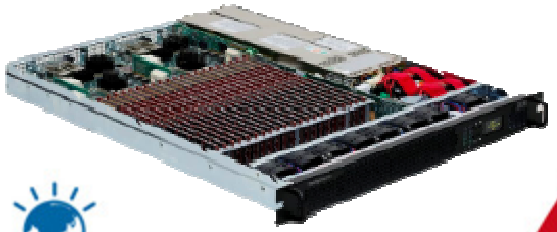
IBM FlashSystems with IBM i

Proof of Concept & Performance Testing

Fabian Michel

Client Technical Architect



Abstract

This document summarizes the results of a first Proof of Concept and performance testing of an environment running IBM i with an IBM FlashSystem attached to an IBM Storwize V7000.

The goals were:

- To demonstrate the simplicity of the setup
- To validate if IBM FlashSystems could help improving performance on an IBM i environment, and
- To use performance data with existing tools such as the IBM i SSD Analyzer Tool or iDoctor to help you predict what kind of workloads or jobs would benefit the most from IBM Flash technology.

Agenda

- Quick Introduction to IBM FlashSystems - The Theory
- Current Official Support Status of IBM FlashSystems with IBM i
- Proof of Concept configuration setup
- Tests performed
- Impact on the workloads – The Practice
- Conclusions

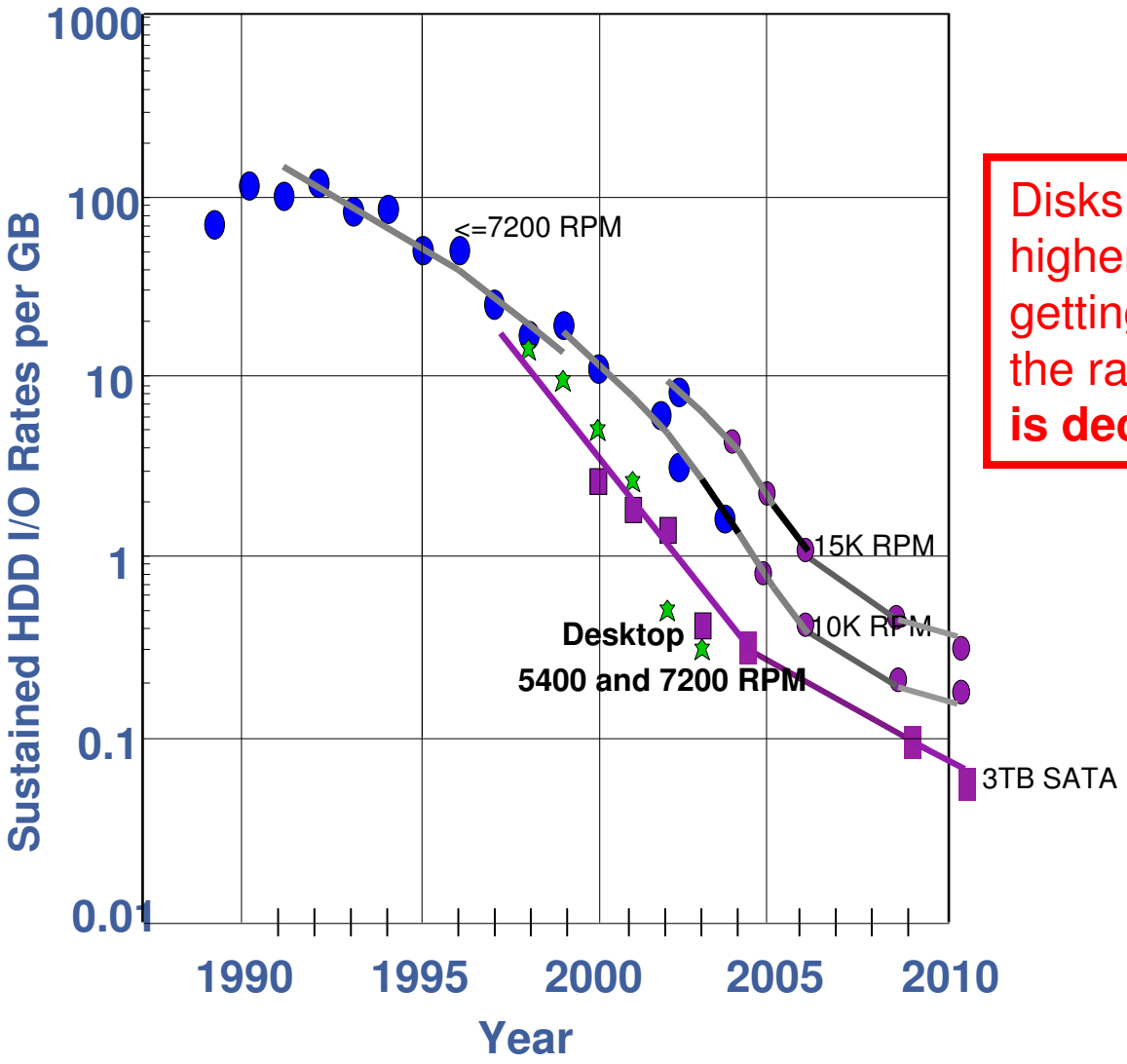
What our clients struggle with…

In the last 10 years:

- CPU speed: increased roughly **8-10x**
- DRAM speed: increased roughly **7-9x**
- Network speed: increased roughly **100x**
- Bus speed: increased roughly **20x**
- Storage speed: increased *only* **1.2x… until now!**

All parts of the infrastructure improve when storage improves !

Ratio of spinning disk I/O performance to capacity



Disks are getting denser at a higher rate than they are getting faster. Consequently, the ratio of **I/Os per GB** stored **is decreasing**.

What Makes the IBM FlashSystems a Better SSD?

Microseconds, not Milliseconds…

Database: Logs (redo/undo)
Tables
Temp files
Indexes

Represent about 80% of disk or I/O activity, but only about 2-5% of all their data. Shared Storage vs. Server dedicated

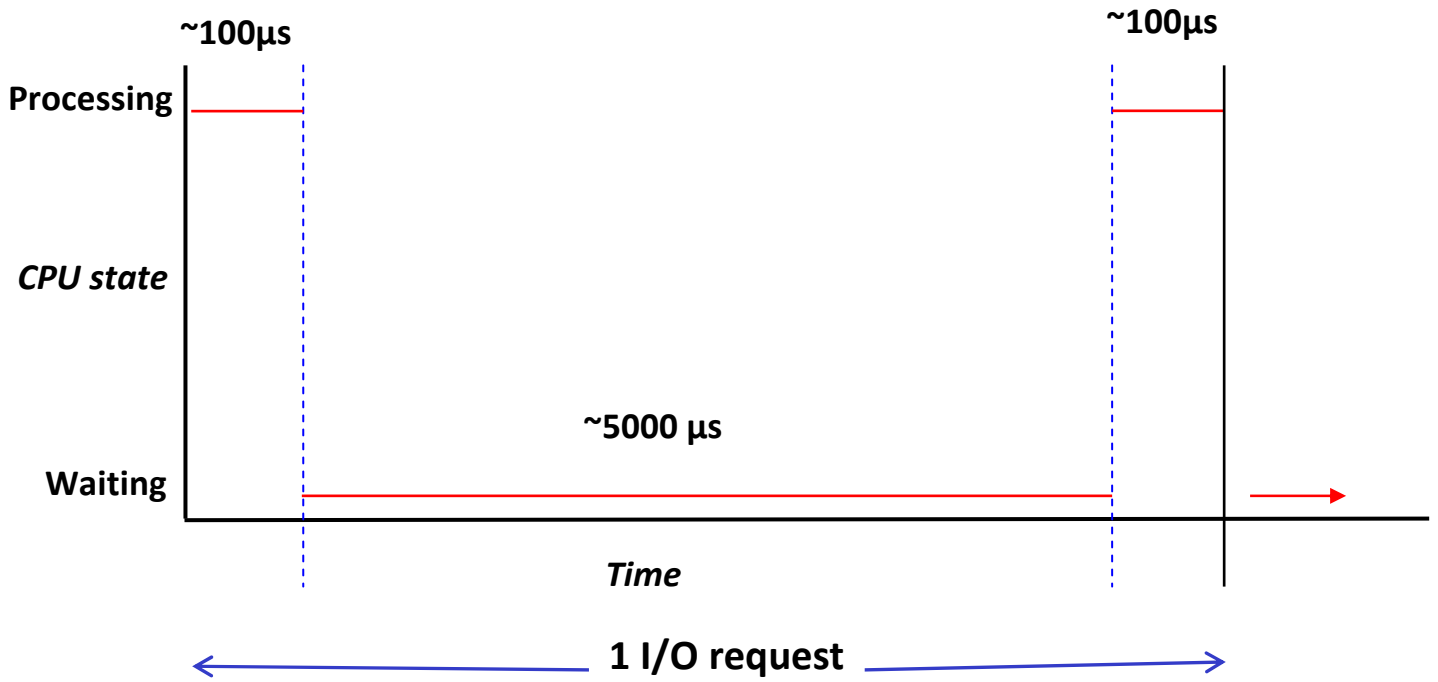
Tiering Storage According to Use/Speed

Flash System	RAID Cache	SSD	RAID	Tape
				
60/1000's of a **milli**seconds (60 **MICRO**seconds)	~1 millisecond	1-2 milliseconds	5-15 milliseconds	Seconds – minutes

I/O with disk

1. Issue I/O request (~ 100 μs)
2. Wait for I/O to be serviced (~ 5,000 μs)
3. Process I/O (~ 100 μs)

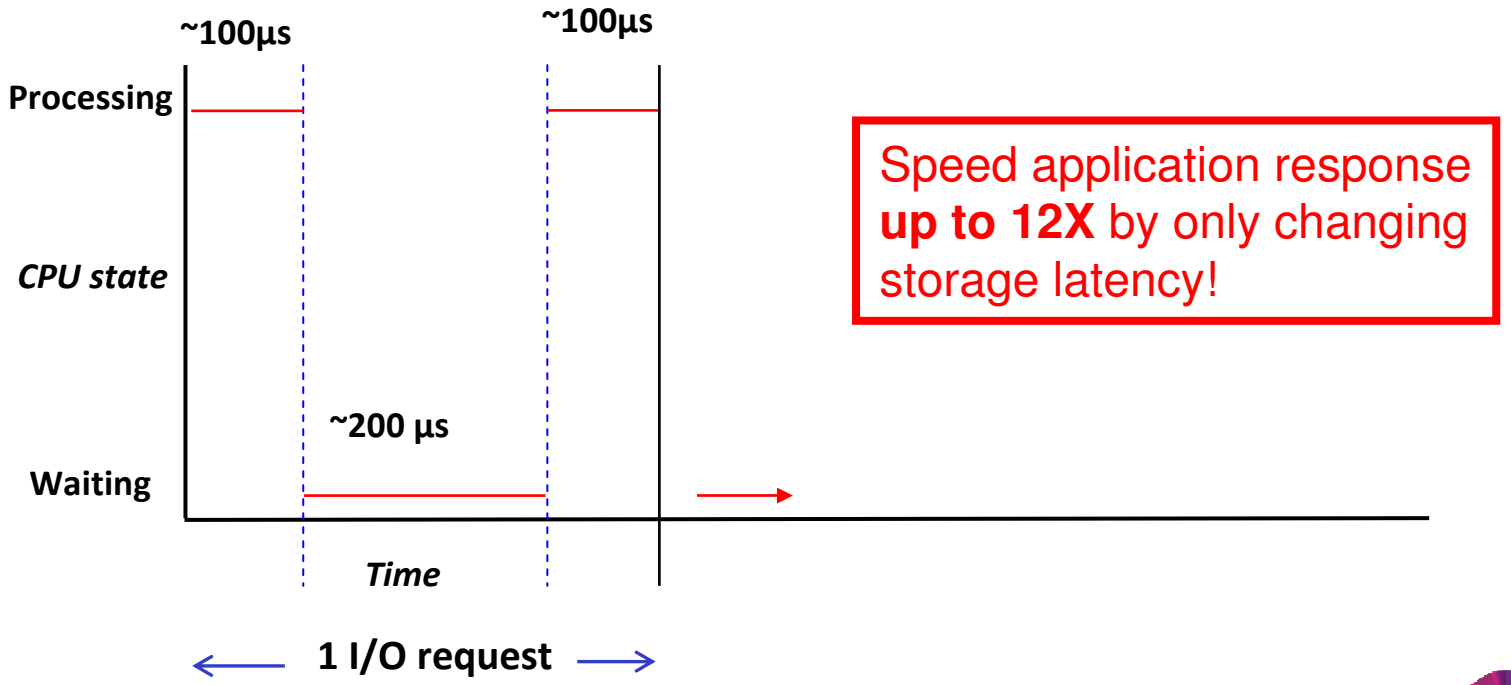
- Time to process 1 I/O request = 100 μs + 5,000 μs + 100 μs = 5,200 μs
- CPU utilization = Wait time / Processing time = 200 / 5,200 = **~4%**



