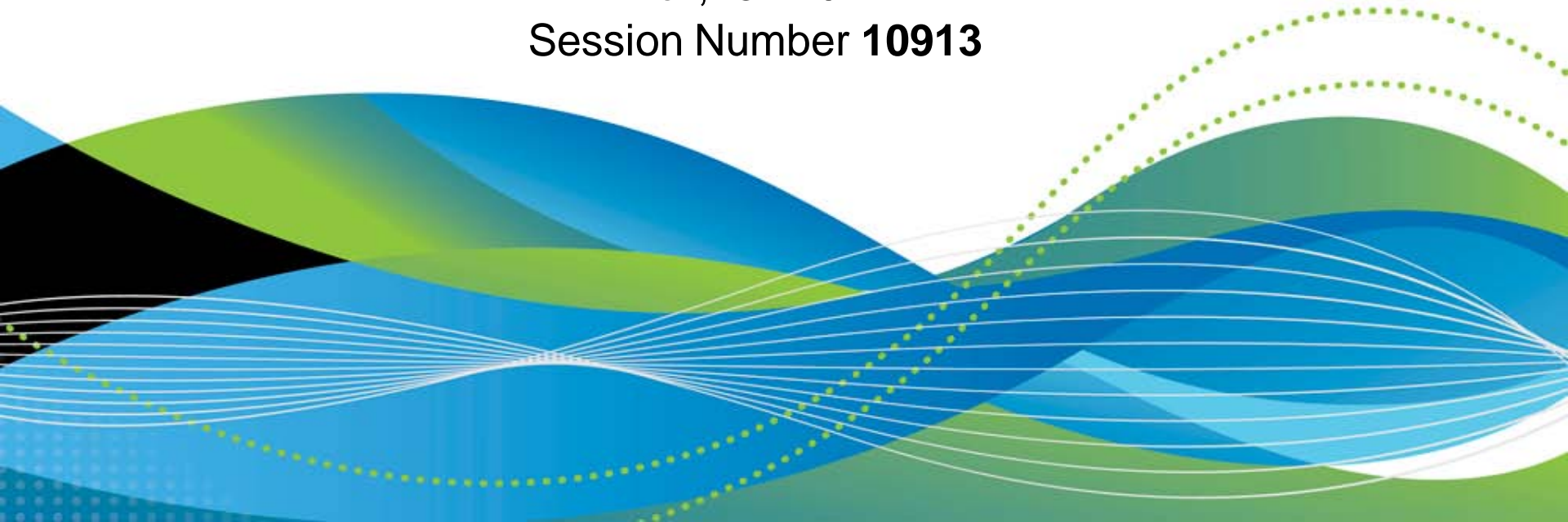


Mainframe Solid State Disk User Experience and Tools to Store and Analyze RMF Data

Sridher Manivel
Lowe's

Mar, 13th 2012
Session Number **10913**



Agenda

- SSD architecture
- SSD's benefits and its applicability to mainframe storage
- PoC
- Tiering
- Implementation
- Choosing right data
- Tools
- Key things to remember

SSD Architecture

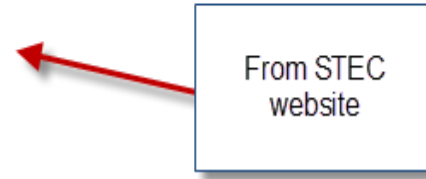
- SLC
 - Single Level Cell, single bit per cell – more write cycles
- MLC
 - Multi Level Cell, multiple bits per cell
- Over provisioning and Wear leveling
 - More Storage than what is quoted (or what is available for config). Rewrites routed to new blocks
- Questions to Vendor
 - Ask what type it is, ask how disk replacements are made (no mechanical failures). SLC is better than MLC, costlier than MLC. MLC is still evolving with different types of implementation

SSD Architecture

IOPS & Throughput

Prominent product specifications include:

- Up to 52,000 Sustained Random Read IOPS
- Up to 17,000 Sustained Random Write IOPS
- Up to 250MB/sec sustained, sequential reads; 200MB/sec sustained, sequential writes
- Interfaces: Fibre Channel, SAS and SATA
- Form Factor: 3.5-inch standard HDD dimensions
- Weight of less than 0.4 kg



Spinning disk

- 200 IOPS
- 100-200 MB/sec

SSD Architecture

IOPS & Throughput

Based on IO meter tests





Writing to internal disk (32k) 60 MB/sec

Reading from internal disk (32k) 75 MB/sec

USB Max IOPS with 4K 1700 with 0.6msec

Writing to USB Flash (32k) 16 MB/sec

Reading from USB Flash (32k) 26 MB/sec

Class	Speed
 Class 2	2 MB/s
 Class 4	4 MB/s
 Class 6	6 MB/s
 Class 10	10 MB/s

from
Wikipedia

Notice read to write throughput difference

Notice throughput difference between enterprise SSD and USB drive

SSD Benefits and Mainframe Storage

- Key distinction between HDD & SSD
 - Random access **response time** (no seek time)
 - **IOPS** per disk – cost per IO
 - Consistent response time
- VSAM datasets with random reads
- DB2 synchronous read I/Os
- Short stroking may be eliminated
- Drive size, IOPS per GB
 - Bigger drives, contention, sharing drives with critical workload, sequential and random workloads
- Move critical tables which match selection criteria to SSD to meet SLAs – more later
- Unlike distributed platform, centralized allocation policy

