

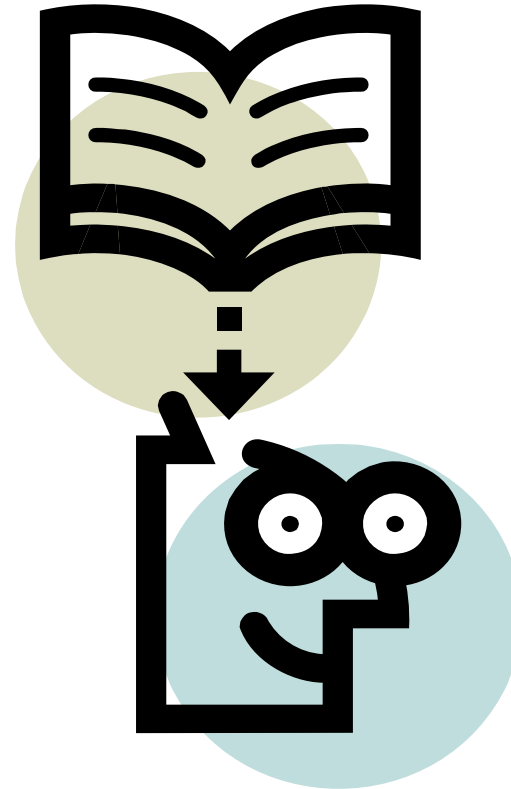


DB2 for z/OS V8 Early Experiences

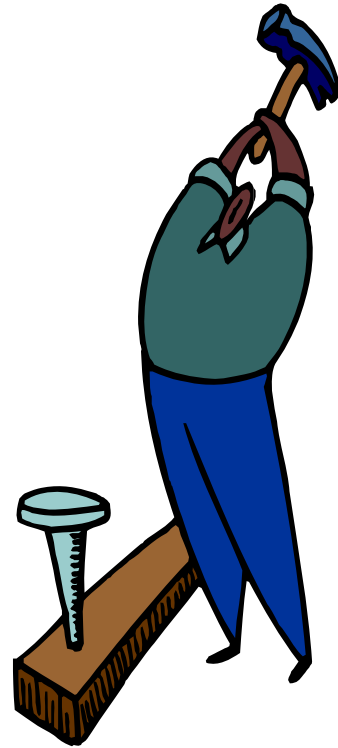
John J. Campbell
Distinguished Engineer
DB2 for z/OS Development

Agenda

- Quick Hits
- Code Page Issues
- Unicode Catalog
- Migration Plans
- Resource Utilization
- SQL and Performance
- Summary



Quick Hits



Inhibitors

- Cleaning up code page issues
- Planning and preparation for significant new release
- Vendor code
- Cleaning up V8 incompatible changes
- Old COBOL
- Real storage
- Shortage of change windows
- 4Q2004 Change Freeze



Secrets to a successful migration

- Spend time and effort to keep fire away from production
- Code page clean up
- Pre-migration catalog migration testing
- Batch regression testing
- Monitor and control virtual and real storage usage prior to and during V8 COMPAT mode
- Take advantage of CST/RSU plus Enhanced HOLDDATA
- Reorganize Catalog and Directory objects under V7 prior to start of V8 migration
- Rebind all plans with no PKLIST prior to start of V8 migration
- Set zparm ABIND=COEXIST when running in release coexistence
- Keep period of coexistence reasonable short
- Rebind packages and plans which are associated with high volume and performance sensitive applications in V8 COMPAT mode
- Reorganize Catalog objects with hash links under V8 COMPAT mode before entering V8 ENFM
- Schedule an outage for ENFM/NFM and get it out of the way