I/O with flash

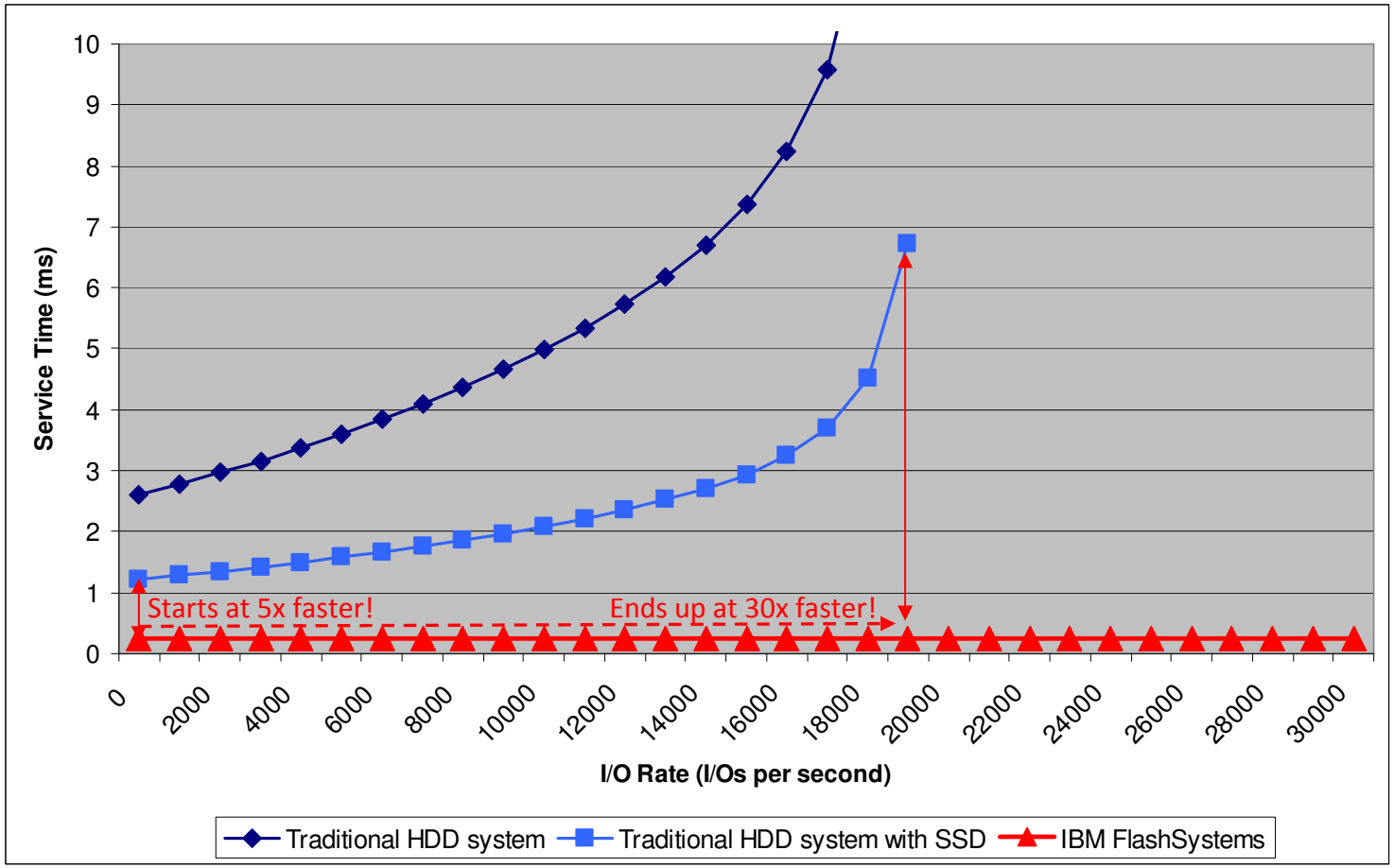
1. Issue I/O request (~ 100 μs)
2. Wait for I/O to be serviced (~ 200 μs)
3. Process I/O (~ 100 μs)

- Time to process 1 I/O request = 100 μs + 200 μs + 100 μs = 400 μs
- CPU utilization = Wait time / Processing time = 200 / 400 = **~50%**



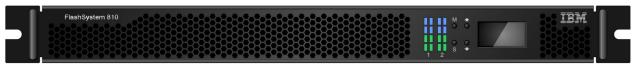
Speed application response **up to 12X** by only changing storage latency!

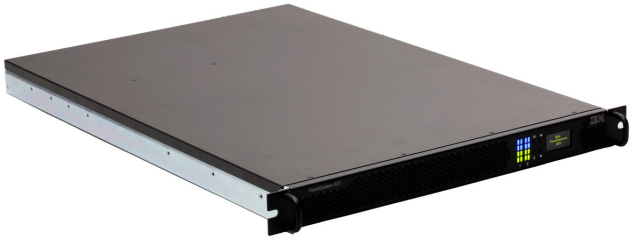
IBM FlashSystems - Service Time per I/O Rate



(*) This is a theoretical example for illustration purposes only, results will vary depending on configurations and workloads.

IBM FlashSystems

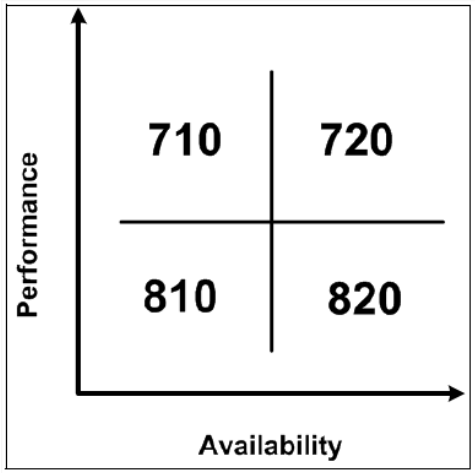




710/810
SLC (710) / eMLC (810) Flash
100/25 us R/W Latency
1-5 or 2-10 TB
450K/400K IOPS (4K)
5/4 GB/s
1U rackmount, 4x 8Gb FC ports, 4x 40Gb QDR InfiniBand
List Price Range (NA) $49K- $149K

Model	TB	Flash	Price
710	1,2,3,4, 5	SLC	45-149K
810	2,4,6,8, 10	eMLC	45-149K
720	5 or 10	SLC	174-325K
820	10 or 20	eMLC	174-324K

Model 810 with 6 TB was used for this PoC



720/820
SLC (720) / eMLC (820) Flash
100/25 us R/W Latency
5, 10, or 20 TB w/HA (6/12/24 TB non-HA)
500K/450K IOPS (4K)
5/4 GB/s
1U rackmount, 4x 8Gb FC ports, 4x 40Gb QDR InfiniBand
List Price Range (NA) $174K- $324K

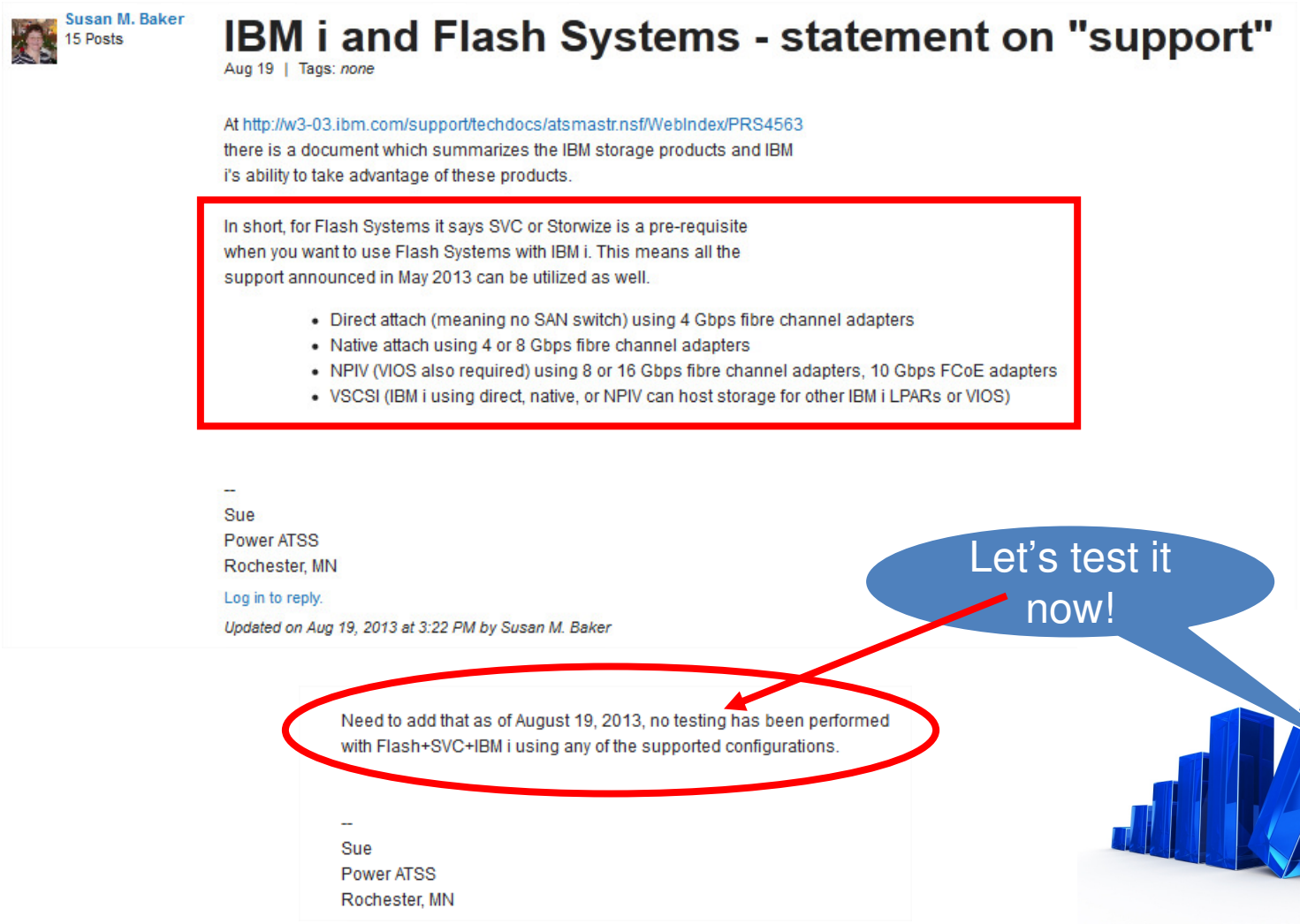
IBM i POWER External Storage Support Matrix

Last updated August 6, 2013		Rack / Tower Systems										Notes
		Server model			IBM i version			IBM i attach				IBM i 6.1 or later is a prerequisite for VIOS vSCSI, VIOS NPIV, and POWER7 systems. Native attach means the fibre channel IOA is allocated to the IBM i LPAR.
Storage Family		POWER5	POWER6	POWER7	5.4	6.1	7.1	Direct	Native	VIOS vSCSI	VIOS NPIV	
Storwize	V3500	⊘	✓	✓	⊘	✓	✓	⊘	⊘	✓	✓	**NPIV** support requires IBM i 7.1 TR6 or later
	V3700	⊘	✓	✓	⊘	✓	✓	✓	✓	✓	✓	**NPIV** support requires IBM i 7.1 TR6 or later **NATIVE** support requires - POWER7 or POWER7+ servers - IBM i 7.1 TR6 plus PTFs MF56600, MF56753, MF56854 or their supersedes - and 6.4.1.4 or later SVC/V3700/V7000 firmware.
	V7000	⊘	✓	✓	⊘	✓	✓	✓	✓	✓	✓	
SVC		⊘	✓	✓	⊘	✓	✓	✓	✓	✓	✓	**DIRECT** attach available for POWER7 systems when utilizing PCIe 4Gb fibre channel adapters (feature 5774 or 5276)
Flash Systems		⊘	⊘	⊘	⊘	⊘	⊘	⊘	⊘	⊘	⊘	There is no support without SVC or Storwize and SVC/Storwize firmware level 6.4.1.5 or higher is required. See support documentation at the URLs listed on the notes page.

The most current version of this document can be located at one of the following URLs:

- http://www.ibm.com/support/techdocs/atsmastr.nsf/WebIndex/PRS4563
- https://www-304.ibm.com/partnerworld/wps/servlet/ContentHandler/tech_PRS4563
- http://w3.ibm.com/support/techdocs/atsmastr.nsf/WebIndex/PRS4563

Current official status



Susan M. Baker
15 Posts

IBM i and Flash Systems - statement on "support"

Aug 19 | Tags: *none*

At http://w3-03.ibm.com/support/techdocs/atsmastr.nsf/WebIndex/PRS4563
there is a document which summarizes the IBM storage products and IBM
i's ability to take advantage of these products.

In short, for Flash Systems it says SVC or Storwize is a pre-requisite
when you want to use Flash Systems with IBM i. This means all the
support announced in May 2013 can be utilized as well.

- Direct attach (meaning no SAN switch) using 4 Gbps fibre channel adapters
- Native attach using 4 or 8 Gbps fibre channel adapters
- NPIV (VIOS also required) using 8 or 16 Gbps fibre channel adapters, 10 Gbps FCoE adapters
- VSCSI (IBM i using direct, native, or NPIV can host storage for other IBM i LPARs or VIOS)

--
Sue
Power ATSS
Rochester, MN

Log in to reply.

Updated on Aug 19, 2013 at 3:22 PM by Susan M. Baker

Let's test it now!

Need to add that as of August 19, 2013, no testing has been performed with Flash+SVC+IBM i using any of the supported configurations.

