Starter's guide to REAL memory management with DB2 for z/OS

John Campbell
IBM DB2 for z/OS Development
Session Code: 9007
Platform: DB2 for z/OS
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.

Trademarks
IBM, the IBM logo, ibm.com, DB2, FlashCopy, GDPS, Redbooks, System z and z/OS are trademarks of International Business Machines Corp., registered in many jurisdictions worldwide. Other product and service names might be trademarks of IBM or other companies. A current list of IBM trademarks is available on the Web at "Copyright and trademark information" at www.ibm.com/legal/copytrade.shtml.
Agenda

- Common problems
- Real memory monitoring and tuning principle
- DUMP settings
- Tools of choice
- Large size real memory page frames
- Appendix
Common problems

- **Too many installations are running ‘one DUMP away from a disaster’**
  - Not good practice to rely on paging to absorb peak demands
    - DUMP capture, workload failover, system slowdown
  - Can lead to excessive paging and severe **performance issues**
  - Can lead to long DUMP processing times and cause **major disruption**
    - Once paging begins, it is possible to have DUMP processing take 10s of seconds to minutes
    - High risk of sysplex-wide slowdown/hang as p-lock negotiation is affected
  - Ultimately, **can take the LPAR out**
    - Once all AUX is consumed, the LPAR goes into a wait state
Common problems ...

- **Wasted opportunities for CPU reduction**
  - Reluctance to use bigger or more buffer pools
  - Reluctance to use buffer pool long-term page fix
  - Many performance opportunities in DB2 require real memory
    - e.g. RELEASE(DEALLOCATE)
  - Increasingly larger real memory sizes will be required for future performance and scalability improvements
    - e.g. V12 index Fast Traverse Blocks
Real memory monitoring and tuning principle

- **Make sure you have enough real memory to**
  1. Cover your workload peak demands (peak working set)
  2. Allow a DUMP to be captured quickly in memory without causing any disruption
  3. Provide enough spare memory for workload failover or abnormal slowdowns

- **In other words, design and tune for zero paging, even during DUMP capture**
  - Not just about minimising paging rates
  - You want to avoid DB2 being paged out
    - In case of DUMP, each page may have to be paged in, which may severely impact capture time
      - No prefetch on AUX storage, so all synchronous I/Os
How much real memory is configured on the LPAR?

Command

D M=STOR

IEE174I 11.51.10 DISPLAY M
REAL STORAGE STATUS
ONLINE-NOT RECONFIGURABLE
0M-67584M
ONLINE-RECONFIGURABLE
NONE
PENDING OFFLINE
NONE
2048M IN OFFLINE STORAGE ELEMENT(S)
0M UNASSIGNED STORAGE
STORAGE INCREMENT SIZE IS 256M
DUMP settings

Command: D DUMP,O

---

IEE857I 14.38.39 DUMP OPTION
SYSABEND- ADD PARMLIB OPTIONS SDATA=(LSQA,TRT,CB,ENQ,DM,IO,ERR,SUM),
        PDATA=(SA,REGS,LPA,JPA,PSW,SPLS)
SYSUDUMP- ADD PARMLIB OPTIONS SDATA=(SUM), NO PDATA OPTIONS
SYSDUMP- ADD PARMLIB OPTIONS (NUC,SQA,LSQA,SWA,TRT,RGN,SUM)
SDUMP- ADD OPTIONS (ALLPSA,SQA,LSQA,RGN,LPA,TRT,CSA,SWA,SUMDUMP,
        ALLNUC,Q=YES,GRSQ,COUPLE,XESDATA,WLM,SERVERS),
        TYPE=(XMEME),BUFFERS=00000000K,
        MAXSPACE=00020480M, MSGTIME=00000 MINUTES,
        MAXSNDSP=015 SECONDS, AUXMGMT=ON, DEFERTND=NO
SYSFAIL NO STRLIST OPTIONS
ABDUMP- TIMEENQ=0240 SECONDS

**MAXSPACE** restricts the virtual storage available to the DUMPSRV address space

**AUXMGMT** specifies when SDUMP data captures should stop
Default is ON (recommended setting)
- No new dumps are allowed when AUX storage utilisation reaches 50%
- Current dump data capture stops when AUX storage utilisation reaches 68%
- Once the limit is exceeded, new dumps will not be processed until the AUX storage utilisation drops below 35%
DUMP settings ...

- **Make sure AUXMGMT = ON (default)**
  - Stop current as well as new DUMPs when excessive paging on the system
- **Do not oversize AUX otherwise safety nets like AUXMGMT, ENF55 will not help you**
- **Make sure MAXSPACE is set properly and defensively**
  - General default: Plan for approx. 16GB per DB2 member of the same data sharing group running on the same LPAR
    - Allow for DUMP wildcarding
    - Could be adjusted after benchmarking DUMP sizes in the production environment
  - Over-sizing MAXSPACE might result in excessive paging on the system
  - Under-sizing MAXSPACE will result in partial dumps and seriously compromises problem determination
- **Dumps should be taken very quickly (<10 secs) almost without any one noticing i.e., little or no disruption to the subject LPAR and to the rest of the Sysplex**
  - Consider automation to kill dumps taking longer than 10 secs
Storage-Class Memory (SCM) on Flash Express cards

