EPV Technologies

Newsletter

July 2023

THIS MONTH HIGHLIGHTS

- Damned statistics
- Python code can be offloaded to zIIP

Damned statistics

"Lies, damned lies, and statistics" is a phrase describing the persuasive power of statistics which is sometimes used to support weak arguments.
This sentence is often used to doubt statistical results.

However, well understood and well used statistics are a powerful tool that can

be used to analyze almost anything.

Statistics are essential in all capacity management activities, in fact, all these activities rely primarily on SMF records.
SMF stands for System Management Facility but it could also be called Statistics Measurement Facility because it provides thousands of important metrics and statistics about z/OS resource usage and workload activity.

This paper is intended for young z/OS performance analysts.
Its goal is to clarify the meaning of the most used statistics and show examples of appropriate use.

If you want to receive the paper you can reply to this e-mail writing **"Damned statistics"** in the subject

---

Python code can be offloaded to zIIP

Up to 70% of the Python code can be offloaded to zIIP in z/OS 2.4 or 2.5 with the following maintenance:
- BCP APAR OA63406 with PTF UJ92511 for z/OS 2.4
- BCP APAR OA63406 with PTF UJ92512 for z/OS 2.5
- Open Enterprise SDK for Python APAR PH52983

We are implementing zHyperLink. Should I use disk I/O statistics to measure response time improvement, or it should be done differently? I have a feeling that

usual metrics don't reflect possible improvement correctly. I expect improvement in working with DB2 data sets. I would really appreciate your opinion on this matter.

EPV Technical Support answer

There is a new set of SMF metrics available for zHyperLink I/Os (see below) and there are reports in EPV showing them.

SMF74SBR	Number of synchronous I/O read bytes transferred.
SMF74SBW	Number of synchronous I/O write bytes transferred.
SMF74SQR	Number of successfully completed synchronous I/O read requests.
SMF74SQW	Number of successfully completed synchronous I/O write requests.
SMF74SPR	Processing time (in 0.5 microsecond units) for successful synchronous I/O read requests.
SMF74SPW	Processing time (in 0.5 microsecond units) for successful synchronous I/O write requests.
SMF74SFTR	Elapsed time (in 0.5 microsecond units) for unsuccessful synchronous I/O read requests.
SMF74SFTW	Elapsed time (in 0.5 microsecond units) for unsuccessful synchronous I/O write requests.
SMF74SLBR	Number of synchronous I/O read link busy conditions.
SMF74SLBW	Number of synchronous I/O write link busy conditions.
SMF74SCMR	Number of cache miss conditions for synchronous I/O read requests.
SMF74SNIS	Number of synchronous I/O write requests where the write data could not be immediately stored.
SMF74STOR	Number of synchronous I/O read timeout conditions.
SMF74STOW	Number of synchronous I/O write timeout conditions.
SMF74SOR	Number of synchronous I/O read requests rejected for reasons other than link busy, read cache miss or timeout conditions.
SMF74SOW	Number of synchronous I/O write requests rejected for reasons other than link busy, timeout or deferred write conditions.

Remember that, depending on the workload, on the same disk volume you may have traditional (asynchronous) I/Os or zHyperLink (synchronous) I/Os.

It may happen also in disks dedicated to Db2 logs where only writes to the log can use zHyperLink.
In this case you will have reads still measured the old way (where response/connect/pending and disconnect time are still relevant) and writes measured with the new zHyperLink metrics.

If Db2 is enabled to use zHyperLink also for synchronous reads you will have a different situation where asynchronous reads and writes are still managed the old way.

In general, I think the best approach could be to make a general average of read and write response time for the Db2 disk volumes and check it before and after the

zHyperLink activation. To do that you have to put together synchronous and asynchronous measurements.

Optimal Block Size

Setting an optimal block size when reading or writing a dataset is needed to save disk space, get good I/O performance thus reducing job elapsed time.

When working with sequential datasets the solution is simple: use BLKSIZE=0 in your JCL.

We made a test with two jobs executing the IEBGENER utility to copy an SMF dataset using the following DCBs for the output dataset:

- JOB1: LRECL=32760,RECFM=VBS,BLKSIZE=4096
- JOB2: LRECL=32760,RECFM=VBS,BLKSIZE=0

We run the test more times. These are the results:

	DISK SPACE USED (CYLs)	EXCPs (K)	ELAPSED seconds
JOB1	2392	493	41

JOB2	2096	195	33

With this simple change we got a reduction of:

- about 12% of the disk space used;
- about 20% of the elapsed time;
- about 60% of the number of EXCPs.

The reduction in the number of EXCPs also resulted in a slight reduction of the CPU time.

Quotes

Katsumoto : "You believe a man can change his destiny?"
Algren : "I think a man does what he can, until his destiny is revealed."

The Last Samurai

If you've received this mail by mistake, or you don't want to receive any more such messages, please send an e-mail to epv.info@epvtech.com with subject "REMOVE". You'll be promptly removed from the list. If you want to subscribe to this list you can do that simply by sending an e-mail to epv.info@epvtech.com with a subject "SUBSCRIBE".

If you've received this mail by mistake, or you don't want to receive any more such messages, please send an e-mail to epv.info@epvtech.com with subject "REMOVE". You'll be promptly removed from the list. If you want to subscribe to this list you can do that simply by sending an e-mail to epv.info@epvtech.com with a subject "SUBSCRIBE".

Our mailing address is:
EPV Technologies
Viale Angelico, 54
Roma, RM 00195
Italy

Add us to your address book

Our mailing address is:
EPV Technologies
Viale Angelico, 54
Roma, RM 00195
Italy

Images designed by : Freepik, Flaticon