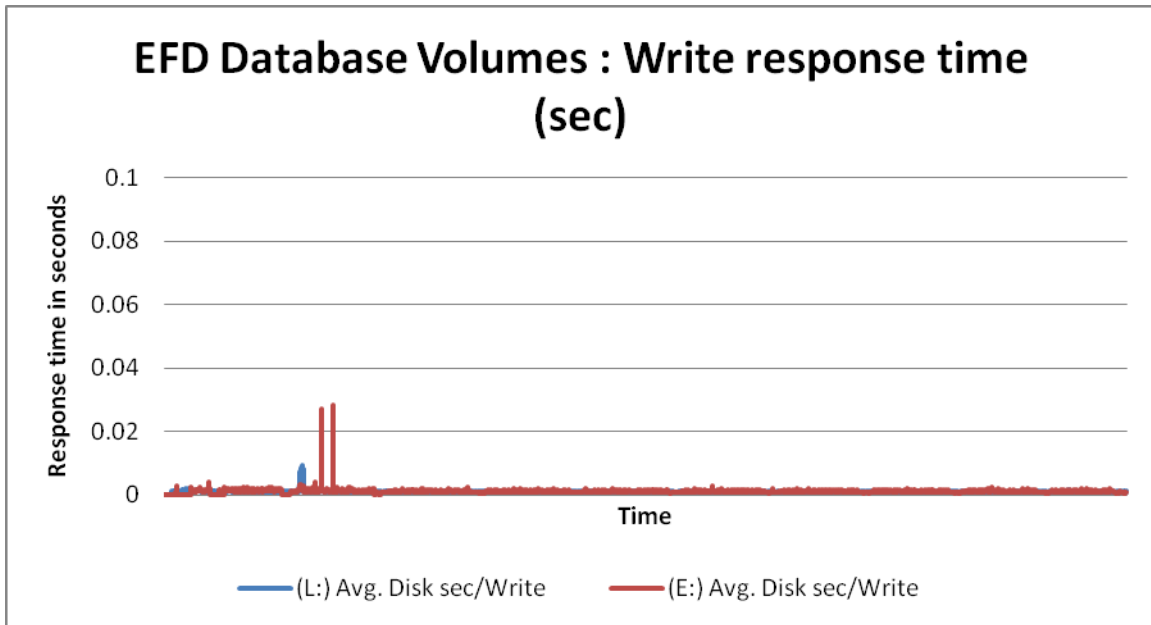
While in general, the transaction log typically has a sequential write stream during normal operations, SQL Server does generate read requests against the transaction log for operations such as transaction log backups that take place during this workload.



**Figure 8. Response time for read operations with EFD**

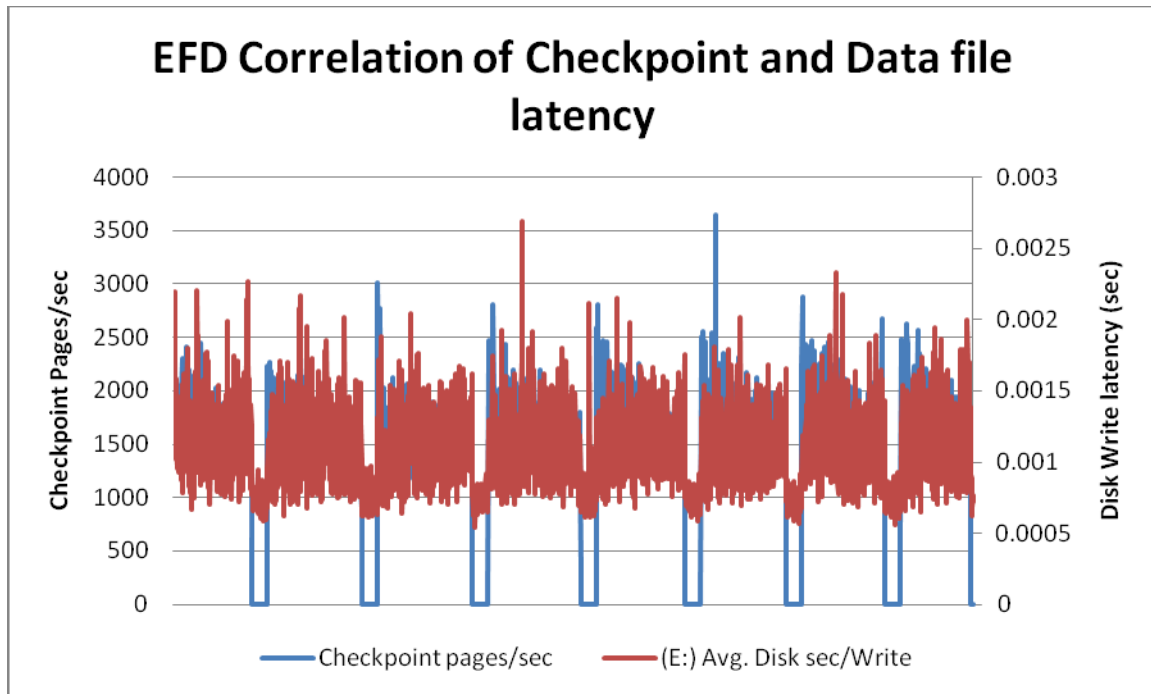
Write response times for the data files and transaction log are detailed in Figure 9. The counters for read and write response times from the host include all latencies along the I/O channel from SQL Server through