Where it helps and where it doesn't

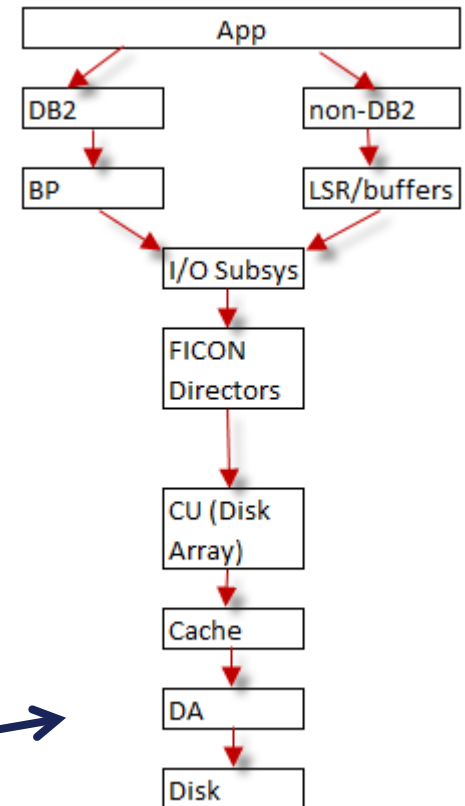
- *Random Read I/Os*
- *Datasets with low cache hit ratio and high I/O rate*
- *Datasets with high read percentage*
- *Conn, Disc, IOSQ, PEND*
- Very Large datasets or Very large partitioned table spaces
 - You can save buffer pool for other workloads
- May not get optimal result with
 - Sequential I/Os
 - Write intensive I/Os
 - Datasets with cache hit ratio relatively high (> 50%)
 - Not in lieu of dedicated buffer pool. This may result in degradation.

Where it helps and where it doesn't

I/O path of read miss (z/OS buffer) I/Os

Disk read time	I/Os	Cache hit	Backend I/Os	Elapsed time for backend I/Os (cache miss I/Os)	Elapsed time in seconds
6	100000	90%	10000	60000	60
1	100000	90%	10000	10000	10

Disk read time	I/Os	Cache hit	Backend I/Os	Elapsed time for backend I/Os (cache miss I/Os)	Elapsed time in seconds
6	100000	30%	70000	420000	420
1	100000	30%	70000	70000	70



