Auto Alter – for DB2 Couch Potatoes

Judy Ruby-Brown
IBM Corp.
zGrowth Washington Systems Center

Monday, March 2, 2015
Session Number 16547

#SHAREorg

SHARE is an independent volunteer-run information technology association that provides education, professional networking and industry influence.

SHARE in Seattle 2015

Acknowledgements and Disclaimers

Availability. References in this presentation to IBM products, programs, or services do not imply that they will be available in all countries in which IBM operates.

The workshops, sessions and materials have been prepared by IBM or the session speakers and reflect their own views. They are provided for informational purposes only, and are neither intended to, nor shall have the effect of being, legal or other guidance or advice to any participant. While efforts were made to verify the completeness and accuracy of the information contained in this presentation, it is provided AS-IS without warranty of any kind, express or implied. IBM shall not be responsible for any damages arising out of the use of, or otherwise related to, this presentation or any other materials. Nothing contained in this presentation is intended to, nor shall have the effect of, creating any warranties or representations from IBM or its suppliers or licensors, or altering the terms and conditions of the applicable license agreement governing the use of IBM software.

All customer examples described are presented as illustrations of how those customers have used IBM products and the results they may have achieved. Actual environmental costs and performance characteristics may vary by customer. Nothing contained in these materials is intended to, nor shall have the effect of, stating or implying that any activities undertaken by you will result in any specific sales, revenue growth or other results.

IBM, the IBM logo, ibm.com, DB2, and FlashCopy are trademarks or registered trademarks of International Business Machines Corporation in the United States, other countries, or both. If these and other IBM trademarked terms are marked on their first occurrence in this information with a trademark symbol (® or TM), these symbols indicate U.S. registered or common law trademarks owned by IBM at the time this information was published. Such trademarks may also be registered or common law trademarks in other countries. A current list of IBM trademarks is available on the Web at

- "Copyright and trademark information" at www.ibm.com/legal/copytrade.shtml
- Other company, product, or service names may be trademarks or service marks of others.

Is this Presentation right for you?

- You are familiar with 3 types of DB2 Data Sharing Structures
  - Lock, SCA, Group Buffer Pools (GBP)
  - They enable essentially ONE DB2 subsystem to run across many LPARs
    - with integrity (locks, exception conditions)
    - and coherency (you always get the most current copy of an updated page, no matter which DB2 member updated it)

- You know that Coupling Facility (CF) GBP structures contain DB2 pages and pointers to members of the data sharing group whose local buffer pools hold them.
  - Pages are z/OS “elements” (4K) - Directory entries are “entries” (≈250 bytes)
  - DB2 needs some ratio of directory entries to pages – default is 5:1 but is that right?

- You wonder how to arrive at the correct number of directory entries

- Notes: Slides with a title of “Notes” are for your reference and are not shown in this presentation

How do you tune your DB2 GBPs?

- Deafening Silence….
- Correct answer: NO ONE DOES

- Why?
- Answer is political, not technical
  - **z/OS "owns" CFRM policy, RMF CF Reports, and CF Structures**
  - z/OS doesn't "do" DB2
- So let's do something "autonomic" (from management)

