Performance Value of Solid State Drives using IBM i

May 2009

By Mark Anderson, Robert Gagliardi, Henry May, Ginny McCright, Steve Tlusty, Wesley Varela

IBM Systems and Technology Group

Executive Overview

Solid State Drives (SSDs) offer a number of advantages over traditional hard disk drives (HDDs). With no seek time or rotational delays, SSDs can deliver substantially better I/O performance than HDDs. Capable of driving tens of thousands of I/O operations per second (IOPS), as opposed to hundreds for HDDs, SSDs break through performance bottlenecks of I/O-bound applications. Applications that require dozens and dozens of "extra" HDDs for performance can meet their I/O performance requirements with far fewer SSDs, resulting in energy, space, and cost savings.

Solid State Drive technology was introduced more than three decades ago. Until recently, however, the high cost-per-gigabyte and limited capacity of SSDs restricted deployment of these drives to niche markets or military applications. Recent advances in SSD technology and economies of scale have driven down the cost of SSDs, making them a viable storage option for many I/O intensive enterprise applications.

While the cost of SSDs is trending downward, the $/GB for SSDs is still substantially higher than that of HDDs. Thus, it is not yet cost-effective for most applications to replace all HDDs with SSDs. Fortunately, it is often not necessary to replace all HDDs with SSDs. For instance, infrequently accessed (cold) data can reside on lower cost HDDs while frequently accessed (hot) data can be moved to SSDs for maximum performance. Many applications have a high percentage of cold data compared to hot data allowing SSD usage to be leveraged very effectively. The appropriate mix of SSDs and HDDs is then used to strike a proper balance between performance and cost.

To demonstrate the benefits of SSDs, we ran experiments comparing SSDs with HDDs. The experiments showed a significant performance advantage with SSDs which resulted in a substantial reduction in the number of drives needed to meet the desired level of performance. Fewer drives translate into a smaller physical footprint, reduced energy consumption, and less hardware to maintain. The experiments also showed better application response times for SSDs, which leads to increased productivity and higher customer satisfaction.

This paper focuses on applications in an IBM i operating system environment. Other papers are available for other IBM environments. IBM i 5.4 and IBM i 6.1 offer some very powerful functions that can allow you to easily, productively and cost effectively implement SSD in your environment. The paper describes how to deploy SSDs in a tiered storage environment to allow you to leverage your existing storage with SSDs for maximum performance and minimum cost. This paper also discusses IBM i tools and services available to assist you in deploying and managing a storage solution with SSDs.

The integrated Storage Management capability of IBM i 5.4 and 6.1 has added new SSD function to the existing ASP Balancer allowing "hot" data to be easily identified and placed on SSDs. It also leverages the increased speed of SSDs by automatically placing specific system-critical, high-use objects on these faster drives. Additionally, a new "media preference" option has been created as part of an integrated solution with DB2 and Libraries.

Introduction

Some common terms that are used throughout the paper are defined here:

- HDD is short for Hard Disk Drive
- SSD is short for Solid State Drive
- IOPS is short for IOs Per Second
- DASD is short for Direct Access Storage Device
- ASP is short for Auxiliary Storage Pool

Comparing HDD and SSD High Level View

As is always the case with computers, programs and the data they access need to be loaded into memory. When these programs are not in use, they reside on HDDs today. The response time of your applications can be directly mapped to several factors, available CPU(s) time slices, size and availability of memory, size and location of cache and, of course, HDD speed. With each execution of an instruction that requires memory to be accessed, the resulting wait for the data would vary from a few nanoseconds to a few microseconds depending on where "*in memory*" the needed data is located. However, if the needed data is located on a physical disk(s) the latency increases dramatically.

In this simple example, let us assume that an application makes 1000 data accesses. 999 of those attempts were "*in memory*" resulting in an average access time of 500 nanoseconds. Even if 1 of the attempts has to retrieve the data from an HDD, we add roughly 10 millisecond latency. Therefore, 95% of the total access time was 1 access to an HDD, making overall access time seem very large.

As is apparent from the above example, application response time could be dominated by a few accesses to physical disks. Over the last several decades, software companies have spent a considerable amount of time, energy and money to develop predicative algorithms that would limit the high number of access to the HDDs by predicting and prefetching data that it believed the program might need.

HDDs are essentially spinning platters that consist of tracks and sectors of data that are read by a moving physical arm. When data is needed from an HDD, the arm must move to the correct track and the platter must rotate to the correct sector(s) for the data to be read. Even with 15K RPM spinning platters, physical access time to the HDD in the above example is a dominating 95% of the accumulated access time.

SSDs, on the other hand, use solid state persistent memory to store data. SSDs have no moving parts which result both in significantly lower access times (100 microsecond range), and reduced power consumption and noise. Since SSDs do not have to power spinning platters or moving arms the overall watts/hour needed to power an SDD is significantly less than that of a typical 3.5-inch HDD.

Identifying client applications that can benefit from SSDs

While a single SSD can perform as many operations with better response time than many HDDs, no device has infinite capacity. Therefore, planning is required to arrive at an optimal configuration. In general, workloads with a large number of random synchronous reads that are

not sourced from adapter cache will benefit the most from SSDs. IBM i has a variety of tools that facilitate collection of the data necessary to determine if SSDs will produce benefit and in designing an optimal configuration.

The first thing to look at, as with most performance questions, is Collection Services. Read and write rates, as well as overall service time, can be retrieved from the Collections Services database with simple queries or performance tools (57xx-PT1) reports. I/O Adapter (IOA) cache, device cache and seek statistics are also available for storage subsystems that allow collection of that data. Wait bucket data is available to determine the amount of time jobs are waiting on disk reads. Analysis of these values provide good insights on your I/O subsystem and can indicate if SSD usage would be a significant boost to I/O performance.

Analysis that is more detailed can be performed with Performance Explorer disk traces. This data can be used to identify hot data by disk extent or object. This allows the number of SSDs required to be more accurately predicted. Performance consultants familiar with the tool will generally be required to assist with this very detailed trace analysis

SSD Integration in IBM i

As IBM i has it's own storage manager and DB2 for i built in, the integration of SSDs has been a fairly simple task. The functions provided for management of SSDs and adjusting their impact on Applications and Database are very simple and easy to use.

There are three basic methodologies to place data on SSD.

- ASP Balancer – Enhanced for SSDs
- Library and SSD Integration
- DB2 and SSD Integration

Two of the methods listed above were used to move hot data to SSDs on the following scenarios. Here is a brief description about those methods while the full description and details about all three methodologies can be found towards the end of the document after we go through the performance results.

ASP Balancer – Enhanced for SSDs

This option balances the data by moving the most frequently accessed "hot" data to the faster SSD units, while moving the least frequently accessed "cold" data to the slower HDDs. This is accomplished by using two commands Trace ASP Balance (TRCASPBAL) and the ASP Balance (STRASPBAL) command, specifying TYPE(*HSM)[1]. Based on the usage statistics gathered with the trace command, the balance command moves the hot data to the SSD drives. At the same time, other balancer tasks move the cold data from the SSD drives to the slower HDD drives.

DB2 and SSD Integration

This option allows a user to specify what data should be allocated on SSD. DB2 has provided the capability to specify a "media preference" as an attribute of a database table, partition, or index. This option can be used on the CRTxx/CHGxx CL commands and/or the SQL statements used to create or alter tables and indexes.

[1] The type *HSM is for IBM i releases 5.4 and 6.1. This will change to *SSD in a future release

Quantifying the Benefits of SSDs using CPW-like OLTP Workload

On-line Transaction Processing (OLTP) applications are typically thought of as having a large number of users concurrently executing transactions to a database. The transactions vary by application, but usually each transaction produces a substantial amount of I/O to the disk subsystem. Such transactions expect that the response time is not only low, but also consistent throughout the business day.