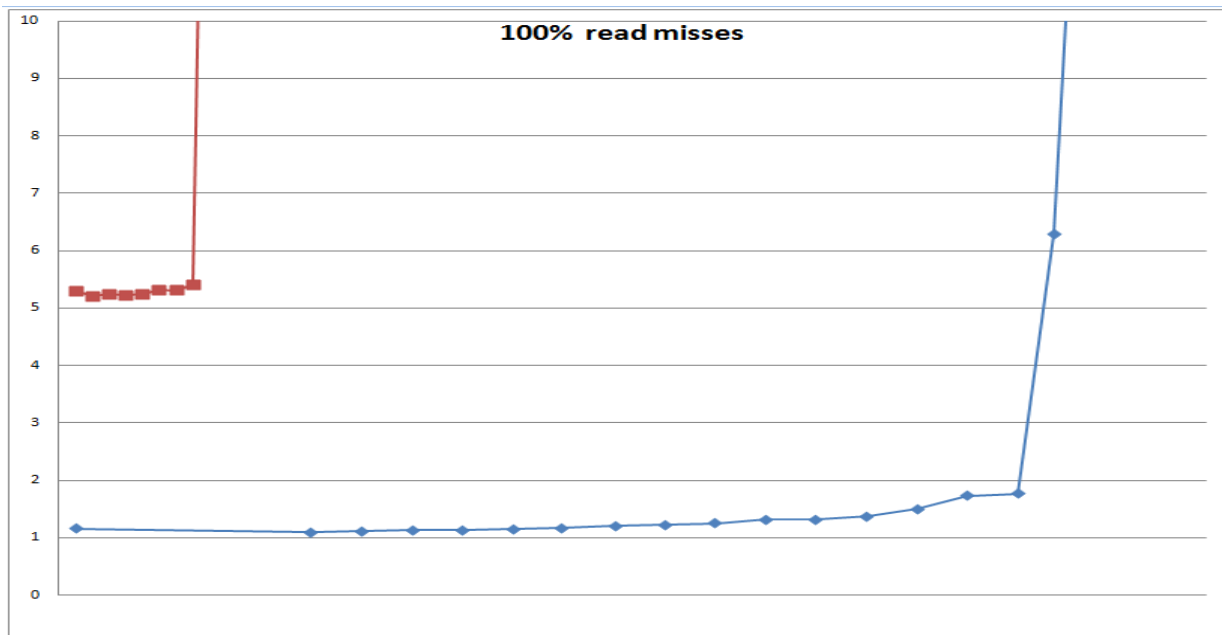
PoC

- How do you know the backend is supported by or mapped to SSD drives?
 - Get UCBs associated to SSD drives
 - DS QD,nnnn,ATTRIBUTE

```
DS QD, ██████,ATTRIBUTE
IEE459I 18.36.38 DEVSERV QDASD 594
UNIT VOLSER SCUTYPE DEVTYPE
ATTRIBUTE/FEATURE YES/NO
0 ██████ ██████ 2107921 2107900
SOLID STATE DRIVES Y
*** 1 DEVICE(S) MET THE SEL
*** 0 DEVICE(S) FAILED EXTEN
```

PoC

- Method 1:
 - With PA I/O driver tool[®], use H4 test. 100% read miss test
 - Expect approx 4000 IOPS per physical disk
 - Expect response time < 2 msec
 - It is important to find knee of the curve

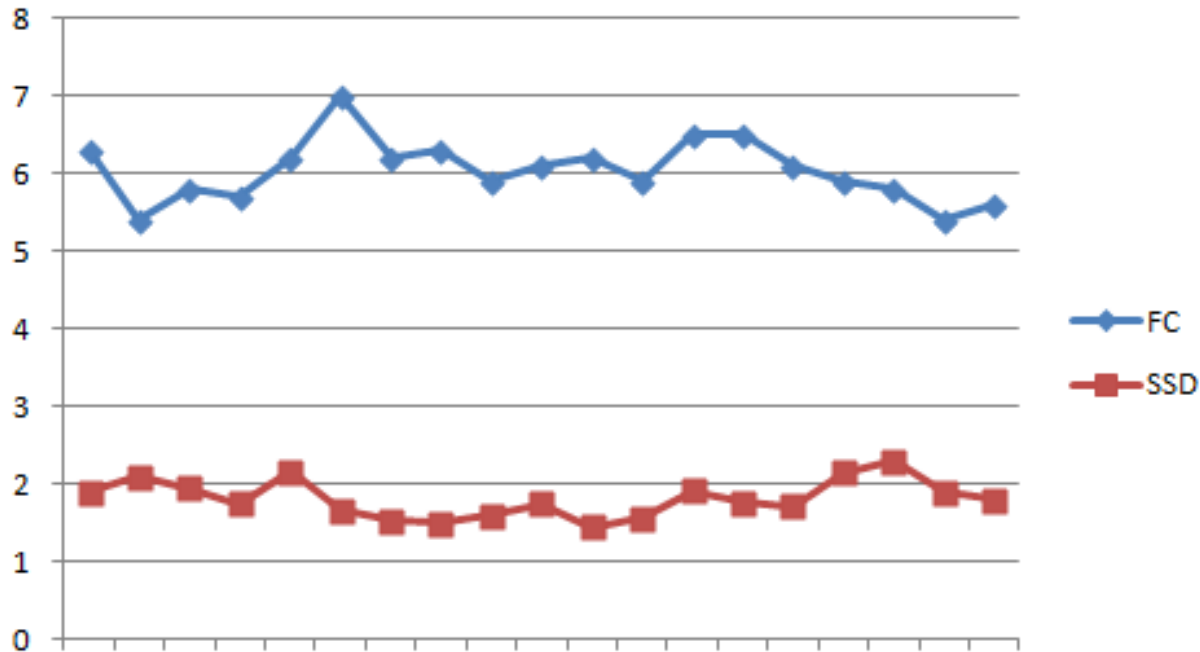


PoC (contd)

- Method 2:
 - Allocate a larger KSDS VSAM on one of the logical disk
 - Prepare a sequential data and populate KSDS with data
 - Scramble sequential data, will be used as input for SSDTEST
 - Use SSDTEST assembler load module
 - Run tests against spinning disk and SSD disk in parallel
 - Checks the time delta and reports for every 1000 I/Os
 - This calculates to 6 msec avg resp. time
 - For SSD, this should be < 2 msec
 - For Spinning disk, this should be ~ 6 msec
 - Notice IOPS difference between the tests