- SCM is optional auxiliary storage, in addition to page data sets

<table>
<thead>
<tr>
<th>Command</th>
<th>\texttt{D M=SCM(DETAIL)}</th>
</tr>
</thead>
<tbody>
<tr>
<td>STORAGE-CLASS MEMORY STATUS - INCREMENT DETAIL</td>
<td></td>
</tr>
<tr>
<td>ADDRESS</td>
<td>IN USE</td>
</tr>
<tr>
<td>0G</td>
<td>0%</td>
</tr>
<tr>
<td>16G</td>
<td>0%</td>
</tr>
<tr>
<td>32G</td>
<td>0%</td>
</tr>
<tr>
<td>48G</td>
<td>0%</td>
</tr>
<tr>
<td>64G</td>
<td>0%</td>
</tr>
<tr>
<td>ONLINE: 80G</td>
<td>OFFLINE-AVAILABLE: 16G</td>
</tr>
<tr>
<td>0% IN USE</td>
<td></td>
</tr>
</tbody>
</table>

- Common misconception that SCM is there to absorb DUMPs
  - Paging to SCM is much faster than paging to AUX (DASD)
  - But it is most unlikely that, when a large DUMP is taken, it will be fast enough to avoid disruption on the LPAR and possible ‘sympathy sickness’ across the data sharing group

- SCM should be seen as an ‘insurance policy’ to protect the system against unintended paging in exceptional circumstances
Tools of choice
#1 tool of choice

- **IFCID 225**
  - Collected as part of DB2 Statistics Class 1
  - Do not just look at snapshots or Statistics reports
  - **Create graphs** from DB2 start to DB2 shutdown to study the evolutionary trend

- **Options to pull IFCID 225 data**
  - Performance database where Statistics data is kept at one-minute intervals
  - Sample REXX program MEMU2 to pull the data via IFI and create a .csv file
    - Available for download on IBM My developerWorks – New version supporting DB2 12!
      Make sure MEMU2 job runs with high dispatching priority otherwise system may stall on latch
  - OMPE Spreadsheet Input Data Generator to post process SMF data and create .csv files
    - APAR PM73732 and OMPE v510 PTF UK90267 / OMPE v511 PTF UK90268
    - Sample JCL provided in Appendix
Memory map

- **31-bit and 64-bit private memory for each DB2 system address space**
  - No sharing across DB2 system address spaces
  - Largest contributor since DB2 10: DBM1 64-bit private
    - Buffer pools, dynamic statement cache, skeleton pool, DBD pool, compression dictionaries
    - LOB and XML memory usage
- **64-bit shared memory (HVSHARE)**
  - Allows sharing across DB2 system address spaces and utilities (token required to connect)
    - No need to use expensive cross memory moves
    - Reduced complexity, addressable storage is always available
  - Use of 64-bit shared memory now dominates in DB2
    - Thread-related storage (system and non-system agents)
    - Shared stack (save areas, working storage)
- **31-bit and 64-bit common memory – CSA/ECSA and HCSA (HVCOMMON)**
  - Small use of CSA/ECSA (but if many DB2 subsystems per LPAR then aggregate requirement may be significant)
  - Limited use of 64-bit common in DB2 V10, but log buffers are moved there in V11 CM
Graph #1 – Real memory counters

DBM1 REAL in use for 31 and 64-bit priv
DBM1 REAL in use for 64-bit priv
DBM1 REAL in use for 64-bit priv w/o BP

DBM1 REAL in use for 31-bit priv = DBM1 REAL in use for 31 and 64-bit priv – DBM1 REAL in use for 64-bit priv
DBM1 REAL in use for BP = DBM1 REAL in use for 64-bit priv – DBM1 REAL in use for 64-bit priv w/o BP

DIST REAL in use for 64-bit priv
DIST REAL in use for 31 and 64-bit priv

DIST REAL in use for 31-bit priv = DIST REAL in use for 31 and 64-bit priv – DIST REAL in use for 64-bit priv

REAL in use for 64-bit shared
REAL in use for 64-bit shared stack
REAL in use for 64-bit common

See appendix for IFCID 225 field names and mapping reference table for MEMU2 and OMPE performance database
Graph #2 – AUX storage counters

- DBM1 AUX in use for 31 and 64-bit priv \(^{(1)}\) \(^{(2)}\)
- DBM1 AUX in use for 64-bit priv \(^{(1)}\) \(^{(2)}\)
- DBM1 AUX in use for 64-bit priv w/o BP

DBM1 AUX in use for 31-bit priv = DBM1 AUX in use for 31 and 64-bit priv – DBM1 AUX in use for 64-bit priv
DBM1 AUX in use for BP= DBM1 AUX in use for 64-bit priv – DBM1 AUX in use for 64-bit priv w/o BP

- DIST AUX in use for 64-bit priv \(^{(1)}\) \(^{(2)}\)
- DIST AUX in use for 31 and 64-bit priv \(^{(1)}\) \(^{(2)}\)

DIST AUX in use for 31-bit priv = DIST AUX in use for 31 and 64-bit priv – DIST AUX in use for 64-bit priv

- AUX in use for 64-bit shared
- AUX in use for 64-bit shared stack
- AUX in use for 64-bit common

**WARNING!**

\(^{(1)}\) There could be double accounting with REAL (if a frame was paged out and paged back in, it may show up both in REAL and AUX)

