DB2 for z/OS
Best Practices

Log Activity and Miscellaneous

John Campbell
Distinguished Engineer
DB2 for z/OS Development

CAMPBELJ@uk.ibm.com

Disclaimer/Trademarks

THE INFORMATION CONTAINED IN THIS DOCUMENT HAS NOT BEEN SUBMITTED TO ANY FORMAL IBM TEST AND IS DISTRIBUTED AS IS. THE USE OF THIS INFORMATION OR THE IMPLEMENTATION OF ANY OF THESE TECHNIQUES IS A CUSTOMER RESPONSIBILITY AND DEPENDS ON THE CUSTOMER'S ABILITY TO EVALUATE AND INTEGRATE THEM INTO THE CUSTOMER'S OPERATIONAL ENVIRONMENT. WHILE IBM MAY HAVE REVIEWED EACH ITEM FOR ACCURACY IN A SPECIFIC SITUATION, THERE IS NO GUARANTEE THAT THE SAME OR SIMILAR RESULTS WILL BE OBTAINED ELSEWHERE. ANYONE ATTEMPTING TO ADAPT THESE TECHNIQUES TO THEIR OWN ENVIRONMENTS DO SO AT THEIR OWN RISK.

ANY PERFORMANCE DATA CONTAINED IN THIS DOCUMENT WERE DETERMINED IN VARIOUS CONTROLLED LABORATORY ENVIRONMENTS AND ARE FOR REFERENCE PURPOSES ONLY. CUSTOMERS SHOULD NOT ADAPT THESE PERFORMANCE NUMBERS TO THEIR OWN ENVIRONMENTS AS SYSTEM PERFORMANCE STANDARDS. THE RESULTS THAT MAY BE OBTAINED IN OTHER OPERATING ENVIRONMENTS MAY VARY SIGNIFICANTLY. USERS OF THIS DOCUMENT SHOULD VERIFY THE APPLICABLE DATA FOR THEIR SPECIFIC ENVIRONMENT.

Writing to the Active Log

- When does DB2 write to the active log?
  - UR-related events
    - End Commit Ph1
    - Begin Commit Ph2 (2 log writes for 2 phase commit)
    - Ph1/Ph2 combined for 1 phase commit (1 log write for 1 phase commit)
  - Database writes
    - "Write ahead" logging protocol
      - All undo/redo records pertaining to the page being written must be hardened on disk before the page is written to DASD or CF
      - Page P-lock negotiation and index leaf-page split in data sharing
        - 2 forced physical log writes per index leaf-page split
    - Force log up to PGLOGRBA (RBA/LRSN of last update to the page)
  - System checkpoints

Writing to the Active Log …

- When does DB2 write to the active log? …
  - Log write threshold reached
    - Fixed "20 buffer" value V7 and above
    - Prior to V7 controlled by ZPARM WRTHRSH
  - ARCHIVE LOG command
  - IFI read for IFCID 306 from another member
- Log writes are chained, up to 128 4K CIs per write I/O
  - It takes at least 4 CIs to get parallel overlap and start benefiting from DFSMS striping
    - First and last CIs are always done serially
- For dual logging
  - Full CI writes are fully overlapped
  - CI re-writes are done serially

Log Write Statistics

Field Name	*Description*
QJSTWTB	UNAVAILABLE OUTPUT LOG BUFFER
QJSTBPAG	LOG OUTPUT BUFFER PAGED IN
QJSTWRNW	NOWAIT LOG WRITE REQUESTS
QJSTWRF	FORCE LOG WRITE REQUESTS
QJSTBFFL	ACTIVE LOG OUTPUT CI CREATED
QJSTLOGW	LOG WRITE I/O REQUESTS
QJSTCIWR	LOG CI WRITTEN

```
LOG ACTIVITY                QUANTITY  /SECOND  /THREAD  /COMMIT
-------------------------   --------  -------  -------  -------
...

UNAVAILABLE OUTPUT LOG BUFF    10.00     0.00     0.00     0.00
OUTPUT LOG BUFFER PAGED IN      0.00     0.00     0.00     0.00

LOG RECORDS CREATED          23673.6K  3315.64   402.30    98.14
LOG CI CREATED                1710.1K   239.51    29.06     7.09
LOG WRITE I/O REQ (LOG1&2)    1816.3K   254.39    30.87     7.53
LOG CI WRITTEN (LOG1&2)       4383.3K   613.91    74.49    18.17
LOG RATE FOR 1 LOG (MB)           N/A     1.20      N/A      N/A
LOG WRITE SUSPENDED            623.8K    87.37    10.60     2.59
```