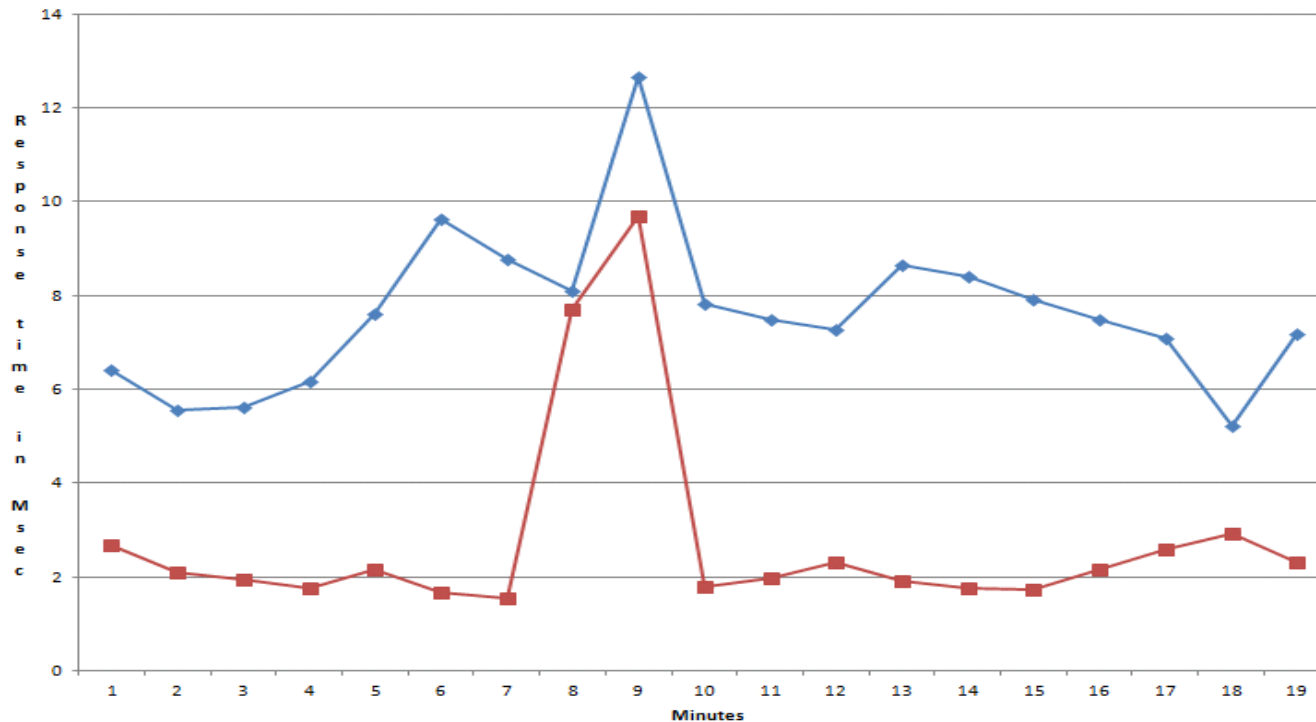
PoC (contd)

- Results with random I/Os – from Method 2



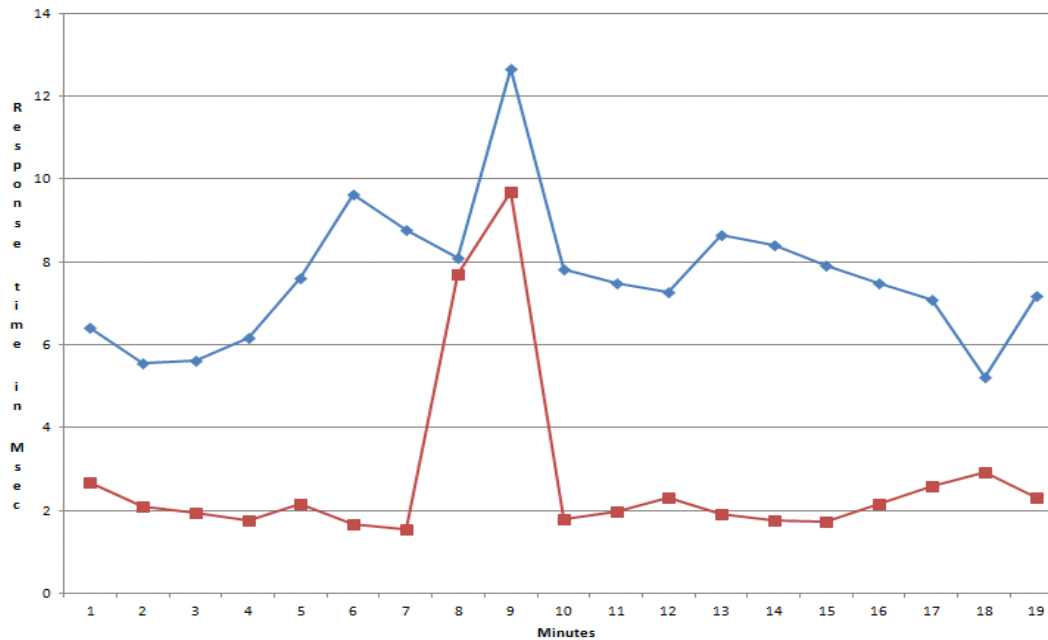
PoC (contd)

- What happened to the response time?