\(^{(2)}\) Does not include frames that are in SCM

See appendix for IFCID 225 field names and mapping reference table for MEMU2 and OMPE performance database
<table>
<thead>
<tr>
<th>Time (UTC)</th>
<th>Total AUX in use - DB2</th>
<th>ASID DBM1 AUX in use for 31-bit priv (MB)</th>
<th>ASID DBM1 AUX in use for 64-bit priv w/o BP (MB)</th>
<th>ASID DIST AUX in use for 31-bit priv (MB)</th>
<th>ASID DIST AUX in use for 64-bit priv (MB)</th>
<th>AUX in use for 64-bit shared (MB)</th>
<th>AUX in use for 64-bit common (MB)</th>
</tr>
</thead>
<tbody>
<tr>
<td>09:23:01</td>
<td>17</td>
<td>17</td>
<td>17</td>
<td>17</td>
<td>17</td>
<td>17</td>
<td>17</td>
</tr>
<tr>
<td>09:28:02</td>
<td>17</td>
<td>17</td>
<td>17</td>
<td>17</td>
<td>17</td>
<td>17</td>
<td>17</td>
</tr>
<tr>
<td>10:03:02</td>
<td>17</td>
<td>17</td>
<td>17</td>
<td>17</td>
<td>17</td>
<td>17</td>
<td>17</td>
</tr>
<tr>
<td>10:28:03</td>
<td>17</td>
<td>17</td>
<td>17</td>
<td>17</td>
<td>17</td>
<td>17</td>
<td>17</td>
</tr>
<tr>
<td>11:03:04</td>
<td>17</td>
<td>17</td>
<td>17</td>
<td>17</td>
<td>17</td>
<td>17</td>
<td>17</td>
</tr>
<tr>
<td>11:28:05</td>
<td>17</td>
<td>17</td>
<td>17</td>
<td>17</td>
<td>17</td>
<td>17</td>
<td>17</td>
</tr>
<tr>
<td>12:03:06</td>
<td>17</td>
<td>17</td>
<td>17</td>
<td>17</td>
<td>17</td>
<td>17</td>
<td>17</td>
</tr>
<tr>
<td>12:28:07</td>
<td>17</td>
<td>17</td>
<td>17</td>
<td>17</td>
<td>17</td>
<td>17</td>
<td>17</td>
</tr>
<tr>
<td>13:03:08</td>
<td>17</td>
<td>17</td>
<td>17</td>
<td>17</td>
<td>17</td>
<td>17</td>
<td>17</td>
</tr>
<tr>
<td>13:28:09</td>
<td>17</td>
<td>17</td>
<td>17</td>
<td>17</td>
<td>17</td>
<td>17</td>
<td>17</td>
</tr>
<tr>
<td>14:03:10</td>
<td>17</td>
<td>17</td>
<td>17</td>
<td>17</td>
<td>17</td>
<td>17</td>
<td>17</td>
</tr>
<tr>
<td>14:28:11</td>
<td>17</td>
<td>17</td>
<td>17</td>
<td>17</td>
<td>17</td>
<td>17</td>
<td>17</td>
</tr>
<tr>
<td>15:03:12</td>
<td>17</td>
<td>17</td>
<td>17</td>
<td>17</td>
<td>17</td>
<td>17</td>
<td>17</td>
</tr>
<tr>
<td>15:28:13</td>
<td>17</td>
<td>17</td>
<td>17</td>
<td>17</td>
<td>17</td>
<td>17</td>
<td>17</td>
</tr>
<tr>
<td>16:03:14</td>
<td>17</td>
<td>17</td>
<td>17</td>
<td>17</td>
<td>17</td>
<td>17</td>
<td>17</td>
</tr>
<tr>
<td>16:28:15</td>
<td>17</td>
<td>17</td>
<td>17</td>
<td>17</td>
<td>17</td>
<td>17</td>
<td>17</td>
</tr>
<tr>
<td>17:03:16</td>
<td>17</td>
<td>17</td>
<td>17</td>
<td>17</td>
<td>17</td>
<td>17</td>
<td>17</td>
</tr>
<tr>
<td>17:28:17</td>
<td>17</td>
<td>17</td>
<td>17</td>
<td>17</td>
<td>17</td>
<td>17</td>
<td>17</td>
</tr>
<tr>
<td>18:03:18</td>
<td>17</td>
<td>17</td>
<td>17</td>
<td>17</td>
<td>17</td>
<td>17</td>
<td>17</td>
</tr>
<tr>
<td>18:28:19</td>
<td>17</td>
<td>17</td>
<td>17</td>
<td>17</td>
<td>17</td>
<td>17</td>
<td>17</td>
</tr>
<tr>
<td>19:03:20</td>
<td>17</td>
<td>17</td>
<td>17</td>
<td>17</td>
<td>17</td>
<td>17</td>
<td>17</td>
</tr>
<tr>
<td>19:28:21</td>
<td>17</td>
<td>17</td>
<td>17</td>
<td>17</td>
<td>17</td>
<td>17</td>
<td>17</td>
</tr>
<tr>
<td>20:03:22</td>
<td>17</td>
<td>17</td>
<td>17</td>
<td>17</td>
<td>17</td>
<td>17</td>
<td>17</td>
</tr>
<tr>
<td>20:28:23</td>
<td>17</td>
<td>17</td>
<td>17</td>
<td>17</td>
<td>17</td>
<td>17</td>
<td>17</td>
</tr>
<tr>
<td>21:03:24</td>
<td>17</td>
<td>17</td>
<td>17</td>
<td>17</td>
<td>17</td>
<td>17</td>
<td>17</td>
</tr>
<tr>
<td>21:28:25</td>
<td>17</td>
<td>17</td>
<td>17</td>
<td>17</td>
<td>17</td>
<td>17</td>
<td>17</td>
</tr>
<tr>
<td>22:03:26</td>
<td>17</td>
<td>17</td>
<td>17</td>
<td>17</td>
<td>17</td>
<td>17</td>
<td>17</td>
</tr>
<tr>
<td>22:28:27</td>
<td>17</td>
<td>17</td>
<td>17</td>
<td>17</td>
<td>17</td>
<td>17</td>
<td>17</td>
</tr>
<tr>
<td>23:03:28</td>
<td>17</td>
<td>17</td>
<td>17</td>
<td>17</td>
<td>17</td>
<td>17</td>
<td>17</td>
</tr>
<tr>
<td>23:28:29</td>
<td>17</td>
<td>17</td>
<td>17</td>
<td>17</td>
<td>17</td>
<td>17</td>
<td>17</td>
</tr>
</tbody>
</table>
In this particular case, all the buffer pools were defined as PGFIX(YES), so it is the rest of DB2 that is paged out (thread control blocks, buffer pool control blocks, etc.)
Graph #3 – QW0225_REALAVAIL (S)
Graph #3 – QW0225_REALAVAIL (S) ...

