



Advanced Technical Support

Data Sharing Health Checks... What We Have Learned

Judy Ruby-Brown
IBM Dallas Systems Center
Advanced Technical Support



© 2006 IBM Corporation

Notes

- This Health Check is performed by IBM Dallas Systems Center and IBM Washington Systems Center cross-disciplinary teams. The customers have complex, data sharing environments in the Americas.
- Focus is on parallel sysplex infrastructure, not application behaviors.
- Customer participation is usually DB2 systems staff
- Presentation is based on experience of last 3 years over 15 customer health checks. As of June, 2006 primary release is DB2 V7
- Some V8 enhancements pointed out where appropriate

Agenda

- Keeping up with Structure Sizing
- Autonomic Tuning of GBP structures
- DBM1 Virtual Storage – How much for “real work”
- What’s wrong with ICF-only configurations?
- Usual Suspects - Runaways, rollbacks and others



Advanced Technical Support

Keeping up with Structure Sizing

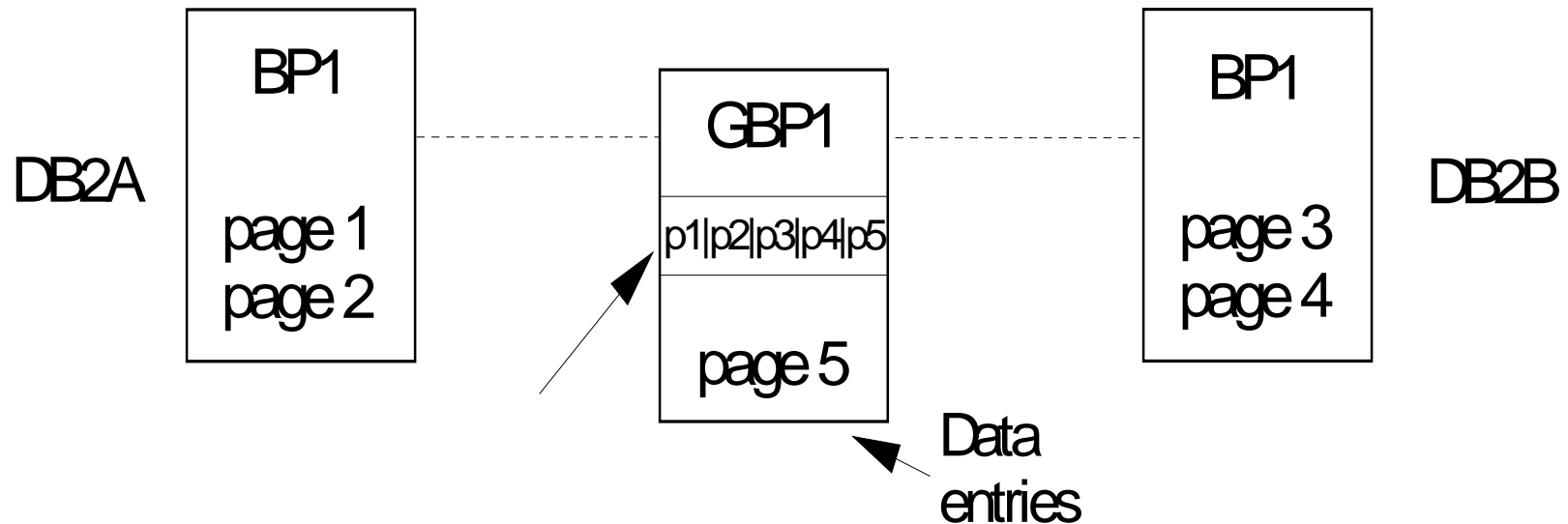
Group Buffer Pools and Lock Structure



© 2006 IBM Corporation

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

- 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.

Initial GBP Sizing

- Several ways to estimate GBP size before DS enabled
 - Goal: Safety – “Don’t shoot yourself in the foot”
- CFSIZER - <http://www.ibm.com/servers/eserver/zseries/cfsizer/>
 - Advantage: connects to live server for calculation of latest CFCC
- Formulae in *Data Sharing: Planning and Administration*

- **Dallas Systems Center formula for INITSIZE of a GBP:**
 - Sum VP + HP pages for number of members expected
 - Convert to K (multiply by 4 for 4K page)
 - Take 40% of result
 - Provides enough directory entries to avoid XIs
 - >4 members, formula over allocates GBP

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 previous slide showed.

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.

Recent note (04/2006): Original DSC formula took 1/3 of the total. It worked 1996-2006. It failed due to internal structure increases, likely since CFCC 12. At this time, you have two options: use CFSIZER, or increase the 33 1/3% to 40%. Either one increases the storage used by about 12%. But neither one guarantees no XIs due to directory reclaims.

Why these are bad..

1. *Performance*: XIs due to Directory Reclaims in RMF interval
 - should be 0!
 - Read page from DASD to local BP -Register it in GBP
 - CPU and I/O overhead if not enough directory entries:
 - 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
2. *Availability*: Writes failed due to lack of storage – should be 0!
 - Potential outage if pages added to LPL

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**
- 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 (drains entire object < V8)

What about Lock Structure?

- Monitor with RMF CF Activity Report for False Contention > 1%
 - FC/Total Requests on right hand side of report
 - Since z/OS 1.2, Heuristic Algorithm, DB2PM calculated huge % or not at all, so I ignore it
- Can specify LTE=# of lock table entries desired on each IRLM proc

- Assume a 4-way DS, FC=3% and LTE=64M according to RMF. Structure statement says:
 - INITSIZE=256,000K,SIZE=256,000K and yields
 - Lock Table of 128M with 64M LTE and Modify Lock list (RLE) = 128M
- We want to increase the LTE to 128M to reduce FC. Next slide...

2 ways to get 128M Lock Table Entries

- Specify the structure size either of two ways:
 - INITSIZE=512,000K,SIZE=512,000K
 - Lock Table of 256M with 128M LTE and a Modify Lock list (RLE) = 256M
 - or
 - INITSIZE=384,000K,SIZE=384,000K
 - Lock Table of 256M with 128M LTE and a Modify Lock list (RLE)=128M
 - Must rebuild structure to change Lock Table

- Specify number of entries you want using LTE=128 on irlm startup
 - **Warning: make sure LTE in sync with CFRM INITSIZE**
 - If LTE > , can allocate too few RLE or can abend w/o PK19928(UK13975 HIPER 5/17/06)

What about Lock Structure?