PoC (contd)

- Random and Sequential
 - Mixing Sequential workload will negate the purpose of moving critical workload to SSD drives
 - RMF 15 min invl may not show this



PoC (contd)

- Test I/Os with internal copies (such as FlashCopy, Time Finder Mirror, Clones, ShadowImage, etc)
- Test I/Os with any other workload in the array, for example, synchronous or asynchronous replication
- If possible, test with DB2 I/Os by allocating data on SSD drives

Facts



- Sequential performance of FC drives beat SSD drives – specifically with partitioned table spaces spanning across several spinning disks
- Don't expect 50000 IOPS!
- Minimum of 3x response time improvement & 20-25x IOPS improvement with random read I/Os

Tiering

- Array level
 - Implemented by all major vendors (still evolving, some don't support mainframe yet)
 - Regularly monitors the backend I/Os by proprietary algorithms and move chunks of data from one device type to SSD pool & vice versa
 - Ability to influence the window, ability to pin certain data in SSD
 - No host awareness of where data is
- Hybrid approach : PCI Flash for Open systems

Tiering

- EMC
 - Analyzes at 768K tracks level, moves 10 chunks of 12 track at a time
- IBM
 - Analyzes and moves 1GB of chunks at a time
- HDS
 - Analyzes and moves 42MB chunks
- No direct influence from z/OS level. Arrays monitor I/O activity and makes decision to move based upon the policy (customizable)
- DB2 Re-org, VSAM Re-org, non-VSAM reallocation will move data from one logical volume to another. How does the array keep track of this?

Tiering

- z/OS level
 - Allocate all SSD logical devices (backed by SSD drives) to SMS storage group.
 - Control allocations with SMS classes & ACS changes
 - Complete awareness of where data is – from z/OS host
 - Requires more effort to move data in and out of SSD pool
 - but z/OS RMF helps with this

Implementation

- Spread SSD drives across all backend DAs (Backend Device Adapters)
 - Very important, otherwise DAs will be a bottleneck
 - Besides serving read misses, staging and writing, they handle other I/Os too
- Leave few drive slots free for adding SSD capacity later
- Dedicated Storage Group with FILTLIST
- Direct allocations based on STORCLAS
- DB2 online reorgs to move data from FC pool to SSD pool
- VSAM Re-org to move data

Choosing candidates



- Talk to Application Developers
- Use Tools such as CA Insight[®] or equivalent to check DB2 sync I/Os and average response time
- FLASHDA
 - SAS based tool from IBM. TYPE 42 sub type 6 and TYPE 74 records as input and produces report to look at
 - <http://www-03.ibm.com/systems/z/os/zos/downloads/flashda.html>
- Talk to your vendor – they have custom tools that work with TYPE42 & TYPE74 data
- Better accuracy with smaller interval

TYPE 42

- MXG® SAS
 - If MXG is not available, use any SMF reporting program to focus on the fields described here or use FLASHDA
- Pick a window to analyze
 - Where SSD may help, such as heavy batch with high random activity to certain tables
- Start tracking TYPE42 for the selected objects
 - With the input from application developers
- Identify most affected datasets with random read activity
 - Approach developers with the data to check if response time improvement will be beneficial

TYPE 42

- Key Metrics from MXG® fields
 - IBM added new fields with **OA25559**, especially for SSD – separates read from write – S42DSRDD & S42DSRDT
- Find weighted average for the fields
 - AVGDISMS, S42DSRDD, AVGRSPMS, CHITPCT, DASDRATE, S42DSRDT, SEQIOS,
- Find Random Ratio
 - $RRATIO = (\text{Sum of IO} - \text{Sum of SEQIOS}) / \text{Sum of IO}$
- Find Read percentage
 - $(S42DSRDT / \text{IOCOUNT}) * 100$
- Eliminate non-application I/O activity, such as DBA jobs, backup jobs to reduce/eliminate skewing

TYPE 42

- Total backend wait time = Average Read Disc time * Read I/Os
 - $S42DSRDD * S42DSRDT$
- If needed, group table spaces to find average
 - (no sense in moving part of partitioned table space to SSD, in some cases, it may make sense)

TYPE 42

- MXG/SAS to calculate values and store data in VSAM
- Let this process collect data for a couple of days/weeks
- Downloaded the data to spread sheet for analysis
(or)
- REXX/ISPF/Assembler based tool to analyze the data
 - *Criteria: Random Read, low cache hit ratio, high I/O rate, high read percentage and high disconnect time*

TYPE42 Tool Tips

- Use FILTER to analyze the data
- Use DSNNAME to focus on a dataset prefix or dataset
- Use SORT commands to sort any columns
- Use DATE to focus on a particular date
- Use GENHTML command to write results to an html file
 - With HTTP server installed, view the results in a web browser, export the table to a spread sheet and generate graphs