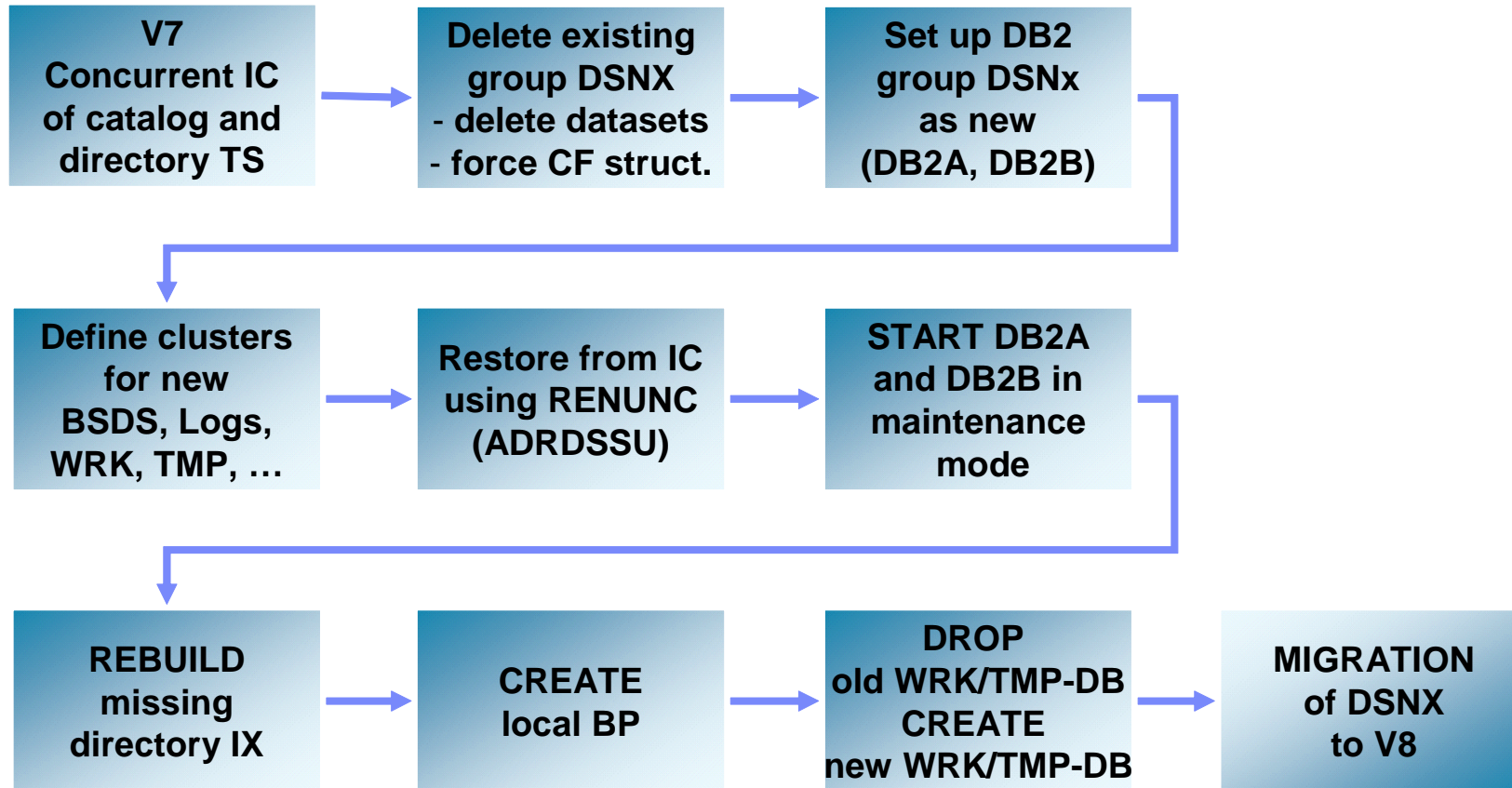


Pre-migration catalog migration testing

- Clone a production catalog and practice migration on the cloned copy without risk and before migrating the catalog of any productive system
 - ▶ Options: Concurrent Copy, Flashcopy, DFSMS Dump, DSNTIJIC
 - ▶ Practice migration:
 - V7-> V8 plus CATMAINT
 - V8 CMP mode -> V8 ENFM mode -> V8 NFM mode
 - Rebind plans and packages
 - Explain all views
- Possible extend this test to an extended regression test system
- Engage other parties into the migration and pre migration process and testing
 - ▶ DBA's, Application developers....



Pre-migration catalog migration testing ...