the operating system to the driver and to the storage array. In the case of write activity, bursts of write workloads can generate queues at all levels, and this can adversely affect response times.



**Figure 9. Response time for write operations with EFD**

When considering write operations occurring against database data files, it is important to note the nature of these write operations. SQL Server will execute periodic write operations to the data files through checkpoint operations. These checkpoints occur as needed and represent a coordinated point of consistency between the transaction log and the data files. The checkpoint write operations occur on a cyclical basis to ensure that SQL Server is able to meet the Recovery Time Interval setting for the database. If this is not explicitly set it will be determined automatically by SQL Server based on transaction log activity. Figure 10 shows a correlation between checkpoint page accumulations, and related write operations caused by checkpoint flushes and write latency for a subset of the entire workload run shown in earlier figures.



**Figure 10. Correlation of checkpoint operations and disk write latency with EFD**

It is important to understand the characteristics of the write activity when a checkpoint occurs. Within an extremely short duration, thousands of write operations that are generally random and small block (8 KB) in nature, constituting in excess of 40 MB of data, are flushed to the data file volumes. By design, checkpoint operations issue a large amount of I/O operations in order to flush dirty buffers in as short a period of time as possible. Although the checkpoint mechanism has logic that throttles I/O volumes based on real-time disk response times it is not uncommon to see some increase in response time during these operations.

It should also be noted that the performance of read operations is shielded from the write workload, as can be seen by the very low read latencies shown in Figure 8 on page 14. However closer examination does show that read requests have higher latencies when the burst of write operations occur. This is as a result of queuing in the I/O path as both read and write requests share the same I/O paths.

### Traditional hard disk drive configuration

To contrast the EFD configuration, a configuration based on traditional spinning HDDs was constructed for use by the same physical server. Given the nature of the workload was write intensive, and based on expectations of customer production applications, a configuration based on a RAID 1 protection scheme was defined. In an attempt to determine an adequate number of HDDs required, a calculation based on that defined by the “Host I/O operations versus physical I/O operations” section on page 9 was made.

From the EFD configuration testing, it was seen that the total I/O workload as seen by the host was around 11,700 IOPS with an 80% read component, and therefore a 20% write component.

To determine the number of physical spindles required for the HDD test, it is necessary to first determine the I/O capacity of the physical spindles. In general, for 300 GB 15,000 rpm drives whose capacities are fully utilized (i.e., fully stroked) a value of 180 IOPS is generally recommended as an appropriate I/O rate at an 8 KB I/O size with acceptable latency. Using this 180 IOPS value to determine the required number of spindles would result in 78 physical spindles required to service the workload. Conversely, the storage allocation available for this number of spindles would result in around 11.7 TB of storage allocation when using RAID 1 (around 23.4 TB raw).

Since the capacity requirements for the database files was minimal (300 GB) sizing based on the disk capacity being fully utilized is not necessary as the data will utilize only a small portion of each physical

spindle. In this case the disks will benefit from lower seek time (short-stroked) and can be sized with a high IOPS per physical disk. As a result, an estimation of around twice the expected I/O per drive was anticipated (360 IOPS) as a result of the short-stroking of the HDDs.