--
Sue
Power ATSS
Rochester, MN

https://w3-connections.ibm.com/forums/html/topic?id=bb1e9d49-0829-4c8c-bf1d-7a89ed4bbc4b

Configuration setup for this PoC

- IBM i v7.1 running on a Flex System POWER7 node p260 with 8 cores and 96GB of memory allocated to the logical partition
- VIOS version : v2.2.2.2
- Redundant Flex Chassis FC3171 SAN switches @ 8Gb
- IBM Storwize V7000 with 24x300GB 10krpm SAS HDD in RAID5
- 10 LUNs (volumes) of 70GB dedicated to the IBM i partition (vSCSI)
- 10 LUNs created on an IBM FlashSystem 810 equipped with 6TB
- I used IBM Storwize V7000 Mirrored volumes to redirect all READS operations to the IBM FlashSystem 810 defined as primary copy

Mirrored volumes

By using volume mirroring, a volume can have two physical copies. Each volume copy can belong to a different storage pool, and each copy has the same virtual capacity as the volume. In the management GUI, an asterisk (*) indicates the primary copy of the mirrored volume. The primary copy indicates the preferred volume for read requests.

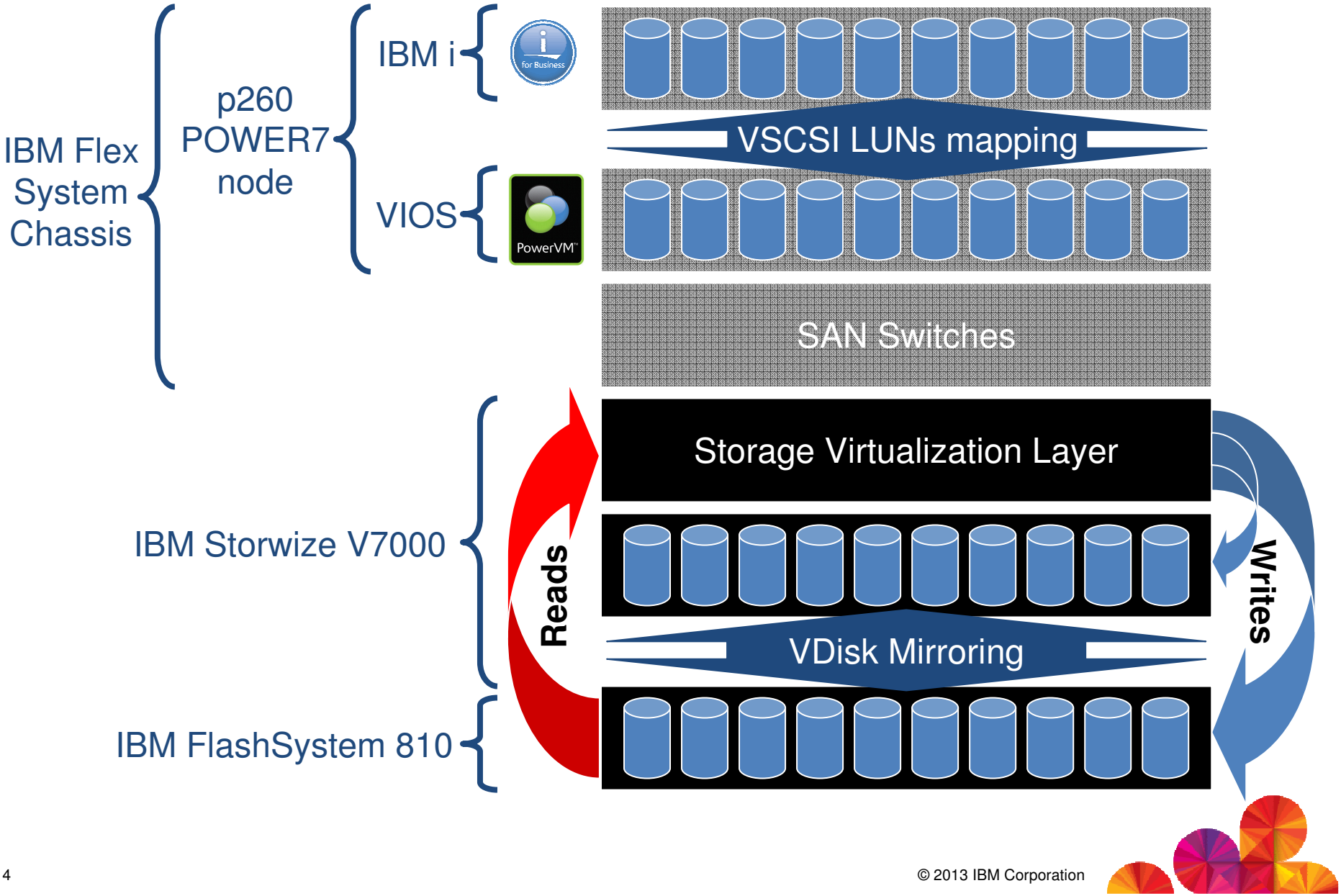
When a server writes to a mirrored volume, the system writes the data to both copies. When a server reads a mirrored volume, the system picks one of the copies to read. If one of the mirrored volume copies is temporarily unavailable; for example, because the storage system that provides the storage pool is unavailable, the volume remains accessible to servers. The system remembers which areas of the volume are written and resynchronizes these areas when both copies are available.

You can create a volume with one or two copies, and you can convert a non-mirrored volume into a mirrored volume by adding a copy. When a copy is added in this way, the Storwize® V7000 clustered system synchronizes the new copy so that it is the same as the existing volume. Servers can access the volume during this synchronization process.

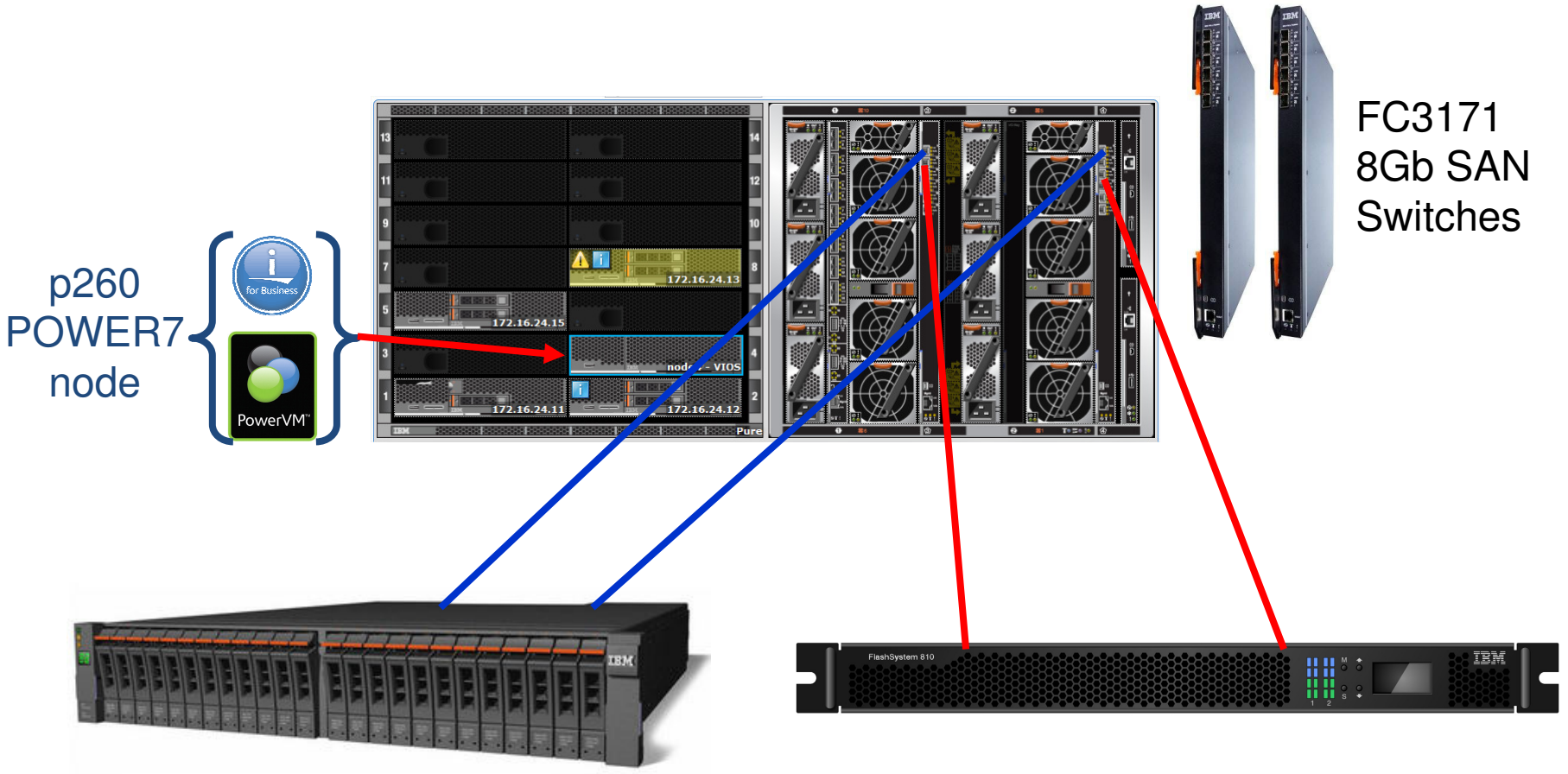
You can convert a mirrored volume into a non-mirrored volume by deleting one copy or by splitting one copy to create a new non-mirrored volume.

http://pic.dhe.ibm.com/infocenter/storwize/ic/topic/com.ibm.storwize.v7000.doc/svc_vdiskmirroring_3r7ceb.html

Setup – Logical View



HW Components overview

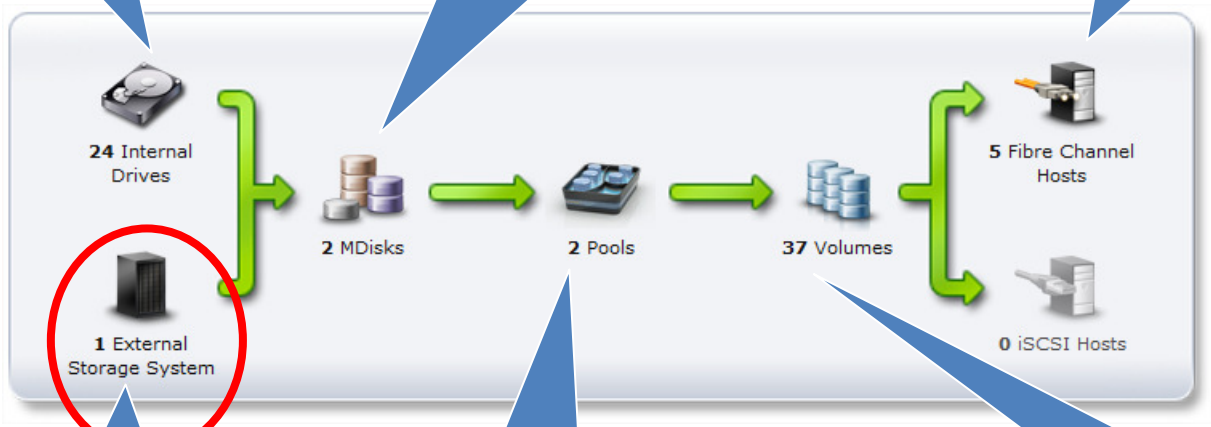


IBM Storwize V7000 Configuration Overview

The IBM Storwize V7000 internal HDDs

I created 2 MDisks : 1 for HDD & 1 for Flash

I mapped the 10 VDisk mirrored volumes to the IBM i host (VIOS)



The IBM FlashSystem 810

I set the 2 MDisks into 2 Pools : PoolPure1 for HDD and Flash_pool1 for Flash

I created 10 mirrored Volumes for IBM i : 10 volumes on HDDs copied to 10 volumes on FlashSystem 810

(*) There were other hosts running in this setup, that's why you see more hosts and volumes.