- Monitor with RMF CF Activity Report for False Contention > 1%
 - FC/Total Requests on right hand side of report
 - Since z/OS 1.2, Heuristic Algorithm, DB2PM calculated huge % or not at all, so I ignore
- Can specify LTE=# of lock table entries desired on IRLM proc
- If 4-way DS, FC=3% and LTE=64M according to RMF. Structure statement says:
 - INITSIZE=256,000K,SIZE=256,000K yields
 - Lock Table of 128M with 64M LTE and Modify Lock list (RLE) = 128M
- We want to increase the LTE to 128M to reduce FC. Next slide...

2 ways to get 128M Lock Table Entries

- Double the structure size:
 - INITSIZE=512,000K,SIZE=512,000K
 - Lock Table of 256M with 128M LTE and a Modify Lock list (RLE) = 256M
 - or
 - INITSIZE=384,000K,SIZE=384,000K
 - Lock Table of 256M with 128M LTE and a Modify Lock list (RLE)=128M
 - Must rebuild structure to change Lock Table

- Specify number of entries you want using LTE-128 on irlm startup
 - **Warning: make sure LTE in sync with CFRM INITSIZE**
 - If LTE > , can allocate too few RLE or can abend w/o PK19928(UK13975 HIPER 5/17/06)

Notes

- While Lock Table itself must be a power of 2, you can avoid creating an extraordinarily large STRUCTURE by using the LTE parameter on the IRLM startup procedure to specify the entries you want.
 - The example shows how you can keep the same Modify Lock List (RLE) size, assuming it is correct, and double the Lock Table, while conserving CF storage.
- The LTE specification takes precedence over STRUCTURE unless there is insufficient INITSIZE that it cannot be allocated and should fall back to a 50:50 split. Without HIPER PQ19928 (closed 4/27/06), an abend is likely.
 - If it can be allocated, you are left with fewer RLEs than planned
- Even without LTE specified (LTE=0), INITSIZE does not have to be an even multiple of 2. IRLM essentially divides the allocation by 2 and rounds up to the next power of 2 or down to the previous power of 2, whichever is closer.
 - Other important service is PK14969 (F601)

Locking..

- Most Global Lock Contention >5% is usually XES locking and will not decrease until V8 NFM
 - By 2Q 2006, I have not seen a V8 NFM customer
 - LTE on IRLM proc can give finer tuning than doubling size

- Problem: mismatched CPU to CF and/or distance
 - Fast z9xx processors to -1 or -2 generation CFs
 - Heuristic algorithm of XES spares spin time on z990 and converts sync lock request to async
 - If % is large, lock request time goes up
 - One user had z900 > 9672-R6 and instead of 40 usec, **75%-85% were converted to async** at 120 usec/request



Advanced Technical Support


Auto Alter – for Couch Potatoes

Autonomic Tuning for GBPs



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)

Recent Experiences with very large customers

- Millions “XI due to directory reclaims”
- Failure to notice -
(2 days results using
-DIS GBPOOL(*) GDETAIL(*)
TYPE(GCONN) command

	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

	21 Feb,2005 to 29 Mar, 2005			
	Allocated Size (KB)	SIZE (KB)	Dir to Data Ratio	Writes Failed - Storage
GBP0	200,192	240,000	5	-
GBP1	368,640	510,000	5	-
GBP2	300,032	360,000	5	6
GBP3	400,128	480,000	5	-
GBP4	1,000,192	1,200,000	5	21
GBP5	4,096	4,800	5	-
GBP6	8,192	9,600	5	-
GBP9	200,192	240,000	5	-
GBP10	400,192	480,000	5	-
GBP32K	16,128	19,200	5	-
Totals	2,897,984	3,543,600		

Goal: Pro-active tuning

- Allocate large DB2 GBPs (example: 2x current allocation) to handle long term growth for
 - Increases in local BPs
 - Additional data sharing members

- Let Auto Alter tune the directory to data ratio
- Later (6 months?) 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 structures
 - 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**
- Function is not DB2-specific. Works for all structures in the CF

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=100000K
 - SIZE=200000K
 - PREF=(CF2,CF1)
 - FULLTHRESHOLD=80
 - MINSIZE=100000K
 - **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
 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 and will avoid it if at all possible

- ✓ With OA08486 (F0412) - 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
 - Structure Full is interested only in CHANGED entries

- If reclaims occur too frequently, XES increases the number of directory entries while decreasing the number of (data page) elements
 - Up to 40:1 ratio

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
 - 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 (> 500,000K)
- Default FULLTHRESHOLD to 80%

- Dribble Castout to avoid hitting thresholds

Threshold	Recommended	Default
GBP Checkpoint	4 minutes	8 minutes
CLASST (~ VDWQT)	1-5%	10%
GBPOOLT (~ DWT)	10-25%	50%

What about CLASST?

- Premise: match CLASST to VDWQT per Data Sharing Guide
- Seen a lot of 0% - What does it mean?
 - When 40 pages in castout class are updated, DB2 casts out
 - **Except** – If 1% of #pages in GBP is < 40 pages
 - Example: if CLASST=0 and #pages in GBP=1400
 - **Castout occurs at 14 pages since 1% is < 40 pages**
 - Note: “number of data pages” is different from INITSIZE – use DIS GBPOOL command to determine number of actual pages (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
- Castout has more overhead than just writing to DASD

Recommendation for CLASST

- Stick with defaults unless you know your data.
 - 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
 - Many customers do not know data or contents mixed – safe to use default
-
- Safe to use low values 1-2% if GBP INITSIZE > 500,000K
 - Large GBP is 1,000,000K (1GB)

Do not use

- If < CFCC 12
 - Limited to 2GB Control Storage
 - Since CFCC 12 in 2003, most users are CFCC 13 or 14
- If CF available storage is <10%
 - Auto Alter reduces the size of structures below INITSIZE (up to MINSIZE), attempting to get 10% available storage in the CF
- If 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

Do not use...

