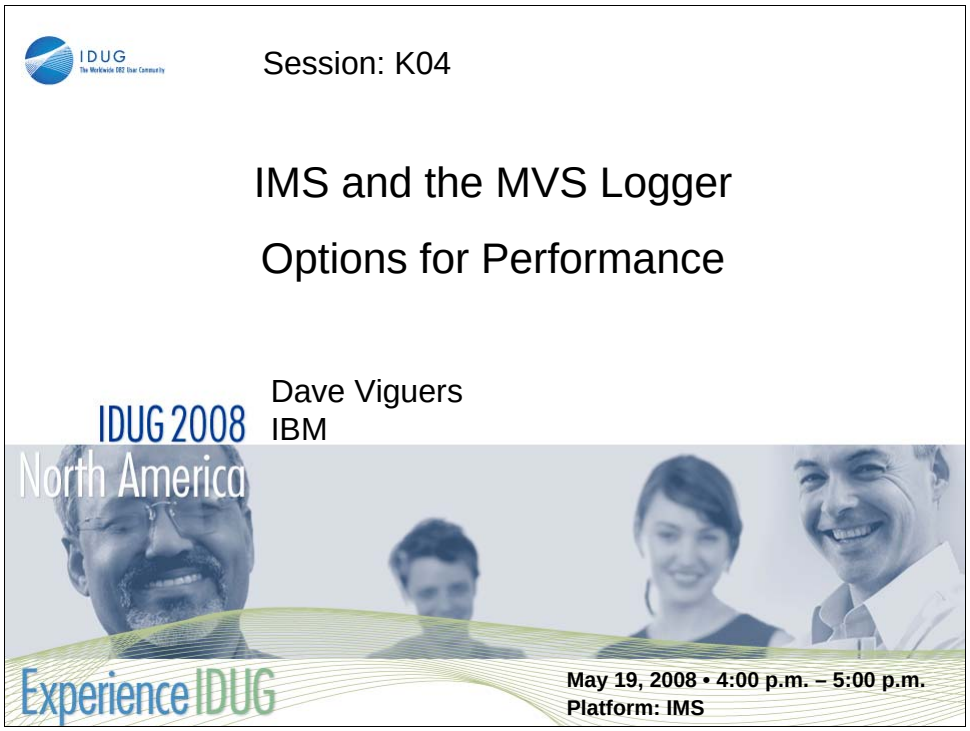


IMS continues to expand its use of the MVS system logger for new functions. The primary performance critical use however is still for CQS logging of input and output messages in a shared queues environment. This presentation will look at how IMS uses the MVS logger and considerations for optimizing the performance in different environments.

IMS and the MVS Logger - Options for Performance

- Understand how IMS uses the MVS system logger
- Learn the tuning parameters within IMS related to the system logger
- Learn the tuning parameters for the MVS system logger
- Understand the impact of certain options on performance
- Learn how to find and interpret performance related information for the logger

IMS users of the MVS Logger

- As of IMS V10 there are two users
  - CQS logging with shared queues
    - Logs operations to the structures
      - Input and output messages
      - Operations to those messages
        - Move, delete, etc.
  - Operations manager logging
    - Logs IMS commands, & responses

OM Logging Overview

- OM logs commands and responses that use the OM API
- Logging is optional
  - AUDITLOG=*name* in CSLOIxxx member
- A single MVS logstream is used for all OM's in the same IMSPLEX group
- MVS Logger offloads to DASD when structure threshold is reached
- IMS (OM) does not perform any delete processing
  - Use RETPD(n) and AUTODELETE(YES) in MVS logger

The 'name' on AUDITLOG= is the 1-26 character name of the MVS logstream.

The logger offload threshold is specified in the logger policy.

RETPD is specified in days and applied to an entire data set, not an individual message. Since IMS does not perform delete processing it is important to specify AUTODELETE(YES) and an appropriate value for RETPD depending on how long you desire to keep this audit trail.