IBM Storwize V7000 Internal Storage (HDDs)

278.90 GB, SAS
10000 rpm
io_grp0

Actions

Drive ID	Capacity	Use	Status	MDisk Name	Enclosure ID	Drive Slot
0	278.90 GB	Candidate	Online		1	23
1	278.90 GB	Candidate	Online		1	24
2	278.90 GB	Candidate	Online		1	22
3	278.90 GB	Candidate	Online		1	20
4	278.90 GB	Candidate	Online		1	21
5	278.90 GB	Candidate	Online		1	18
6	278.90 GB	Candidate	Online		1	17
7	278.90 GB	Candidate	Online		1	19
8	278.90 GB	Candidate	Online		1	16
9	278.90 GB	Candidate	Online		1	15
10	278.90 GB	Candidate	Online		1	14
11	278.90 GB	Candidate	Online		1	13
12	278.90 GB	Member	Online	mdisk1	1	12
13	278.90 GB	Member	Online	mdisk1	1	11
14	278.90 GB	Member	Online	mdisk1	1	9
15	278.90 GB	Member	Online	mdisk1	1	10
16	278.90 GB	Member	Online	mdisk1	1	8
17	278.90 GB	Member	Online	mdisk1	1	5
18	278.90 GB	Member	Online	mdisk1	1	7
19	278.90 GB	Member	Online	mdisk1	1	6
20	278.90 GB	Member	Online	mdisk1	1	4
21	278.90 GB	Member	Online	mdisk1	1	3
22	278.90 GB	Member	Online	mdisk1	1	2
23	278.90 GB	Member	Online	mdisk1	1	1

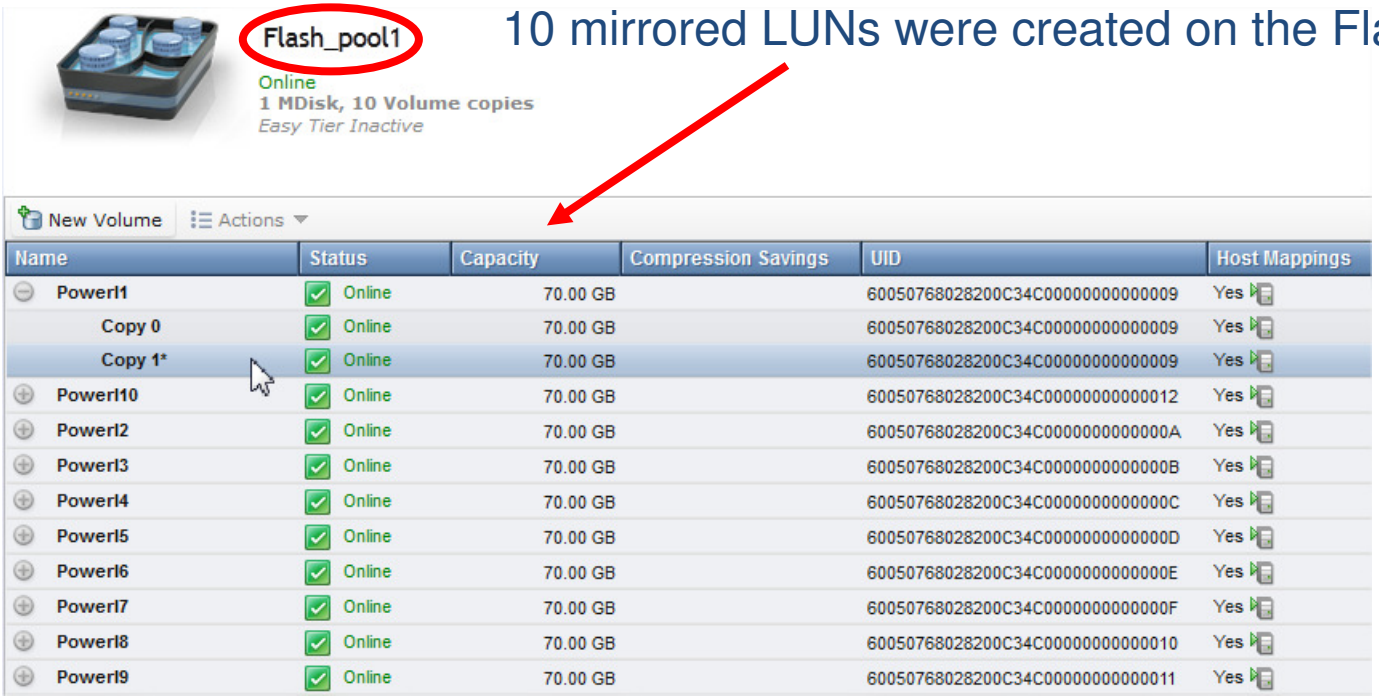
PoolPure1
Online
1 MDisk, 29 Volume copies
Easy Tier Inactive

New Volume | Actions

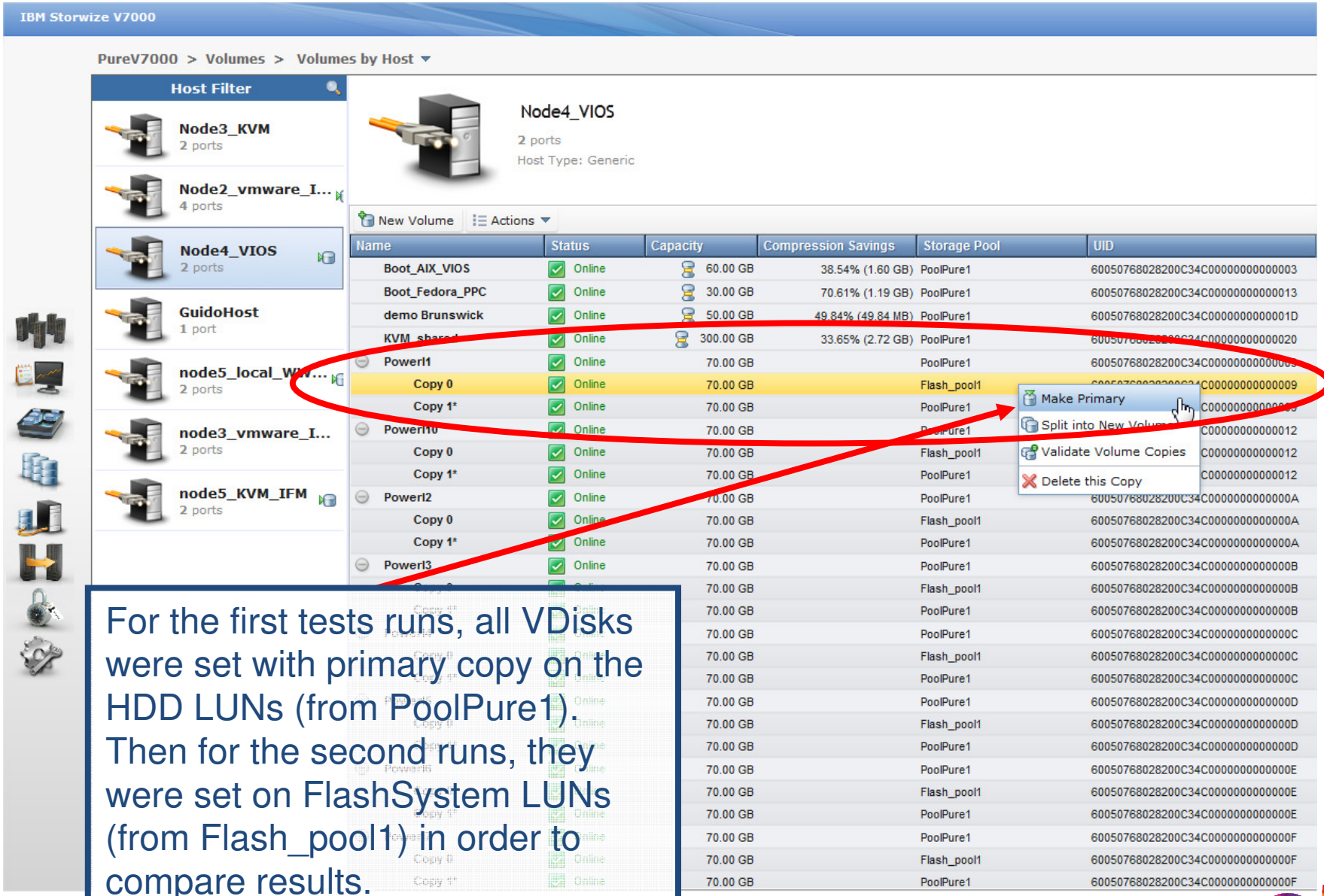
Name	Status	Capacity
Powerl1	Online	70.00 GB
Powerl10	Online	70.00 GB
Powerl2	Online	70.00 GB
Powerl3	Online	70.00 GB
Powerl4	Online	70.00 GB
Powerl5	Online	70.00 GB
Powerl6	Online	70.00 GB
Powerl7	Online	70.00 GB
Powerl8	Online	70.00 GB
Powerl9	Online	70.00 GB

The 10 IBM i LUNs were created on mdisk1 (12 physical 300 GB SAS disks) and set into PoolPure1

IBM Storwize V7000 External Storage – IBM FlashSystem 810



VDisk mirroring setup



For the first tests runs, all VDisks were set with primary copy on the HDD LUNs (from PoolPure1). Then for the second runs, they were set on FlashSystem LUNs (from Flash_pool1) in order to compare results.

Comparing IBM Storwize V7000 internal SAS HDDs with IBM FlashSystem 810

Tests performed and testing procedures

Tests descriptions

- **Test 1** : A complete power down restart cycle of IBM i – self explaining, a good mix of CPU, memory, read and write IOs.
- **Test 2** : The goal was to test streaming IOs (large block size). I used a backup to a SAVF of a library containing 5 large files for a total of 13GB (2,6 GB average file size).
- **Test 3** : Here, the objective was to test an analytic query like workload performing a lot of random reads. In order to do this I used the GO DISKTASKS option 1. It submits a job that collects information about the disk space, this function captures information on all inactive objects.
  - http://www-912.ibm.com/s_dir/slkbase.NSF/0/16c246792830c6b186256a5b005dcec8?OpenDocument
- **Test 4** : This time, the goal was to test a large set of relatively small file sizes, so less streaming when compared with Test 2. I used a backup to a SAVF of a large IFS directory (/QIBM) containing a mix of 190.713 files of miscellaneous sizes for a total of 18 GB (99KB average file size).

Tests descriptions(2)

Testing procedure :

After an IPL, I performed in sequence Test 2, Test 4, Test 3, then I issued a PWRDWNSYS RESTART(*YES) for TEST 1 and cycled the tests sequence

What was measured / compared ?

- Elapsed time
- IO/sec
- Response time

What scenarios were tested ?

1. The baseline, using V7000 HDD LUNs (primary/preferred LUN on HDD)
2. Then, setting the primary (preferred LUN) for each VDisk to the FlashSystem LUNs. This in order to redirect all READS operations to Flash memory only while WRITES operations remain in sync on both copies.
3. The impact of Real Time Compression (RTC)
4. The impact of disabling the IBM Storwize V7000 cache
5. Splitting the VDisk mirror link to use the IBM FlashSystem only (with V7000 cache disabled or enabled)

I took some screenshots during the runs…

TEST3 (GO DISKTASKS) with HDD as primary…

```
DISKTASKS                                  Disk Space Tasks

To select one of the following, type its number

     1. Collect disk space information
```

Unit	Type	Size (M)	% Used	I/O Rqs	Request Size (K)	Read Rqs	Write Rqs	Read (K)	Write (K)	% Busy
1	6B22	66810	23.1	85.2	6.8	63.2	21.9	7.6	4.5	43
2	6B22	66810	23.2	113.2	7.1	85.9	27.3	7.9	4.5	57
3	6B22	66810	23.1	117.2	7.1	90.2	26.9	7.9	4.5	62
4	6B22	66810	23.1	113.2	6.9	86.2	26.9	7.6	4.5	59
5	6B22	66810	23.1	117.2	6.8	90.6	26.6	7.5	4.5	60
6	6B22	66810	23.1	107.1	6.8	84.1	23.0	7.4	4.5	57
7	6B22	66810	23.1	132.6	6.4	78.7	53.9	7.7	4.5	61
8	6B22	66810	23.1	96.0	7.3	76.5	19.4	8.1	4.5	53
9	6B22	66810	23.1	111.4	6.9	84.1	27.3	7.7	4.5	59
10	6B22	66810	23.1	121.5	6.4	88.4	33.0	7.1	4.5	57

```
Job 066182/QSECOFR/QEZDKSPDT started on 22/08/13 at 17:25:44 in subsystem
Job 066182/QSECOFR/QEZDKSPDT ended on 22/08/13 at 17:39:47; [illegible].634 seconds
```

Small block size, typical of random DB activity. Around 120 I/Os per second per disk unit, mostly READS. The elapsed time of this run was around 15 min.

…TEST3 (GO DISKTASKS) with Flash as primary

```
DISKTASKS                                  Disk Space Tasks

To select one of the following, type its number

     1. Collect disk space information
```

```
Elapsed time:   00:00:06

                 Size     %      I/O    Request    Read   Write    Read   Write     %
Unit  Type        (M)   Used     Rqs  Size (K)     Rqs    Rqs     (K)    (K)  Busy
   1  6B22      66810   23.1   536.1       6.2   377.6  158.5     7.0    4.5    33
   2  6B22      66810   23.1   769.1       6.7   567.5  201.6     7.4    4.5    42
   3  6B22      66810   23.1   828.8       6.6   609.8  218.9     7.3    4.5    42
   4  6B22      66810   23.1   810.5       6.4   588.5  222.0     7.2    4.5    46
   5  6B22      66810   23.1   739.1       6.5   560.1  179.0     7.1    4.5    41
   6  6B22      66810   23.1   752.7       6.5   574.9  177.8     7.2    4.5    42
   7  6B22      66810   23.1   995.0       6.0   585.9  409.1     7.1    4.5    48
   8  6B22      66810   23.1   738.7       6.5   552.2  186.4     7.2    4.5    43
   9  6B22      66810   23.1   757.2       6.6   542.4  214.7     7.5    4.5    42
  10  6B22      66810   23.1   831.1       6.5   595.3  235.7     7.3    4.5    42
```

```
Job 065523/QSECOFR/QEZDKSPDT started on 22/08/13 at 16:34:52 in subsystem QSY
Job 065523/QSECOFR/QEZDKSPDT ended on 22/08/13 at 16:37:17; 5,562 seconds use
```

It looks already promising! Now around 800 I/Os per second per disk unit. Elapsed time went down to 2 min 25 sec.