Introducing SSDs into a production environment can offer many benefits including:

- An increase in I/O and data throughput
- A reduction in application and disk response time
- A reduction in energy and lab space
- A reduction in number of HDDs needed
- In some situations a reduction in purchase cost

As you will see below, numerous configurations were used to show the value of introducing SSDs into a production environment. Each scenario has a detailed description of the configuration, results and a summary section. The scenarios are intended to show the value of SSDs in different configurations as well as the benefit of using the IBM i ASP Balancer or the DB2 media preference flag to move hot data to SSDs.

The system configuration for the OLTP workload scenarios below were run on a POWER 570 Model 9117-MMA using IBM i v6.1 unless otherwise noted. RAID 5 protection is used on all the configurations in the following sections.

The CPW-like workload that is used to drive the storage I/O subsystem is not a direct map to CPW ratings POWER systems have. There were numerous changes to the workload to stress the disk subsystem so the SSDs could be portrayed in a valuable manner while keeping the OLTP characteristics reasonably intact. For all the scenarios listed below, the storage I/O subsystems that are compared are built to be their own ASP (Auxiliary Storage Pool) and contained the database the workload ran against. Each throughput point on the graphs below represent a fixed amount of simulated users per point, so the averages are calculated over the same user points for both the HDD and SSD configurations.

Scenario 1: 8 SAS SSDs vs. 36 SAS HDDs

For this scenario the database ASP contained 8 SAS SSDs within an 5886 EXP12S DASD drawer, connected using 1 x 5908[2] 1.5G SAS RAID controller compared to 36 SAS HDDs, see figure 1 below.

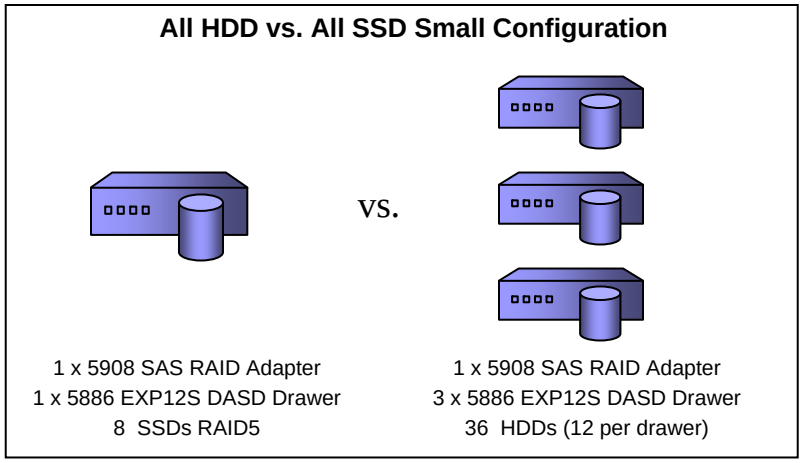

Figure 1: All HDD vs. All SSD Small Configuration

Results for Scenario 1

The purpose of this scenario is to show the difference between running 8 SAS SSDs on 1 5908 compared to 36 SAS HDDs on 1 5908 adapter. Figure 2 below shows the application response time curve as throughput is increased. The application started to show a much larger increase in response time earlier than the 8 SSDs showed, while the total throughput was equivalent up to the last throughput point. When the response time spiked higher on the HDD configuration, the SSD application response time continued to stay at a very reasonable rate.

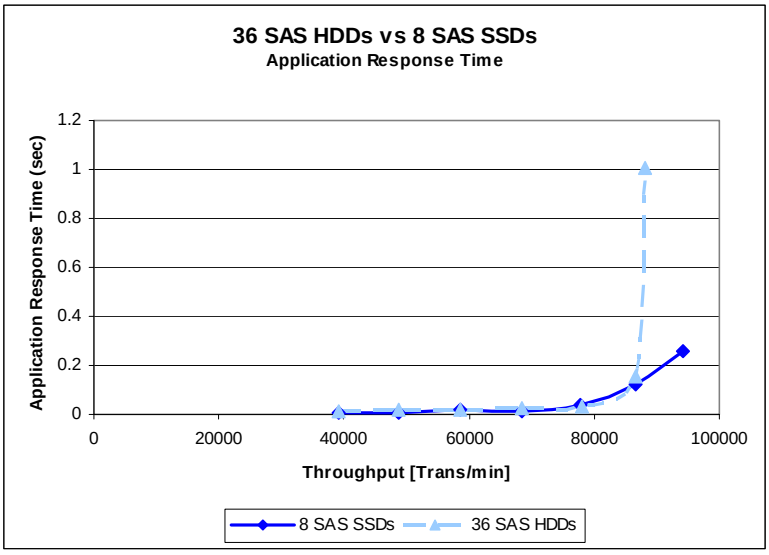

Figure 2: 36 SAS HDDs vs. 8 SAS SSDs small configuration

[2] Other feature codes like 5904 and 5906 are available. Please use the Infocenter for further information about feature codes and card placement
http://publib.boulder.ibm.com/infocenter/systems/scope/hw/index.jsp?topic=/iphcd/pcibyfeature.htm

Figure 3 below shows average transaction throughput per drive as well as average application response time. The efficiency of the drives can also be measured in terms of transaction throughput per drive. Because SSD response time is so much faster than HDD response time, many more HDDs are required to achieve similar throughput to what SSDs can sustain. The SSD ASP provided a 64% lower average application response time and a 4.5x improvement in average throughput per drive over the 36 HDD ASP.

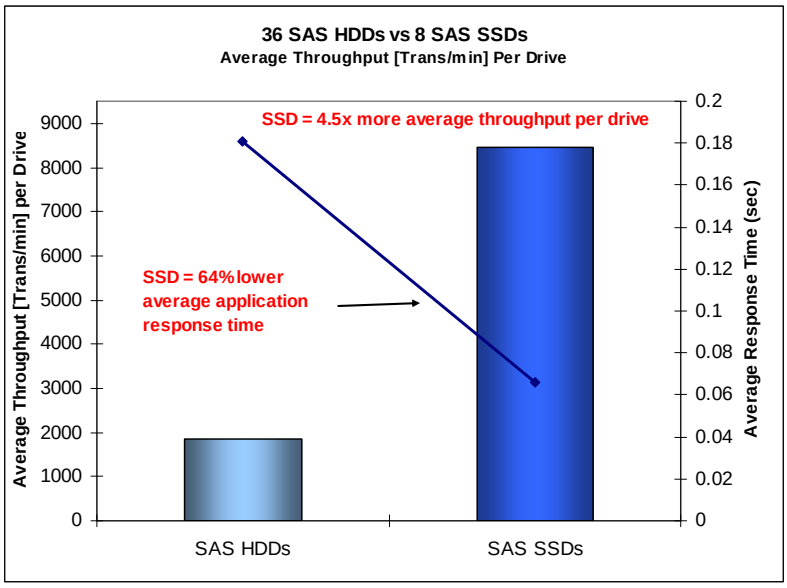

Figure 3: SAS HDDs vs. SAS SSDs. Average throughput [Trans/min] per Drive

Scenario 2: 12 SAS SSDs vs. 108 SCSI HDDs

For this scenario the database ASP contained 12 SAS SSDs (6 SSDs within each 5886 EXP12S DASD drawer), connected using 2 x 5908 1.5G SAS RAID controllers compared to 108 SCSI HDDs. See figure 4 below.