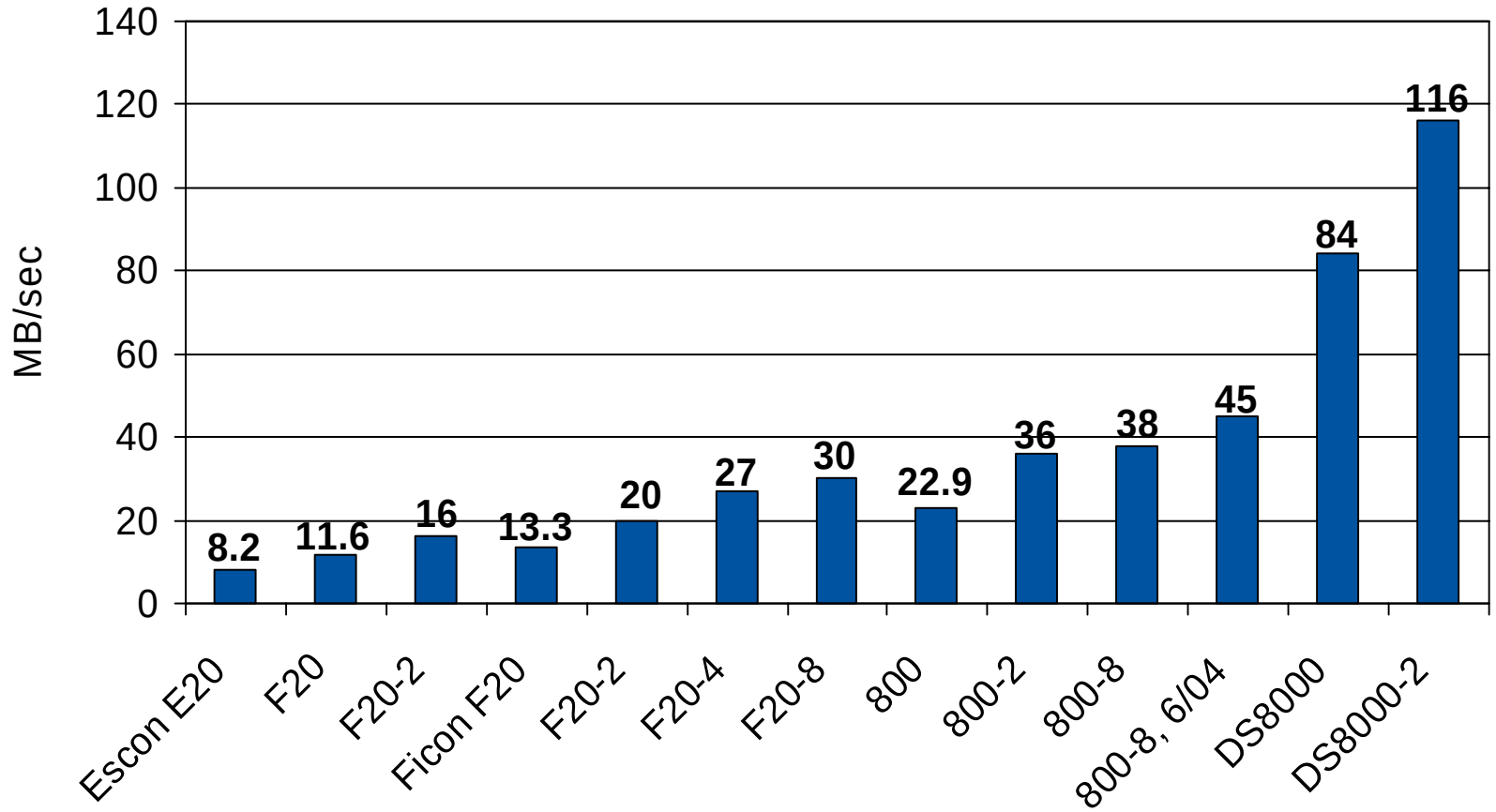
- Output Log Buffer size
  - Controlled by ZPARM OUTBUFF (storage allocated in MSTR address space)

Increase if UNAVAIL OUTPUT LOG BUF > 0 and log data volume << max
Decrease if OUTPUT LOG BUFFER PAGED IN > 1-5% of LOG RECORDS CREATED

  - More output log buffer space may also help for log reads
- Log data volume = LOG CI CREATED * 4KB / stats_interval
  - Start paying attention if >10MB/sec

Maximum Observed Rate of Active Log Write

- First 3 use ESCON channel, the rest is FICON channel
- -2, -4, -8 indicate 2, 4, or 8 I/O stripes