DFSORT and DB2 Utilities

- V8 removes the pre-req for an external sort product
- V8 will only use the DFSort package: SORT, MERGE functions only
 - ▶ Aim to improve reliability and performance
 - ▶ DFSort code is part of a standard z/OS install
 - ▶ DB2 UDB for z/OS Utilities Suite has a license API to use DFSort when invoked by the utilities
 - ▶ Must have access to DFSort R14 plus APAR PQ68263 applied



DFSORT and DB2 Utilities Setup

- If DFSORT is installed as the primary sort package and multiple versions of DFSORT exist
 - ▶ Place DFSORT R14 plus APAR PQ68263 first in system search order
- If DFSORT is NOT installed as the primary sort package, DFSORT R14 plus APAR PQ68263 must be accessible via
 - ▶ DFSORT SORTLPA library is added to the LPALST and SICELINK library is added to the LNKLST, or ...
 - ▶ DFSORT libraries SICELINK and SORTLPA, in that order, are added to the LNKLST, or ...
 - ▶ DFSORT libraries SICELINK and SORTLPA, in that order, are added to the JOBLIB or STEPLIB DD statement, or ...
 - ▶ DFSORT modules are in a private library that is equivalent to one of the above configurations.



Console Security and SYSOPR

- Commands coming through console come through with id of SYSOPR
- Old behavior (V7 and prior)
 - ▶ For logged on user (TSO/SDSF), id passed through for console commands is still SYSOPR
 - ▶ zparm SYSOPRn=SYSOPR allows any user to enter commands from console
- New behavior (V8)
 - ▶ For logged on user, id passed through for console commands is now the user and not SYSOPR
- RACF groups and secondary authids now supported



Dumps and Standalone Dumps

- Allow enough REAL/paging space
- Allow enough DASD
 - ▶ Striped volumes for dumps bigger than a single volume
 - ▶ For example, in SMS ACS routine let the volumes do the work until 2.5GB, then switch to striped volumes
 - ▶ `FILTLIST DB2DUMP INCLUDE (DB2DUMP.**)`
 - ▶ `WHEN (&DSN=&DB2DUMP)`
 - ▶ `IF &SIZE < 2500MB THEN SET &DATACLAS = "`
 - ▶ `ELSE SET &DATACLAS = 'STRIPE'`
- Use DUMP groups
 - ▶ Allows 48GB
 - ▶ All volumes can be written concurrently (much faster)
 - ▶ Allows 16 volumes of the same device type



Zparm Default Changes

- Describe for static V8 is set to yes - some DB2 supplied stored procedures (Management Clients Package)
- Log apply storage in V8 is 100 - people tend to use 10, which is too low. This is needed for fast log apply during recovery Also, Backup/Restore System needs it for 4x better performance

Panel	Field	Parameter	Version 7 default value	Version 8 default value
DSNTIP7	USER LOB VALUE STORAGE	LOBVALA	2048	10240
DSNTIPE	MAX USERS	CTHREAD	70	200
	MAX REMOTE ACTIVE	MAXDBAT	64	200
	MAX REMOTE CONNECTED	CONDBAT	64	10000
	MAX TSO CONNECT	IDFORE	40	50
	MAX BATCH CONNECT	IDBACK	20	50
DSNTIPN	DDF/RRSAF ACCUM	ACCUMACC	NO	10
DSNTIP8	CACHE DYNAMIC SQL	CACHEDYN	NO	YES
DSNTIPP	PLAN AUTH CACHE	AUTHCACH	1024	3072
DSNTIPL	LOG APPLY STORAGE	LOGAPSTG	0	100
	CHECKPOINT FREQ	CHKFREQ	50000	500000
DSNTIPA	BLOCK SIZE	BLKSIZE	28672	24576
DSNTIPR	DDF THREADS	CMTSTAT	ACTIVE	INACTIVE
	IDLE THREAD TIMEOUT	IDTHTOIN	0	120
	EXTENDED SECURITY	EXTSEC	NO	YES
DSNTIP5	TCP/IP KEEPALIVE	TCPKPALV	ENABLE	120
DSNTIPC	MAXIMUM OPEN DATA SETS	DSMAX	3000	10000
	EDMPOOL STORAGE SIZE	EDMPOOL	7312	32768 ¹

Note:

¹ The installation CLIST calculates the default value for the EDMPOOL parameter.

Should I use a PDS or PDSE?