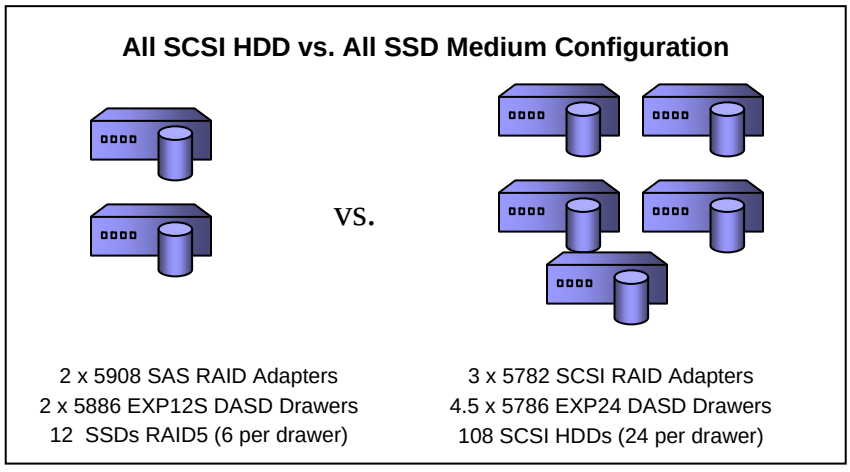

Figure 4: All SCSI HDD vs. All SSD Medium Configuration

Results for Scenario 2

The purpose of this scenario is to show the performance benefit of an all SAS SSDs ASP compared to an all SCSI HDD ASP in a medium size legacy HDD configuration. Figure 5 below shows how the knee of the response time curve for the 12 SSDs is moved out so the workload was able to achieve more throughput while keeping application response time below 1 second.

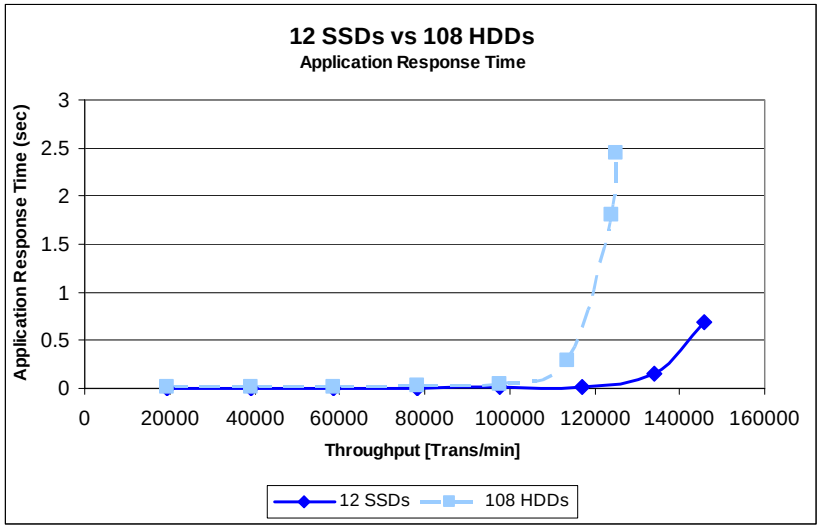

Figure 5: 12 SSDs vs. 108 SCSI HDDs Response time curve

Figure 5 above is showing how the throughput on the SSD configuration is higher as more simulated users are added (the points where the 2 curves start to diverge). Since each throughput point on the graph is a fixed set of simulated users, it is fair to focus on the lower throughput points to see the application response time improvement the SSD configuration achieves. Figure 6 below shows that the SSD configuration can lower application response time by up to 80+%.

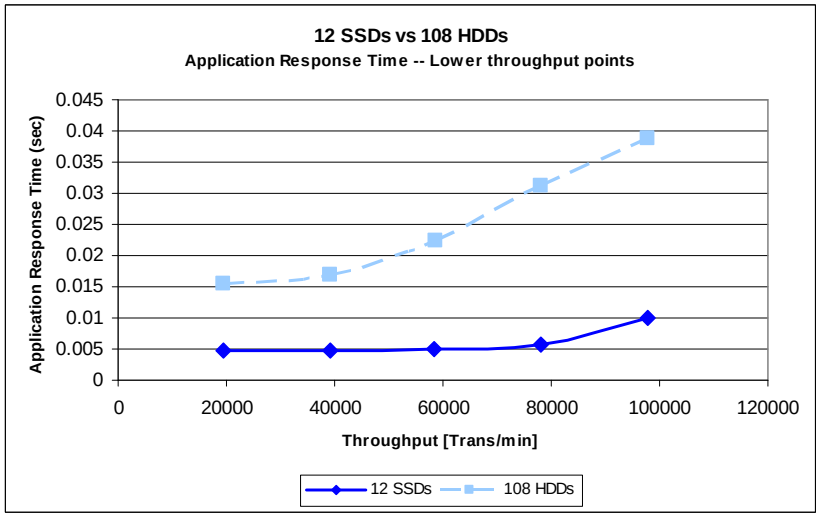

Figure 6: 12 SSDs vs. 108 SCSI HDDs Response time curve for lower Throughput [Trans/min] points

Again, if we take a closer look at the average throughput per drive across the response time curve from figure 5, the SSD configuration out performs the HDD configuration by 9x, see figure 7 below. The average application response time also decreased by 81% in this scenario, which is why the total throughput per drive was able to see the substantial improvement. At the time of the experiments, purchase price for the SSD configuration is also roughly 24% lower when compared to the SCSI HDD configuration.

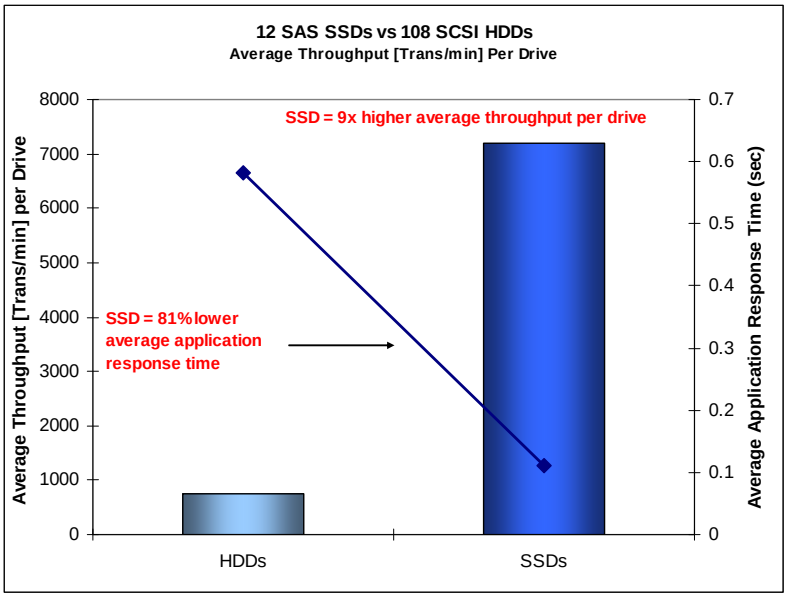

Figure 7: 12 SSDs vs. 108 HDDs. Average throughput [Trans/min] per drive

Scenario 3: Mixed ASP Configuration 36 SAS HDDs + 8 SAS SSDs vs. 108 SCSI HDDs

For this scenario the database ASP contained 8 SAS SSDs in an 5886 EXP12S DASD drawer plus 36 SAS HDDs (12 per drawer), connected using 2 x 5908 1.5G SAS RAID controllers compared to 108 SCSI HDDs, see figure 8 below.