- This presentation helps you to have a foot in each camp

Basic DB2 GBP architecture

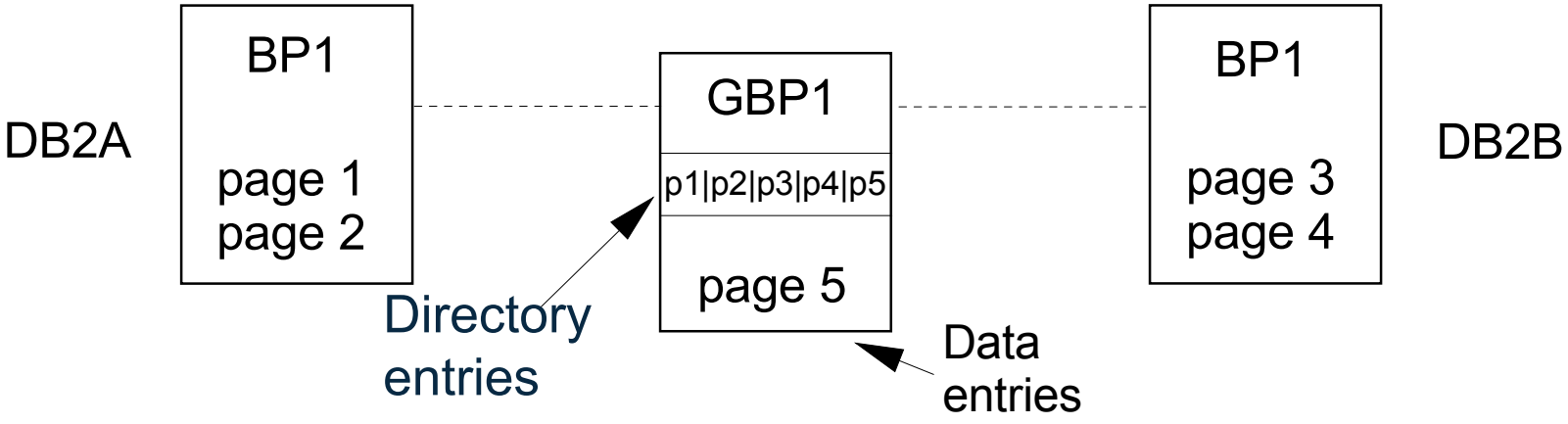
- DB2 looks for a page..
  1. In local buffer pool (nanosecond access)
  2. In group buffer pool (microsecond access)
  3. On DASD (millisecond access)

- Is the page in the local buffer pool (BP)?
  - If yes, is it valid?
    - YES, bingo!
    - NO, look in the GBP and if not there, then do a sync read from DASD

- When a page is updated in BP, it is written at commit to the GBP.
  - All members that have that page receive invalidate (XI) signal
  - Buffer with the committed page can now be used for another page
    - Eventually page in the GBP is written to DASD via the DB2 Castout

Group Buffer Pool (GBP) Sizing

- **Objective: Size GBP to avoid directory entry reclaims**
  - Because GBP directory entry reclaims can cause page invalidation, which can lead to re-reads of clean (unchanged) pages from DASD
- **Key to Success: A directory entry for every local buffer and GBP data entry (a pointer to every cached page)**



Notes

- Each page in a local BP has a directory entry….and so does a page in the GBP!
- Note that page 5 in the GBP does not reside in either DB2A or DB2B, but it also must have a directory entry.
- In this example each page is unique in the data sharing group. This represents the “worst case” situation.

Two problems to avoid..

Using DB2 –DIS GBPOOL(*) TYPE(GCONN) GDETAIL(*)

1. *Performance*: "XIs due to Directory Reclaims" – should be 0!
   - Read page from DASD to local BP -Register it in GBP
   - If too few directory entries, more CPU and I/O overhead:
     - Clean page invalidated when entry stolen before use
     - DB2 wants page but invalid in local BP
       - » Causes 2 CF accesses and 1 sync DASD page read
   - Waste of local BP space!
     - Invalidated data means that these BP pages did not exist at this time
     - To use bigger BPs effectively, you MUST have more directory entries in the GBP structure.

2. *Availability*: "Writes failed due to lack of storage" – goal is 0!
   - Potential outage if pages added to LPL even though DB2 removes them automatically

Notes

- When an XI for directory entry reclaim occurs, IF and WHEN the page is subsequently referenced, additional activity (e.g. DASD read and write/re-register in CF) incurred that would not otherwise have had to occur. This is a **performance issue**
  - This field is also found in IFCID 230 (QBGBRXI) – GBP Attributes

- Writes failed – DB2 tries several times to write page but if unsuccessful, page added to LPL and **lack of availability** until it is recovered from LPL (automatically by DB2)

GBP Sizing

- Goal: Safety – “Don’t shoot yourself in the foot”

CFSizer - http://www.ibm.com/systems/support/z/cfsizer/

CFSizer connects to live server for calculation of latest CFCC

- You input the sum of local buffers for all members and page size
  - i.e. 2 members with 200,000 buffers for BP2 = 400,000 buffers
  - Specify “heavy” vs.. “typical” data sharing in CFSizer for safety
- Take the number it outputs - round to even number for INITSIZE
- Double it for SIZE, and enable Auto Alter (strong recommendation)
- Use frequent –DIS GBPOOL(*) TYPE(GCONN) GDETAIL(*) to monitor for the allocated size=SIZE

- After you have been in data sharing, use current values to resize .