OM Logging

- Specify MAXBUFSIZE of 32760 to prevent truncation
- Volume will depend on number of OM's as well as the volume and type of commands
  - START DB DI21PART would be minimal
  - DISPLAY DB ALL might be a bit more
- OM logging is not in the critical transaction processing path like our next topic

OM will segment data longer than 32760 and if the MAXBUFSIZE is smaller than this your command output may be truncated.

The amount of generated data will vary considerably based on the number of commands and the amount of command output generated.

CQS Logging Overview

- IMS passes message to CQS
  - CQS logs message then puts on SMQ structure
- IMS Requests message for processing
  - CQS logs read/move from ready to lock queue
  - CQS passes message to IMS to Appl
- Application ISRT's output message
  - CQS logs message and puts on SMQ structure
  - Input message is deleted and logged as such
- IMS requests output message to send
  - CQS logs read/move from lterm ready to lock queue
  - Output message is deleted and logged when delivered

CQS logging is synchronous to the task requesting this function meaning the data is hardened to the logger structure and optionally the staging data sets before processing of the message can resume.

CQS Logging Basics

- Basically 4.x structure accesses per transaction
  - The x is for the DELETE's which are batched
- IMS requires a logger structure
  - DASD only log streams not supported
- The MVS logger will also duplex the log stream(s)
  - Duplexing can take different forms
    - Dataspace
    - Staging Data Sets
    - System managed structure duplexing
  - Performance implications covered shortly

Because DASD log streams are unique to a specific system IMS requires that the structure be used so that the log records from all systems are included should recovery be necessary.

The logger duplexes differently depending on various settings and the HW configuration. These options are covered in subsequent slides.

Tuning from the IMS/CQS side

- Keep the number of Structure accesses to the minimum
  - Minimum is 4.x as mentioned before
  - However, there could be more if messages span multiple QBUFS requiring more than 1 put to the SMQ structure
    - Meaning more than 1 log record also
- Evaluate your SMSG, LGMSG, and QBUF size
  - Use IMSPA MSGQ report with INTERVAL

Message Queue Utilization-IMS1

Msg Length Interval	Msg Avg Length	Input Transaction Count	ShMsg	LgMsg	-Message Switch-- Count	ShMsg	LgMsg	-Program Switch-- Count	ShMsg	LgMsg	-Output Message-- Count	ShMsg	LgMsg	------Totals----- Count	ShMsg	LgMsg	Pct
00000-01023	939	59871	59871	-	4105	4100	5	1559	1559	-	21566	21566	-	87101	87096	5	8
01024-02047	1441	369K	352K	17134	27234	25300	1934	2779	2771	8	279K	277K	2168	679K	658K	21244	62
02048-03071	2614	75805	-	75805	3559	26	3546	624	-	624	98327	-	98327	178K	26	178K	16
03072-04095	3649	2	-	2	2135	114	2079	23130	-	23130	28149	-	28149	53416	114	53360	5
04096-05119	4653	-	-	-	4374	177	4315	130	-	130	11061	-	11061	15565	177	15506	1

In order to minimize the number of structure accesses for best performance the size of your messages should be taken into consideration before setting the SMQG, LGMSG and QBUF sizes. The IMSPA MSGQ report may be used for this. While you may need multiple QBUFS to hold larger messages it is usually possible to optimize the storage use and number of CF accesses with a little analysis.

MVS Logger Basics

- One full function log stream for each SQ group
  - Optionally a FP log stream if using SEMH
- Each MVS Logger writes to the same logstream
  - Log records are kept in sequence across the sysplex
- CQS issues IXGWRITE to interface to logger
  - Also IXGBRWSE and IXGDELET
- Logger writes to data space AND structure
  - Maybe staging data sets instead of data space
  - Depends on options defined
- Logger returns to CQS who then operates on the SMQ structures
- Logger offloads data to DASD when pre-defined threshold is reached
  - CQS will tell logger to delete data older than the oldest structure checkpoint

There is one MVS logger address space for each LPAR. Each subsystem using the logger will interface with it using the IXGWRITE and other logger macros. From an IMS/CQS perspective you could think of these log writes as being similar to an IMS CHECKWRITE.