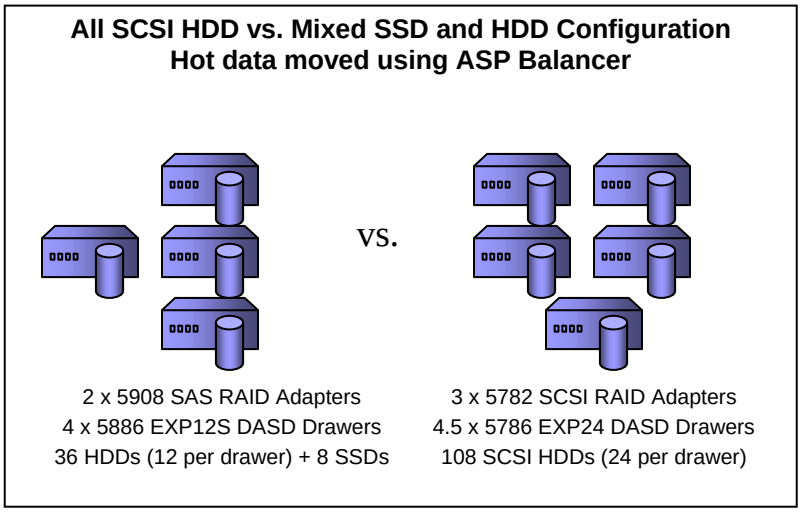

Figure 8: Mixed HDD & SSD Configuration vs. 108 SCSI HDDs

Results for Scenario 3

The purpose of this scenario is to show the benefit of using the IBM i ASP Balancer described in a previous section, to move the most frequently read data onto the SSDs while the cold data was kept on the HDDs. As figure 9 shows, the HDD config starts to hit the knee of the curve at a throughput of about 170,000 while the mixed config just slightly starts to rise. The mixed config is able to achieve a throughput increase of about 20% while the application response time was under 0.5 seconds while the equivalent throughput point on the HDD curve is showing a 4.5 second response time.

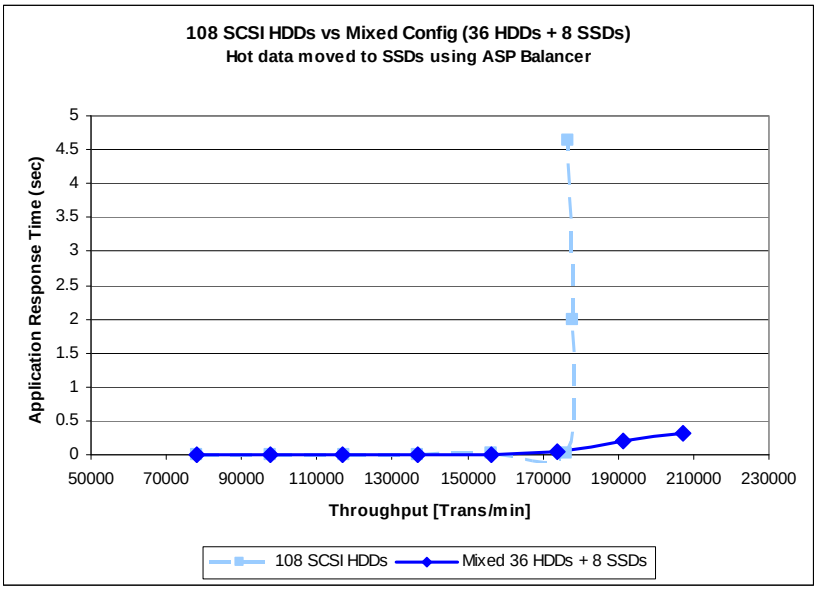

Figure 9: 108 HDD vs. Mixed config with hot data moved to SSDs using the ASP Balancer

Since the mixed configuration was able to keep a reasonable application response time well past the HDD configuration, figure 10 shows the value of being able to put the hot data onto SSDs while keeping the cold data on HDDs. The mixed configuration on average produced 2.5x higher throughput per drive while lowering the average application response time by 91%. By tracing the application over a certain time and balancing the hot data on SSDs and cold data on HDDs, more throughput can be achieved even when the total number of disks in the database ASP was decreased by roughly 60%. At the time of the experiments, purchase price for the mixed SSD/HDD configuration is also roughly 30% lower when compared to the SCSI HDD configuration.

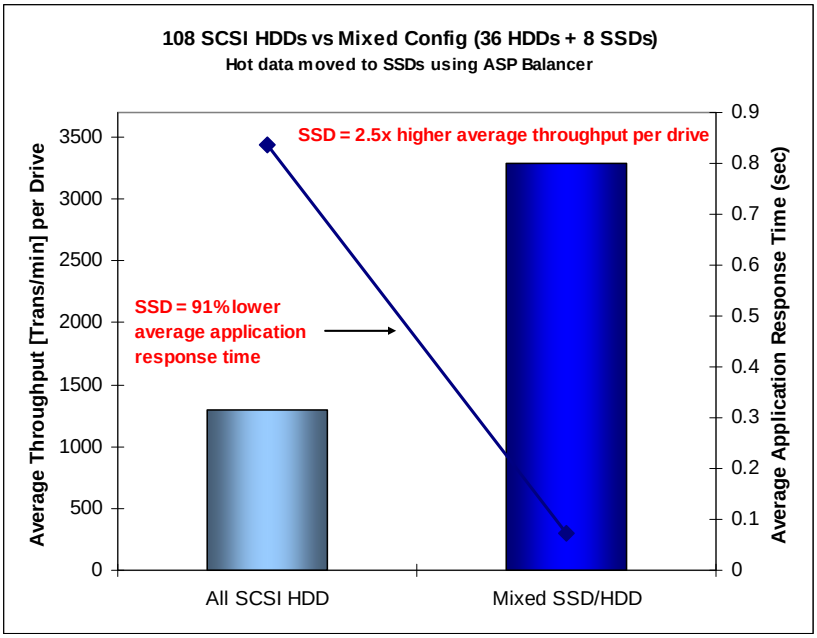

Figure 10: 108 HDD vs. Mixed ASP Average Throughput [Trans/min] per drive with hot data moved to SSDs using ASP Balancer

Another benefit of SSD configurations is the space and energy consumption savings. Figure 11 shows how the watts consumed per transaction/second of the Mixed SSD/HDD subsystem are 64% lower than the HDD subsystem. The system CEC and other ASP's were not part of the energy metrics. For energy estimates, please use the IBM Systems Energy Estimator available at this link: http://www.ibm.com/systems/support/tools/estimator/energy

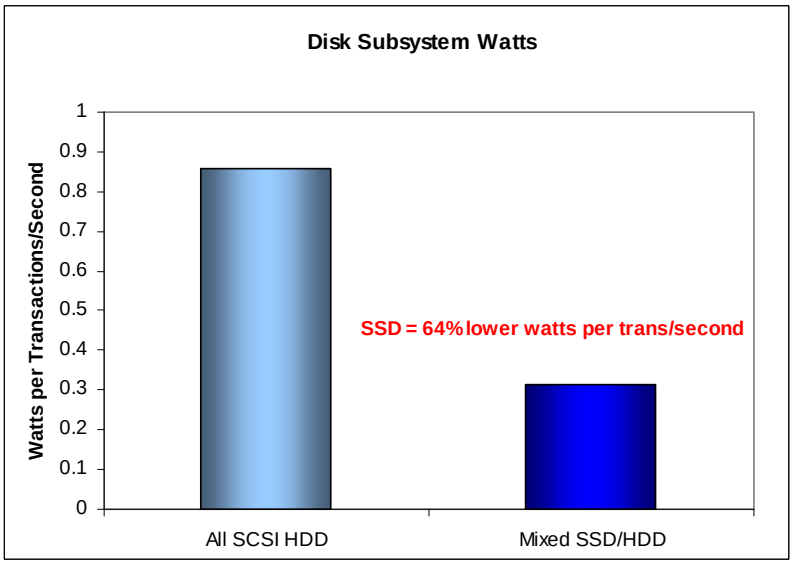

Figure 11: Disk Subsystem Watts per Transactions/Second

Scenario 4: Mixed ASP Configuration vs. All SCSI HDD using DB2 and SSD Integration

For this scenario the database ASP contained 8 SAS SSDs in an 5886 EXP12S DASD drawer plus 36 SAS HDDs (12 per drawer), connected using 2 x 5908 1.5G SAS RAID controllers compared to 108 SCSI HDDs, see figure 12 below. The hot database files were moved to the SSDs using the CHGPF command and given a media preference of *SSD.