- See next page for example of undersized GBP in 4-way group
 - Local buffer pool size = 30,000 pages
 - (INITSIZE should have been 175,000K)

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

Notes

- <http://www-1.ibm.com/servers/eserver/zseries/cfsizer/>
- CFSIZER (<2005) calculated sizes of 175,616 for both INITSIZE and SIZE.
- Calculation could have been... with local VP size=30,000 pages (4k)
- INITSIZE= 30000 for 4 way is $120,000 * 4(\text{convert to K}) = 480,000 * 40\% = 192,000\text{K}$

Example: Small GBP (40MB vs 175MB)

```
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 without Structure Full

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)
- Disadvantages
 - Low setting of FULLTHRESHOLD causes disruptive Structure Full activity

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*, SA22-7625-10, section 4.2.2.5, titled “Allowing a Structure to Be Altered Automatically”.

Conclusions...

- Considerations
 - Set FULLTHRESHOLD for activation in emergency (80-90%)
 - DB2 has full GBP messages (DSNB319A-75% & DSNB325A-90%)
 - Do not make it close to GBPOOLT. A sudden spurt in pages and a temporary lag in Castout can put you over the edge
 - Small GBPs may have few pages between GBPOOLT and current elements if low FULLTHRESHOLD
 - If you plan to use for proactive tuning for increases in BPs, new members, allocate GBP much larger initially
 - Do not use if you have minimal CF storage or want to have tight control

Notes

- If you make 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.
- 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.



Advanced Technical Support

Virtual Storage in DBM1

Using MEMU2 to identify peak virtual storage consumption to determine safe CTHREAD and avoid failures



© 2006 IBM Corporation

How much work thru a DB2 member?

- CTHREAD is a measure of work through DB2, but chosen arbitrarily
 - True also for MAXDBAT
- Some customers know number is wrong but not what is right
 - Abends due to either E2003 or E20016 abends of user transactions
 - 878 abends of DBM1 if no storage available for must complete work (backout)
- RMF VSTOR identifies virtual storage in DBM1, but not how DB2 uses it.
- IFCID225 provided in DB2 V7/V8 provides major consumers
 - DB2PM outputs in Statistics Long (DBM1 Storage) or Record Trace
 - Most other monitors do not have capability
- ✓ MEMU2 Rexx (asis code) outputs primarily IFCID225 data, but also includes some data from IFCIDs 1 and 202
- Rest of section describes how...

Memu – Memu2.zip package

- As-is REXX code (no support provided) from John Campbell, documented by Judy Ruby-Brown – available at DB2 Trading post (url at end of this section)
 - MEMU2 REXX – outputs IFCID225 info invoked as batch job
 - MEMUSAGE REXX – outputs IFCID225 (now) if invoked from TSO Option 6 (once)
 - Memu2.jcl.txt – JCL to invoke MEMU2 REXX
 - Memory Usage-V3-IFCID225.doc – documentation to install, modify, and use
- MEMU2 REXX –
 - Invoked via JCL on single member basis
 - Outputs IFCID225 info to comma delimited data set
 - 5 minute interval (default) independent of STATIME
 - 12 intervals default (one hour)
 - JCL specifies SSID and overrides to above
- Needs PQ85764 (UQ87093) from F404 for DB2 V7, Statistics Class 6. After PQ99658, collection of IFCID225 is triggered by Statistics Class 1

JCL to invoke MEMU2 for 1 day