REAL memory available on LPAR SYSP

GREEN condition
Graph #3 – QW0225_REALAVAIL (S) ...

REAL memory available on LPAR SYSR

AMBER condition (long drops and/or drops to very low values)
Next steps

- Find the time intervals when problematic long, deep drops or page-out occurred
- Use this information as a starting point to find the root cause of these drops
  - What type of activity drives sudden increase in memory consumption?
- Usual suspects
  - DFSORT e.g. DB2 utilities
  - DUMP capture
  - Recycling of CICS regions
  - Big batch job
  - Shift in workload
#2 tool of choice

- **RMF**
  - Monitor III – STORF (Option 3.7)
  - Storage usage by frames

Real memory consumption per jobname
→ **private** memory
#2 tool of choice

- **RMF**
  - Monitor III – STORM (Option 3.7A)
  - 64-Bit Shared and Common

64-bit shared and 64-bit common memory usage

Do not forget to include them in the overall real memory consumption!
Control use of real memory by DFSORT

- DFSORT default settings

**Before z/OS 2.1**

- EXPMAX=MAX ← Make maximum use of memory
- EXPOLD=MAX ← Allow paging of old frames
- EXPRES=0 ← No reserve for new work

Real memory allocated in fixed increment sizes

**After z/OS 2.1**

- EXPMAX=MAX
- EXPOLD=50%
- EXPRES=10%
- TUNE=STOR (new setting)

Gradual allocation of real memory in smaller increments

% for EXPMAX, EXPRES, EXPOLD = % of current resources instead of % of configured memory

- Recommendations to control use of real memory by DFSORT
  - Set EXPOLD=0 to prevent DFSORT from taking old frames from other workloads
  - Set EXPMAX down to limit maximum DFSORT usage
  - Set EXPRES=n {reserve enough for MAXSPACE}

Sample JCL to extract DFSORT installation defaults

```plaintext
//SHOWDEF EXEC PGM=ICETOOL
//TOOLMSG DD SYSOUT=*  //DFSMSG DD SYSOUT=*  //LIST1 DD SYSOUT=*  //TOOLIN DD *
DEFAULTS LIST(LIST1) /*
```
Control use of real memory by DB2

- Unless you have ample real memory headroom and can tolerate some memory growth, recommend running with REALSTORAGE_MANAGEMENT=AUTO (default)
  - With RSM=AUTO, DB2 will regularly discard unused real memory frames
    - RSM=AUTO with no paging (AUTO-OFF) → ‘Soft Discard’ at Thread Deallocation or after 120 commits
    - RSM=AUTO with paging (AUTO-ON) → ‘Soft Discard’ at Thread Deallocation or after 30 commits – STACK also DISCARDED
  - Pros of the ‘Soft Discard’
    - Reduced REAL memory use
    - Reduced exposure to SPIN lock contention
  - Cons:
    - Small CPU overhead in MSTR/DIST SRB time (which can be minimised with good thread reuse)
    - 64-bit shared and common real memory counters are not accurate until paging occurs
      - During ‘Soft Discard’, DB2 uses DISCARD with KEEPREAL=YES
      - The memory is only ‘virtually freed’
      - RSM flags the page as freed or unused, but the frame is still charged against DB2
Control use of real memory by DB2 ...

- **Use automation to trap ‘contraction mode’ messages → very severe condition**
  - When significant paging is detected (ENF 55 signal) or REALSTORAGE_MAX is reached, DB2 enters ‘contraction mode’ (‘Hard Discard’)
    - Applies to REALSTORAGE_MANAGEMENT=AUTO|ON|OFF
    - During ‘Hard Discard’, DB2 uses DISCARD with KEEPREAL=NO
    - KEEPREAL(NO) tells RSM to free and reclaim the page immediately

```
DSNV516I  -TZE1 DSNVMON - BEGINNING STORAGE CONTRACTION MODE
DSNV517I  -TZE1 DSNVMON - ENDING STORAGE CONTRACTION MODE
```
Control use of real memory by DB2 ...