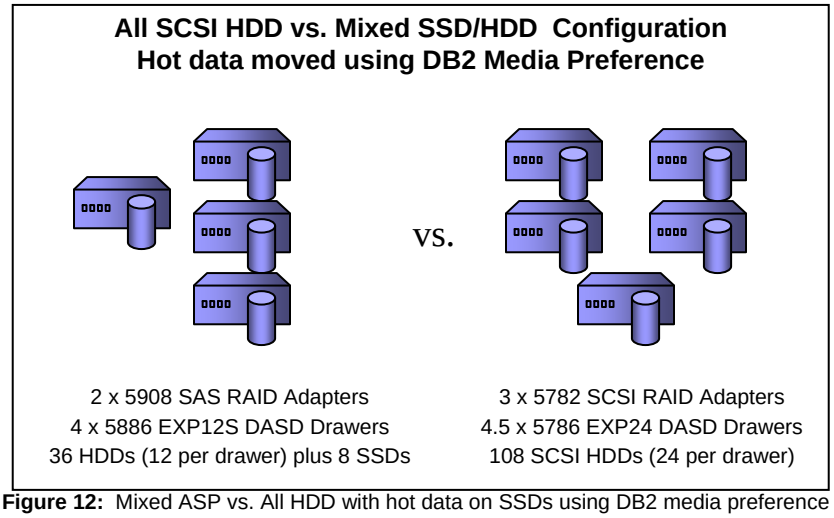

Figure 12: Mixed ASP vs. All HDD with hot data on SSDs using DB2 media preference

Results for Scenario 4

The purpose of this scenario is to show the benefit of using the media preference for a DB2 database table, to move the hot tables to the SSDs while leaving the cold data on HDDs. See the section below titled *DB2 and SSD Integration* to see how hot data placement can be performed. Figure 13 below once again shows the application response time curves as more work was added to the system. Once again, by placing the hot database tables on SSDs more throughput can be achieved even when the total number of devices in the database ASP was decreased by roughly 60%.

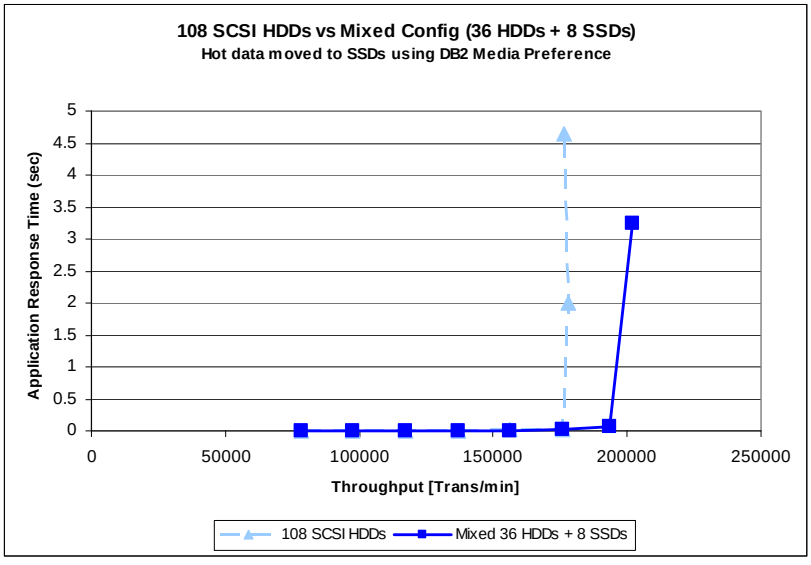

Figure 13: 108 HDD vs. Mixed config with hot data moved to SSDs using DB2 media preference

Figure 14 below examines the average throughput per drive once the hot data was placed onto the SSDs. The mixed ASP was able to maintain 2.5x higher transactions per drive on average, over the all HDD ASP, while the average application response time was reduced by 50%. As with the previous scenario, the purchase price for the mixed SSD/HDD configuration is also roughly 30% lower when compared to the SCSI HDD configuration.

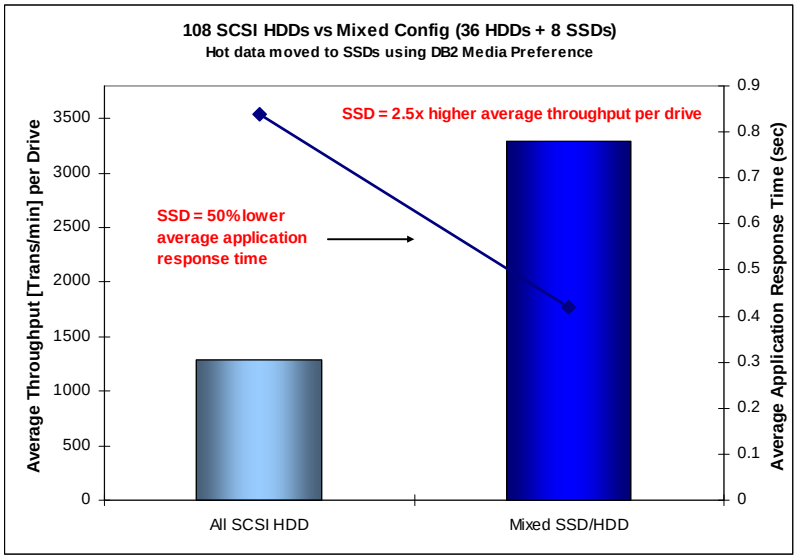

Figure 14: SCSI HDD ASP vs Mixed SSD/HDD ASP. Average Throughput [Trans/min] per Drive

As mentioned before, another benefit of an SSD over an HDD is the number of IOPS SSDs are able to handle while keeping response time low. Figure 15 shows that on average, the database ASP response time is 50% lower and IOPS per drive increase by 3x. This again is impressive since the Mixed SSD/HDD configuration has 60% fewer drives.

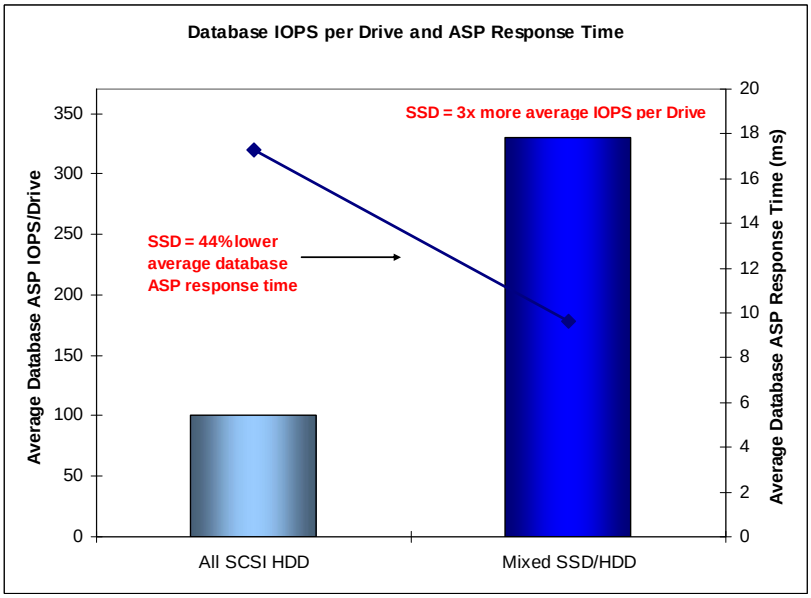

Figure 15: SCSI HDD vs Mixed SSD/HDD Average Database ASP IOPS per Drive

Quantifying the Benefits of SSDs using SAP BW Workload

The SAP Business Warehouse (BW) application, like most Online Analytical Processing (OLAP) applications, generates a variety of queries that can lead to large amounts of I/O as the business users are unleashed to discover new information from the vast amount of data that is stored in the warehouse. The queries that require a lot of paging can benefit from the use of SSDs. This section will explore the performance of a SAP BW workload in three environments: pure HDD, pure SSD and a hybrid environment of both HDDs and SSDs.