To calculate the actual physical workload, the RAID adjusted calculation is required. As customers typically run production SQL Server workloads on RAID 1 configurations, this was the selected configuration for the HDD tests. In actuality, the configuration used for the HDD testing was based on striped metavolumes. Striped metavolumes are created by the Symmetrix array striping across a set of mirrored devices, which may be referred to as a RAID 1/0 configuration.

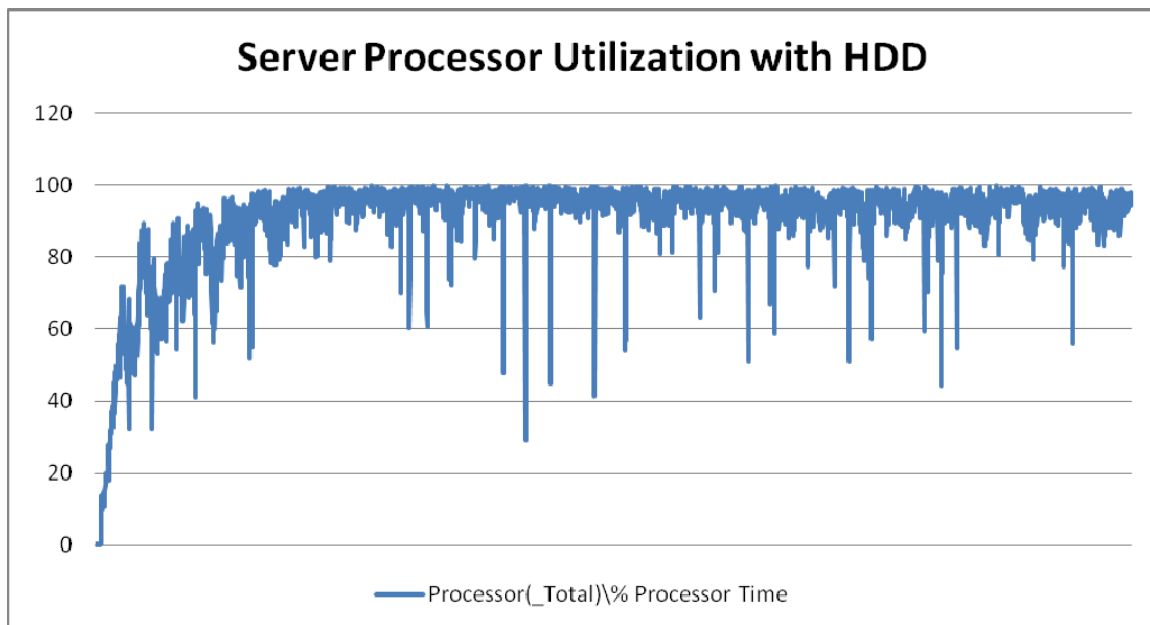
Testing in the EFD configuration revealed that the total I/O workload for the database files, as seen by the host, was an average of 11,700 IOPS with an 80% read and 20% write component at an 8 KB average I/O size. Read cache hit ratios in the vicinity of 25% were seen, and this value is used to adjust the physical read workload. Additionally, the transaction log file was to be placed on dedicated spindles, and was removed from the data file workload. The transaction log write activity was an average of 310 IOPS, and would adequately be satisfied by a two-member striped metavolume using RAID 1.

Using the data provided by the EFD testing, and the modifications indicated, the total IOPS using the described calculation was