MVS Logger Structure (LOGR)

- Defined in Logger Policy AND CFRM Policy
- Logger Policy definition
  - AVGBUFSIZE
    - Used to determine INITIAL entry/element ratio
    - Logger monitors every 30 minutes and may adjust
    - Run IXCMIAPU utility to get current value
  - MAXBUFSIZE
    - Determines element size
    - 65,276 or lower means 256 byte elements
    - Larger than 65,276 will cause 512 byte elements
    - Probably no reason to use 512 byte elements
  - LOGSNUM
    - 1 is recommended here

Using 256 byte elements is usually best since this will reduce wasted space and does not cause any noticeable additional processing overhead.

Multiple log streams (full function and fast path) in a structure are not recommended and could potentially cause bottlenecks.

MVS Logger Structure (CFRM)

- CFRM definition
  - INITSIZE
    - The initial allocation size of the structure
    - Specify smaller than SIZE to allow alter
  - SIZE
    - The largest value that this structure can be
  - MINSIZE
    - Set this to make sure system or operator does not alter too small
  - DUPLEX
    - Whether or not to use or allow system managed duplexing
    - Recommendation later
  - ALLOWAUTOALT
    - Let Logger handle this so set to NO

The MVS logger monitors both the amount of space being used in the structure AND the entry/element ratio and will adjust as necessary therefore auto alter is not recommended.

Logstream Definition (LOGR)

- HIGHOFFLOAD / LOWOFFLOAD
  - 0 for low – 50 to 70 for high
- LS_SIZE
  - Size of each offload data set
  - Larger size means less frequent allocations
- DUPLEXMODE
  - COND or UNCOND
- STG_SIZE
  - Size to allocate staging data sets
  - Needs to be much larger than structure
    - 4k blocks means unused space
- STG_DUPLEX
  - Yes or No to duplex to staging data sets
- RETPD
  - How many days to keep logger data
  - Unless needed for auditing set this to 0

The values for offload apply to both the structure and staging data sets.

When writing to the staging data sets the logger will always write in 4k increments so there is typically a significant amount of unused space.

RETPD should be 0 for CQS but needs to be set for the audit log of OM.

Performance implications

- Many Parameters interact
- SIZE (structure) and HIGHOFFLOAD
  - Impacts frequency of offload
  - Amount of data offloaded
  - Can also impact MVS storage usage
- STG_SIZE and HIGHOFFLOAD
  - STG_SIZE needs to be larger than structure
  - Logger offloads when either structure of staging reaches threshold

A larger structure can mean less frequent offloads however it can also mean more storage usage in the data space so be careful not to make so large as to possibly cause paging.

Performance implications

- LS_SIZE
  - Offload data set size (in 4k)
  - Large to allow multiple offloads per extent
    - Each offload is HIGHOFFLOAD % of either staging or structure
  - Large to reduce number of allocations
    - Allocations take longer than just adding to existing

LS_SIZE will interact with the structure size and/or staging data set sizes along with the offload threshold.

Performance implications

- DASD
  - Staging data sets are critical
    - Affects online transactions
      - Response time
      - Region occupancy
  - Offload data sets not quite as critical
    - Unless you fill up the structure before offload completes
    - Processing being stopped is usually not considered good performance

Staging data sets are like Checkwrites in IMS and are critical to transaction performance. More about this in subsequent slides. The Logger is quite efficient with offloads to VSAM Linear data sets but offload must be able to keep ahead of the actual logging volume just like IMS archive.