TEST4 (IFS to SAVF) – HDD as primary

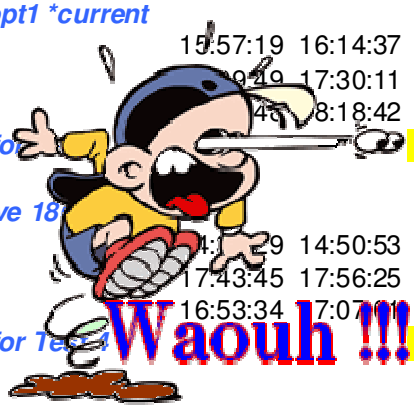
Elapsed time: 00:00:10

Unit	Type	Size (M)	% Used	I/O Rqs	Request Size (K)	Read Rqs	Write Rqs	Read (K)	Write (K)	% Busy
1	6B22	66810	25.4	171.1	48.8	100.5	70.6	39.7	61.6	63
2	6B22	66810	25.4	186.4	46.8	98.6	87.8	43.9	49.9	83
3	6B22	66810	25.4	178.9	49.1	100.6	78.3	42.5	57.6	75
4	6B22	66810	25.4	163.1	51.8	93.5	69.5	41.4	65.7	65
5	6B22	66810	25.4	181.2	48.6	99.5	81.6	44.8	53.4	78
6	6B22	66810	25.4	172.4	48.8	96.8	75.5	40.1	59.9	74
7	6B22	66810	25.4	169.9	48.9	98.0	71.8	39.2	62.1	68
8	6B22	66810	25.4	177.0	47.3	96.4	80.5	40.6	55.4	72
9	6B22	66810	25.4	176.7	49.0	97.8	78.8	42.1	57.4	71
10	6B22	66810	25.4	172.5	48.7	97.7	74.8	41.3	58.3	70

As we can observe, a save operation has a relatively large block size. This is what is called streaming. A good mix or READS and WRITES, which sounds logical since we are reading from and saving to disks.

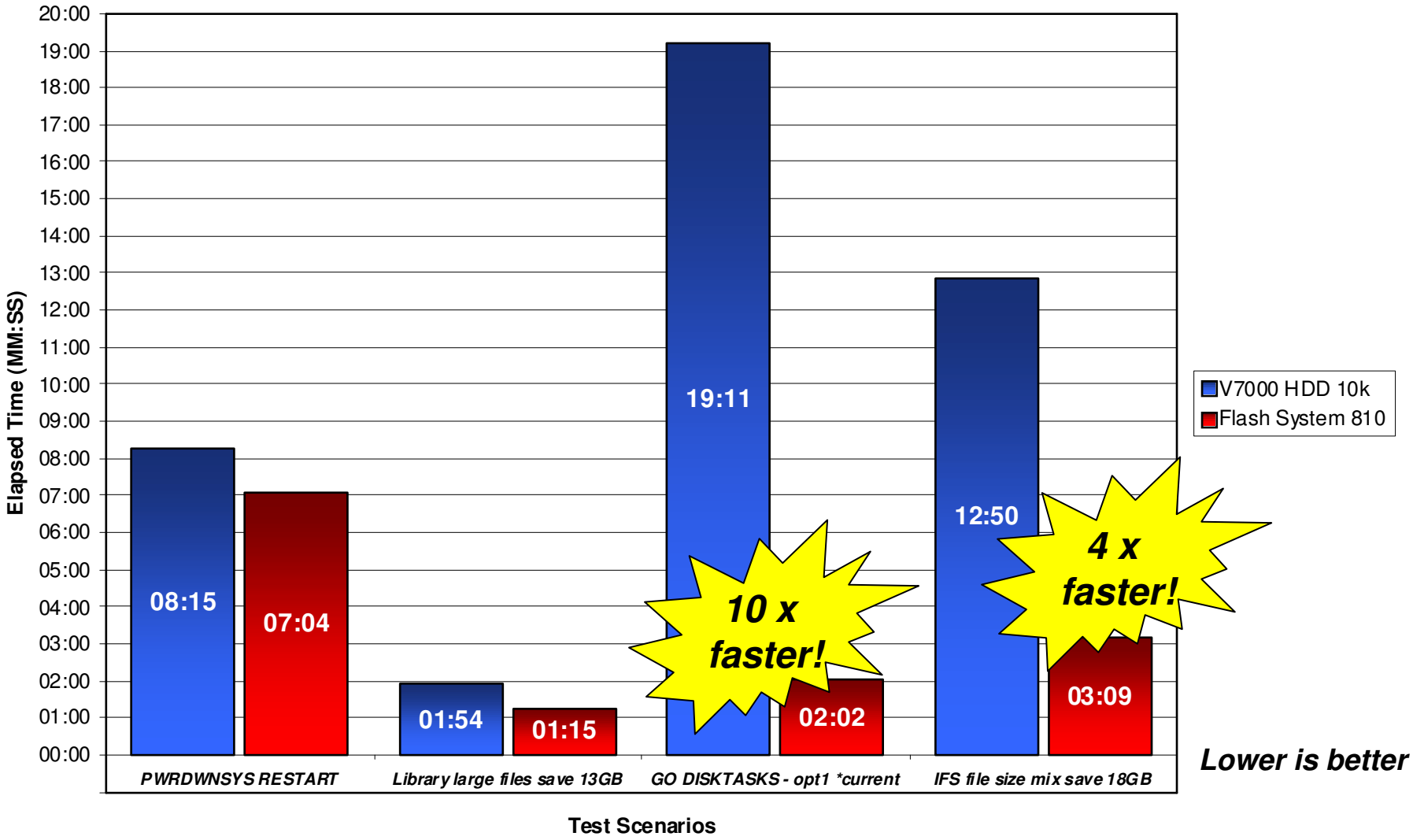
Results – V7000 HDD vs FlashSystem

Test #	Test description	IBM Storwize V7000 HDDs			IBM FlashSystem810			Improvement
Test 1	PWRDWNSYS RESTART	Start	End	Elapsed	Start	End	Elapsed	
	Test 1 Run 1	16:37:28	16:46:04	00:08:36	21:01:55	21:09:11	00:07:16	
	Test 1 Run 2	17:31:29	17:39:38	00:08:09	19:11:23	19:18:34	00:07:11	
	Test 1 Run 3	18:20:59	18:29:00	00:08:01	20:21:39	20:28:23	00:06:44	
	Average of 3 runs for Test 1			00:08:15	Average		00:07:04	117%
Test 2	Library large files save 13GB							
	Test 2 Run 1	15:44:21	15:46:17	00:01:56	11:56:29	11:57:49	00:01:20	
	Test 2 Run 2	16:22:38	16:24:36	00:01:58	18:58:33	18:59:47	00:01:14	
	Test 2 Run 3	16:49:52	16:51:41	00:01:49	20:45:57	20:47:09	00:01:12	
	Average of 3 runs for Test 2			00:01:54	Average		00:01:15	152%
Test 3	GO DISKTASKS - opt1 *current							
	Test 3 Run 1	15:57:19	16:14:37	00:17:18	11:59:03	12:00:53	00:01:50	
	Test 3 Run 2	[illegible]	17:30:11	00:20:22	20:56:36	20:58:53	00:02:17	
	Test 3 Run 3	[illegible]	18:18:42	00:19:54	20:16:09	20:18:07	00:01:58	
	Average of 3 runs for			00:19:11	Average		00:02:02	946%
Test 4	IFS file size mix save 18							
	Test 4 Run 1	[illegible]	14:50:53	00:12:24	20:48:40	20:51:48	00:03:08	
	Test 4 Run 2	17:43:45	17:56:25	00:12:40	19:01:12	19:04:20	00:03:08	
	Test 4 Run 3	16:53:34	[illegible]	00:13:27	19:23:41	19:26:51	00:03:10	
	Average of 3 runs for Test 4			00:12:50	Average		00:03:09	408%



I compared the results obtained with the V7000 VDisk mirror set to use HDD volumes as primary copy versus FlashSystem volumes defined as primary (preferred) copy. This in order to redirect all READS operation to FlashSystem.
I performed 3 runs for all 4 tests and I took the average value.

Chart



Additional scenarios tested

- In order to compare the impact, I also performed several runs with the following IBM Storwize V7000 features :
  - Real Time Compression (RTC) active on the V7000 HDD VDisk mirror volumes.
  - Disabling the V7000 cache (IBM FlashSystem memory is supposed to have a lower latency than cache memory).

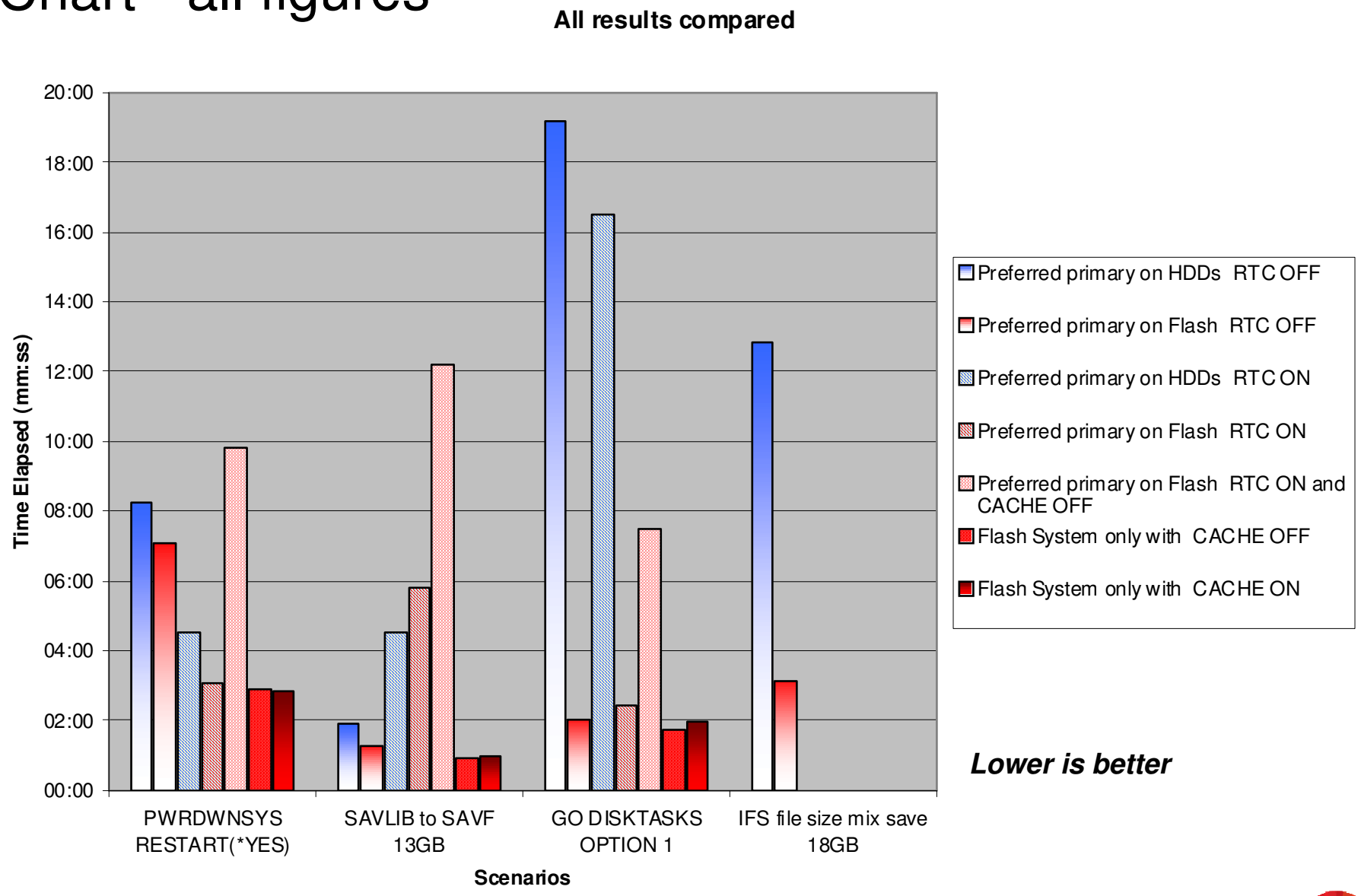
Comparing the different results

	Preferred primary on HDDs RTC OFF	Preferred primary on Flash RTC OFF	Preferred primary on HDDs RTC ON	Preferred primary on Flash RTC ON	Preferred primary on Flash RTC ON and CACHE OFF	Flash System only with CACHE OFF	Flash System only with CACHE ON
Test 1 *PWRDWNSYS RESTART(*YES)*	8 m 15 s	7 m 4 s	4 m 33 s	3 m 4 s	9 m 49 s	2 m 54 s	2 m 50 s
Test 2 *SAVLIB to SAVF 13GB*	1 m 54 s	1 m 15 s	4 m 33 s	5 m 47 s	12 m 11 s	54 s	58 s
Test 3 *GO DISKTASKS OPTION 1*	19 m 11 s	2 m 2 s	16 m 31 s	2 m 25 s	7 m 31 s	1 m 43 s	1 m 59 s
Test 4 *IFS file size mix save 18GB*	12 m 50 s	3 m 9 s	*I had no time left to complete this test ☹*				

VDisk mirroring with preferred copy on HDD vs preferred copy on IBM FlashSystem

VDisk mirror with Flash as primary is close to "Flash only" performance

Chart - all figures



Lower is better

Conclusions

- As expected, IBM FlashSystem is always boosting **performance** :
  - Modest improvement for a complete shutdown – restart cycle (**1,17x**)
  - Good improvement for save operations (**1,5x** in streaming), especially with IFS large number of small files (**4x**)
  - Impressive improvement for analytics and queries like workload (up to **10x**)
  - The save tests were certainly slowed down by the synchronous writes to HDD
- With **Real Time Compression** turned on :
  - The addition of a VDisk mirror on IBM FlashSystems is improving DB performance compared to HDD only
  - But please, notice that the activation of RTC is in general negatively affecting the performance compared to the run without RTC (except for the shutdown-restart of the system where RTC seems to have a positive impact)
- With the IBM Storwize **V7000 cache de-activated** :
  - The runtime of the FlashSystem was also slightly better, this confirms the very low latency of Flash memory compared to cache memory
- The **main conclusion** to draw here is :
  - The use IBM FlashSystems as VDisk mirrors of existing HDD LUNs performs really close to a “dedicated IBM FlashSystems” performance, benefiting from the best of both worlds : the very low latency of IBM FlashSystems and the functionalities and flexibility of the IBM Storwize V7000