Notes

This slide is put in for users who haven't implemented data sharing yet and want to know how to size them initially. This presentation assumes that the user is already data sharing and thus, we work with the sizes we are given. But it is handy to know how to size, as a sanity check in case numbers are way off, as a future slide will show.

The key to avoiding GBP directory entry reclaims is to have a directory entry for every different page that could be cached in a local buffer pool or in the associated group buffer pool. Generally speaking, this means having a directory entry for every "slot" (local buffer pool buffer or GBP data entry) into which a page can be read.

Formulae in *Data Sharing: Planning and Administration* are more difficult. CFSizer is good, easy to use and did not exist when data sharing was introduced. It does not guarantee no XIs due to directory reclaims. That is why Auto Alter is recommended.

Experiences with very large customers

- Millions “XI due to directory reclaims”
  - Wastes local BP storage in each member
- Failure to notice – 2 days results

  -DIS GBPOOL(*) GDETAIL(*) TYPE(GCONN) command

- Note: This example is not current.
- In 2015 they are usually much larger, in hundreds of MB or in GB, but this illustrates the extreme condition of XIs.

	Allocated Size (KB)	SIZE (KB)	Xis due to directory Reclaims
GBP0	79,192	79,192	12,243
GBP1	1,280	1,280	150
GBP2	-	-	-
GBP3	144,128	144,128	1.9M
GBP4	179,200	179,200	39.8M
GBP5	247,808	247,808	464K
GBP6	36,352	36,362	**144.6M**
GBP7	76,800	76,800	36.8M
GBP8	36,352	36,352	**173.2M**
GBP9	9,472	9,472	-
GBP16	8,704	8,704	542,463
GBP17	13,824	13,824	18.7M
GBP18	7,168	7,168	2.4M
GBP19	29,184	29,184	**91.9M**
GBP20	21,760	21,760	**107.4M**
Totals	891,224	891,234	

And “Write failures due to lack of storage”

	Allocated Size (KB)	SIZE (KB)	Dir to Data Ratio	XIs due to directory Reclaims	Writes Failed - Storage	GBP Check point time
GBP0	150,271	400,000	4	-	-	4
GBP2	**8,785,408**	12,000,000	1	-	5	4
GBP3	2,178,816	3,600,000	1	-	-	4
GBP4	**2,000,128**	2,000,000	1	47,418	27	4
GBP5	275,200	550,000	5	-	-	4
GBP32K	100,096	200,000	8	-	-	4
Totals	13,489,919	18,750,000				

Notes

- Observe that GBP2 is extremely large – currently 8.7GB with a SIZE of 12 GB, yet it still had experienced a few write failures.
- GBP8 has many XIs due to directory reclaims, in addition to write failures. This is indicative of a too small GBP. Its current allocated size is the same as its maximum size limit (SIZE).
- Both GBPs show a directory to data ratio of 1. This occurs when a GBP hits SIZE at some point in time. When there is no more space that can be allocated in a GBP and more pages are needed, the only alteration that can occur is to reduce the number of directory entries in favor of data pages. The lowest ratio possible is 1:1.

Goal: Pro-active tuning

- Allocate large DB2 GBPs (initial size) to handle growth for
  - Increases in local BPs
  - Additional data sharing members

- Let Auto Alter tune the directory to data ratio
- Later (6 months? 1 year?) DB2 and z/OS staff get up off the couch and update if needed

- Rest of this section tells how…

What is Auto Alter?