Scenario 5: SAP BW Workload

The tests described below do not use the SAP BI Mixed Load (BI-MXL) Standard Application Benchmark as designed and are not SAP certified benchmarks, nor should they be compared to any SAP certified benchmarks. The workload used is a modified version of the SAP BI-MXL workload that is redesigned to stress the disk subsystem. Although it is still very similar in business functions performed, it is not the Standard Application Benchmark.

The SAP BI-MXL workload we modified was developed by SAP to simulate a customer environment. The workload simulates work performed by a business end user in an Online Analytical Processing (OLAP) environment; therefore, it performs many reads against the disk subsystem and only a few writes. This environment contains 10 SD InfoCubes containing 300 million rows each. Each user loops through 11 steps that simulate 2 typical drilldown scenarios and 2 additional reports. The throughput is measured in dialog steps per hour.

The following steps describe the operations done within the SAP application during the BI-MXL workload.

Step	***Operation***
1	Select a year
2	Drill down to a specific country
3	Drill down to a specific sales organization
4	Drill down to a specific distribution channel
5	Drill down to type, version and material number
6	Change material number and customer number
7	Start new query including formulas
8	Start new query retrieving yearly and quarterly sales volumes for different ranges
9	Start new query for country, sales organization and distribution channel
10	Expand to material level 2
11	Expand to material level 3

The standard BI-MXL workload usually uses a “think time” variable to determine the amount of time the user takes between steps. One example of our modifications is that this variable was set to zero to increase the amount of work performed by the disk subsystem.

Hardware Configuration

All tests were performed on a POWER 550 8204-E8A using IBM i v6.1. All disk units are contained in the system ASP, RAID5 protected and attached using 5908 SAS RAID controllers. The pure HDD environment uses 4 controllers with 24 HDDs per controller. The pure SSD environment uses 2 controllers with 8 SSDs per controller. All HDDs were 140 GB SAS 15k RPM drives and all SSDs were 70 GB SAS drives.

The mixed environment introduces an additional controller with 4 SSDs to the pure HDD environment with the expectation of providing a turbo boost to throughput and response time (user and disk). After the empty SSDs are added to the ASP, the data is balanced using the ASP Balancer tool described in the section below titled *"ASP Balancer – Enhanced for SSDs"*.

.

Results

The following table shows the measurement results for each of the scenarios.

	HDD	***SSD***	***HDD+SSD***
Users	20	20	20
Dialog Steps/Hour	4962	5802	5576
User Response Time (seconds)	14.6	12.6	12.9
CPU Utilization	82.2	92.3	89.9
Drives	96	16	96 HDD + 4 SSD
I/O per second	17724	30512	20533
DASD Response Time (ms)	14.9	4.2	9.3

Table 1: SAP BW Workload Results

The use of all SSDs provided the best results with a 17% increase in throughput and a 14% decrease in user response time while having 83% fewer drives than the HDD environment. The addition of 4 SSDs into a mixed environment did provide a boost in performance with a 12% increase in throughput and 12% decrease in user response time as well as 38% lower DASD response time. Figure 16 below shows the improvement achieved in both throughput and response time by switching to either a pure SSD environment or by adding a few new SSDs to an existing environment.

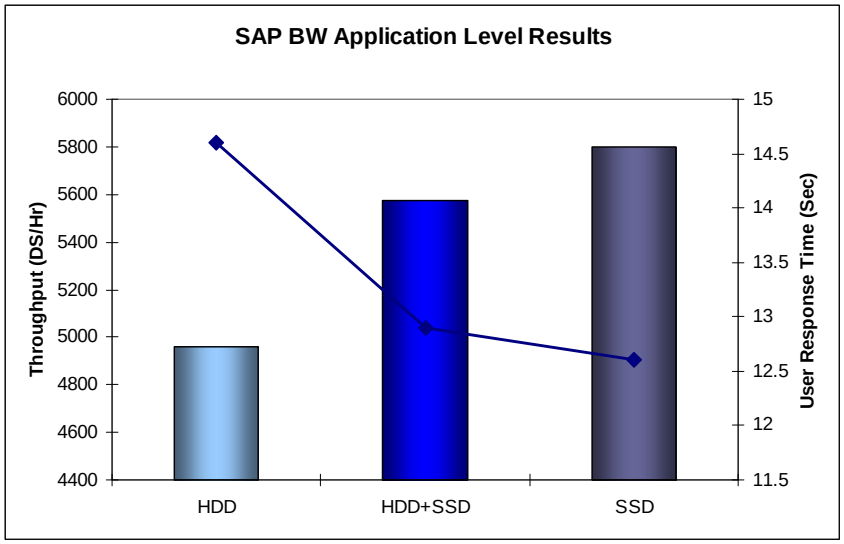

Figure 16: SAP BW Workload results

The total disk space used for the whole system was 487 GB. After the balancer completed moving hot data to the new SSDs in the mixed environment, the SSDs contained 25 GB of data, or 11.9% of the available SSD capacity. Figure 17 below shows how the ASP Balancer evenly spread the I/O across the SSD and HDD units within the mixed configuration. The 96 HDDs and 4 SSDs roughly did equivalent IOPS while the SSDs maintained a 6ms disk response time.

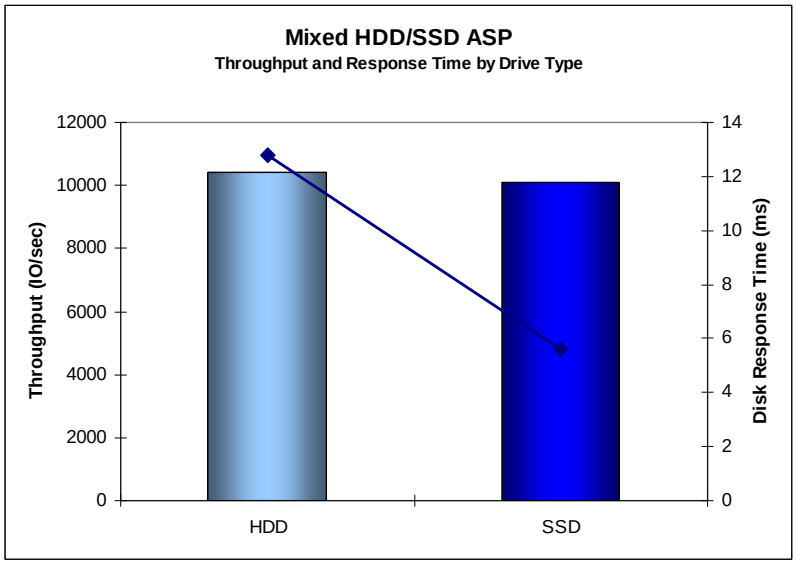

Figure 17: SAP BW Workload Mixed HDD/SSD ASP

Summary of above scenarios

The following table summarizes the scenarios from the previous pages in a simple table so all performance numbers can be seen together.