Remarks

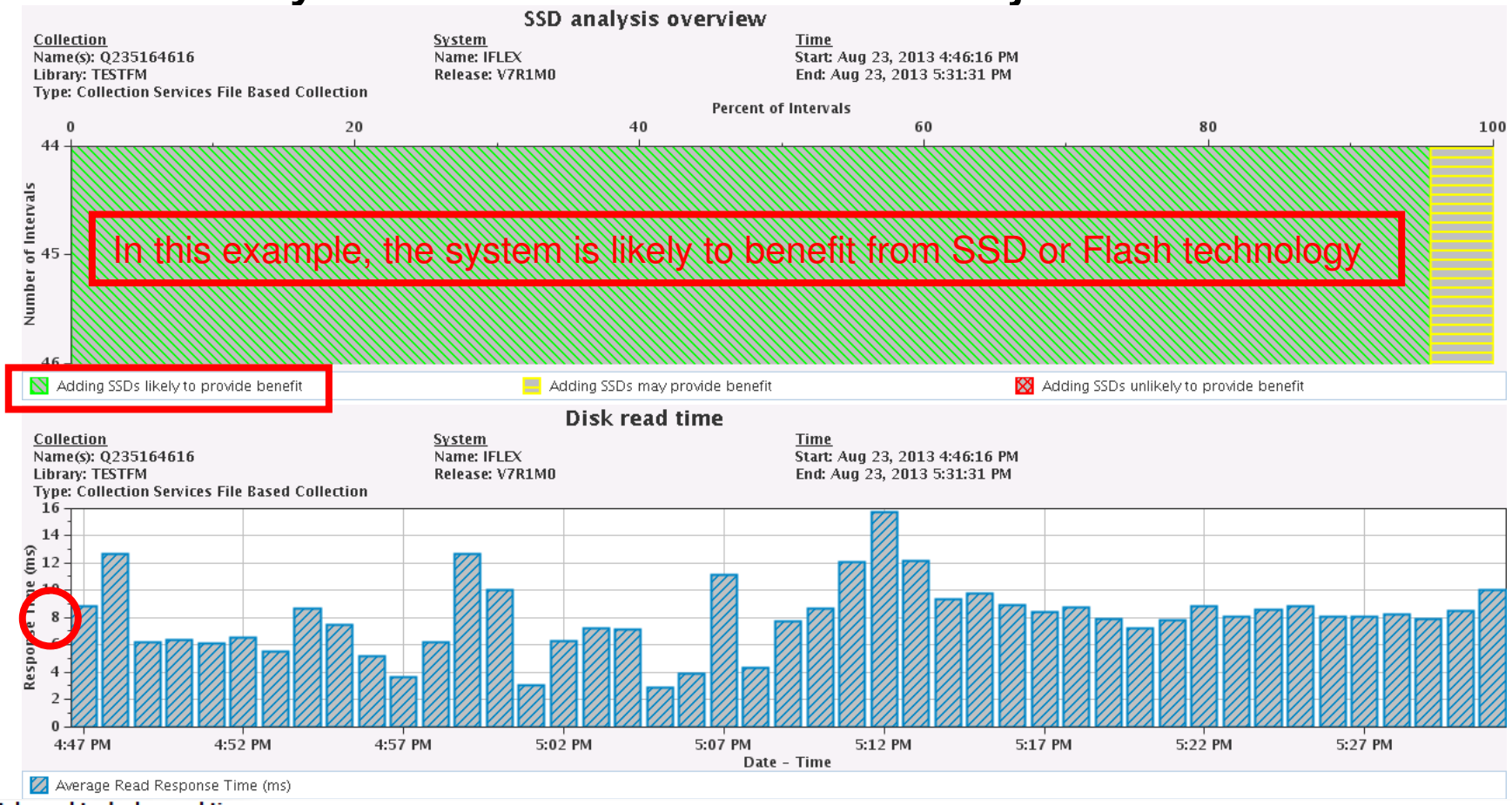
- In the tests I ran, a “dedicated IBM FlashSystems” means an IBM FlashSytem virtualized by an IBM Storwize V7000.
- According to the interoperability matrix, native direct attached IBM FlashSystems (without an IBM Storwize V7000 or SAN Volume Controller) is not supported by IBM i at the time being. As such, I could not test it and could not demonstrate if IBM FlashSystems would possibly perform better than in a virtualized configuration.
- I used only 1TB out of the available 6TB Flash memory on the IBM FlashSystem 810. Our Advanced Technical Support specialists told me that by using more than 50% of the usable IBM FlashSystems capacity, the write performance could possibly deteriorate as it would leave less free cells available for garbage collection.

Host performance data views using the SSD Analyzer Tool

http://www-03.ibm.com/support/techdocs/atsmastr.nsf/WebIndex/PRS3780

The IBM i SSD Analyzer Tool is designed to help you determine if SSDs could help improve performance on your system(s). The tool works with the performance data that is collected on your system by collection services.

SSD analysis tool – QEZDKSPDT job on HDD first

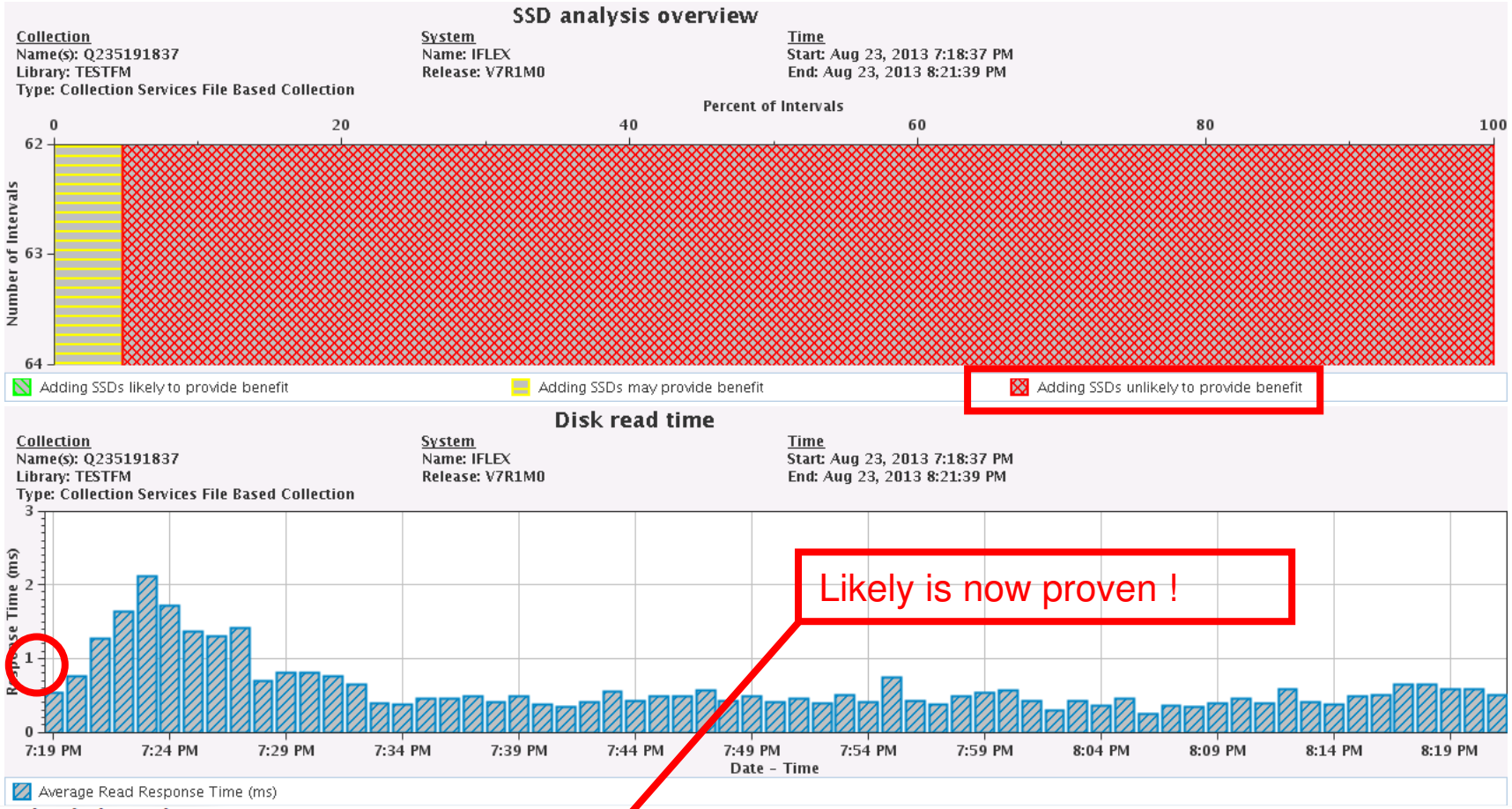


Jobs and tasks by read time

--- Select Action ---

Select	Job name/user /number - thread id	Disk read time (Seconds)	Disk non-read time (Seconds)	CPU time (Seconds)	Read time per operation (Average seconds)	Disk read time / CPU (Average)	Idle time (Seconds)
☐	QPADEV0002/QSECOFR /069539 - 00000001	800.56	5.68	21.16	0	38	1206.1
☑	QEZDKSPDT/QSECOFR	490.49	0.43	5.52	0.01	89	0

SSD analysis tool (2) – then on FlashSystem



Jobs and tasks by read time

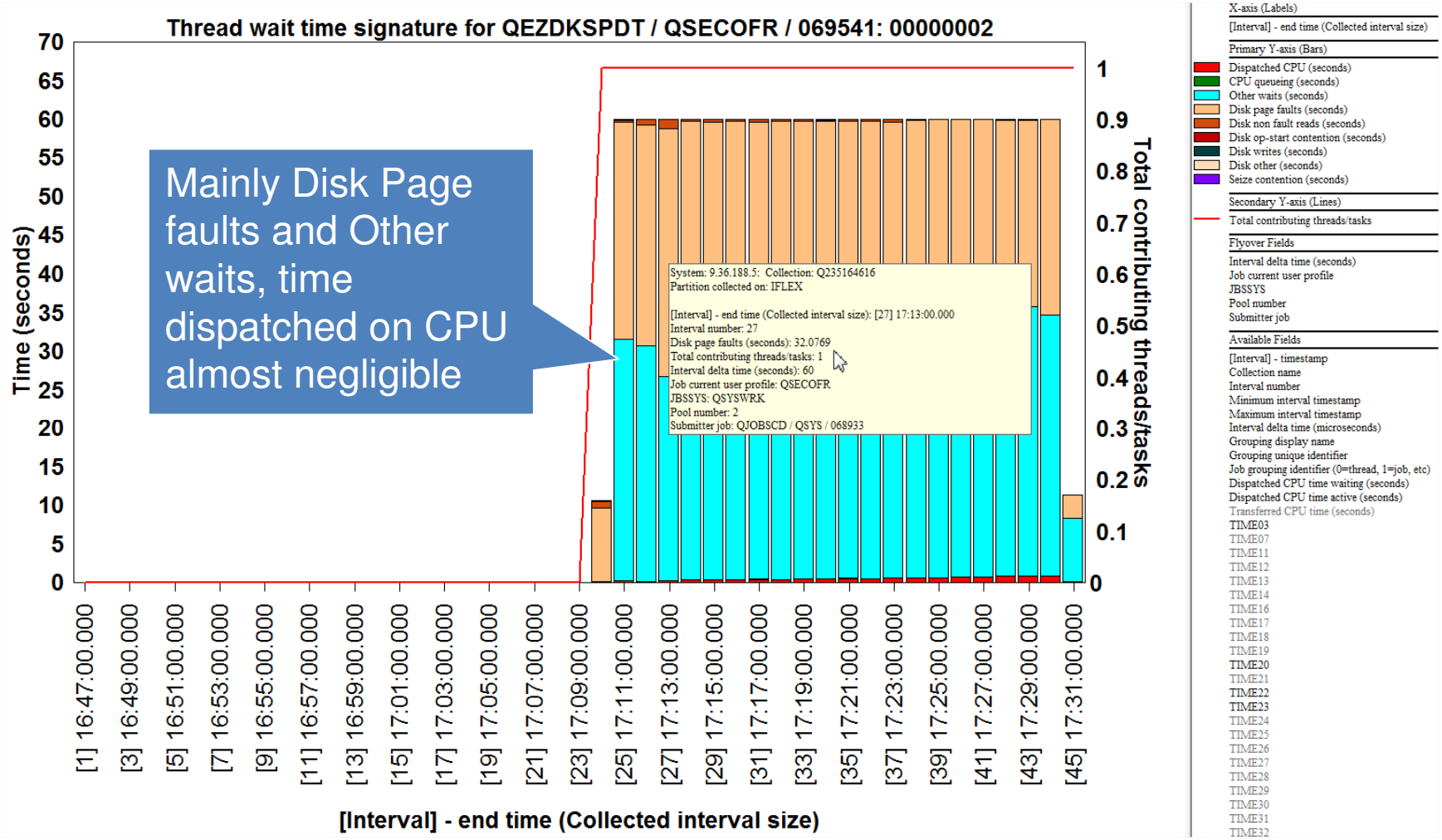
--- Select Action ---

Select	Job name/user /number - thread id	Disk read time (Seconds)	Disk non-read time (Seconds)	CPU time (Seconds)	Read time per operation (Average seconds)	Disk read time / CPU (Average)	Idle time (Seconds)
☐	QPADEV0002/QSECOFR /072018 - 00000003	130.76	9.38	21.2	0	6	502.42
☐	QEZDKSPDT/QSECOFR /072178 - 00000004	42.07	0.53	5.62	0	7	0