- Autonomic effort by XES to avoid filling up any kind of structure
  - If all data elements (pages) are changed, writes cannot occur
  - If all directory elements are marked changed, new pages cannot be registered
- Auto Alter has algorithms that
  - can increase or decrease number of entries and/or elements to avoid structure full conditions
  - can increase or decrease the size of the structure
- Can alter dynamically the precise directory to data ratio for GBPs

- **Design point is for gradual growth, not spikes**

How to Specify Auto Alter

- You implement through **STRUCTURE** statement in CFRM policy
  - Specify **ALLOWAUTOALT(YES)** – permit the altering of this structure
  - Specify **FULLTHRESHOLD** (if you don't, 80% is the default) – when either entries or elements exceed this threshold
  - Specify **MINSIZE** (if you don't, 75% of INITSIZE is default) – XES can not alter structure lower than this size
- Example
  - STRUCTURE NAME(DB2GR0B_GBP3)
    - INITSIZE=100M
    - SIZE=200M
    - PREF=(CF2,CF1)
    - FULLTHRESHOLD=80
    - MINSIZE=100M
    - **ALLOWAUTOALT(YES)**
    - DUPLEX(ENABLED)
    - REBUILDPERCENT(1)

DB2 Structures support Auto Alter

- LOCK1 – effective on Modify Lock List (a.k.a. Record List Entries)
  - Lock Table Entries cannot be changed without a rebuild
- SCA – can be increased

- Main value is for Group Buffer Pools (GBPs). Why?
  - People don't tune GBPs
    - Organizational Division of labor
      - DB2 DBAs responsible for local BPs – forget about GBPs
      - z/OS responsible for GBPs – and they own the CFRM Policy
  - DB2 needs ?? more directory entries than data page elements
  - Each –ALTER to change directory entries means manual GBP rebuild
- Works for duplexed GBPs

Structure Full Algorithm

- Auto Alter has two algorithms
  1. Structure Full avoidance
     - Page 13
  2. (Directory/entry) reclaim avoidance

     Subordinate to 1
     - Page 12

- Structure Full avoidance uses Full Structure Monitoring statistics to monitor both **changed** entries and elements:
  - If either one exceeds FULLTHRESHOLD, XES views impending catastrophe to avoid if at all possible

✓ If either one or the other exceeds FULLTHRESHOLD, XES increases size slightly (about 10%) and also “juggles” entries and elements
  - increasing one and decreasing the other

Reclaim Avoidance

- Uses statistics to determine if (directory) entry reclaim is occurring
  - For any reason (changed or unchanged)
  - Structure Full is interested only in CHANGED entries
- When 95% entries are in use, XES increases the number of directory entries while decreasing the number of (data page) elements
  - Up to 255:1 ratio (z/OS 1.13 – before that 40:1 was limit)

Structure Size Manipulation Summary

- If either changed entries or elements exceed FULLTHRESHOLD, XES will increase the structure size to gain enough capacity
- Auto Alter can decrease a structure size
  - Only if CF itself is under stress (rarely) (<10% free storage)
  - It will “steal” storage from all structures with ALLOWAUTOALT (YES)
    - But will not reduce them to less than MINSIZE value
- **Reclaim Avoidance does not change the size of a structure**
  - But increases (directory) entries and reduces (data page) elements

DB2's mechanisms to avoid Structure Full

- With large GBPs, lots of shared data sets, fast processors, tune aggressively
  - Large (>1G)
  - Pretty large (> 500M)
- Default FULLTHRESHOLD to 80%
  - Dribble Castout to avoid hitting thresholds

Threshold	Recommended
GBP Checkpoint	<=2-4 minutes
CLASST (~ VDWQT)	1-5%
CLASST – DB2 11	Can be <1%
GBPOOLT (~ DWQT)	10-30%

DB2 11 CLASST More Granularity

- CLASST (class-threshold1,class-threshold2)
- Class-threshold1- percentage of the number of data entries as used in previous versions of DB2
  - between 0 and 90, default is 5 (that is, 5%)
  - Example: CLASST(3,0)
    - castout when 3% of the GBP pages in castout class are updated
- Class-threshold2 is an absolute number of pages.
  - Useful for low threshold value for a large GBP and class-threshold1 is not granular enough between 0 and 1.
    - between 0 and 32767, default is 0.
  - Example: CLASST(0,2500)
    - starts castout when the number of pages in class reaches 2500.
  - Used only when class-threshold1 is 0.
    - If the value of class-threshold1 is non-zero, class-threshold2 is ignored.
    - If both class-threshold1 and class-threshold2 are 0, the class-threshold1 value of 0 is used.

What about CLASST = 0 or (0,0)?