Scenarios	Configuration: HDD only (all 15k rpm)	Configuration: with SSD	Performance with SSD (Note: These results measured using sample workloads designed to exercise I/O and may differ noticeably from customer workloads.)	OS & sample workloads used
Scenario 1: HDD to SSD	1.5GB SCSI adapter + 36 SCSI HDD	1.5GB SAS adapter + 8 SSD	• 78% fewer drives • 64% lower average application response time • 4.5X higher average throughput per drive • 5.5X higher IOPS/device • 30% lower average ASP disk response time	IBM i & OLTP/DB
Scenario 2: SCSI HDD to SSD	(4) 1.5GB SCSI adapters + 108 SCSI HDD	(2) 1.5GB SAS adapters + 12 SSD	• 89% fewer drives • 81% lower average application response time • 9X higher average throughput per drive • 12X higher IOPS/device • 65% lower average ASP disk response time • 24% lower storage subsystem purchase cost	IBM i & OLTP/DB
Scenario 3: SCSI HDD to Mixed HDD/SSD ASP Balancer	(4) 1.5GB SCSI adapters + 108 SCSI HDD	(2) 1.5GB SAS adapters + 8 SSD + 36 HDD	• 60% fewer drives • 91% lower average application response time • 2.5X higher average throughput per drive • 2.4X higher IOPS/device • 50% lower average ASP disk response time • 30% lower storage subsystem purchase cost • 64% lower storage subsystem energy usage	IBM i & OLTP/DB
Scenario 4: SCSI HDD to Mixed HDD/SSD DB2 media preference flag	(4) 1.5GB SCSI adapters + 108 SCSI HDD	(2) 1.5GB SAS adapters + 8 SSD + 36 HDD	• 60% fewer drives • 50% lower average application response time • 2.5X higher average throughput per drive • 3X more IOPS/device • 44% lower average ASP disk response time • 30% lower storage subsystem purchase cost • 64% lower storage subsystem energy usage	IBM i & OLTP/DB
Scenario 5: SAS HDD to SSD	(4) 1.5GB SCSI adapter + 96 SCSI HDD	(2) 1.5GB SAS adapters + 16 SSD	• 83% fewer drives • 14% lower user response time • 17% higher dialog steps/hour • 10X higher IOPS/device • 72% lower disk response time	IBM i & SAP BW workload
Scenario 5: SAS HDD to Mixed HDD/SSD ASP Balancer	(4) 1.5GB SCSI adapter + 96 SCSI HDD	(5) 1.5GB SAS adapters + 96 HDD + 4 SSD	• 12% lower user response time • 12% higher dialog steps/hour • 11% higher IOPS/device • 38% lower disk response time	IBM i & SAP BW workload

IBM i Tools and Commands

Now that we have seen the benefit of integrating SSDs into a production environment, let us take a closer look at how to implement these functions.

Again, there are three basic methodologies to place data on SSD.

- ASP Balancer – Enhanced for SSDs (used in scenario 3 & 5)
- Library and SSD Integration
- DB2 and SSD Integration (used in scenario 4)

ASP Balancer – Enhanced for SSDs

IBM i users have long been able to balance the data in their Auxiliary Storage Pools (ASPs) based on the capacity or the usage of each drive. With the arrival of SSDs, the Hierarchical Storage Management (HSM) balancing option has been enhanced to allow users to balance data based on media preference, or the type of drive being used. The HSM option balances the data by moving the most frequently accessed "hot" data to the faster SSD units, while moving the least frequently accessed "cold" data to the slower, spinning disks. The ASP Balancer works with all types of ASPs, including the System ASP, user ASPs and Independent ASPs (IASPs). It can also be used on Integrated File System file systems, such as the QSYS and QDLS.

Like the Usage type of balancing method, balancing the hot and cold data requires two steps: Trace and Balance. First, run the Trace ASP Balance (TRCASPBAL) command to collect usage statistics. The Trace function monitors how frequently data is accessed on each drive. It counts the number of I/O Requests for each 1 MB "stripe" of each disk unit in the ASP. It also tracks how often each disk drive in the ASP is accessed.

After the data has been collected with the Trace command, the user runs the ASP Balance (STRASPBAL) command, specifying TYPE(*HSM)[1]. Based on the usage statistics gathered with the trace command, the balance command moves the hot data to the SSD drives. At the same time, other balancer tasks move the cold data from the SSD drives to the slower HDD drives.

The balance command works best when it is run soon after the trace command completes. This ensures that the data is balanced based on statistics that characterize the workload of the system. The trace and balance commands can be run on a periodic basis to ensure that the hot data continues to reside on the SSD drives.

Following is an example in which a user runs a trace for three hours and then runs the balancer until it is complete:

1. **Start a trace session on ASP 1 for three hours (180 minutes)**
   TRCASPBAL SET(*ON) ASP(1) TIMLMT(180)

2. **Run the balancer on ASP 1 until it is finished**
   STRASPBAL TYPE(*HSM) ASP(1) TIMLMT(*NOMAX)

The time required to move the data depends on a number of factors, including the number and size of the drives in the ASP. If necessary, a user can run the balance function for a few hours, stop it when the system is at peak usage, and then start it again when the system is less busy. When the balance function restarts, it will continue from the point where it left off. This allows the user to balance the drives with a low impact to the system.

Here is an example in which a user runs a trace for four hours, runs the balancer for three hours, then stops it and restarts it again, allowing it to complete.

1. **Start a trace session on ASP 4 for four hours (240 minutes)**
   TRCASPBAL SET(*ON) ASP(4) TIMLMT(240)

2. **Run the balancer on ASP 4 for three hours**
   STRASPBAL TYPE(*HSM) ASP(4) TIMLMT(180)

3. **Stop the balancer**
   ENDASPBAL ASP(4)

4. **Restart the balancer on ASP 4, and allow it to continue until the data is completely balanced.**
   STRASPBAL TYPE(*HSM) ASP(4) TIMLMT(*NOMAX)
   Note: The balancer continues from the point where it left off in Step 2.

The introduction of the media preference enhancement to the ASP Balancer allows IBM i users to increase the overall ASP performance of their SSD and HDD drives in hybrid environments. Every user environment is unique. Using the ASP Balancer to move data between SSDs and HDDs requires careful planning to maximize the benefit. Customers should contact Advanced Technical Support or IBM Systems Lab Services and Training for more information about how to design and configure the system to obtain optimal performance when using the ASP Balancer.

Library and SSD Integration

A library can be created with an ASP attribute. If a user configures a set of SSDs into a separate ASP, this allows every object created into the library to be stored on those SSDs.

For example, in CL:

CRTLIB x **ASP**(n)

In SQL, the same capability is also supported:

CREATE SCHEMA x **IN ASP** n

This technique is reasonable if all the objects in the library have critical performance requirements. However, this technique also has several drawbacks.

- The ASP of a library cannot be changed, so save and restore would be necessary to move an existing library and its objects to another ASP.
- File objects cannot be moved between ASPs, so save and restore is necessary to move any existing files to a library in a different ASP.
- An entire file network (a file and all the dependent files) must be on the same ASP. This means that you would need to make sure that enough SSD disk space was available for all the files in the network even if some of those related logical files or indexes did not have critical performance requirements.
- In SQL, CREATE SCHEMA to an ASP with SSDs would also create journal receivers into the library and hence journal receivers would take up space on the ASP. Journal receivers are not good candidates for SSD because journal entries are primarily sequentially written and sequentially read. Since they are written and read in a sequential manner, allocating them on SSD will typically not result in a significant performance improvement.

- If you determine that only a subset of the objects in the library should be on SSD, and you want to move a subset to another library, you may need to make application changes where the library name is explicitly specified.

Due to all these restrictions, using this technique may not be practical. However, you could use this technique to move an entire library of low priority objects or rarely accessed data to an ASP that consists only of traditional disks.

DB2 and SSD Integration

Like many new technologies, the initial cost of SSDs is relatively high compared to traditional disks. Since the cost is higher, customers are more likely to initially purchase and deploy a small number of SSDs into their current set of traditional disks. To get the most value out of SSDs, customers will want to make sure that the data accessed by those applications that have the most critical performance requirements is on SSD.

To allow a user to specify what data should be allocated on SSD, DB2 has provided the capability to specify a "media preference" as an attribute of a database table, partition, or index. It should be noted that this attribute specifies that storage allocations on SSD are preferred, but if no SSD disks are available or if the SSD disks do not have enough space left to allocate the entire object, at least some part of the object will be allocated on traditional disks.