- Refrain from using PGFIX(YES) if you don’t have enough real memory headroom

- If buffer pools are defined with PGFIX(YES), consider specifying z/OS WLM STORAGE CRITICAL for the DB2 system address spaces
  - Long-term storage protection for the DB2 address spaces
  - Objective: Help safeguard the rest of DB2
    - Try to keep thread control blocks, EDM and other needed parts of DB2 in real memory
    - Might help prevent performance problems as the Online day starts and DB2 has to be rapidly paged back in
    - But not 100% guarantee that DB2 will not be paged out!

- If you have multiple DB2 subsystems with differing importance running on the LPAR, restrict the use of PGFIX(YES) and WLM STORAGE CRITICAL to the most important one
Control use of real memory by DB2 ...

- **Review your DB2 memory budget**
  - Settings that can have a direct impact on real memory usage if they need to be backed
    - Buffer pool sizes (VPSIZE)
    - Global dynamic statement cache (ZPARM EDMSTMTC)
    - Local dynamic statement cache (ZPARM MAXKEEPD)
    - DBD pool size (ZPARM EDMDBDC)
    - Skeleton pool (ZPARM EDM_SKELETON_POOL)
    - RID pool (ZPARM MAXRBLK)
    - Sort pool (ZPARM SRTPOOL – per thread setting)
    - Log output log buffer (ZPARM OUTBUFF)
    - Hashing entries in dynamic statement cache (ZPARM CACHE_DEP_TRACK_STOR_LIM) → Default 2GB, SAP recommends 10GB
    - LOB value storage (ZPARM LOBVALS) → Default=2GB
    - XML value storage (ZPARM XMLVALS) → Default=10GB
  - Indirectly ZPARMs CTHREAD, MAXDBAT, DSMAX will also affect the overall budget
QW0225_REALAVAIL (S)

REAL memory available on LPAR SYSQ

MAXSPACE
Next steps

- Buy more memory!
Large size real memory page frames

• **Benefit of large size (1M or 2G) real memory page frames**
  • The Translation Lookaside Buffer (TLB) is a cache used to speed up the conversion of virtual addresses into real addresses
  • With the introduction of 64-bit real and virtual addressing, the TLB coverage has dramatically shrunk, leading to performance degradation
  • Large size page frames help increase TLB coverage without having to enlarge the TLB size
  • Result: Better performance by decreasing the number of TLB misses

• **Common problem**
  • LFAREA is grossly over-configured, which might result in a shortage of 4K size frames and lead to expensive breakdown of 1M size large frames, expensive page movement for 4K page fixes, and/or premature paging
Large size real memory page frames ...

- **LFAREA – 1M/2G large frame area**
  - Fixed 1M/2G page frames
  - Defined in IEASYSxx parmlib member
  - ‘Old’ syntax (still supported)
    - LFAREA = (xM | xG | xT | x%)
    - Pct formula: (x% of online memory available at IPL) – 2G
    - Max LFAREA is (80% * online real memory available at IPL) – 2G
  - New syntax
    - LFAREA = (1M=(a [,b]) | 1M=(a% [,b%]) | 2G=(a [,b]) | 2G=(a% [,b%])
    - Pct formula: x% of (online memory available at IPL – 4G)
    - Max LFAREA is 80% of (online memory available at IPL time – 4G)
  - Only changeable by IPL
  - If the LFAREA is overcommitted, DB2 will use 4K and/or 1M size page frames
Large size real memory page frames ...

- **Quad area**
  - 12.5% of online memory at IPL time

- **PLArea – Pageable 1M large frame area**
  - Pageable 1M page frames
  - Allocated on SCM-capable machines (zEC12/zBC12 and above)
    - If Flash Express is installed, these large pages may be paged to and from SCM
    - If Flash Express is not installed, then if those pages are ever paged out, they will be demoted to 4K size page frames and will remain 4K size until the next IPL
  - System-defined size
    - Approximately 12.5% of online memory at IPL time – adjusted to what fits after Quad and LFArea are built
  - Pageable 1M frames overflow into the LFAREA when PLArea is depleted
Large size real memory page frames ...

- **Let's do some maths...**
  - **Starting position**
    
    | Component          | Value (GB) |
    |--------------------|------------|
    | Online memory      | 50.0       |
    | LFAREA             | 0          |
    | QUAD               | 6.3        |
    | 1MB PAGEABLE       | 6.3        |
    | 4KB FRAMES         | 37.4       |

- **If you were to add 100GB to the LPAR and define it all as LFAREA**
  
<table>
<thead>
<tr>
<th>Component</th>
<th>Value (GB)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Online memory</td>
<td>150.0</td>
</tr>
<tr>
<td>LFAREA</td>
<td>100.0</td>
</tr>
<tr>
<td>QUAD</td>
<td>18.8</td>
</tr>
<tr>
<td>1MB PAGEABLE</td>
<td>18.8</td>
</tr>
<tr>
<td>4KB FRAMES</td>
<td>12.4</td>
</tr>
</tbody>
</table>

  Probably not enough 4K frames to handle the 4K workload needs, including taking dumps quickly, without having to break down free 1M frames.

---

**Do not forget that Quad area and Pageable 1M area grow proportionally with additional REAL memory!**
Large size real memory page frames ...