```
//JRBMEMU JOB (????), 'JUDY R-B',MSGCLASS=0,REGION=0M,
// CLASS=A,NOTIFY=???????
/*JOBPARM SYSAFF=SYSD
/* COMMENT THE /*OUTP THE FIRST TIME, SINCE IT IS NOT CREATED
//DEL001 EXEC PGM=IEFBR14
//OUTP DD DISP=(MOD,DELETE,DELETE),
// DSN=JUDYRB.MEMUSAGE.OUTPUT
//MEMU2 EXEC PGM=IKJEFT01,DYNAMNBR=25,ACCT=SHORT,
// REGION=4096K
//STEPLIB DD DSN=DSN810.SDSNLOAD,DISP=SHR
//SYSEXEC DD DSN=JUDYRB.WSC.JOBS,DISP=SHR WHERE 80 CHAR REXX LIVES
//OUTP DD DISP=(,CATLG,DELETE),
// DSN=JUDYRB.MEMUSAGE.OUTPUT,
// DCB=(RECFM=VB,LRECL=4096,BLKSIZE=0),
// SPACE=(TRK,(5,1),RLSE),UNIT=SYSDA
//SYSUDUMP DD SYSOUT=*
//SYSTSPRT DD SYSOUT=*
//SYSOUT DD SYSOUT=*
//SYSTSIN DD *
MEMU2 JUDY 288
```

DB2PE Sample – Statistics Trace | Report

DBM1 AND MVS STORAGE BELOW 2 GB	QUANTITY	DBM1 AND MVS STORAGE BELOW 2 GB	QUANTITY
TOTAL DBM1 STORAGE BELOW 2 GB	(MB) 773.05	24 BIT LOW PRIVATE	(MB) 0.23
TOTAL GETMAINED STORAGE	(MB) 575.00	24 BIT HIGH PRIVATE	(MB) 2.25
VIRTUAL BUFFER POOLS	(MB) 429.69	31 BIT EXTENDED LOW PRIVATE	(MB) 27.38
VIRTUAL POOL CONTROL BLOCKS	(MB) 13.43	31 BIT EXTENDED HIGH PRIVATE	(MB) 954.23
EDM POOL	(MB) 117.19	EXTENDED REGION SIZE (MAX)	(MB) 1714.00
COMPRESSION DICTIONARY	(MB) 2.35	EXTENDED CSA SIZE	(MB) 200.06
CASTOUT BUFFERS	(MB) 9.13		
DATA SPACE LOOKASIDE BUFFER	(MB) 0.00	AVERAGE THREAD FOOTPRINT	(MB) 3.61
HIPERPOOL CONTROL BLOCKS	(MB) 0.05	MAX NUMBER OF POSSIBLE THREADS	236.12
DATA SPACE BP CONTROL BLOCKS	(MB) 0.00		
TOTAL VARIABLE STORAGE	(MB) 139.53		
TOTAL AGENT LOCAL STORAGE	(MB) 53.94		
TOTAL AGENT SYSTEM STORAGE	(MB) 32.35		
NUMBER OF PREFETCH ENGINES	77.00		
NUMBER OF DEFERRED WRITE ENGINES	300.00		
NUMBER OF CASTOUT ENGINES	73.00		
NUMBER OF GBP WRITE ENGINES	58.00		
NUMBER OF P-LOCK/NOTIFY EXIT ENGINES	9.00		
TOTAL AGENT NON-SYSTEM STORAGE	(MB) 21.60		
TOTAL NUMBER OF ACTIVE USER THREADS	29.67		
RDS OP POOL	(MB) 34.54		
RID POOL	(MB) 16.97		
PIPE MANAGER SUB POOL	(MB) 0.00		
LOCAL DYNAMIC STMT CACHE CNTL BLKS	(MB) 0.99		
THREAD COPIES OF CACHED SQL STMTS	(MB) 0.00		
IN USE STORAGE	(MB) N/A		
STATEMENTS COUNT	N/A		
HWM FOR ALLOCATED STATEMENTS	(MB) N/A		
STATEMENT COUNT AT HWM	N/A		
DATE AT HWM	N/A		
TIME AT HWM	N/A		
BUFFER & DATA MANAGER TRACE TBL	(MB) 9.41		
TOTAL FIXED STORAGE	(MB) 3.80		
TOTAL GETMAINED STACK STORAGE	(MB) 54.71		
STORAGE CUSHION	(MB) 112.04		

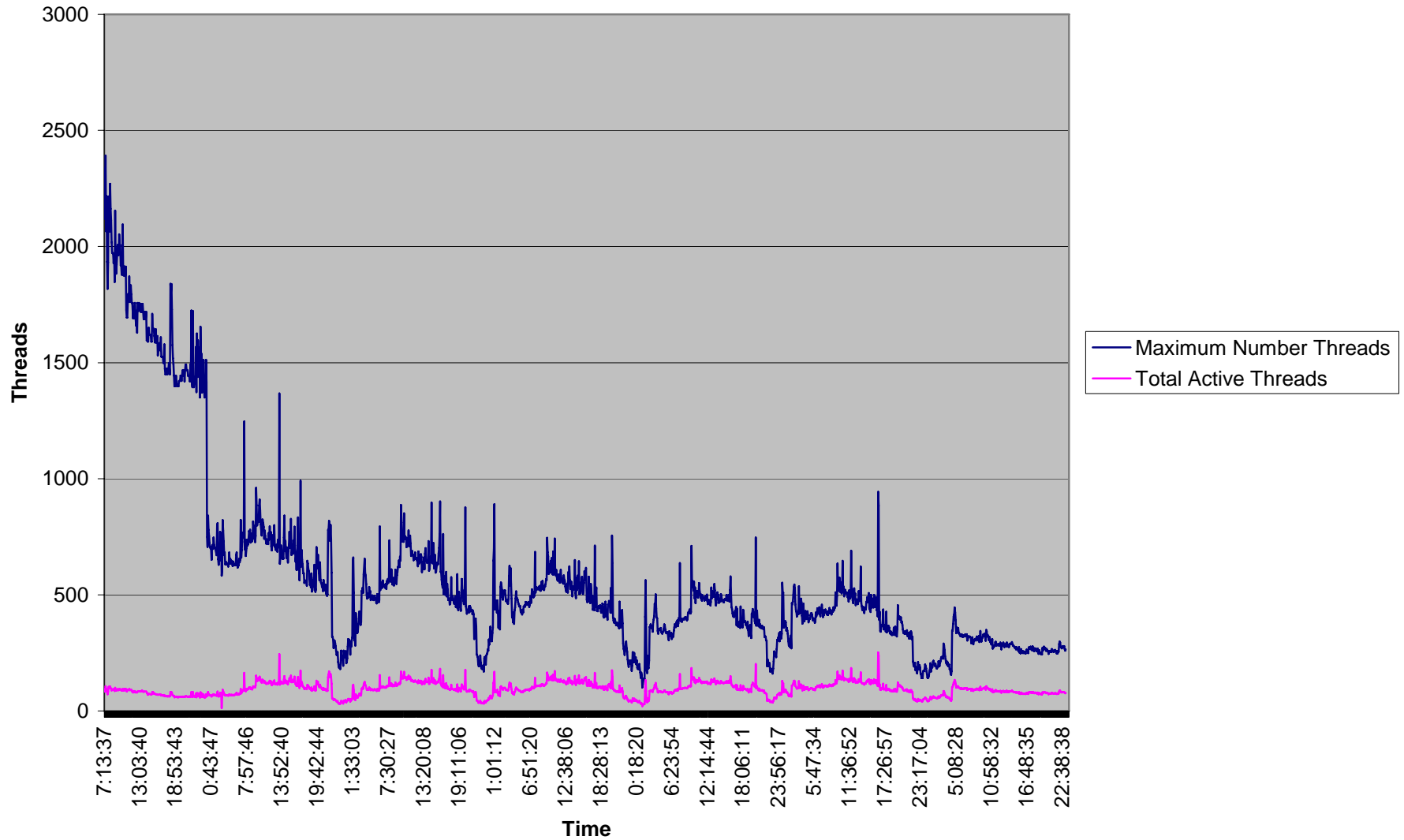
Create Spreadsheet from MEMU2 output

- Download MEMUSAGE.OUTPUT to hard drive
- Open file in Microsoft Excel and Text Wizard walks you through creation of spreadsheet from comma delimited input
- Add some columns at end to make calculations easier
 - Basic Storage Cushion
 - Upper Limit Total
 - Total Fixed
 - Upper Limit Variable
 - Thread Footprint
 - Max # Threads
 - Total Active Threads (Allied Threads + Active DBATs)
 - Column before first one “Time”. Label it “Date” and enter it as number, not date type, of ‘yyyymmdd’

Notes – Extra column contents

- Basic Storage Cushion = $\text{ROUND}(\text{MIN}(200, (\text{MVS Extended Region}(\text{Epvt})) * 0.15), 0)$
 - note that MVS extended Region is the same as EPVT, so that parenthesis is just a shorthand notation of it, not a part of formula
 - “200” is the hard coded storage cushion
- Upper Limit Total
 - MVS Extended Region - Basic Storage Cushion
- Total Fixed
 - $\text{=Round}(\text{Total Getmained Storage} + \text{Total Stack Storage} + \text{Total Fixed Storage} + \text{MVS Low Private Storage}, 0)$
- Upper Limit Variable
 - MVS Extended Region - Basic Storage Cushion - Total Fixed Storage
- Thread Footprint
 - $\text{=Round}((\text{Total Variable} - \text{Total Agent System Storage}) / (\# \text{ Active Allied Threads} + \text{Max \# Active DBATS})^2)$