- Premise: match CLASST to VDWQT per *Data Sharing Guide*
- Seen a lot of 0% - What does it mean?
  - When 32 pages in castout class are updated, DB2 casts out
  - **Except** – If 1% of #pages in GBP is < 32 pages
    - Example: if CLASST=0 and #pages in GBP=1400
    - **Castout occurs at 14 pages since 1% is < 32 pages**
  - Note: “number of data pages” is different from INITSIZE – use DIS GBPOOL command to determine number of actual pages
    - Example: INITSIZE=11520 resulted in 1125 pages at a 15:1 ratio
- 1%
  - Since DB2 gathers 128 pages at a time for castout, don’t use 1% unless actual “number of data pages” > 12800.

Notes

- *Data Sharing: Planning and Administration* suggests using VDWQT=> CLASST and DWT=> GBPOOLT
    - Not usually good if VDWQT is 0% or 1%.
- VDWQT = 0 was popular in early 90s
    - Old DASD or poorly performing DASD couldn't handle spikes when standard thresholds hit (10%)
    - Increased CPU in DBM1 ASID vs.. 10% default

- Use 0 if you have badly performing DASD, else increase VDWQT for local BPs
  - Can get local BP Write Engines not available with so many writes
- Castout has more overhead than just writing to DASD

Recommendation for CLASST

- Stick with defaults unless you know your data.
- If no reuse of data, no benefit to keep it in GBP, write out ASAP, set CLASST=1% as long as 12800 “real” pages
- If reuse is high, you can raise, but not enough to cause it to hit GBPOOLT

- Most customers I see do not know data, or contents mixed due to functional split – safer to use default

- Safe to use low values 1-2% if GBP INITSIZE > 500M
- Large GBP is 1,000M (1G)

Do not use Auto Alter if

- CF available storage is <10%
  - Auto Alter reduces the size of “alterable” structures below INITSIZE (up to MINSIZE), attempting to get 10% available storage in the CF

- Not enough storage to size structure, especially in Test environments
  - XES reaches SIZE quickly
  - Reclaim avoidance results in constant XES attempts to increase directory entries and reduce data pages
    - Reclaim avoidance alone does not allow structure size increase
  - Attempts usually fruitless - produce alarming console messages
  - Hint: test one structure, correctly sized for workload, instead of all

Situations to avoid

- See next page for example of undersized GBP in 4-way group
  - Local buffer pool size = 30,000 pages
  - (INITSIZE should have been 125,000K)

- STRUCTURE NAME(DB2GR0B_GBP3)
  - INITSIZE=**30720K**
  - SIZE=**39976K**
  - FULLTHRESHOLD=90
  - ALLOWAUTOALT(YES)
  - DUPLEX(ENABLED)
  - REBUILDPERCENT(1)

Notes

- CFSizer is currently at http://www.ibm.com/systems/support/z/cfsizer/
- CFSizer calculated 4 x 30,000 at 116,736 for INITSIZE and 228,352 for SIZE.
  - We will always round to even number, such as INITSIZE=125,000 and SIZE=250,000 to make it easy on US!

Example: Small GBP (40MB vs. 125MB)

```
IXC588I AUTOMATIC ALTER PROCESSING INITIATED
FOR STRUCTURE DB2GR0B_GBP3.
CURRENT SIZE: 39936 K
TARGET SIZE: 39936 K
TARGET ENTRY TO ELEMENT RATIO: 36275 : 16488
36275/16488 = 2.20

IXC590I AUTOMATIC ALTER PROCESSING FOR STRUCTURE DB2GR0B_GBP3
COMPLETED. TARGET NOT ATTAINED.
CURRENT SIZE: 39936 K TARGET: 39936 K
CURRENT ENTRY COUNT: 17269 TARGET: 17891
CURRENT ELEMENT COUNT: 8181 TARGET: 8130
ALTER OF REBUILD-OLD STRUCTURE INSTANCE WAS COMPLETED.
17269/8181 = 2.11 -- thus target was not attained
```