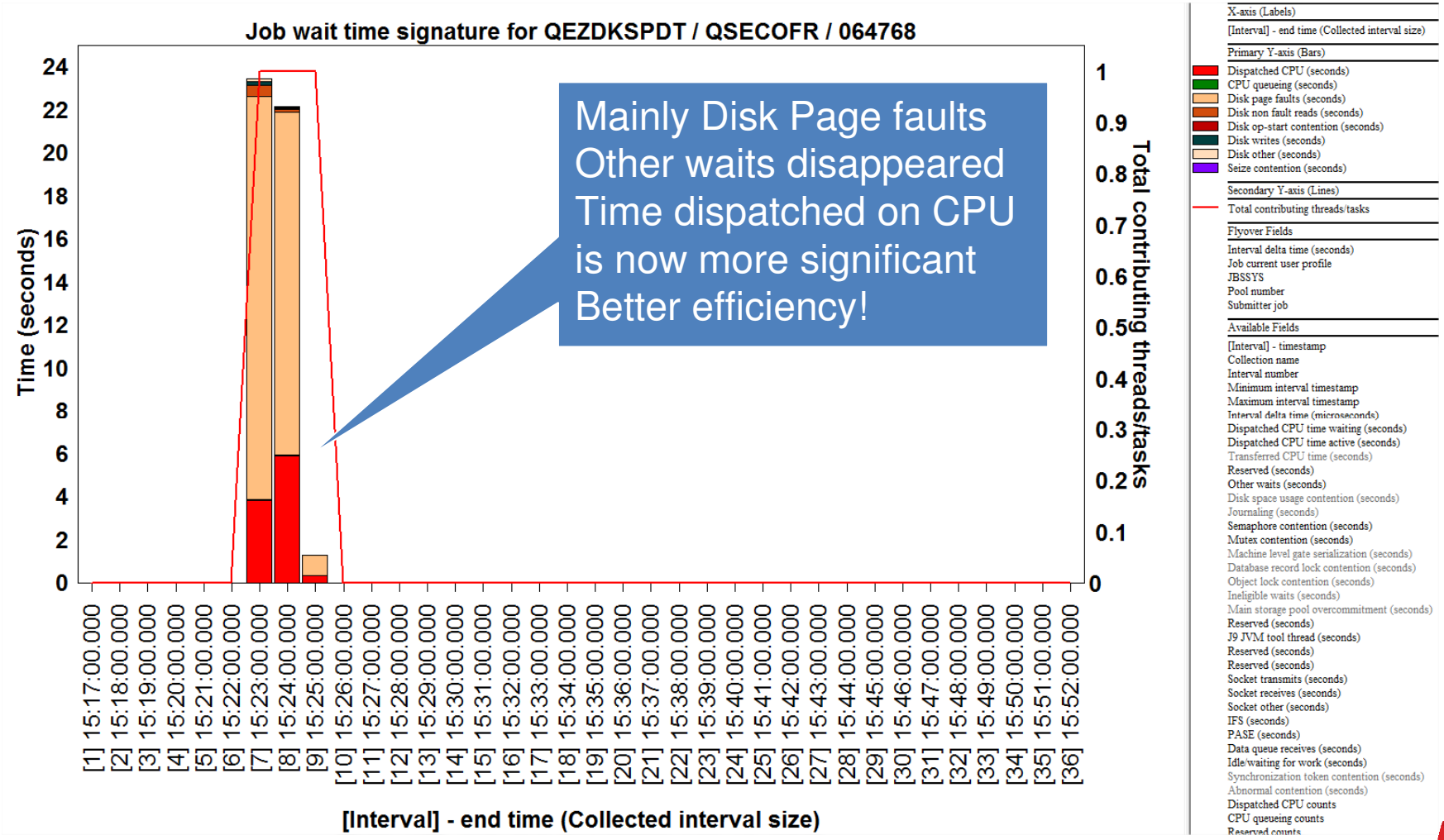
Host performance data views using iDoctor
https://www-912.ibm.com/i_dir/idoctor.nsf

IBM iDoctor for IBM i is a suite of performance tools that can be used by the performance expert or novice to collect, investigate and analyze performance data on System i. The tools are used to monitor overall system health at a high "overview" level or to drill down to the performance details within job(s), disk unit(s) and/or programs over data collected during performance situations.