 - Max # of Active DBATs is used instead of Current Active # DBATS. We don't know when those DBATs were active, but storage was consumed on their behalf and we cannot assume it has been released.
- Max # Threads
 - $\text{=Round}((\text{Upper Limit Variable} / \text{Thread Footprint}), 0)$
- Total Active threads (Number of Active Allied Threads + Current number of active DBATS)
- Insert a column before the first one “Time” and label it “Date”. Enter date as yyyyymmdd but just as number, not formatted. Propagate it down the rows.

DB2B



Notes 1

- This is a V7 chart
- This chart represents a week whose beginning is DB2 startup. Note the lower line shows gentle repetitions, indicative of a repeating workload for each day.
- DB2 was brought up at the start of this chart. We ignore the first day, while DB2 settles in.
- At 13:52 we have high value, but it is not repeated at all in our chart.
- 17:26:57 on the right hand side shows high of 940. This is good because several days have gone by and there is still enough storage to maintain 940 threads.
- There are several points in the middle around 19:11:06 that have 890, and that is the chosen value, which you split between CTHREAD and MAXDBAT..

Notes 2 – How to create a chart

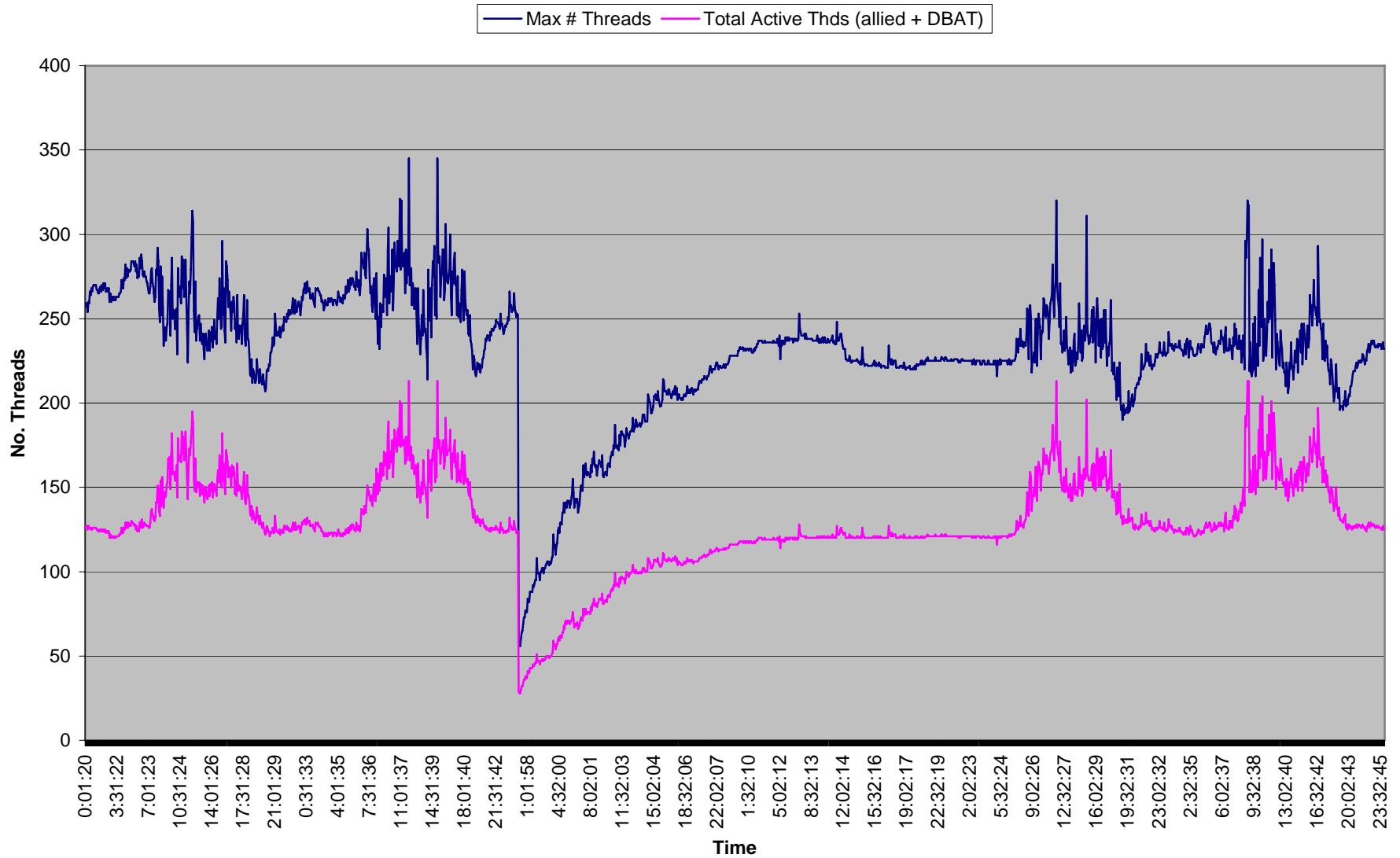
- Select only the column with Time, and the last two columns Max threads and Current Active Threads.
- Click the Chart toolbar and select “line drawing”.
 - Default was not used. One at top left was used.
 - Follow the “next” prompts until “Step 3 of 4 – Chart Options”
 - Legend – change radio button to Top
 - Titles – key the following
 - DB2 SSID and whatever else is meaningful for you
 - Category(X) axis = Time
 - Category(Y) axis = No. Threads
 - On “Step 4 of 4”, change radio button to “Insert as new sheet”
 - Then Finish. There will be a new worksheet with the chart just created. You can make it larger by clicking on the chart and using View >> Sized with Window.

Dataspace Lookaside Pool Size	Current Number of Active DBATs	Maximum Number of Active DBATs	Full System Contraction	Storage Critical	Basic Storage Cushion	Upper Limit Total	Total Fixed	Upper Limit Variable	Thread Footprint	Maximum Number Threads	Total Active Threads
36	3	9	0	0	197	1117	218	899	1.2	749	11
36	4	9	0	0	197	1117	218	899	1.37	656	10
36	3	9	0	0	197	1117	219	898	1.36	660	9
36	3	9	0	0	197	1117	219	898	1.05	855	13
36	3	9	0	0	197	1117	220	897	1.33	674	9
36	9	9	0	0	197	1117	220	897	1.51	594	15
36	8	9	0	0	197	1117	223	894	1.48	604	14
36	8	9	0	0	197	1117	224	893	1.28	698	16
36	8	9	0	0	197	1117	224	893	1.4	638	15
36	4	9	0	0	197	1117	224	893	1.26	709	12
36	6	9	0	0	197	1117	224	893	1.43	624	12
36	4	9	0	0	197	1117	224	893	1.42	629	10
36	3	9	0	0	197	1117	224	893	1.22	732	11
36	4	9	0	0	197	1117	224	893	1.3	687	11
36	3	9	0	0	197	1117	224	893	1.4	638	9
36	3	9	0	0	197	1117	224	893	1.46	612	9
36	3	9	0	0	197	1117	224	893	1.37	652	10
36	4	9	0	0	197	1117	224	893	1.25	714	13
36	3	9	0	0	197	1117	224	893	1.21	738	12
36	3	9	0	0	197	1117	225	892	1.39	642	10
36	3	9	0	0	197	1117	225	892	1.39	642	10
36	4	9	0	0	197	1117	225	892	1.25	714	13
36	3	9	0	0	197	1117	225	892	1.49	599	9
36	4	9	0	0	197	1117	225	892	1.53	583	10
36	3	9	0	0	197	1117	225	892	1.32	676	11
36	4	9	0	0	197	1117	225	892	1.51	591	10
36	3	9	0	0	197	1117	225	892	1.5	595	9
36	4	9	0	0	197	1117	225	892	1.29	691	13
36	2	9	0	0	197	1117	225	892	1.51	591	8
36	3	9	0	0	197	1117	226	891	1.45	614	10

General Guidelines for Charts

- Throw away the first day or so while DB2 is settling in.
- Throw away abnormally low values for Current Active Threads (especially after an interval of high values).
 - If #users is small, footprint is abnormally large as current threads and max DBATs active are the divisor.
 - Max # Threads will be lower.