TYPE42 tool tips

- Use FILTERs to analyze/filter entries
- Change DISC (average read disconnect time), IORT (average I/O rate), RH (average read hit), RP (average read percentage), RR (average random ratio)

```
COMMAND ==> TYPE 42 ANALYSIS Row 1 to 16 of 16
Filters ( OFF ) ==> DISC > 4 IORT > 10 RH < 50 RP > 80 RR > 70
Dataset name Disc IORT RH RP RR Date
```

TYPE42 tool tips

- Use `DSNAME *` to retrieve all records, sort the list by `SORT SUMRDISC D`
- This should sort the list by datasets with most I/O wait time (sum of `RDD*RDT`)

```

TYPE 42 ANALYSIS                                     Row 1 to 19 of 5,377
COMMAND ==> DSNAME *
Filters ( OFF ) ==> DISC > 4   IORT > 10   RH < 50   RP > 80   RR > 70
Dataset name                               Disc IORT RH  RP RR  Date

```

TYPE42 tool tips

- DSNNAME dsname ← to focus on a dataset
 - Hint: Perhaps after talking to Applications Support Group
 - You may want to tweak MXG/SAS to force this entry to go to VSAM – default is top 300
 - You may use this command in conjunction with GENHTML command to export the data to a spreadsheet to generate graphs for users

```

COMMAND ==>
Filters ( ON ) ==> DISC > 4   IORT > 10   RH < 50   RP > 80   RR > 70
TYPE 42 ANALYSIS
Row 1 to 3 of 3
Dataset name      Disc IORT  RH   RP  RR  Date
5.10 35   26  96  77
5.42 25   19  98  90
5.75 12   13  98  95
  
```

TYPE42 tool tips

- Use DATE to list all the candidates from a particular date and focus on one by using DSNNAME dsn command. You may need to keep the filter off.

```
COMMAND ==> date 12-02-19 Row 1 to 19 of 33  
Filters ( OFF ) ==> DISC > 4   IORT > 10   RH < 50   RP > 80   RR > 70  
Dataset name                               Disc IORT RH  RP RR  Date
```

TYPE42 tool tips

- DSNNAME command with FILTER off

```

TYPE 42 ANALYSIS                                     Row 1 to 16 of 16
COMMAND ==>
Filters ( OFF ) ==> DISC > 4   IORT > 10   RH < 50   RP > 80   RR > 70
Dataset name                                         Disc IORT RH   RP RR   Date
12-02-01
12-02-02
12-02-03
12-02-04
12-02-05
12-02-06
12-02-08
12-02-10
12-02-11
12-02-12
12-02-13
12-02-14
12-02-15
12-02-16
12-02-17
12-02-18
***** Bottom of data *****

```

TYPE42 tool tips

- GENHTML

```

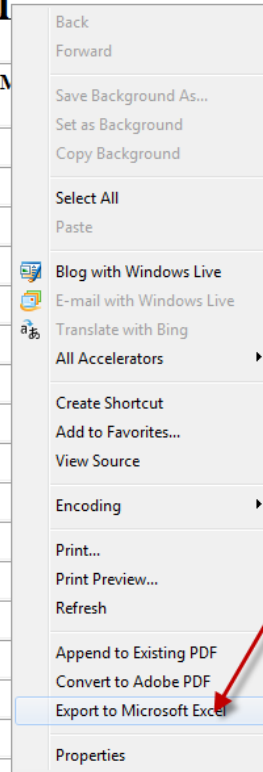
TYPE 42 ANALYSIS                                     Html written
COMMAND ==> GENHTML_
Filters ( OFF ) ==> DISC > 4   IORT > 10   RH < 50   RP > 80   RR > 70
Dataset name                               Disc IORT RH   RP RR   Date
12-02-01
12-02-02
  