- **DB2 can exploit fixed 1M and 2G page frames for page-fixed buffer pools**
  - Requirements
    - Buffer pools must be defined as PGFIX=YES
    - 1M size page frames requires DB2 10, z10 and above
    - 2G size page frames requires DB2 11, zEC12 and above
  - DB2 11 supports FRAMESIZE parameter (4K, 1M, 2G) at the BP level for flexibility
- **DB2 11 can also exploit fixed 1M page frames for the log output buffer (OUTBUFF)**
- **DB2 can exploit pageable 1M frames for buffer pools control blocks (PMBs)**
  - PMBs consume only 4-5% of buffer pool allocations but they are referenced very heavily
  - This is where customers get the most benefit
  - Requirements
    - DB2 10 with APAR PM85944 or DB2 11
    - Buffer pools can be defined as either PGFIX=YES or PGFIX=NO, and can be backed by either 4K, 1M or 2G size page frames
Large size real memory page frames ...

- **Estimating ‘optimal’ LFAREA**
  - Total of
    - (Sum of VPSIZE*page size from candidate local buffer pools) * 1.05
    - Plus 20MB for z/OS usage
    - Plus log output buffer size (OUTBUFF) if running DB2 11
    - Plus non-DB2 usage e.g., Java heap sizes
    - Plus any over flow from PLArea (Pageable Large Area)

- **Recommendation**
  - Define LFAREA based on what you can afford after considering total real memory demands for 4K frames above 2G
    - Must consider operating system memory needs
      - RSM requirement for memory mapping (approximately 1/64 total online real memory at IPL)
    - System address spaces memory usage (DB2, CICS, etc.)
    - Must also include enough spare 4K size frames for taking dumps quickly
What happens in case of shortage of 4K frames?

- **2 types of 4K requests**
  - **Preferred requests**
    - Used by non-swapable address spaces, can be fixed long term → DB2 uses preferred requests
  - **Non-Preferred requests**
    - Used by swapable address spaces, used short term and not fixed

- **Pageable 1M large frame area**
  - Pageable 1M frames can be broken down to satisfy demand for 4K preferred or non-preferred requests

- **Quad area**
  - Quad frames can be broken down to satisfy demand for 4K non-preferred requests only
  - Breaking news for z/OS 2.2 only!
    - Several customers have reported excessive paging after migration to z/OS 2.2
    - Analysis showed an abundance of available Quad frames that are not used by RSM
    - Fixed by APAR OA50945 (HIPER)
What happens in case of shortage of 4K frames? ...

- **LFAREA and INCLUDEMAFC option**
  - Available frame count (AFC) is used to determine when storage management should begin paging frames
  - Pre-z/OS V2R2:
    - By default, AFC does not include LFAREA 1M pages
      - 1M fixed frames are preserved for fixed 1M requests
      - In case of shortage of 4K frames, paging can occur even with an abundance of available 1M fixed frames
    - INCLUDE1MAFC on LFAREA parameter means AFC includes the LFAREA 1M pages
      - New keyword added by APAR OA41968
      - 1M fixed frames can be broken down to satisfy demand for 4K non-preferred requests only
      - Paging is delayed
  - z/OS V2R2 → Two options when defining the LFAREA: INCLUDEMAFC(YES|NO)
    - YES is the default
    - Recommendation is to use INCLUDEMAFC(YES) so that unused 1M pages can be broken down to satisfy demand for 4K non-preferred requests
Large size real memory page frames – Monitoring

- **Useful commands**
  - Find out how many real memory page frames of each size are being used
    - Especially useful when running multiple DB2 subsystems on the same LPAR
  - **DB2 10** - `DISPLAY BUFFERPOOL(BPx) SERVICE(4)`
    - See DSNB999I message
  - **DB2 11** - `DISPLAY BUFFERPOOL(BPx) DETAIL(*)`
    - See DSNB546I messages

```
DSNB999I +D2V1 DSNB1DBP SERVICE( 4 ) OUTPUT
DSNB999I +D2V1 4K PAGES 0
DSNB999I +D2V1 1M PAGES 1476

DSNB546I +PDJ1 PREFERRED FRAME SIZE 1M
793600 BUFFERS USING 1M FRAME SIZE ALLOCATED
DSNB546I +PDJ1 PREFERRED FRAME SIZE 1M
2956400 BUFFERS USING 4K FRAME SIZE ALLOCATED
```
Large size real memory page frames – Monitoring ...

- **Useful commands ...**
  - MVS DISPLAY VIRTSTOR,LFAREA
    - Show total LFAREA, allocation split across different size page frames, what is available
    - See IAR019I message

```
IAR019I  10.48.56 DISPLAY VIRTSTOR
SOURCE =  C2
TOTAL LFAREA = 14336M , 0G
LFAREA AVAILABLE = 0M , 0G
LFAREA ALLOCATED (1M) = 14329M
LFAREA ALLOCATED (4K) = 0M
MAX LFAREA ALLOCATED (1M) = 14329M
MAX LFAREA ALLOCATED (4K) = 0M
LFAREA ALLOCATED (PAGEABLE1M) = 7M
MAX LFAREA ALLOCATED (PAGEABLE1M) = 7M
LFAREA ALLOCATED NUMBER OF 2G PAGES = 0
MAX LFAREA ALLOCATED NUMBER OF 2G PAGES = 0
```