- Pay attention to month end or other known peaks, weekends where REORG and heavy I/C can increase VSTOR stress.
- Look for peaks after DB2 has been up for awhile. That gives a good feeling that the Max can still be maintained.
- Get several plot points to validate your choice, to be conservative. The most accurate storage is obtained as you get close to the maximum number of active threads, because DBM1 storage is a more accurate reflection.
- Apportion chosen maximum thread value between CTHREAD and MAXDBAT.
- NOTE: Assumption is charted workload does not change. Changes mean that these measurements must be redone.

DB2C - peak week - Active vs. Max Threads



Notes

- This DB2 V7 system has been up for awhile, but was determined by user to be peak for a week. User knew this by virtue of a knowing the primary workload. Still MEMU2 was run for a month to verify.
- The abrupt drop in both lines reflected the peak was in business days. They were split by a weekend. No measurements were reported on Sunday.
- The Value for CTHREAD was 200 and the lower pink line shows that DB2 hit CTHREAD several times during this peak. As such, the max thread value is somewhat artificial.
- DB2PM showed at a peak time 30% or 125K threads were “queued at Thread Create” in 30 minute period.
- Proceeding cautiously, 325 was chosen. MEMU2 should be rerun following this change to determine if It is possible to increase CTHREAD beyond 325.

Summary

- Customers can be shocked to find out how much storage compressed data consumes. One compressed all d/s > 10MB in its tool.
- MEMU2 shows how many times critical storage compaction occurred
- Use several common CTHREAD values for cloned, diverse workloads?
 - 2 Batch members, 3 online members, 2 distributed members
 - A common CTHREAD value for all can leave a lot of capacity on the table
 - ✓ Use one CTHREAD for each of 3 workload types to provide failover
- See *DB2 V8 Performance Topics*, SG24-6465, section 4.3, for DBM1 virtual storage detail
- The MEMU package is now available on the DB2 Trading Post at
 - <http://www.ibm.com/software/data/db2/zos/exHome.html>
 - Works for both V7 and V8



Advanced Technical Support

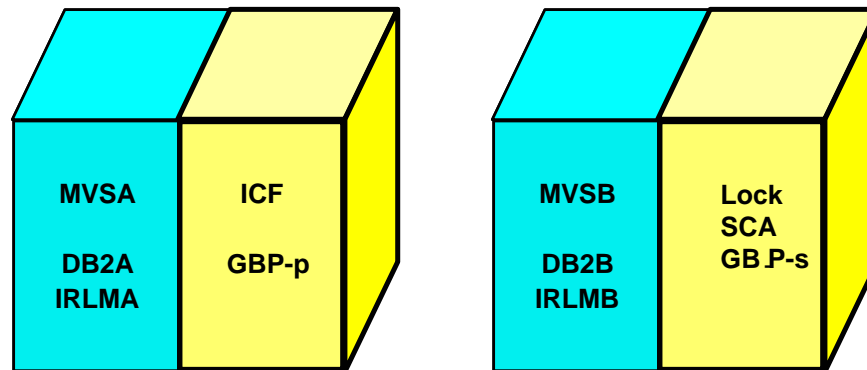
What's wrong with ICF-only Configurations

Internal Coupling Facility (ICF) - Most common parallel sysplex CF configuration



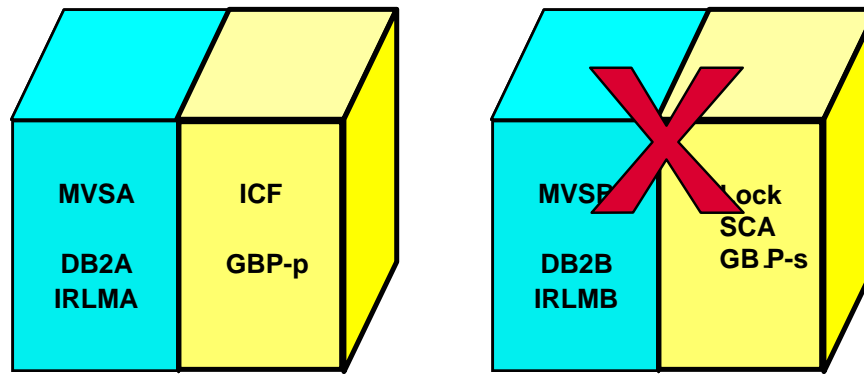
© 2006 IBM Corporation

Typical Configuration of 2000s



- Fewer processors with many fast engines – z900, z990, System z9
- ICFs require fewer links than external CFs
- Parallel Sysplex has fast recovery from single failure
 - If a single CF fails,
 - Lock/SCA rebuild in other CF
 - GBPs fall back to simplex
 - If MVSB fails, ARM restarts it on MVSA quickly

If box with z/OS image and SCA/Lock fails



- **Double failure** (Sysplex recovers quickly from SPOF, but not double)
- Lock/SCA rebuilt in ICF in other processor. BUT
 1. Each IRLM contributes its part of the lock structure for fast rebuild
 2. IRLMB cannot respond or participate until MVSB is failed
 3. Rebuild cannot complete, SO fallback to original SCA/Lock structures - Not accessible
- Data sharing group fails - DB2A/IRLMA – to preserve integrity
- Eventually MVSB fails (fast with ISOLATETIME vs PROMPT in SFM policy). DB2B/IRLMB fail

Restarting the group

- ✓ Outage can be short – After MVSB/DB2B fails, ARM can restart DB2B on other processor fast
- Start of DB2A is a group restart to rebuild the SCA/LOCK
- With well behaved units of work, takes slightly more time than normal restart
 - If long running threads, can take a long time – must read log from oldest UR to recreate LOCK structure
- GBPs in simplex (nothing in GRECP/LPL)

- What solutions are possible?