```


TYPE42 tool tips

- GENHTML

SSD REPORT

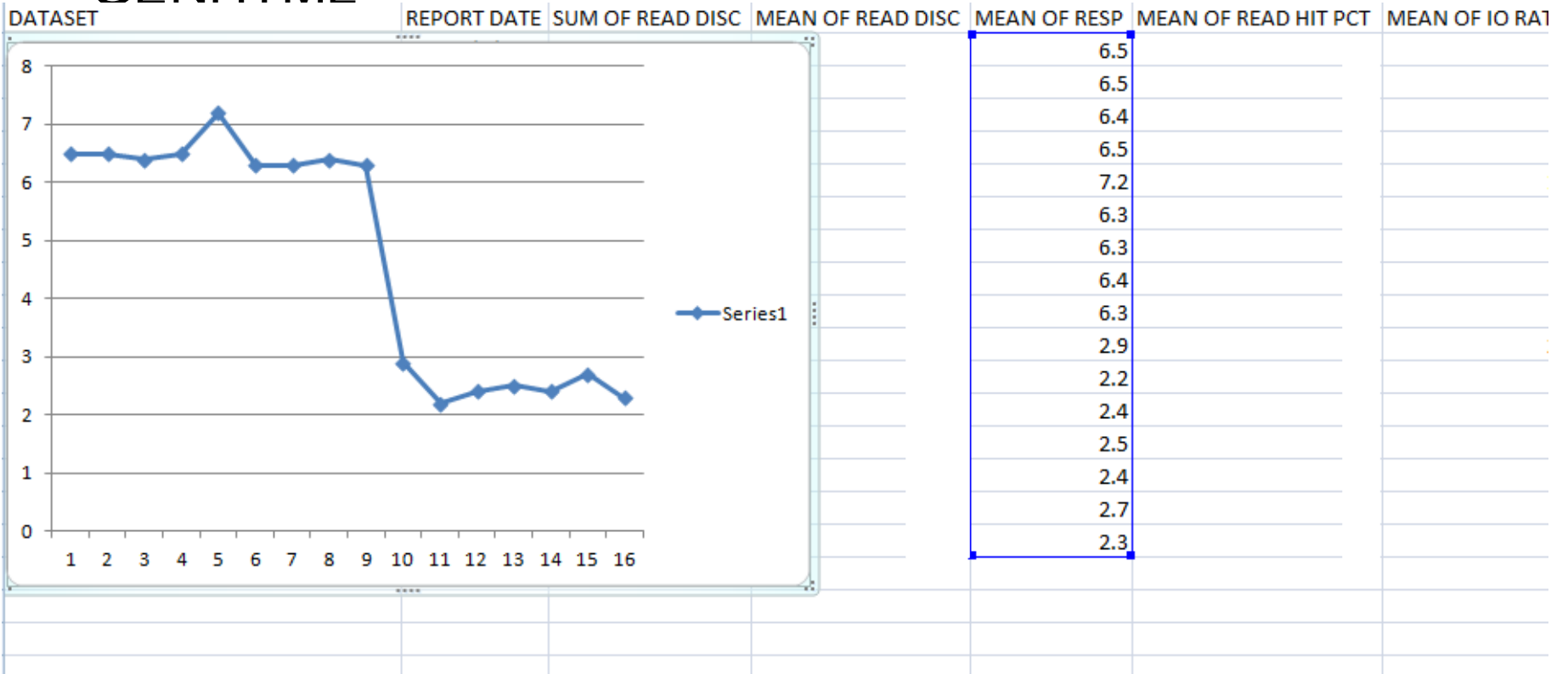
DATASET	REPORT DATE	SUM OF READ DISC	MEAN OF READ DISC	MEAN OF RESP	M	OF IO	MEAN OF RANDOM RATIO	MEAN OF % READ
	12-02-01			6.5				
	12-02-02			6.5				
	12-02-03			6.4				
	12-02-04			6.5				
	12-02-05			7.2				
	12-02-06			6.3				
	12-02-08			6.3				
	12-02-10			6.4				
	12-02-11			6.3				
	12-02-12			2.9				
	12-02-13			2.2				
	12-02-14			2.4				
	12-02-15			2.5				
	12-02-16			2.4				
	12-02-17			2.7				
	12-02-18			2.3				



- Back
- Forward
- Save Background As...
- Set as Background
- Copy Background
- Select All
- Paste
- Blog with Windows Live
- E-mail with Windows Live
- Translate with Bing
- All Accelerators
- Create Shortcut
- Add to Favorites...
- View Source
- Encoding
- Print...
- Print Preview...
- Refresh
- Append to Existing PDF
- Convert to Adobe PDF
- Export to Microsoft Excel
- Properties