Structure already at SIZE, so nowhere to go

Target attained..

```
IXC588I AUTOMATIC ALTER PROCESSING INITIATED
FOR STRUCTURE DB2GR0B_GBP3.
CURRENT SIZE: 39936 K
TARGET SIZE: 39936 K
TARGET ENTRY TO ELEMENT RATIO: 37991 : 16362
37991/16362 = 2.32

IXC590I AUTOMATIC ALTER PROCESSING FOR STRUCTURE DB2GR0B_GBP3
COMPLETED. TARGET ATTAINED.
CURRENT SIZE: 39936 K TARGET: 39936 K
CURRENT ENTRY COUNT: 18732 TARGET: 18732
CURRENT ELEMENT COUNT: 8063 TARGET: 8063
ALTER OF REBUILD-OLD STRUCTURE INSTANCE WAS COMPLETED.
18732/8063 = 2.32
```

Continual Reclaim Avoidance occurs

```
IXC588I AUTOMATIC ALTER PROCESSING INITIATED
FOR STRUCTURE DB2GR0B_GBP3
CURRENT SIZE: 39936 K
TARGET SIZE: 39936 K
TARGET ENTRY TO ELEMENT RATIO: 27469 : 7559
27469/7559=3.63

IXC590I AUTOMATIC ALTER PROCESSING FOR STRUCTURE DB2GR0B_GBP3
COMPLETED. TARGET NOT ATTAINED.
CURRENT SIZE: 39936 K TARGET: 39936 K
CURRENT ENTRY COUNT: 26532 TARGET: 26910
CURRENT ELEMENT COUNT: 7433 TARGET: 7402
ALTER OF REBUILD-OLD STRUCTURE INSTANCE WAS COMPLETE
26532/7433=3.56
```

Reclaim avoidance occurred 7 times in 10 minutes

OA42412 – Closed 5/16/2014

- Abstract: *Corrections to Auto Alter processing for "small" structures*
- Prior example shows problems when structure size is significantly smaller than CFSizer calculation (40G vs 120G)
  - SIZE quickly hit, many attempts for reclaim avoidance, but little success
  - User is "wasting" local buffer pool storage
    - Small GBP cannot track interest in the data in the local BPs
  - Proper solution is to increase GBP size
- **Reclaim avoidance will not occur for "small" GBPs < 20M**
- Structure full avoidance (FULLTHRESHOLD) still enabled

OA42412 cover letter as of 10/8/2013

Submitter Page
Description:
EXTERNAL SYMPTOMS:
Exploiter structure full indications.
Example)
*DSNB325A -DP33 DSNB1CNE THERE IS A CRITICAL SHORTAGE OF SPACE IN GROUP BUFFER POOL GBP32K
DSNB327I -DP33 DSNB1CNE GROUP BUFFER POOL GBP32K HAS ADEQUATE FREE SPACE

Possibly also IXC585E messages for FULLTHRESHOLD being exceeded and auto alter messages indicating the structure was altered.
The structure may be "small" in size and the entry to element ratio may be maxed out resulting in a performance concern. Application monitors may indicate a performance problem or delay.

ANALYSIS:
XCF / XES / CFCC will adjust the size and / or ratio of a structure when ALLOWAUTOALT(YES) is specified and the FULLTHRESHOLD for the structure is exceeded.

* z/OS R9 introduced FULLTHRESHOLD
* z/OS R10 introduced ALLOWAUTOALT with a 1:40 maximum structure ratio
* z/OS R13 increased the maximum structure ratio from 1:40 to 1:255

OA42412 (cover letter continued)

Structure Full Algorithm
- Auto Alter has two algorithms
1. Structure Full avoidance
2. (Directory/entry) reclaim avoidance

Subordinate to 1
- Structure Full avoidance uses Full Structure Monitoring statistics to monitor both changed entries and elements:
- If either one exceeds FULLTHRESHOLD, XES views impending catastrophe to avoid if at all possible
- If either one or the other exceeds FULLTHRESHOLD, XES increases size slightly (about 10%) and also juggles entries and elements increasing one and decreasing the other

When a structure is "small" performance issues may surface if the 1:255 entry is achieved by starving either the entries or the elements causing thrashing

System-initiated alter processing has been changed to suppress reclaim avoidance system-initiated alters **when the structure size is less than or equal to 20 megabytes**

OA42412 (cover letter continued)

LOCAL FIX

BYPASS/CIRCUMVENTION:

- Increase the size of the structure in the policy and use SETXCF command to increase the size of the structure.
- If the application has a means of altering the ratio, try to reduce the ratio.