Duplexing

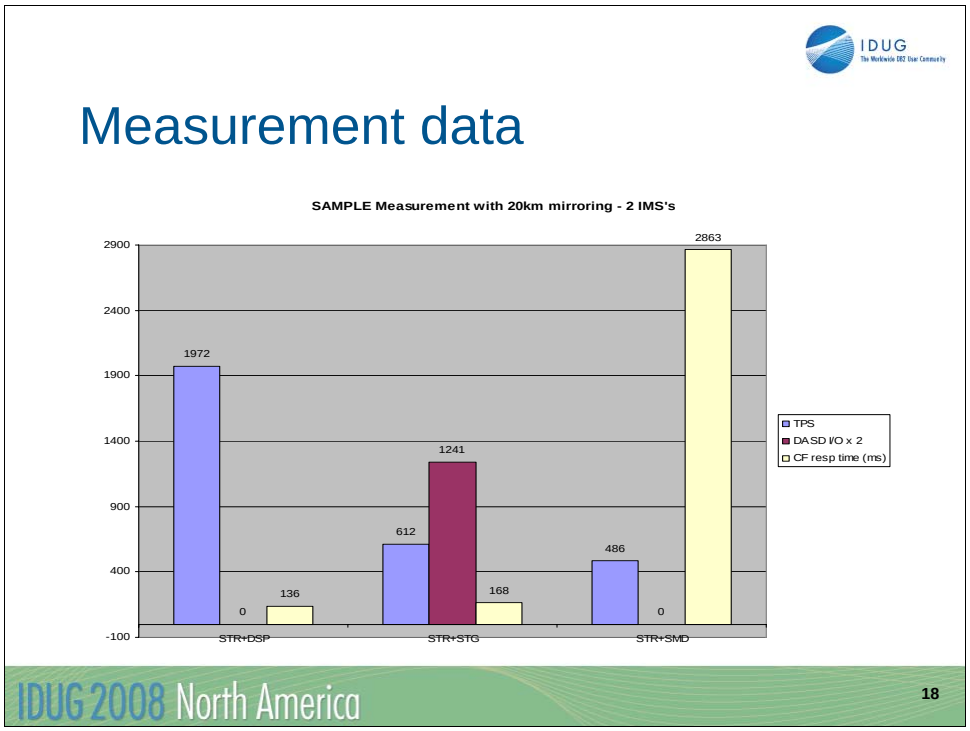
- Dataspace is always best
  - Availability must be considered however
  - ICF's or volitile CF's could require other means
- Staging Data Sets
  - Performance critical path
  - Transaction thruput is limited to DASD speed
  - Striping only helps a bit

For availability it may be necessary when not using external CF's with UPS to use duplexing other than the dataspace. While a dataspace is the best from a performance perspective it may not be a valid option. Depending upon your transaction volume staging data sets or SMD may be quite satisfactory.

Duplexing

- System Managed Duplexing
  - Usually slower than staging data sets
  - Depends on DASD speed of course
- Bottom line from performance perspective
  - Dataspace if possible
  - Staging data sets with fast DASD
  - System Managed Duplexing

System managed duplexing is not nearly as fast as some software duplexing however not all structures for all subsystems have their own duplexing support. Whether SMD or staging data sets are better will depend on the DASD speed as well as the CF and CF link speed.



In this example the method of duplexing was varied. For the first measurement the structure was used with data space duplexing. As you can see this was by far the best performing option.

The second set of bars is for duplexing to the staging data sets. Here you can see that there are approximately 4 I/O's per transaction. In our case the response time for the staging data sets was 2ms with synchronous mirroring (PPRC) at 20km. When using a multi site sysplex this would be the only way to have the data for recovery at both sites.

The 3rd set of bars is when using system managed duplexing. This method requires that the 2 CF's communicate with each other to coordinate the updates being made.

MISC

- RETPD
  - How many days to keep data
  - How much DASD do you have?
  - CQS will keep 2 structure checkpoints worth
  - Set to 0 unless otherwise necessary
    - OM Auditing?
- AUTODELETE
  - No (default) for CQS
  - Yes with RETPD for OM

CQS will keep data from two structure checkpoints at a minimum. Data older than that will be deleted asynchronously but will be on a data set boundary. The only reason to specify a retention period would be for some need other than CQS restart/rebuild processing. For OM you should specify a retention period and AUTODELETE(YES) to avoid filling up the logstream DASD.