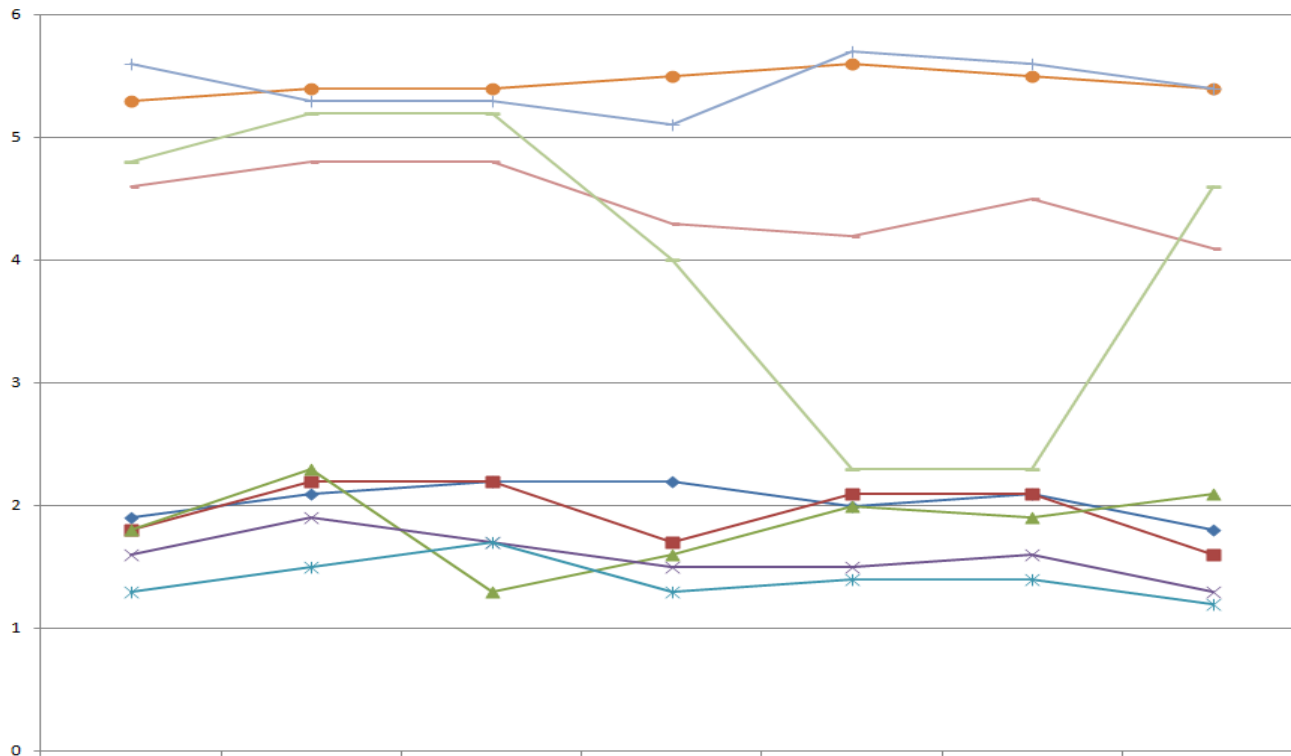
TYPE42 tool tips

- GENHTML



Results - TYPE 42

- Difference between before and after move. Avg. Response time from TYPE42 Sub type 6



From DB2 statistics

- Before move, job elapsed time 15 minutes, after move 5-6 min

```

Menu Print Tools Help CA-Insight for DB2 [redacted] [redacted]
                        r14.0 SP0
_  1 Overview  2 Resp Time  3 Locks  4 Buffers  5 More...

R/HTUDETl          Thread History Summary Overview          Row 7-20/20

Commit - RRSAF...      0          Total      0      0      Aborts      0
DDF TYPE2 - Inact     0          Commits      1
-----
Times in HH:MM:SS.T
Elapsed Time App      6:39.2  Max Pg Locks  1  Select      1  Getpage      1702970
Elapsed Time DB2     6:39.0  Lock Suspnds  0  Fetch       10  Read I/O     169896
CPU Time DB2         36.7    Deadlocks     0  I/U/D       0    Read Eff     10.0
Wait All DB2 I/O     5:59.3  Timeouts      0  Dynamic     0    Pref Reqs   51552
Wt All Lock/Ltch     0.0     Escalations   0  DDL/DCL    0    Buf Updts   49124
Wait Log              0.0     L Prf No Stg  0  Calls       0    BP Warn     3
DB2 Services         0.2     Parallel Err  0  CallFail   0    Avg I/O     0.0016
Wt Data Shr Msgs     0.0
Wait SP/UDF TCB      0.0
Routine Elapsed      0.0
Log Write            6
  
```

Key things to remember

- When you have tight batch schedule, consider SSD
- Do not move all candidates with high disc. time to SSD
- Avoid moving sequential workloads to SSD
- Implement HyperPAVs to avoid shifting the issue from disk to queuing
- Implement zHPF to improve connect time
- Reduce priority for internal array copies
 - Without tuning, this will have negative impact on performance
- Schedule re-org of the tables during non-peak time (non-peak for SSD drives)

Key things to remember

- Schedule re-org of the tables during non-peak time (non-peak for SSD drives)
- Use the tool to monitor moved objects
 - Check Read%, Random I/Os
- Beware of sequential throughput as you consolidate into fewer drives
- Spread SSD across DAs

Key things to remember (cont)

- RAID5 configuration outperforms RAID1 for random reads (because of the number of drives supporting the workload)
 - Eg. 4 x 200GB = 400GB, max approx. IOPS 8,000
 - 4 x 200GB = 600GB, max approx. IOPS 16,000
- Test, Test, Test before moving any workload to understand the limits (knee of the curve)
- Setup RMF report for SSD devices and monitor regularly
- Always check pre and post metrics
- Tweak tool to add new fields you are interested in

Additional resources

<http://www.redbooks.ibm.com/abstracts/redp4537.html>

- This paper says Reorg still matters
- PCI Flash is another hybrid approach, applicable for non-mainframe
- SSD as a staging device within the array (similar to cache, but behind DAs) ?

Q & A

- Session#10913