- DSNALLOC job allocates A/SDSNLOAD as PDSE
 - ▶ Development anticipated DSNXGRDS LMOD to exceed 16m limit during QPP
 - ▶ DSNXGRDS LMOD now not expected to exceed 16m limit in V8
- Can allocate ADSNLOAD as PDSE (advantages: it's not shared, less compressing needed)
- Can allocate SDSNLOAD as PDS or PDSE
- If SDSNLOAD allocated as PDSE:
 - ▶ IMS Attach requires IMS toleration APAR PQ79118
 - ▶ Be aware of operational differences between PDS and PDS/E's (see DB2 V8 program directory section 5.2.2.2)
- Most customers install DB2 systems with PDS



Leftovers

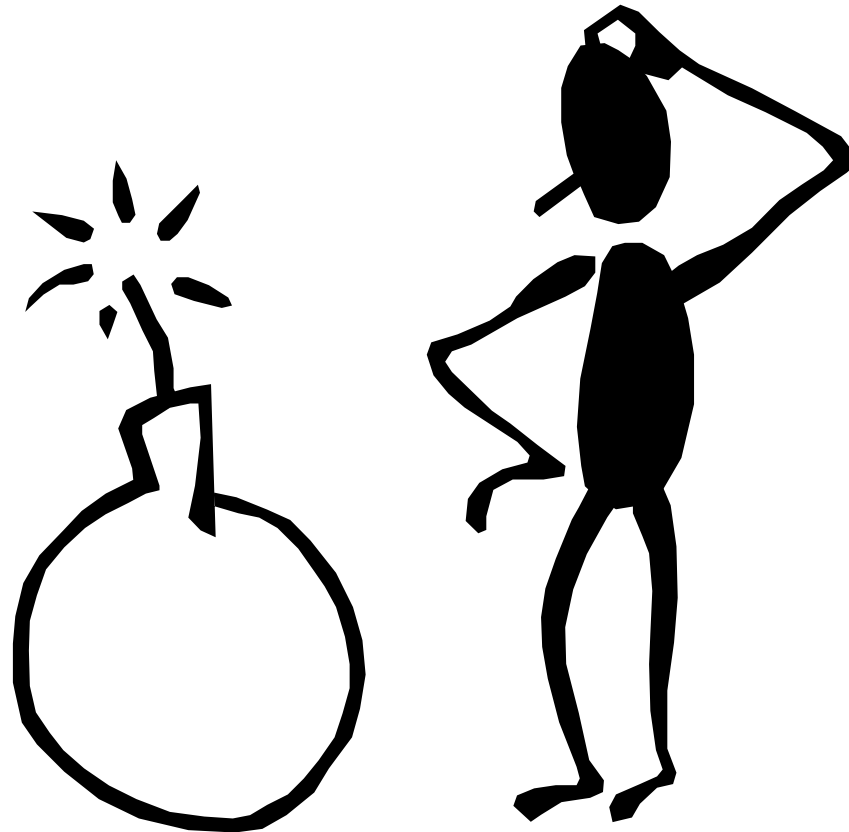
1. Bimodal Migration Accommodation is not supported by DB2
2. ICSF Modules to avoid Nasty Grams - SCSFMOD0 in Link List
3. Use ISPF APAR OA07685 to browse Unicode DBRMs
4. Tools Certification is important, don't wait to verify

http://www-1.ibm.com/support/docview.wss?rs=434&context=SSZJXP&uid=swg21162152&loc=en_US&cs=utf-8&lang=en+en

5. After the V8 migration, every problem becomes your problem
 - ▶ Collect SMF data and reports
 - ▶ EXPLAINS of key packages
 - ▶ Understand system resource utilization before the migration
 - ▶ Standard, sample job stream that exercises the system
6. Regular status calls are very valuable



Code Page Issues



If one encoding scheme is good...

- Potential for incorrect character interpretation (i.e., data corruption) if multiple CCSIDs within one encoding scheme
 - ▶ Terminal emulator using incompatible code page
 - ▶ Incorrect CCSID specification
 - ▶ Change of CCSID specification
- Must be fixed and residual corruption cleaned up ahead of V8
 - ▶ 28 Step or Zap or Unload/Drop/Create/Load? It depends...
- V7 APAR PQ56697 introduces checks for non-zero valid CCSIDs and issues warning when trouble found
- V7 APAR PQ89018 will provide CCSID detection if DECP value does not match the emulator value