ICF Availability Solutions

- Solution 1 – Systems Managed Duplexing for DB2 Lock and other structures
 - Needs 1 add'l ICF engine in each ICF, 2 more ICB links (CF-CF), and 1 host engine in each z990 or System z9
 - DB2 lock request 3x-4x higher than simplex (60 usec vs 9-15 usec) and converted to asynchronous

- ✓ Solution 2 – External CF, such as z890 or z9-BC
 - Requires 2 ICB4 links from each processor – no add'l host overhead
 - SCA/Lock structures now can rebuild into ICF if external CF fails
 - DB2 lock requests remain < 15 usec. Likely synchronous request



Advanced Technical Support

Usual Suspects

Runaways, rollbacks...and others, including V8 CF Batching



© 2006 IBM Corporation

Long Running Updaters - runaways

- Failure to commit cause of
 - Increases locking overhead (GCLSN does not advance and lock avoidance fails)
 - Lock escalation inhibits concurrency

 - Causes timeouts/deadlocks for other workload
 - Utilities can't break in (also long running READERS)
 - High rollback rate – sometimes 10-15%

 - Increase in thread related virtual storage
 - Think back to virtual storage section!

Everyone has them!

- Some haven't set URCHKTH or URLGWTH – some values too large to do any good
 - Intent of parameters - notify users before runaways reach critical mass
- Most do not cancel runaways, leaving exposure to above
 - Some trap messages and produce reports
- Some have limits during “daytime” and remove them at night
 - Problem when batch runs late
- Don't forget readers – Online REORG can't break in for switch phase
 - V8 has IFCID313 to track and ZPARM LRDRTHLD (# min for Read Claim)

Plan to reduce runaways

- ✓ Develop long term project to put commit scope in applications
 - Management must establish reasonable commit scope and support project
 - Set URCHKTH and URLGWTH high at first (i.e. 10 and 100)
 - Avoid flooding consoles with messages
 - Produce reports to identify offenders and “encourage” commits
 - Provide reasonable deadline by which commits must be implemented or jobs will be cancelled
 - Reduce URCHKTH and URLGWTH and continue process goal commit scope is reached

Notes

- Assuming CHKFREQ=5 minutes, if URCHKTH=10, then 50 minutes goes by – long time on zSeries processors – If abends, may take 100 minutes to back out
- DSNR035I for URCHKTH; DSNJ031I for URLGWTH
- Start with URCHKTH at 10 and reduce it over time to 2 or 10 minutes
- Start w/URLGWTH at 100 (100000) and reduce over time to 20 or whatever meets the commit scope for single update job

Other

- High Rollback rates common – one large user had 12% in 30 minutes for 1000 rollback/minute – another had 14% in 15 minute for 20K/minute
 - High % of reads from active log vs. output log buffer (better)
 - Lots of work to undo – resources; process not optimized
 - Determine reasons and work to eliminate

- One disciplined user had <1% peak abend for 15 minutes for 70/minute most of other 6 members' totals were in single digits!

New V8 CF Batching commands

- Design point –
 - take single entry CF request, such as Write and Register page
 - Example: 30 usec sync service time that takes 10 usec on CF CPU
 - Create a single CF request that has a list of actions for multiple pages
 - Example: Bundle 20 WAR requests into WARM - runs in 200 usec on CF
 - Result: No CF CPU reduction but saves host and link time
 - Due to service times, will normally run async
- Invoked in CM mode when GBPs are reallocated
- Tools to use
 - RMF CF Activity Report gives the service times for each GBP
 - DB2PM GBP sections - # of CF Requests in fields
 - Write and register Mult >>>> WARM
 - Read for Castout Mult >>>>RFCOM – read for castout multiple

Calculating effect of CF Batching