$$PhysicalIO = ((11,390 * 80\%) - 2278) + ((11,390 * 20\%) * 2) = 11,390$$

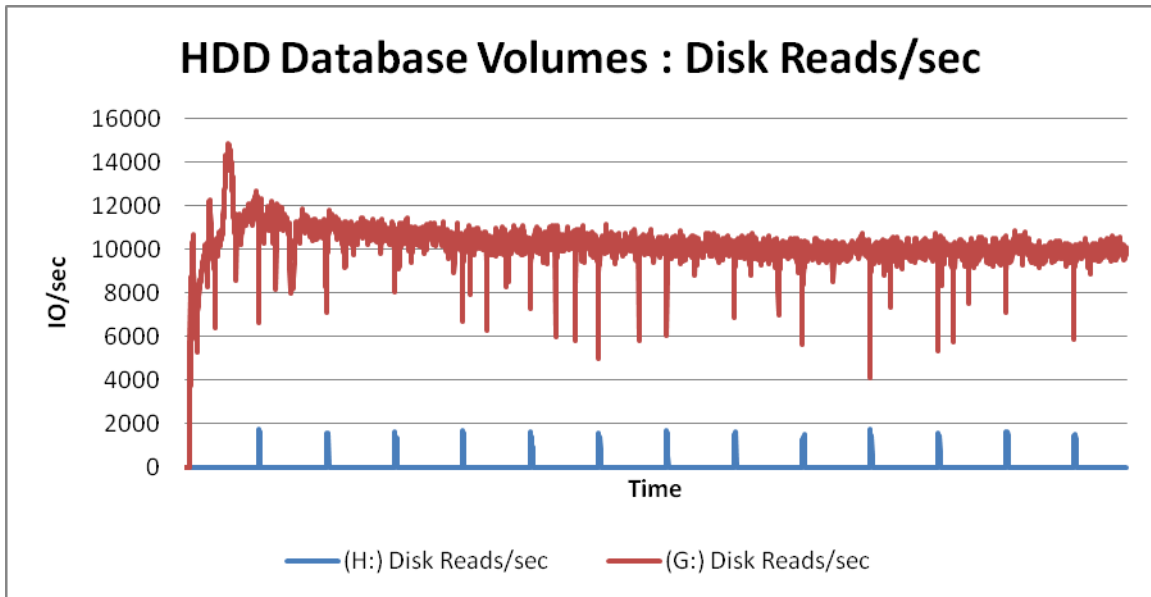
The total RAID adjusted I/O required was therefore 11,390. Based on the I/O per drive in a short-stoked configuration, this would require 34 physical spindles for the data file I/O requirements. In addition the transaction logs would be serviced by utilizing a striped two-member metavolume, resulting in a total HDD configuration of 38 physical HDD spindles.

The tested configuration approached similar utilization rates as seen through CPU utilization in Figure 11. Although it is notable that the workload did not cause CPU utilization to be completely consumed as was seen in the EFD testing.



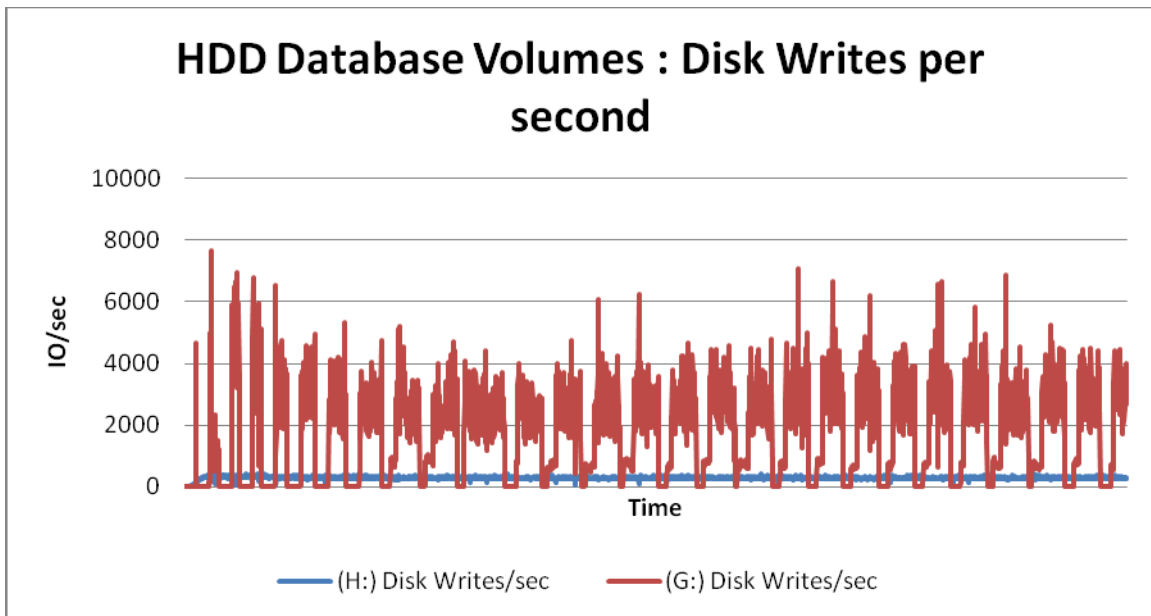
**Figure 11. CPU utilization of the server with HDD**

Read workloads were also seen to have similar rates for both the database data and transaction log file locations as shown in Figure 12.