RECOVERY ACTION:

- Increase the size of the structure in the policy and use SETXCF command to increase the size of the structure.
- If the application has a means of altering the ratio, try to reduce the ratio.

Problems with > 4K page size GBPs

- STRUCTURE NAME(DB2GR0B_GBP32K)
  - INITSIZE=3000K
  - SIZE=10000K
  - PREFLIST (CF01,CF02)
  - ALLOWAUTOALT(YES)
  - DUPLEX(ENABLED)
  - REBUILDPERCENT(1)

  - MINSIZE not specified (default 75% INITSIZE)
  - SIZE >3X INITSIZE
  - 32K page needs **8 elements/page** and 1directory entry
  - When GBP allocated, 68 entries – 117 elements (14 pages)
    - INITSIZE too small!

Problems with > 4K page size GBPs…

- CF storage consumption later exceeded 90%
  - **Auto Alter reduced size of structure to 48 entries/5 elements**
- Not even one page could be written
  - **04E-00C202A5** - Basically out of space
    - data element size was not allocated as DB2 requested (i.e. 8)
  - Write failures result in pages in LPL
    - Even for duplexed GBPs, since there is not enough storage!
- Contributing factor was the SIZE:INITSIZE ratio of 3:1

More OA42412

- Part of problem in previous example is that the math in the algorithm works differently when there is a many to 1 relationship between entries and elements
- One element is always 4K
  - No awareness that a 32K GBP has 8 elements per DB2 page
  - XCF has no idea of pages either
  - If entry:element ratio achieves 255:1, situation is exacerbated

- With OA42412, DB2 passes to XCF the number of elements in its page, called MAXELEMNUM
  - Values are 1 (4K), 2 (8K), 4 (16K), 8 (32K)
  - XCF will not allow directory entries > 255/MAXELEMNUM
    - For GBP32K = 255/8 = 31

Mass Updates can flood a GBP

- Batch jobs and utilities can cause
  - Heavy, sustained GBP page write activities
  - Pages written to GBP faster than engines can cast out
  - Shortage of GBP storage
    - Auto Alter responds by increasing storage up to SIZE and more!
  - Pages may even be added to LPL for repeated write failures
    - Shown in the example on page 13

- Solution: GBP write around for DB2 11
  - z/OS APAR (OA37550) (RSU1212)
  - CF Micro Code Load provide CF command support of conditional writes
    - CFCC 17 +MCL012 service level 10.15 (z196 /z114)
    - CFCC 18 (zEC12)

Example in DB2 (old version)

- 4-way group with BP2 same at 20000 pages
- CFSizer calculated 125,000KB INITSIZE (originally)
- Due to prior growth and IXC185E messages, user increased INITSIZE = 1,000,000KB (1 GB)
- Total pages=342,868, CLASST castout 6860 pages (2%)
- Over short period of batch time on April 30, of 1.5 hours, GBP2 was increased to SIZE, then more pages acquired by exchange entries to elements - and still had 134 Writes failed

	Allocated Size (KB)	SIZE (KB)	Dir to Data Ratio	Xis due to directory Reclaims	Writes Failed - no Storage	GBP Check point time	GBPOOLT % (like DWT)	CLASST % (like VDWQT)	INITSIZE (KB)	SIZE (KB)	Date Allocated
GBP2	1,572,864	1,572,864	1	-	134	8	15	2	125,000	250,000	29-Apr

GBP Write Around Details

- Conditional write to GBP
  - Deferred writes from batch updates
  - DB2 conditionally enables and disables based on thresholds
  - GBP write around protocol
    1. Activated when GBP level threshold 50%, GBP class threshold 20%
    2. Pages are written to DASD via existing deferred write (does not go to GBP)
    3. Disabled when GBP level threshold 40%, GBP class reverse threshold 10%
    4. XI signals sent after DASD write I/O is complete
  - No Write around
    - At commit changed pages written to GBP as usual
    - Index split pages written to GBP
- Benefit:
  - Fewer changed pages in GBPs
  - Desirable pages may remain in GBP
  - Fewer occasions for castout getting behind and hitting FULLTHRESHOLD
  - More gradual growth via Auto Alter when necessary

When duplexed structures don't match – NP!