```

STRUCTURE NAME = DSNDB2P_GBP24      TYPE = CACHE  STATUS = ACTIVE
# REQ ----- REQUESTS -----
SYSTEM      TOTAL          #    % OF  -SERV TIME(MIC)-  REASON  #
NAME        AVG/SEC        REQ  ALL   AVG   STD_DEV          REQ  % OF
                                     /DEL   STD_DEV  /ALL
ABCD        212K   SYNC   187K   6.6   32.5   61.5   NO SCH  1101   0.5
                235.9  ASYNC   25K   0.9   237.1  382.6  PR WT   0     0.0
                CHNGD  1099   0.0   INCLUDED IN ASYNC  PR CMP  0     0.0
                                     DUMP   0     0.0
WXYZ        2615K  SYNC   1789K  63.3   36.6   73.5   NO SCH  65K   2.6
                2905  ASYNC   761K  26.9   204.8  283.6  PR WT   0     0.0
                CHNGD   65K   2.3   INCLUDED IN ASYNC  PR CMP  0     0.0
                                     DUMP   0     0.0
  
```

Of the async requests: $761K - 65K = 696K$

From DB2PM GBP24 sum of 2 WARM/RFCOM fields were 671K and were about 95% of the async requests

Notes

- Of the 761K async requests, 65K do not count because they were waiting for subchannel busy (696K remain)
- Going to the DB2PM reports total “write and register mult” and “read for castout mult”. Those are the requests to the CF for 671K.
 - 671K/696K > 95% is the percentage of async that were for CF Batching
- Do not include PAGES WRITE & REG MULT as they refer to pages, *not* to number of requests
- The rest of the requests were converted to async based on their service times (spare the host CPU spin time is the goal in the heuristic conversion of sync >async).

DB2 restart after z/OS image failure

- Reliance on (slow) manual procedures for restart
 - Manual restart of DB2 on another image (5-10 minutes) *or*
 - IPL image and start DB2 (30-60 minutes)
 - Causes timeouts for workload on *other* members that should be little affected
- ✓ Implement Automated Restart Manager (ARM) for fastest restart (10-15 sec – either Normal or Light)
- Some customers can't implement ARM for cross system restart – requires JES2 MAS (multi access spool)
- ✓ Light improved in V8
 - Stays up for commit coordinator, including DDF) (no new work)
 - Data sets with IX-L locks not held in X mode – now accessible by other DB2s

Others

- Paging critical with BPs in data spaces or for V8
 - <1% okay, 2-4 investigate and tune >5% reduce size
 - Have not seen problems to date – Check anyway
- Package not found in first collection in PKLIST
 - Package allocation attempts vs Package allocation success not 1:1
 - Some customers have 2-3x more attempts vs successes
 - i.e. PKLIST(coll1.*, coll2.*, coll3.*, coll4.*)
- PCLOSET = fewer than 20/minute
 - Reduce open data sets by use of CLOSE (YES) data sets for most t/s
- CLOSE NO only for t/s that are sensitive to physical open
 - Once GBP dependent, they remain so, even if no other members
- Logging
 - Place active Log Copy1 and Copy2 on separate DASD SSIDs
 - Multi-volume archive logs? Reduces # logs registered in BSDS
 - Determine active log residency – relative to image copy cycle for critical t/s

Notes

- Paging for data spaces and V8
 - BPs reads ROT < 1%-5% max
 - Pages read (A) = Sync Reads + Seq Prefetch pages + Dynamic prefetch pages + List Prefetch pages
 - % Paging for Reads = Page Ins for read / A
 - BP writes ROT is < 1%-5% max
 - Pages written (A) = sync Writes + (async writes times pages written/write i/o)
 - % Paging for writes = Page ins for write / A
 - ROTs apply only in steady state. If a buffer pool is being populated, there can be many PAGE-IN's but that is OK.

Notes (continued)

- Access for coll1 is one attempt but no success. If found in coll4, 4 attempts, 1 success
 - Reduce number of collections to a few or
 - Use CURRENT PACKAGESET special register

- Logging
 - Separate SSIDs in case non-recoverable DASD SSID failure (current active log)
 - How many archives created/day and how much of that is active log? Increase active logs?
 - Is DASD log sufficient for IC cycle of critical objects

Multi Site Locking Issues

- For z990 > sync lock requests can be about 10 usec with fast links or up to 20 usec w/slower ones
- System Managed duplexing of Lock/SCA
 - CF – CF signals also needed
 - Locally about 4X a simplex lock (10-15 usec vs. 40-60 usec)
- Distance adds 10 usec/km at best (speed of light) round trip
 - If sites are 30KM (~20 mi) = 300 usec
 - If duplexed, 3x-4X or about **1 ms/lock request**
 - Not practical
- Only customer seen with SM Duplexing - across 5KM
 - 5400 req/sec with 95 usec sync and **225 async** usec 4 way DS Group
 - Most requests async

DSNZPARMS

ZPARAM	New value	Reason
LOGAPSTG	100	FLA for mass object recovery and LPL/GRECP recovery – can improve recovery times by 10x. Make sure storage cushion is 200M, to allow for it when needed)
CTHREAD		Set based on MEMU2 Max Threads - one of 3 major parms for DBM1 virtual storage cushion needed
MAXDBAT		Set based on MEMU2 Max Threads. Does not contribute to CTHREAD, but does for DBM1 (one of 3 major parms for DBM1 virtual storage).
IDBACK		Make sure enough. RRS attach (SPs and JDBC T2 connections) count as batch threads and so do parallel utility tasks. IDFORE+IDBACK <
DSMAX		one of 3 major parms for DBM1 virtual storage
CHKFRFQ	5	Recommended for z990 configurations – affects URCHKTH to be used for long running update job messages
PCLOSEN	5	With CHKFREQ=5, will occur at 25 minutes
PCLOSET		Usual goal is < 10-20/minute
SMFSTATS	1,3,4,5,6	(6 includes IFC225 for Virtual Storage Map of DBM1 and SAP trans DB2)
STATIME	5	better peak identification and 12 SMF records generated vs 4 per hour is
URCHKTH		With CHKFREQ at 5, pick the value according to customer commit start with 10 (100,000) and reduce - no. of updates a single UOW is allowed before warning message issued.
URLGWTH	10	
SYNCVAL	0	synchronize Statistics records to same time and to RMF, usually
LBACKOUT	AUTO	
BACKODUR		Only backs out 50K records/interval value. That is, 20=1M records
IDXBPOOL	49	unassigned BP
TBSBPOOL	49	unassigned BP
DLDFREQ	5	detect volume restored/backlevelled outside DB2 – externalized starting LRSN for log apply with Logonly recovery
OUTBUFF	40000	increases reads found in buffer vs. active log if significant active log reads occur - buffers are in MSTR, not DBM1 VSTOR