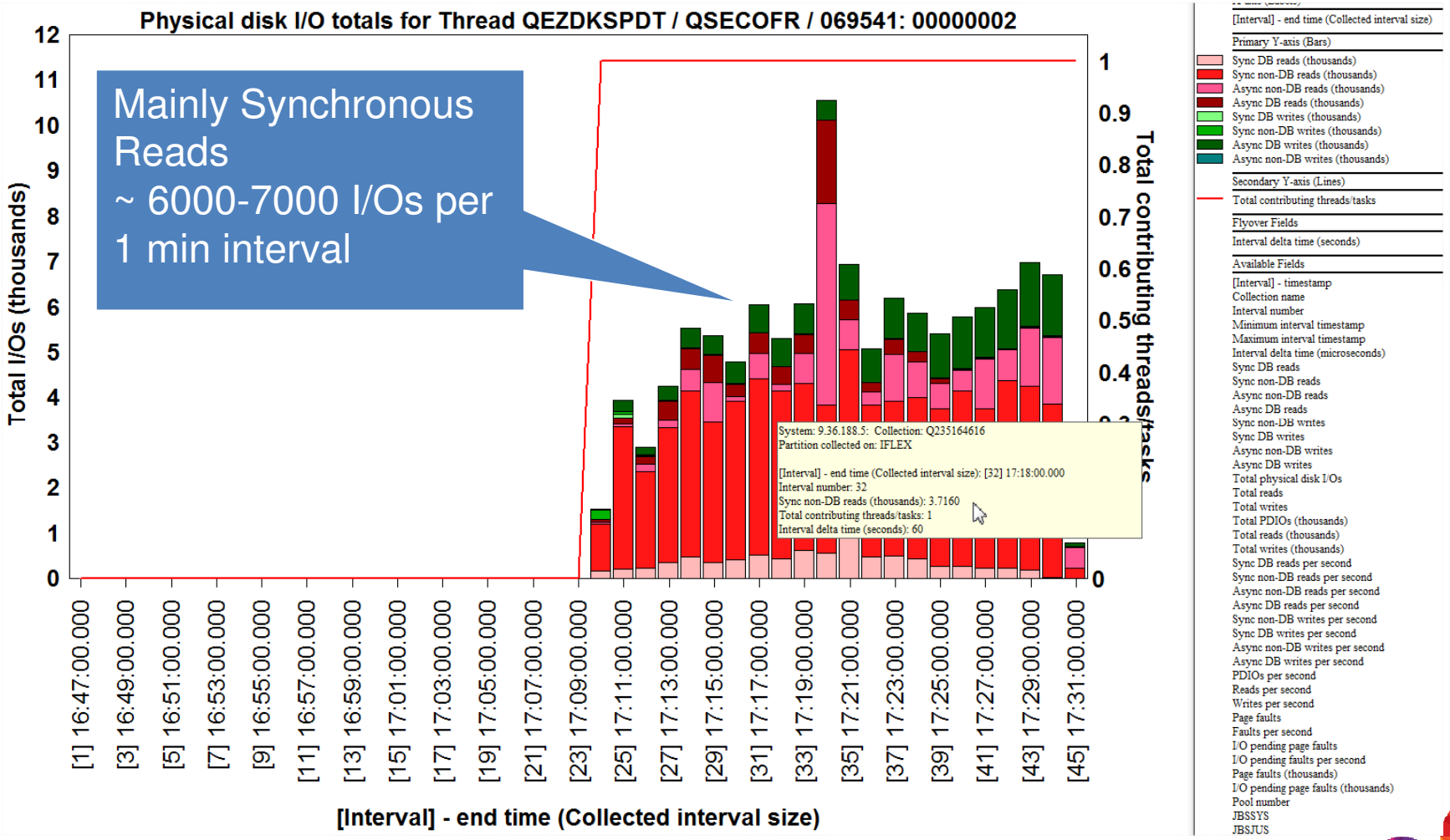
QEZDKSPDT job on HDD…



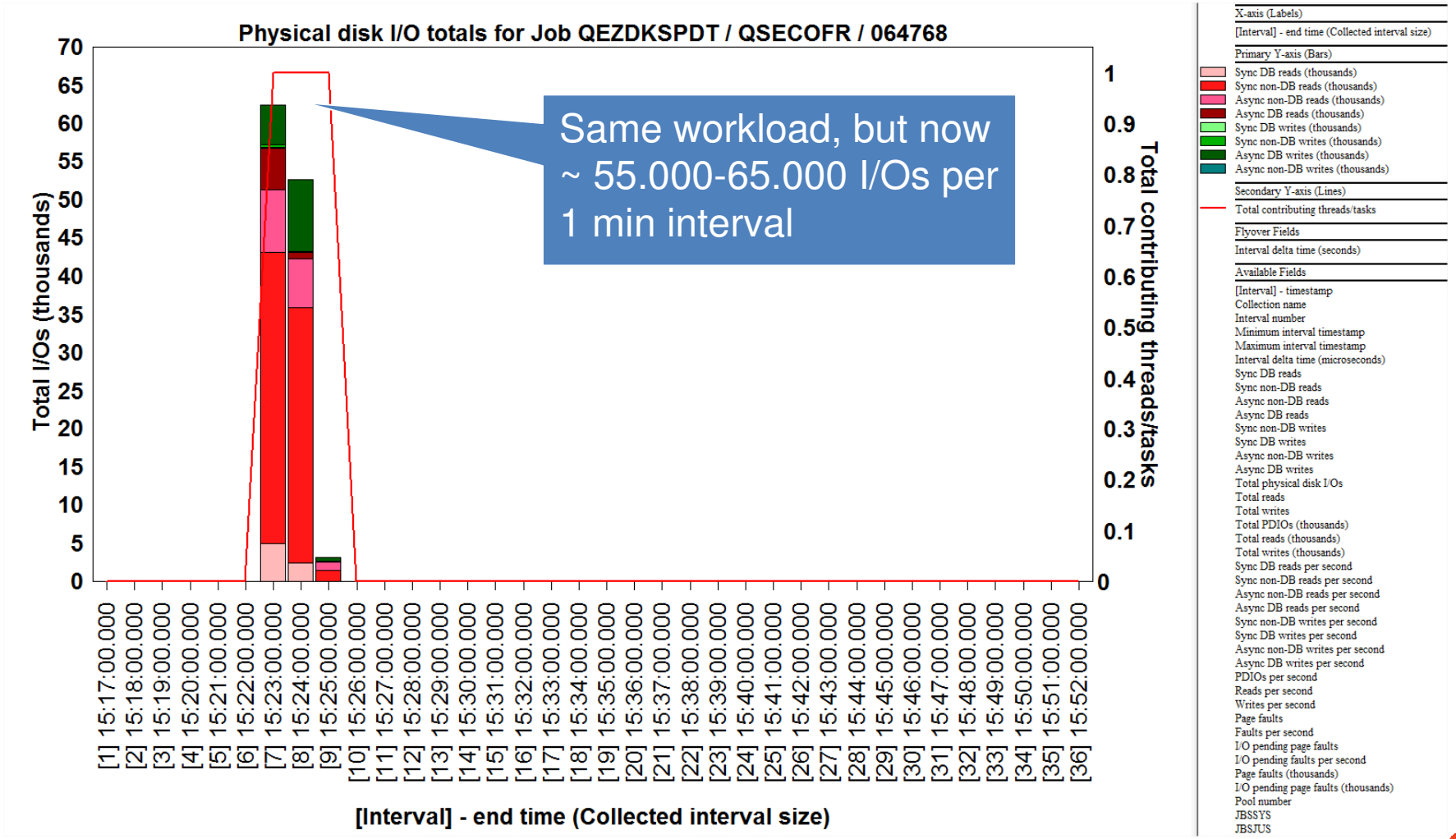
…and QEZDKSPDT job on IBM FlashSystem



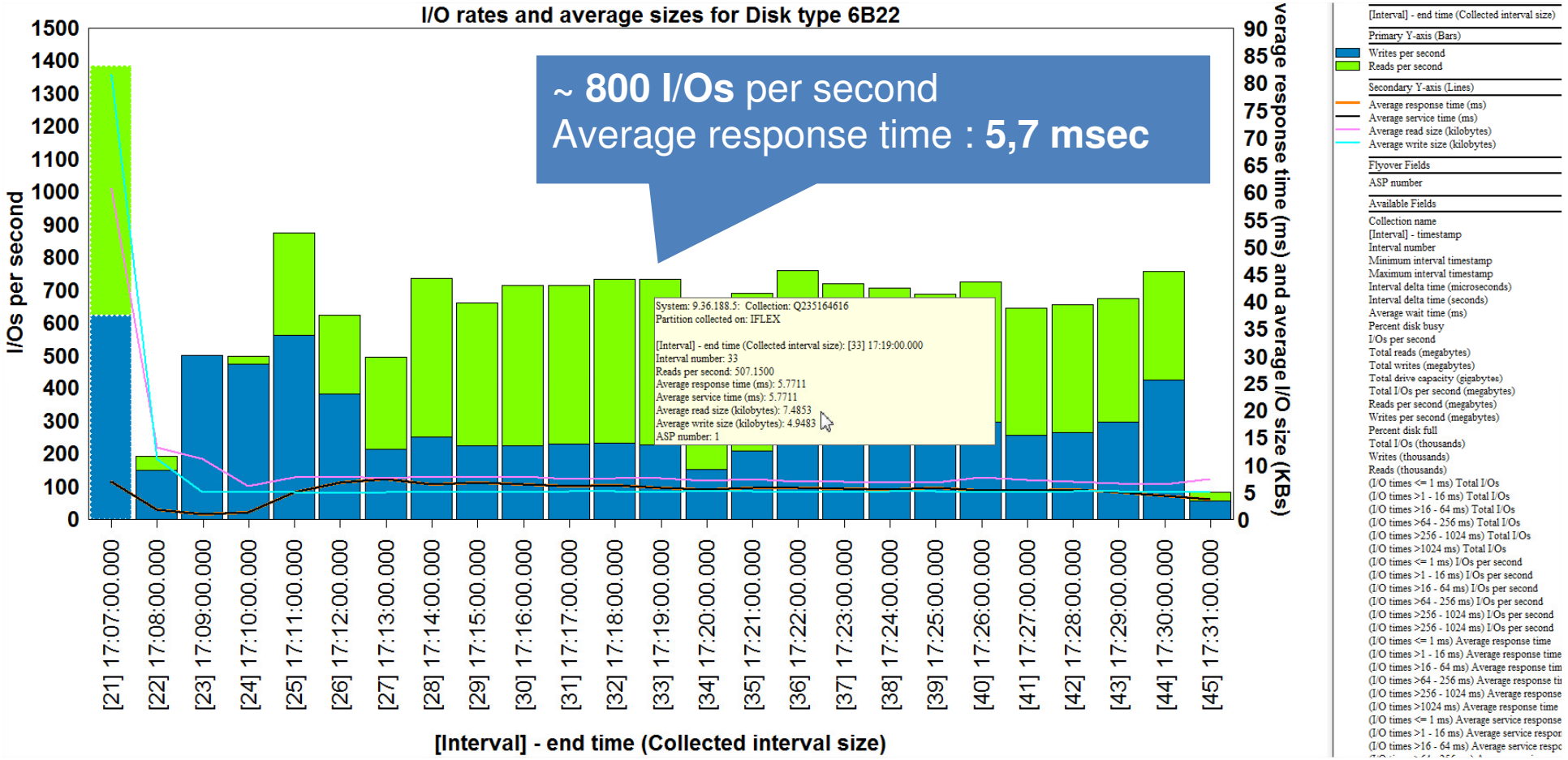
QEZDKSPDT job on HDD…



…and QEZDKSPDT job on IBM FlashSystem

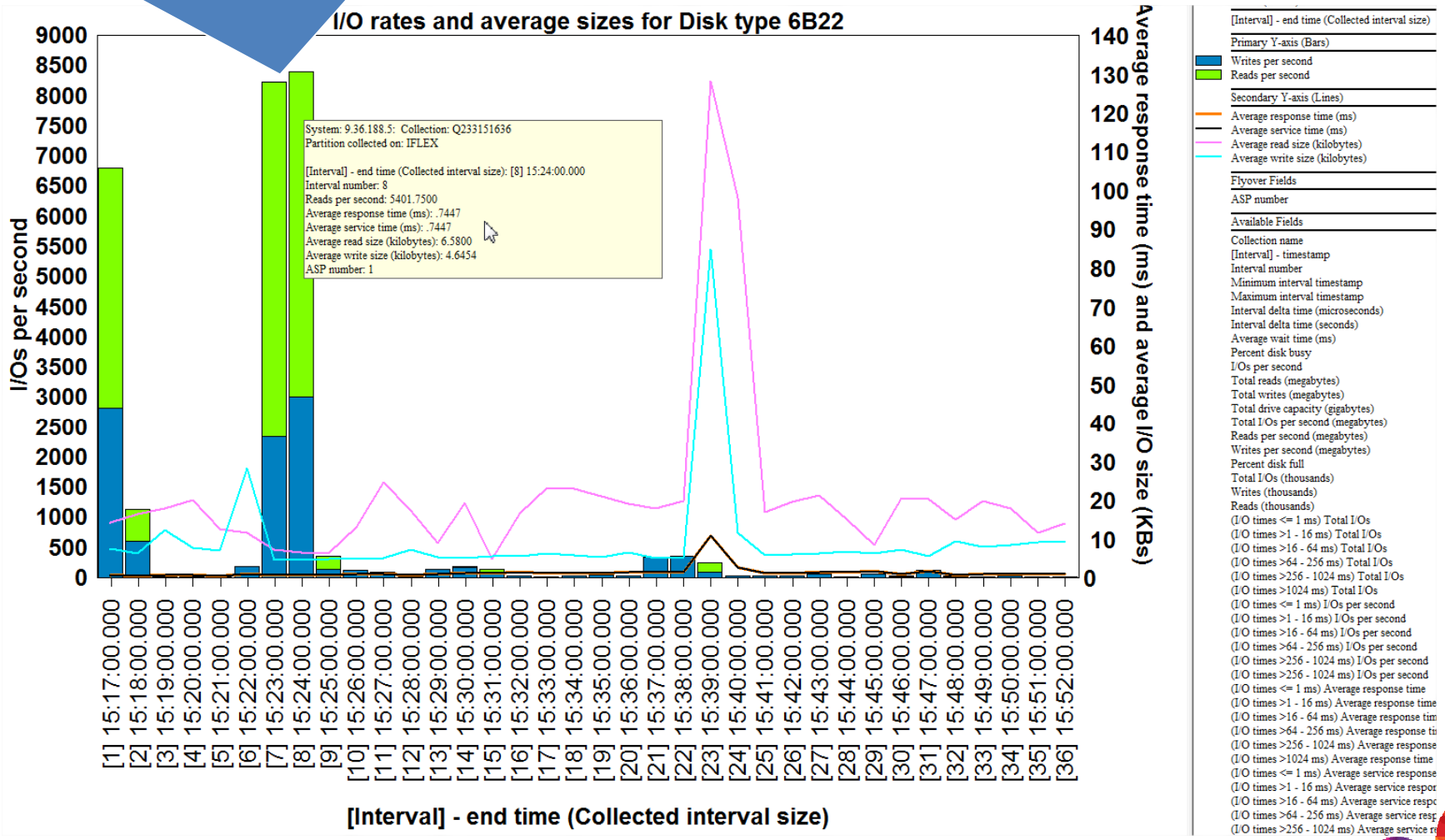


QEZDKSPDT job on HDD…



…and QEZDKSPDT job on IBM FlashSystem

~ **8500 I/Os** per second
Average response time : **0,7 msec (or 700 µsec)**



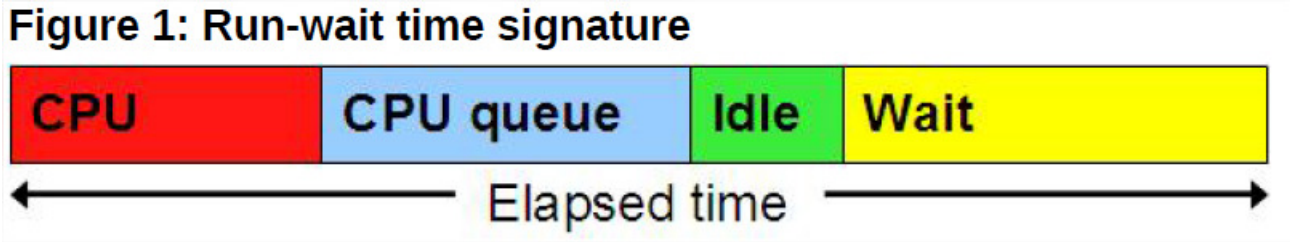
Recommended Reading

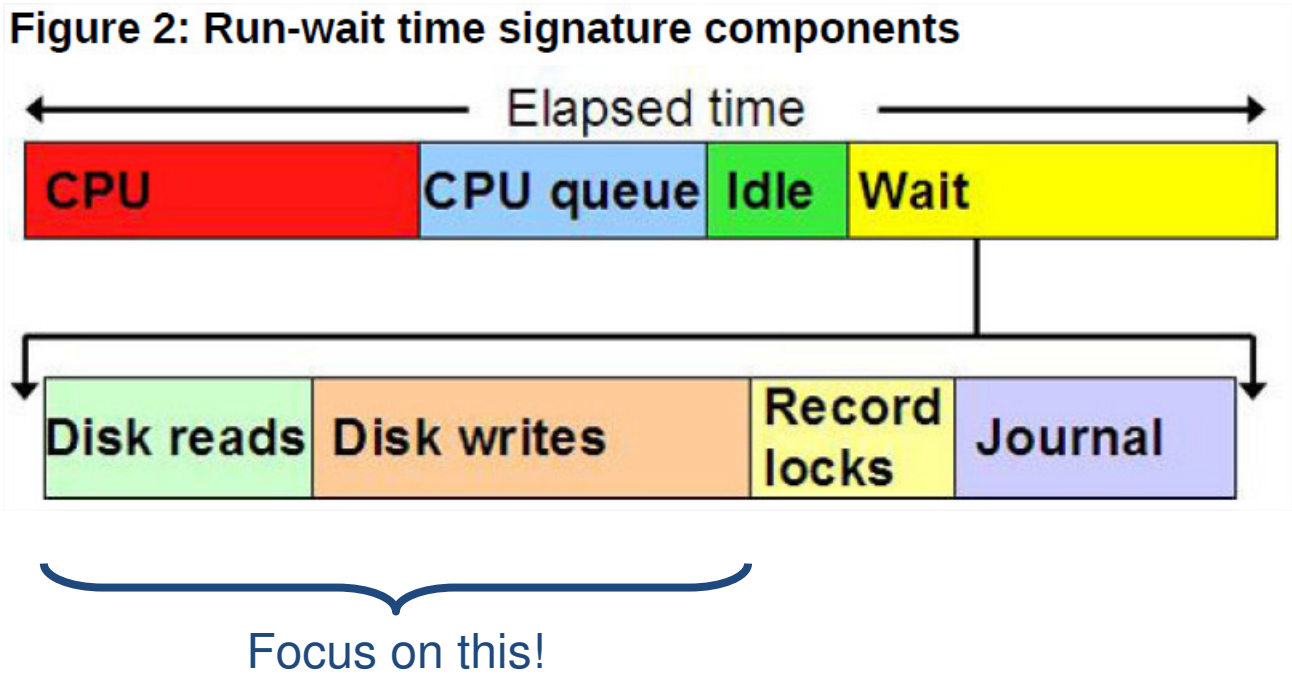
IBM i wait accounting, Thread-level run-wait analysis by **Dawn May**

http://www.ibm.com/developerworks/ibmi/library/i-ibmi-wait-accounting/

IBM i wait accounting is a technology built into the operating system that can identify what every thread or task is doing on the system when it is not using the processor. Wait accounting is a very powerful tool for performance analysis and problem determination. This article describes wait accounting and explains how you can use it to troubleshoot performance problems or to improve the performance of your applications.

Wait time signature analysis





Wait time signature analysis (2)

- **Disk page faults** are those wait points that are related to faulting; page faulting on IBM i is often normal and expected. Page faults occur when data from a disk must be brought into the main storage but it was not explicitly read in. The disk page faults wait bucket can help you identify whether your faulting might be excessive and what is causing it, which can then lead you to assess if you need to take action to reduce the faulting rate. For example, you might need to adjust memory pool sizes to keep frequently used data in memory. There are a variety of techniques that can be used to optimize bringing data from disk into memory, which are beyond the scope of this article.
- **Disk non fault** are explicit synchronous reads that are performed to bring data into the main store from the disk. When the synchronous read operation is performed, the application will wait for the read operation to complete. The *disk non fault reads* wait bucket can show you how much time your application has spent on reading data from the disk and can help you assess whether that time is significant enough to consider application changes, such as asynchronous reads.
- **Disk space usage contention** can occur when an object is created or extended, free disk space has to be located to meet the request and there is some level of serialization that is associated with that. Typically, you should see very little of this type of wait. If it is present in significant percentages, it usually means that your application is performing a high rate of object create/extend/truncate/delete operations. (Note that opening a DB2 file causes a create activity). The size of the disk space requests is not important; it is the rate of the requests that is important.
- **Disk op-start contention** can occur when a disk operation start is delayed due to a very high rate of concurrent disk operations in progress at the moment it is requested. If you see this wait type, you might need to look at the overall disk operations occurring to see if there are significant disk I/O inefficiencies that should be eliminated.
- **Disk writes** are explicit synchronous writes that are performed to store data from the mainstore to the disk. When the synchronous write operation is performed, the application waits for the write operation to complete. The *disk writes* wait bucket can show you how much time your application spends on writing data to the disk and can help you assess whether that time is significant enough to consider application changes, such as asynchronous writes. However, this wait bucket also includes waits that are related to waiting for asynchronous disk writes to complete.
- **Disk other** *wait* is the catch-all grouping for all the other reasons that the system may wait for disk operations.


IBM
Fabian Michel
Client Technical Architect
Senior IT Specialist
IBM Certified
IBM Belgium
Avenue du Bourget, 42
B-1130 Bruxelles
Tel. +32 2 339 38 22
fabian_michel@be.ibm.com