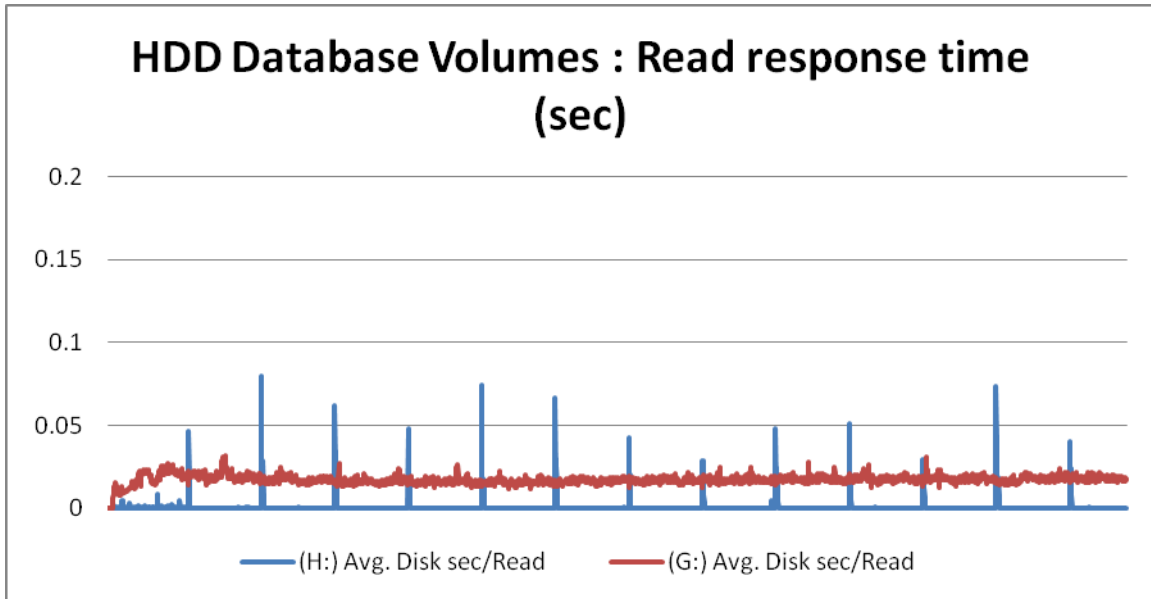
**Figure 12. Host read operations/s for data and log volumes with HDD**

For the HDD configuration, the write workload for the data volume varied markedly from the EFD testing as displayed in Figure 13. Write activity was more sporadic with greater variance between maximum and minimum values for the data file volume. Transaction log write activity was constant, although at a lower level than seen with the EFD configuration.



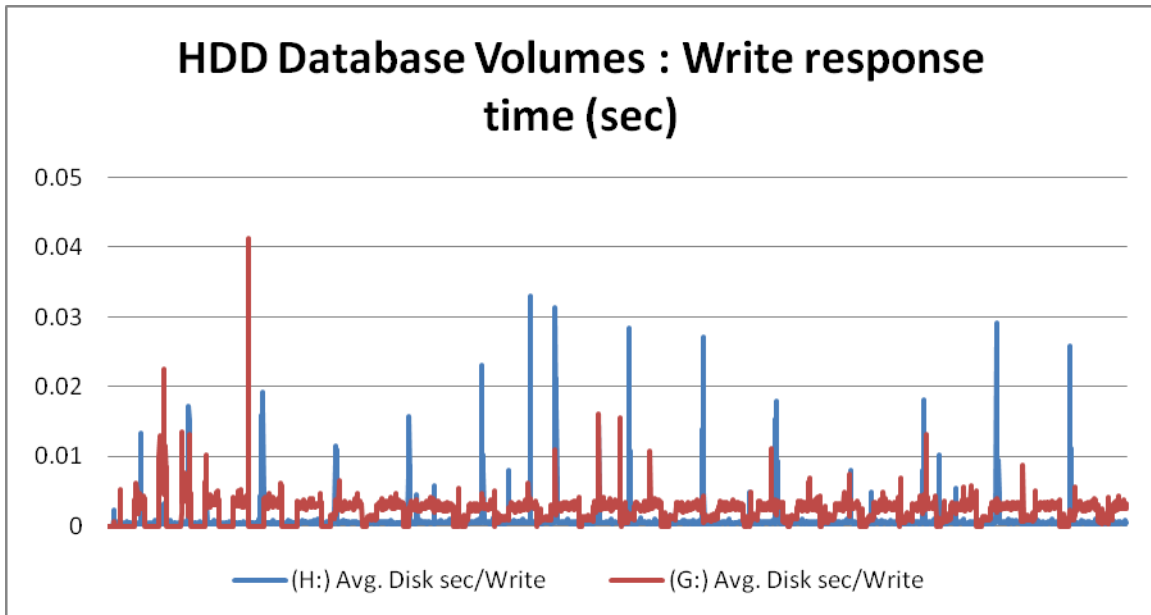
**Figure 13. Host write operations/s for data and log volumes with HDD**