- Sometimes the secondary/new GBP does not exactly match the primary/old GBP (slightly different entry/element ratio and/or structure size)
- When the CF alters the structure, no "quiesce" occurs
  - Updates to both structures continue during alter process
  - Alter process can take several internal iterations between XCF and the CF, during which time in-use counts are changing in both structures
  - DB2 GBPs are "user-managed" duplexed (by DB2, not XCF) vs. "system-managed" exact copies of one another (XCF)
    - Contents of primary and secondary GBP are quite different by design
    - DB2 deletions from secondary after primary castout periodically cleans up some of the discrepancies
  - CF provides only a return code on requested alter action; XCF compares alter target counts and actual achieved counts, and displays "Target Attained/Not Attained" message.
    - When counts don't match, there is no explanation from CFCC as to what prevented the alter from actually attaining the target counts
    - Targets can be attained for one structure instance but not the other
- CFLEVEL or CF service level differences between the two structure instances may also result in discrepancies in size/ratios between them

Recommendations

- More is better – overconfiguring avoids problems; underconfiguring brings them on
- Do not minimize sizes for multi-element GBPs that you think will have little use:
  - Remember that 8K(2), 16K(4), 32K(8) pages require multiple 4K elements
- Keep SIZE < = 2X INITSIZE
  - At time of allocation, part of structure storage is used for “room for expansion” in case structure is increased to its SIZE.
  - If SIZE were 5X (15,000:3000), there are fewer entries/elements on initial allocation than if the ratio is 2:1 (6000,3000), given the same INITSIZE
- Specify MINSIZE same as INITSIZE to avoid a reduction surprise for production structures
- z/OS team should monitor available storage in CF
  - For CF failover
  - Over time more structures may be added to CF, reducing the free storage
- Apply OA42412

Conclusions

- Advantages
  - ✓ Automatic – ease of use – You give guidelines and get out of the way
  - ✓ Alters ratios without structure rebuild (vs. DB2 –ALTER GBPOOL)
  - ✓ Builds better directory/data ratio than manual tuning
  - ✓ Avoids Write Failures due to lack of storage
  - ✓ Can increase Modify Lock List of Lock Structure when necessary (ERP products and apps with infrequent commit and/or row level locking)
- Minor potential disadvantages
  - – Might not respond soon enough to handle spikes in workload such as update intensive batch (CF write around in DB2 11 can help)
  - – Structure size never decreases – some want expansion / contraction over time

Notes

- Designed to take advantage of your extra CF storage and to offload work from you.

- For more information, you may want to take a look at the z/OS manual, *Setting up a Sysplex,* SA22-7625-22*,* Chapter 4, under the topic “Allowing a Structure to Be Altered Automatically”.

Conclusions…

- Considerations
  - Thresholds – keep low enough that structure full avoidance is never needed
    - Allow FULLTHRESHOLD to default (80%)
    - GBPOOLT should be low enough (<30% default is good) that a sudden spurt in pages and temporary lag in castout should not hit FULLTHRESHOLD
  - For proactive tuning for increases in BPs or new members, allocate GBP SIZE 2x that of INITSIZE
  - Do not use if you have minimal CF storage or want tight control
  - **Reclaim avoidance not performed for GBPs < 20 MB** (OA42412)
- This presentation also available on Techdocs: http://www.ibm.com/support/techdocs/atsmastr.nsf/WebIndex/PRS1956 as "Auto Alter - for DB2 Couch Potatoes"

Notes

- If you set FULLTHRESHOLD close to GBPOOLT and if you have been increasing entries and thus decreasing elements, the number of pages between GBPOOLT (50) FULLTHRESHOLD(60) can be too few and will take little to hit the threshold if the GBP is small. Even if it at first appeared ok.
  - This example came from V7 and user set FULLTHRESHOLD too low
- DB2 can handle via CASTOUT writing of pages from the GBP.
- If GBPOOLT is 25%, then allowing FULLTHRESHOLD to default to 80% is also okay.
- DB2 full GBP messages (DSNB319A-75% & DSNB325A-90%). You can ignore the 75% one. If you have FULLTHRESHOLD of 80%, you should not see DSNB325A. If FULLTHRESHOLD is 90% (common for 1G or greater GBPs), then Auto Alter will increase when DSNB325A is issued

QR Code