Log Dataset I/O Tuning

- Avoid I/O interference among Active Log Read/Write and Archive Log Read/Write
- If Log Rate near maximum,
  - Use faster log device
  - Consider use of DFSMS striping
  - Reduce log data volume
    - Use of DB2 Data Compression for insert intensive tables
    - Optimise table design to minimise log record size (BRF only)
    - Be aware of impact of DBD update
- If the DB2 archive log offload process cannot keep up with speed that the DB2 active log fills
  - Consider DFSMS striping of archive log files (V9)

Log Reads

- Log reads are driven by any of the following types of activity
  - Rollback of a unit of work
  - Data recovery
    - RECOVER utility
    - LPL or GRECP recovery
  - Online Reorg LOGAPPLY phase
  - Restart recovery
  - IFI log read interface
    - Data replication
    - IFCID 129: read range of log record CIs from active log
    - IFCID 306: supports archive logs, decompression, and data sharing log merge
      - Option to not merge
  - Standalone log read – DSNJSLR as DSN1LOGP

Log Read Performance

Field Name	*Description*
QJSTRBUF	READS SATISFIED FROM OUTPUT BUF
QJSTRACT	READS SATISFIED FROM ACTIVE LOG
QJSTRARH	READS SATISFIED FROM ARCHIVE LOG

```
LOG ACTIVITY                QUANTITY /SECOND /THREAD /COMMIT
--------------------------  -------- ------- ------- -------
READS SATISFIED-OUTPUT BUFF 27654.00    1.28    0.17    0.06
READS SATISFIED-OUTP.BUF(%)    21.67
READS SATISFIED-ACTIVE LOG  99955.00    4.64    0.60    0.20
READS SATISFIED-ACTV.LOG(%)    78.33
READS SATISFIED-ARCHIVE LOG     0.00    0.00    0.00    0.00
READS SATISFIED-ARCH.LOG(%)     0.00
```

- More output log buffer space may help log read performance
- Active log reads perform best due to
  - Prefetch of log CIs enabled
  - Automatic I/O load balancing between copy1/copy2
  - VSAM striping can be enabled
  - Reduced task switching
- Archives on DASD perform better: tape destroys parallelism
  - Tape requires serialization in data sharing between concurrent Recovers
  - You cannot concurrently share tape volumes across multiple jobs
  - Small number of read devices on backend of current VTS models, which can become a severe bottleneck for DB2 mass recovery

DB2 for z/OS
Best Practices

Miscellaneous

RID List Processing

Field Name	*Description*
QISTRLLM	TERMINATED-EXCEED RDS LIMIT
QISTRPLM	TERMINATED-EXCEED DM LIMIT

```
RID LIST PROCESSING            QUANTITY  /SECOND  /THREAD  /COMMIT
---------------------------    --------  -------  -------  -------
MAX RID BLOCKS ALLOCATED        8469.00      N/A      N/A      N/A
CURRENT RID BLOCKS ALLOCAT.       47.28      N/A      N/A      N/A
TERMINATED-NO STORAGE              0.00     0.00     0.00     0.00
TERMINATED-EXCEED RDS LIMIT      515.00     0.07     0.01     0.00
TERMINATED-EXCEED DM LIMIT         0.00     0.00     0.00     0.00
TERMINATED-EXCEED PROC.LIM.        0.00     0.00     0.00     0.00
```

- RID list processing failures may cause unnecessary CPU resource consumption and possibly unnecessary I/O, as in most cases, DB2 reverts to tablespace scan
  - TERMINATED-EXCEED DM LIMIT
    - Number of RID entries > physical limit (approx. 26M RIDs)
  - TERMINATED-EXCEED RDS LIMIT
    - Number of RIDs that can fit into the guaranteed number of RID blocks > maximum limit (25% of table size)
- Most common reasons
  - Inaccurate or incomplete statistics
    - e.g. old statistics, inadequate or missing distribution statistics collection
  - Use of the LIKE operator in SQL statements
  - Use of host variables or parameter markers for range predicates on SQL statements (BETWEEN, >, <)
- Identify offending applications and SQL statements with accounting reports and/or IFCID 125

Phantom or Orphaned Trace

IFC DEST.	WRITTEN	NOT WRTN	BUF.OVER	NOT ACCP	WRT.FAIL
SMF	18779.00	0.00	0.00	0.00	0.00
GTF	1048.1K	491.00	N/A	491.00	0.00
OP1	1261.8K	56.00	N/A	56.00	N/A
OP2	0.00	0.00	N/A	0.00	N/A
OP3	0.00	0.00	N/A	0.00	N/A
OP4	0.00	0.00	N/A	0.00	N/A
OP5	0.00	0.00	N/A	0.00	N/A
OP6	0.00	0.00	N/A	0.00	N/A
OP7	0.00	1260.3K	N/A	1260.3K	N/A
OP8	0.00	0.00	N/A	0.00	N/A
RES	0.00	N/A	N/A	N/A	N/A
TOTAL	2328.6K	1260.8K		1260.8K	0.00