Along with the total number of I/O operations processed by the configuration, the latency exhibited by the storage configuration is directly related to the overall performance of the environment. For database applications, elongated read response times can adversely affect throughput for the application. Figure 14 shows the overall response time for read operations executed against the data file volume (drive G:) and the transaction log (drive H:). Average read latency for the data file volume was 17 ms.



**Figure 14. Response time for read operations with HDD**

As discussed in the EFD section, write operations against the data files are generally of an asynchronous nature, and may not directly affect overall performance of the database environment. Transaction log writes operations conversely have a direct relationship to overall database performance. It can be seen in Figure 13 that write operations to the data file volume (drive G:) were very low, with an average write response time of 2 ms. Transaction log write operations (drive H:) were processed with an average write response time of 7 ms. The peaks in the transaction log write activity were correlated to large write operations to the disk subsystem during checkpoint operations.



**Figure 15. Response time for write operations with HDD**

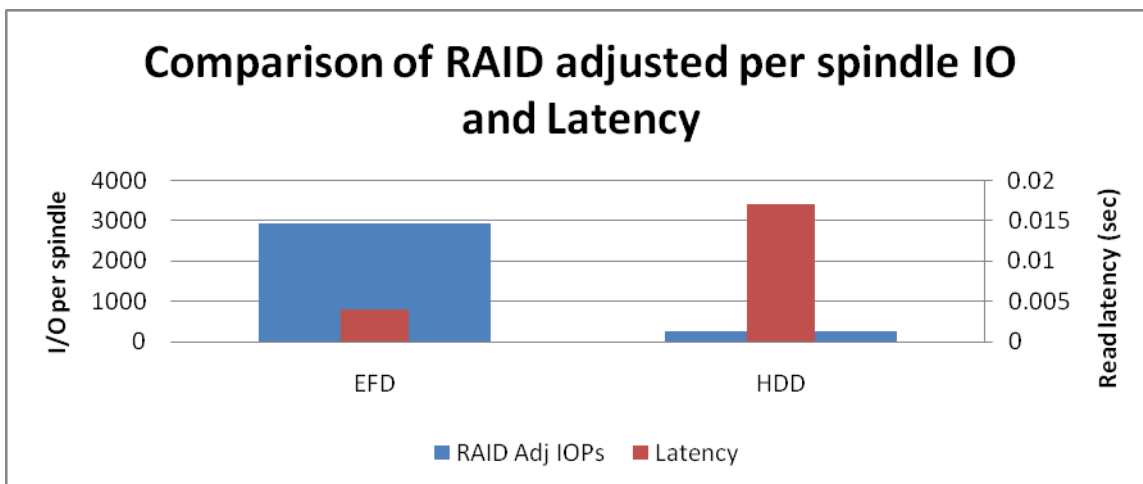
## Conclusion

Enterprise Flash drives provide a significant performance advantage over traditional spinning hard disk drive technologies for OLTP environments utilizing Microsoft SQL Server. With the ever increasing need to provide exceptional performance for business critical environments, EFDs provide a means to extend performance beyond the traditionally defined storage tiers.