Undersized LFAREA on resulting in (small) lost CPU savings

```
IAR019I  10.45.57 DISPLAY VIRTSTOR
SOURCE =  00
TOTAL LFAREA = 7782M , 0G
LFAREA AVAILABLE = 2025M , 0G
LFAREA ALLOCATED (1M) = 5377M
LFAREA ALLOCATED (4K) = 374M
MAX LFAREA ALLOCATED (1M) = 5377M
MAX LFAREA ALLOCATED (4K) = 526M
LFAREA ALLOCATED (PAGEABLE1M) = 6M
MAX LFAREA ALLOCATED (PAGEABLE1M) = 6M
LFAREA ALLOCATED NUMBER OF 2G PAGES = 0
MAX LFAREA ALLOCATED NUMBER OF 2G PAGES = 0
```

Oversized LFAREA resulting in a shortage of 4K size frames and expensive breakdown of 1M size large frames – may also lead to premature paging if INCLUDE1MAFC=NO
### Monitoring REAL/AUX memory usage – Mapping for reference

<table>
<thead>
<tr>
<th>IFCID FIELD</th>
<th>PM FIELD</th>
<th>PDB COLUMN NAME</th>
<th>MEMU2 Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>QW0225RL</td>
<td>QW0225RL</td>
<td>REAL_STORAGE_FRAME</td>
<td>DBM1 REAL in use for 31 and 64-bit priv (MB)</td>
</tr>
<tr>
<td>QW0225AX</td>
<td>QW0225AX</td>
<td>AUX_STORAGE_SLOT</td>
<td>DBM1 AUX in use for 31 and 64-bit priv (MB)</td>
</tr>
<tr>
<td>QW0225HVPagesInReal</td>
<td>SW225VPR</td>
<td>A2GB_REAL_FRAME</td>
<td>DBM1 REAL in use for 64-bit priv (MB)</td>
</tr>
<tr>
<td>QW0225HVAuxSlots</td>
<td>SW225VAS</td>
<td>A2GB_AUX SLOT</td>
<td>DBM1 AUX in use for 64-bit priv (MB)</td>
</tr>
<tr>
<td>QW0225PriStg_Real</td>
<td>SW225PSR</td>
<td>A2GB_REAL_FRAME_TS</td>
<td>DBM1 REAL in use for 64-bit priv w/o BP (MB)</td>
</tr>
<tr>
<td>QW0225PriStg_Aux</td>
<td>SW225PSA</td>
<td>A2GB_AUX SLOT_TS</td>
<td>DBM1 AUX in use for 64-bit priv w/o BP (MB)</td>
</tr>
<tr>
<td>QW0225RL</td>
<td>QW0225RL</td>
<td>DIST_REAL_FRAME</td>
<td>DIST REAL in use for 31 and 64-bit priv (MB)</td>
</tr>
<tr>
<td>QW0225AX</td>
<td>QW0225AX</td>
<td>DIST_AUX SLOT</td>
<td>DIST AUX in use for 31 and 64-bit priv (MB)</td>
</tr>
<tr>
<td>QW0225HVPagesInReal</td>
<td>SW225VPR</td>
<td>A2GB_DIST_REAL_FRM</td>
<td>DIST REAL in use for 64-bit priv (MB)</td>
</tr>
<tr>
<td>QW0225HVAuxSlots</td>
<td>SW225VAS</td>
<td>A2GB_DIST_AUX_SLOT</td>
<td>DIST AUX in use for 64-bit priv (MB)</td>
</tr>
<tr>
<td>QW0225ShrStg_Real</td>
<td>SW225SSR</td>
<td>A2GB_SHR_REALF_TS</td>
<td>REAL in use for 64-bit shared (MB)</td>
</tr>
<tr>
<td>QW0225ShrStg_Aux</td>
<td>SW225SSA</td>
<td>A2GB_SHR_AUXS_TS</td>
<td>AUX in use for 64-bit shared (MB)</td>
</tr>
<tr>
<td>QW0225ShrStkStg_Real</td>
<td>SW225KSR</td>
<td>A2GB_SHR_REALF_STK</td>
<td>REAL in use for 64-bit shared stack (MB)</td>
</tr>
<tr>
<td>QW0225ShrStkStg_Aux</td>
<td>SW225KSA</td>
<td>A2GB_SHR_AUXS_STK</td>
<td>AUX in use for 64-bit shared stack (MB)</td>
</tr>
<tr>
<td>QW0225ComStg_Real</td>
<td>SW225CSR</td>
<td>A2GB_COMMON_REALF</td>
<td>REAL in use for 64-bit common (MB)</td>
</tr>
<tr>
<td>QW0225ComStg_Aux</td>
<td>SW225CSA</td>
<td>A2GB_COMMON_AUXS</td>
<td>AUX in use for 64-bit common (MB)</td>
</tr>
<tr>
<td>QW0225_REALAVAIL</td>
<td>S225RLAV</td>
<td>QW0225_REALAVAIL</td>
<td>REALAVAIL (MB) (S)</td>
</tr>
</tbody>
</table>

Note: All REAL/AUX storage fields in IFCID 225 and OMPE performance database are expressed in 4KB frames or slots – they should be converted to MB (conversion is already done in MEMU2)
Monitoring REAL/AUX memory usage – Based on OMPE PDB

- **Stacked AREA graph** - one for each DB2 member (one sheet per DB2 member)