IFC RECORD COUNTS	WRITTEN	NOT WRTN
SYSTEM RELATED	179.00	0.00
DATABASE RELATED	120.00	0.00
ACCOUNTING	981.7K	536.4K
START TRACE	1.00	3.00
STOP TRACE	3.00	0.00
SYSTEM PARAMETERS	107.00	65.00
SYS.PARMS-BPOOLS	62.00	0.00
AUDIT	11.00	0.00
TOTAL	982.2K	536.4K

- IFC RECORD COUNTS NOT WRTN
  - Phantom or orphaned trace because monitoring (e.g. vendor tool) stopped but the corresponding DB2 trace didn't
- Same CPU overhead as real trace
- Display Trace to check
- V9 (CM) tries to eliminate orphaned trace records

Package List (PKLIST) Search

Field Name	*Description*
QTPKALLA	PACKAGE ALLOCATION ATTEMPT
QTPKALL	PACKAGE ALLOCATION SUCCESS

PLAN/PACKAGE PROCESSING	QUANTITY	/SECOND	/THREAD	/COMMIT
PLAN ALLOCATION ATTEMPTS	166.4K	7.73	1.00	0.34
PLAN ALLOCATION SUCCESSFUL	253.8K	11.78	1.53	0.51
PACKAGE ALLOCATION ATTEMPT	1650.9K	76.64	9.93	3.33
PACKAGE ALLOCATION SUCCESS	1548.0K	71.86	9.31	3.12

- Within each collection (e.g. “COL_a.*, COL_b.*, COL_c.*”), efficient matching index access to find the package, but DB2 goes serially through the PKLIST entries
- Success rate (%) = PACKAGE ALLOC. SUCCESS / PACKAGE ALLOC. ATTEMPT * 100
- Impact of long PKLIST search
  - Additional CPU resource consumption, catalog accesses, and elapsed time
  - Can aggravate DB2 internal latch (LC32) contention
- Recommendations
  - Reduce the number of collections on the PKLIST
    - Scrub all inactive or unused collections on PKLIST
    - Fold in and collapse the number of collections on PKLIST
  - Ruthlessly prioritise and reorder the collection sequence on PKLIST based on frequency of access
  - Use SET CURRENT PACKAGESET special register to direct the search to a specific collection

Disabled SPROCs

Field Name	*Description*
QISTCOLS	# OF COLUMNS (rows x columns) FOR WHICH AN INVALID SPROC WAS ENCOUNTERED

```
---- MISCELLANEOUS -----
BYPASS COL:  1585.00
```

- Many plans/packages have SPROCs for fast column processing
- As a result of invalidation, DB2 has to build SPROCs dynamically at execution time
  - e.g. V7 to V8 migration, V8 to V9 migration
  - Typical CPU performance impact in 0 to 10% range
- Non-zero value for BYPASS COL indicator of problem
- IFCID 224 identifies plans and packages that need rebinding to re-enable SPROCs

Incremental BIND

Field Name	Description
QXINCRB	INCREMENTAL BINDS

```
PLAN/PACKAGE PROCESSING      QUANTITY  /SECOND  /THREAD  /COMMIT
---------------------------  --------  -------  -------  -------
INCREMENTAL BINDS            10138.00     2.82     3.77     0.33
```

- Items that can cause Incremental Bind include
  - Static plan or package with VALIDATE(RUN) and bind time failure
  - Static SQL with REOPT(VARS)
  - Private Protocol in requestor
  - SQL referencing Declared Global Temp Table
  - Possibly DDL statements

Dataset Statistics for I/O Tuning

- Statistics class 8 (IFCID 199)

BPOOL	DATABASE SPACENAM PART	TYPE GBP	SYNCH I/O AVG ASYNC I/O AVG ASY I/O PGS AVG	SYN I/O AVG DELAY SYN I/O MAX DELAY	CURRENT PAGES (VP) CHANGED PAGES (VP) CURRENT PAGES (HP) NUMBER OF GETPAGES
BP10	KAGURA24	TSP	23.35	8	3433
	TETHTS	N	0.01	78	0
	30		32.00		N/A
					2868
BP11	KAGURA24	IDX	102.59	1	18991
	TETHIX1	N	4.04	35	74
	36		5.98		N/A
					245586

Count of Sync I/O per second

Average Sync I/O (ms)