Incorporation of Flash drives into Symmetrix DMX-4 with Engenuity 5773 provides a new Tier 0 storage layer that is capable of delivering very high I/O performance at a very low latency, which can dramatically improve OLTP throughput and maintain very low response times. With comprehensive qualification and testing to ensure reliability and seamless interoperability, Tier 0 is supported by key Symmetrix software applications that enable advanced management tools.

Traditional disk drive technology no longer defines the performance boundaries for mission-critical storage environments. The costly approach of spreading workloads over dozens or hundreds of underutilized disk drives is no longer necessary. Symmetrix now combines the performance and power efficiency of Flash drive technology with traditional disk drive technology in a single array managed with a single set of software tools, to deliver advanced functionality, ultra-performance, and expanded storage tiering options.

Combined with Microsoft SQL Server environments, enterprise Flash drives allow customers to consolidate database environments down to a significantly smaller number of enterprise Flash drives which deliver the required I/O rates and extremely low service time latencies as detailed in Figure 16. With the ability to reduce the total number of physical drives within a storage array, EMC Symmetrix DMX with EFD provides improved TCO attributes.



**Figure 16. Comparison of RAID adjusted, per spindle IOPS and latency**

The overall I/O per spindle improvement from the EFD represents in excess of an 1,140% increase over traditional HDDs, as shown in Figure 16. It is also important to note that while maintaining such high levels of I/O rate per spindle, this was done at significantly lower latencies. For the same workload, the fully allocated EFDs provided in excess of 2,900 IOPS, a response time of around 4 ms. Conversely, each short-stroked HDD was able to sustain around 250 IOPS at a latency of 17 ms.

Storage administrators may effectively design storage layouts using traditional disk drive technology, however, when confronted with the need to optimize for performance, they are faced with the limitations of traditional drive technology. To compensate for physical drive characteristics, such as drive head seek, layouts may need to be configured to reduce this limitation by such strategies as disk drive short-stroking. While effective in improving the performance characteristics, such strategies provide poor utilization rates of storage allocations and deliver low TCO.

EFD implementations utilized with Microsoft SQL Server can deliver the highest levels of I/O performance, with significantly reduced latencies, and reduced power consumption.

---

## Appendix: Flash drives and Information Lifecycle Management strategy

It is common that the most recent data has the most demanding latency requirements over older data. This type of data classification is the beginning of an Information Lifecycle Management strategy, or ILM. Data classification is important in order to provide applications with the most cost-effective storage tier to support their workload needs. It can be done by placing each application on the storage tier that fits it best, but it can also be done by using multiple storage tiers within the same application.

A common way for deploying a single database over multiple tiers is by file type. For example aged partitions of a user database can use SATA drives while transaction logs and data files can use Fibre Channel HDD. It is now possible to add a new storage tier using Flash drive technology and place latency-critical data files, indices, or temp files on them as discussed earlier.

However, when the database is large, in order to make best utilization of drive resources, it may be better to place on Flash drives only the data that is accessed most frequently and/or has the most demanding latency requirements. Many databases can achieve this by using table partitioning.

Using further application analysis techniques customers may be able to determine which filegroups and thus, which files, are the most active with workloads that EFDs are uniquely able to help. Placing the LUNs for these filegroups on EFDs will provide significant benefit while not requiring the entire database to be on enterprise Flash storage.

Table partitioning also creates subsets of the table, usually by date range, which can be placed in different data files, each belonging to a specific storage tier. Table partitioning is commonly used in data warehouses where it enhances index and data scans. However, with an ILM strategy in mind, customers should consider the advantage of using table partitioning for OLTP applications. While partitioning allows the distribution of the data over multiple storage tiers, including Flash drives, it does not address the data movement between tiers. Data migration between storage tiers is out of the scope of this paper. Solutions in this space are available by using Symmetrix Virtual LUN technology, or by using host volume management features. An example of using table partitioning is shown in Figure 17.

**CUSTOMER\_ORDER TABLE**

Partition 1	Partition 2	Partition 3	Partition 4
Current data demanding the highest performance and availability	Less current data demanding high performance and availability	Older data for fulfillments and batch processing. Less critical for business.	Oldest data marked for archiving And compliance. Not critical for running business.
Flash	FC-15K RPM	FC-10K RPM	SATA

**Figure 17. A partitioned table using tiered storage levels**