  - \[(\text{REAL} \_ \text{STORAGE} \_ \text{FRAME} - \text{A2GB} \_ \text{REAL} \_ \text{FRAME}) \times 4/1024\]
  - \[(\text{A2GB} \_ \text{REAL} \_ \text{FRAME} - \text{A2GB} \_ \text{REAL} \_ \text{FRAME} \_ \text{TS}) \times 4/1024\]
  - \[\text{A2GB} \_ \text{REAL} \_ \text{FRAME} \_ \text{TS} \times 4/1024\]
  - \[(\text{DIST} \_ \text{REAL} \_ \text{FRAME} - \text{A2GB} \_ \text{DIST} \_ \text{REAL} \_ \text{FRM}) \times 4/1024\]
  - \[\text{A2GB} \_ \text{DIST} \_ \text{REAL} \_ \text{FRM} \times 4/1024\]
  - \[\text{A2GB} \_ \text{COMMON} \_ \text{REAL} \_ \text{ALF}\times 4/1024\]
  - \[\text{A2GB} \_ \text{SHR} \_ \text{REAL} \_ \text{ALF} \_ \text{TS}\times 4/1024\]
  - \[\text{A2GB} \_ \text{SHR} \_ \text{REAL} \_ \text{ALF} \_ \text{STK}\times 4/1024\]

  - AS DBM1_REAL_PRIV_31BIT_MB
  - AS DBM1_REAL_PRIV_64BIT_BP_MB
  - AS DBM1_REAL_PRIV_64BIT_XBP_MB
  - AS DIST_REAL_PRIV_31BIT_MB
  - AS DIST_REAL_PRIV_64BIT_MB

- **Stacked AREA graph** - one for each DB2 member (one sheet per member)

  - \[(\text{AUX} \_ \text{STORAGE} \_ \text{SLOT} - \text{A2GB} \_ \text{AUX} \_ \text{SLOT}) \times 4/1024\]
  - \[(\text{A2GB} \_ \text{AUX} \_ \text{SLOT} - \text{A2GB} \_ \text{AUX} \_ \text{SLOT} \_ \text{TS}) \times 4/1024\]
  - \[\text{A2GB} \_ \text{AUX} \_ \text{SLOT} \_ \text{TS} \times 4/1024\]
  - \[(\text{DIST} \_ \text{AUX} \_ \text{SLOT} - \text{A2GB} \_ \text{DIST} \_ \text{AUX} \_ \text{SLOT}) \times 4/1024\]
  - \[\text{A2GB} \_ \text{DIST} \_ \text{AUX} \_ \text{SLOT} \times 4/1024\]
  - \[\text{A2GB} \_ \text{COMMON} \_ \text{AUX} \_ \text{SLOTS}\times 4/1024\]
  - \[\text{A2GB} \_ \text{SHR} \_ \text{AUX} \_ \text{SLOTS} \_ \text{TS}\times 4/1024\]
  - \[\text{A2GB} \_ \text{SHR} \_ \text{AUX} \_ \text{SLOTS} \_ \text{STK}\times 4/1024\]

  - AS DBM1_AUX_PRIV_31BIT_MB
  - AS DBM1_AUX_PRIV_64BIT_BP MB
  - AS DBM1_AUX_PRIV_64BIT_XBP MB
  - AS DIST_AUX_PRIV_31BIT MB
  - AS DIST_AUX_PRIV_64BIT_MB

- **Line graph** - one for each LPAR

  - \[\text{QW0225} \_ \text{REALAVAIL} \times 4/1024\]

  - AS REAL_AVAIL_LPAR_MB
OMPE spreadsheet generator

Sample JCL

//******************************************
//* JOBCARD
//******************************************
//JOBLIB DD DISP=SHR, DSN=<DB2OMPE>.TKANMOD
/*
//STATFIL EXEC PGM=FPECMAIN
//INPUTDD DD DISP=SHR, DSN=<USER_HLQ>.SMF100
//SYSPRINT DD SYSOUT=* 
//SYSOUT DD SYSOUT=* 
//JOBSUMDD DD SYSOUT=* 
//STFILDD DD DISP=(NEW, PASS), UNIT=SYSDA, SPACE=(CYL, (200, 100))
//SYSIN DD *
GLOBAL
INCLUDE (SUBSYSTEMID(DB2A))
STATISTICS
FILE DDNAME(STFILDD)
EXEC
/*
//FPEPCSV EXEC PGM=FPEPCSV,
//PARM='SSTG Y Y N , . <DB2OMPE>.TKO2SAMP'
//SYSPRINT DD SYSOUT=* 
//FLDSEL DD DISP=SHR, DSN=<USER_HLQ>.FPECSVU(CSVSTO11)
//INPUT DD DISP=(OLD, DELETE), DSN=* . STATFIL.STFILDD 
//OUTPUT DD DSN=<USER_HLQ>.MEMU2.CSV,
//DISP=(NEW, CATLG, DELETE),
//UNIT=SYSDA, SPACE=(CYL, (200, 100), RLSE),
//DCB=(RECFM=VB, LRECL=32756, BLKSIZE=32760) 
//*
International DB2 Users Group

Sponsored by IBM

28th March 2017 - Chennai

Session 9007
Starter's guide to REAL memory management with DB2 for z/OS

John Campbell
IBM DB2 for z/OS Development
campbelj@uk.ibm.com
@gurudb2