Printing CQS log records

- DFSERA10
  - With CQSERA30 exit routine supplied with IMS
  - DSN=logstream name
- LOG RECORD descriptions
  - Assemble CQSLGREC TYPE=ALL

```
//CQSERA10 JOB ....
//STEP1 EXEC PGM=DFSERA10
//STEPLIB DD DSN=IMSV10.SDFSRESL,DISP=SHR
//SYSPRINT DD SYSOUT=A
//TRPUNCH DD SYSOUT=A,DCB=BLKSIZE=80
//SYSUT1 DD DSN=MSGQ.LOG.STREAM,
// SUBSYS=(LOGR,IXGSEXIT,'FROM=(2008/011,07:00:00),TO=(2008/011,23:30:00),LOCAL')),
// DCB=BLKSIZE=32760
//SYSIN DD *
CONTROL CNTL H=EOF
OPTION PRINT EXITR=CQSERA30
END
/*
```

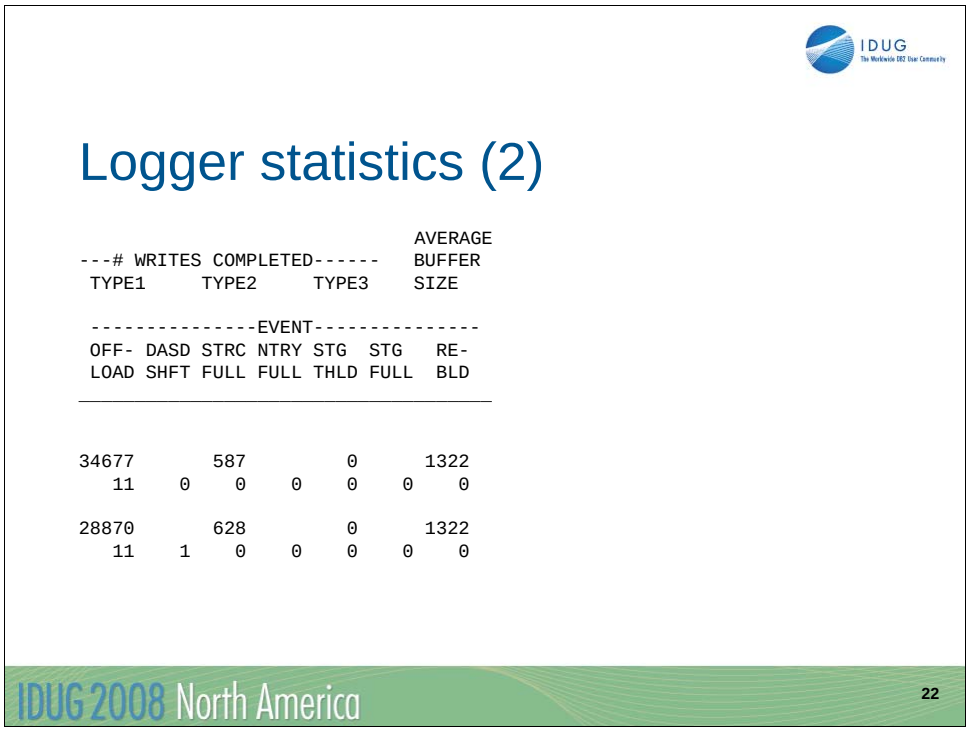

There are also CQS statistics records written to the log. These could be helpful when debugging a performance issue.

Logger Statistics (1)