Creating Database Objects on SSD

A new physical file or logical file can be created with a media preference of SSD. The CL commands CRTPF, CRTLF, and CRTSRCPF support a UNIT(*SSD) and UNIT(*ANY) parameter. For example:

CRTPF department **SRCFILE**(mjasrc/dds) **UNIT**(*SSD)[3]

CRTLF departmentl **SRCFILE**(mjasrc/dds) **UNIT**(*SSD)

Likewise, a new SQL table or SQL index can be created with a media preference of SSD. The SQL statements CREATE TABLE and CREATE INDEX support a UNIT clause. For example,

CREATE TABLE employee (c1 INT) **UNIT** SSD[4]

CREATE INDEX employeeix **ON** mjatst.t2 (c1) **UNIT** SSD

If a specified table or physical file has multiple partitions or members, the media preference specified in the CL commands or SQL statements above applies to all partitions or members of that table or physical file. However, a media preference can also be specified on each partition. For example:

[3] In V5R4, use UNIT(255).

[4] In V5R4, the SQL UNIT clause is not supported. CHGPF and CHGLF can be used to specify a media preference for SQL tables and indexes.

```
CREATE TABLE sales
    (salesdate DATE, storenbr INT, itemnbr INT, quantity INT, price DECIMAL(9,2) )
      PARTITION BY RANGE (salesdate NULLS LAST)
    (PARTITION sales2006 STARTING('2006-01-01') ENDING('2006-12-31') UNIT ANY,
    PARTITION sales2007 STARTING('2007-01-01') ENDING('2007-12-31') UNIT ANY,
    PARTITION sales2008 STARTING('2008-01-01') ENDING('2008-12-31') UNIT SSD)
```

This technique is useful if some partitions of the table have much higher performance requirements than others.

Changing Existing Database Objects on SSD

An existing physical file, SQL table, logical file, or SQL index can be changed to specify a media preference. The CL commands CHGPF, CHGLF, and CHGSRCPF support a UNIT(*SSD) and UNIT(*ANY) parameter. For example:

```
CHGPF employee UNIT(*SSD)[3]
```

```
CHGLF lemployee UNIT(*SSD)
```

Likewise, an existing physical file or SQL table can be changed to specify a media preference. The SQL statement ALTER TABLE supports a UNIT clause. For example,

```
ALTER TABLE employee UNIT SSD[4]Error! Bookmark not defined.
```

When CHGPF UNIT(*SSD) or ALTER TABLE UNIT SSD is specified, DB2 will asynchronously move the file to SSD storage if available. When CHGPF UNIT(*ANY) or ALTER TABLE UNIT ANY is specified, DB2 will asynchronously move the file to traditional disks, if available. An exclusive lock is required to change the media preference, but once changed, the exclusive lock is released and the asynchronous move will continue in background tasks.

Individual partitions of an existing partitioned table can be changed to specify a media preference. The SQL statement ALTER TABLE supports a UNIT clause on a partition. For example:

```
ALTER TABLE sales
   ALTER PARTITION sales2008
      STARTING('2008-01-01') ENDING('2008-12-31') UNIT ANY
  ADD PARTITION sales2009
      STARTING('2009-01-01') ENDING('2009-12-31') UNIT SSD
```

This technique is useful if a new partition of the table has a much higher performance requirement or an existing partition is no longer as performance critical.

Conclusion

Customer I/O demands have outpaced the performance capabilities of traditional HDDs. Latencies associated with spinning platters and moving arms limit the speed of HDD data access. SSDs' near instantaneous data access removes this I/O bottleneck, creating a paradigm shift in I/O performance. Applications throttled by poor I/O performance can benefit greatly from SSDs.

As demonstrated in the above scenarios, SSDs result in a substantial improvement in I/O performance, which translates to increased business output, reduced energy consumption and in some cases decreased cost

The superior performance of SSDs must be balanced with cost. Multi-tiered storage solutions can provide that balance. An application's hot data can be moved to SSDs, while less active data can remain on lower cost HDDs. With the introduction of the commands and tools to help balance the performance critical data onto SSDs customers can benefit greatly.

Customers wishing to learn how solid state disk technology can be best utilized for their IBM i workloads should contact their sales team for IBM products and services. Sales teams needing technical sales assistance should submit a request for technical sales assistance via Deal Hub Connect.

Acknowledgments

Special thanks to all who contributed to this paper and all who contributed during the review process. In addition, special thanks to our performance colleagues who shared sections of their paper titled *"Driving Business Value on Power Systems with Solid State Drives"* written by Douglas, Gao, Romero, et al.

For More Information

IBM Power System Servers	**http://www.ibm.com/systems/power**
IBM Performance Management on IBM i	**http://www.ibm.com/systems/i/advantages/perfmgmt/resource.html**
IBM Systems Energy Estimator	**http://www.ibm.com/systems/support/tools/estimator/energy**
IBM Systems Lab Services and Training	**http://www.ibm.com/systems/services/labservices**

Legal Information

IBM Corporation
Systems and Technology Group
Route 100 Somers, NY 10589

Produced in the United States
April 2009

This publication could include technical inaccuracies or photographic or typographical errors. This publication was produced in the United States. IBM may not offer the products, services or features discussed in this document in other countries, and the information may be subject to change without notice. References herein to IBM products and services do not imply that IBM intends to make them available in other countries. Consult your local IBM business contact for information on the products or services available in your area. Information about non-IBM products is obtained from the manufacturers of those products or their published announcements. IBM has not tested those products and can not confirm the performance, compatibility or any other claims related to non-IBM products. Questions on the capabilities of non-IBM products should be addressed to the suppliers of those products.

Some information in this document addresses anticipated future capabilities. Such information is not intended as a definitive statement of a commitment to specific levels of performance, function or delivery schedules with respect to any future products. Such commitments are only made in IBM product announcements. The information is presented here to communicate IBM's current investment and development activities as a good faith effort to help with our customers' future planning.

IBM, the IBM logo, **ibm.com**, AIX, IBM i are trademarks or registered trademarks of IBM Corporation in the United States, other countries or both. If these and other IBM trademarked terms are marked on their first occurrence in this information with a trademark symbol (® or ™), these symbols indicate U.S. registered or common law trademarks owned by IBM at the time this information was published. Such trademarks may also be registered or common law trademarks in other countries. A current list of IBM trademarks is available on the Web at "Copyright and trademark information" at **ibm.com**/legal/copytrade.shtml.

Linux is a trademark of Linus Torvalds in the United States, other countries, or both. InfiniBand, InfiniBand Trade Association and the INFINIBAND design marks are trademarks and/or service marks of the InfiniBand Trade Association. Other company, product and service names may be trademarks or service marks of others.

When referring to storage capacity, 1 TB equals total GB divided by 1000; accessible capacity may be less. 1GB equals 10*9 Bytes. 1Gb = 10*9 bits.

MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE. Some jurisdictions do not allow disclaimer of express or implied warranties in certain transactions; therefore, this statement may not apply to you.

This publication may contain links to third party sites that are not under the control of or maintained by IBM. Access to any such third party site is at the user's own risk and IBM is not responsible for the accuracy or reliability of any information, data, opinions, advice or statements made on these sites. IBM provides these links merely as a convenience and the inclusion of such links does not imply an endorsement.

Information in this presentation concerning non-IBM products was obtained from the suppliers of these products, published announcement material or other publicly available sources. IBM has not tested these products and cannot confirm the accuracy of performance, compatibility or any other claims related to non-IBM products. Questions on the capabilities of non-IBM products should be addressed to the suppliers of those products.

Performance results set forth in this document are based on measurements and projections using standard IBM benchmarks in a controlled environment. The actual performance that any user will experience will depend on considerations such as the amount of multiprogramming in the user's job stream, the I/O configuration, the storage configuration and the workload processed. Therefore, no assurance can be given that an individual user will achieve performance improvements equivalent to the performance ratios stated here.

SAP, R/3, mySAP, mySAP.com, xApps, xApp, SAP NetWeaver and all SAP product and service names mentioned herein are trademarks or registered trademarks of SAP AG in Germany and in several other Countries all over the world.