- Written to SMF
  - Type 88 records
- Print with IXGRPT1

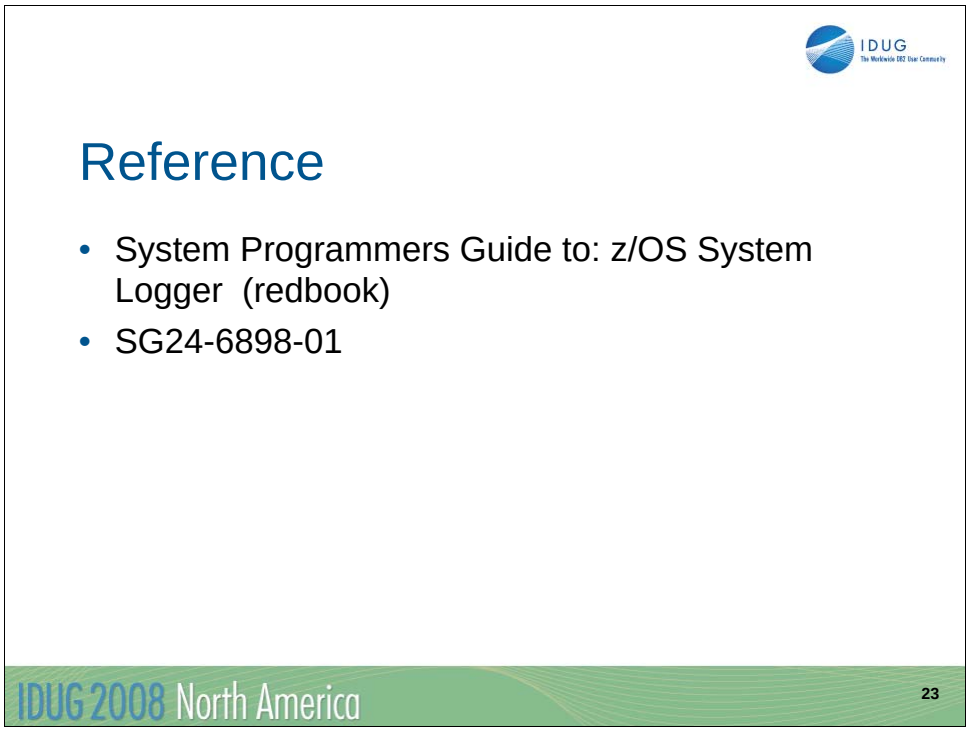
```
 SYSTEM LOGGER ACTIVITY REPORT (IXGRPT1)


                                          BYT WRITTN  BYT WRITTN  BYT WRITTN
                                           BY USERS   TO INTERIM    TO DASD    #WRITES
--LOGSTREAM NAME------------ STRUCTURE NAME--  IXGWRITES   STORAGE                INVOKED

                                          BYT DELETD  # DELETES   BYT DELETD  # DELETS
                                          INTERIM ST   W/O DASD   INTERIM ST      W/
                                          W/O DASD      WRITE      W/DASD        WRITE

-------------------------------------------------------------------------------------

03/06/07  7:55:00 PM  (SMF INTERVAL TIMESTAMP 'C0417BBAD8D00000'X)
  IM0A.MSG.LOGSTR            IM0A_LOGM          311135163   339884032   734566494    235263
                                                        0           0   713006894    538990
03/06/07  7:56:00 PM  (SMF INTERVAL TIMESTAMP 'C0417BF411400000'X)
  IM0A.MSG.LOGSTR            IM0A_LOGM          303507256   331570688   792700054    229499
                                                        0           0   769432174    581697
```


BYT writtn by users – total bytes from CQS. BYT writtn to interim – this includes rounding to the element boundary. BYT writtn to DASD – Offload bytes. #WRITES – IXGWRITES done. BYT DELETD W/O DASD – this will be 0 for CQS. # DELETES W/O DASD – also 0 for CQS. BYT DELETED W/DASD – bytes offloaded. # DELETS W/WRITE – number of delete after offload.



This is the right side of the previous page.

Type1 are the number of normal writes complete below the threshold value. Type2 are the writes at or above threshold but below 90%. Type3 are writes are at or above 90% - shoot for zero. OFFLOAD is the number of offload processes this interval. DASD SHFT is a new data set allocation. STRC FULL is when at limit – should be 0. NTRY FULL – entries at 90% or above, offload initiated. STG THLD – staging data set reached threshold, should be 0. STG FULL – self explanitory. REBLD – any rebuilds done.



This is a good redbook for understanding all about the system logger with a specific section for IMS.

SUMMARY

- Allocate enough DASD to avoid out of space
  - IMS stopping is not good
  - Must hold at least 2+ structure checkpoints
    - Adjust checkpoint interval or space
- LS_SIZE large
  - Avoid unnecessary new allocations
- Set LOWOFFLOAD to 0
- Use external CF's if possible to avoid staging data sets or SMD
- Set IMS SMSG, LGMSG, QBUF sizes to minimize CF access

Session K04

IMS and the MVS Logger –
Options for Performance

Dave Viguers

IBM

dviguers@us.ibm.com