Unicode Catalog



Unicode Catalog

- Run DSNTIJP8 (V7) or DSNTIJPM (V8) ahead of migration to check that there are not multiple CCSIDs within one encoding
- V8 start up checks for:
 - ▶ Non-zero, valid CCSIDs
 - ▶ EBCDIC<->Unicode round trip conversion
- Must configure Unicode Conversion Services for z/OS
 - ▶ Unicode CCSIDs (367,1208,1200) <-> ASCII/EBCDIC CCSIDs
 - ▶ Client CCSIDs <-> Unicode CCSIDs (367,1208,1200)
 - ▶ Must also add to/from Unicode CCSIDs (367,1208,1200)
 - CCSID 37 for DBRMs provided with DB2 install
 - CCSID 500 for DRDA
 - CCSID 1047 for Unix System Services (USS)
 - ▶ Rebuild conversion image, new image picked up by DB2 'on the fly'
- Configure for best conversion performance
 - ▶ z890 and z990 zSeries (hardware instructions instead of mcode/ucode)
 - ▶ z/OS R4



Unicode Catalog ...

- SQL parsed and metadata derived from SQL in Unicode UTF-8 format
- Most character columns converted to Unicode VARCHAR(128)
- Trailing blanks not stripped
- Special characters expand beyond single byte (e.g., #)
- Increase in space: CHAR -> VARCHAR (length bytes)
- Decrease in space: DB2 defined indexes converted to NOT PADDED
- Consider converting user defined indexes to NOT PADDED
- Compatibility (COMPAT) mode
 - ▶ SQL parsed in Unicode UTF-8 format
 - ▶ Metadata derived from SQL converted back to EBCDIC
- New Function Mode (NFM) mode
 - ▶ Catalog converted to Unicode UTF-8 format
 - ▶ Metadata derived from SQL no longer converted back to EBCDIC



Unicode Catalog ...

- Use zparm UIFCIDS option to have IFCID trace data in Unicode UTF-8 format
 - ▶ Default = YES which moves conversion overhead off-line e.g., DB2PE
- Conversion of SQL statements and derived metadata to Unicode
 - ▶ Minor Conversion (fast) for simple alphanumeric characters
 - Optimization for most common single byte CCSIDs (MIXED=NO)
 - Most common case ASCII/EBCDIC->UTF-8, can go other way
 - Internal 256 byte translation tables used by DB2 (TR/TRT)
 - ▶ Major Conversion (slower)
 - Offloaded to Unicode Conversion Services for z/OS
 - Used if EBCDIC/ASCII->UTF-8 cannot be performed inline
 - Always for ASCII/EBCDIC->UTF-16 (no optimization)
- Particular conversion not available (SQLCODE -332)
- SQL SUBSTR behavior can be inconsistent when accessing Catalog data
 - ▶ New character based functions added through APAR PQ88784



Unicode Catalog ...

- If precompile application with NEWFUN(NO)
 - ▶ Disables use of new SQL function for static SQL
 - ▶ Go back through SQL Parser during BIND processing
 - To get Unicode UTF-8 format
 - Incur extra overhead
- After entry to NFM, default for NEWFUN changes to YES
 - ▶ DBRMs switch to Unicode UTF-8 format
 - ▶ Set NEWFUN to NO
 - Get DBRMs in EBCDIC format
- At BIND time
 - ▶ Anything from DBRM in Unicode UTF-8 format goes into SYSSTMT and SYSPACKSTMT in Unicode UTF-8 format
 - ▶ Anything from DBRM in EBCDIC format goes into SYSSTMT and SYSPACKSTMT in EBCDIC format



Migration Plans



Migration Plan Initial Thoughts

- Most customers do not have an operational proving test environment and application regression workloads
 - ▶ Regression and stress test to flush out implementation issues
- Do not want to run for an extended period with different code and function levels across 'test' and 'production' environments
- Have clear separation between 'toleration' and 'exploitation' of features in the new release
 - ▶ Exploitation project would follow on from toleration project when the new release has been running stable in production for a reasonable period (typically complete one processing month cycle) and there is little chance of fallback to previous release of DB2
 - ▶ Exploitation project will focus on a prioritized list of new features offering business value. During the toleration period customer will run with release N-1 of the DB2 Precompiler to prevent new SQL features being used.
- Distinguish carefully
 - ▶ Data sharing 'coexistence'
 - ▶ 'Compatibility mode'



Migration Plan Recommendations

1. Migrate V7 test data sharing group to V8 CM and hold. If want to roll in the new release member by member, migrate the first 2 members, wait, then roll through the remaining members quickly.
2. After (1) has been proven for extended period, migrate production data sharing group to V8 CM and hold. If want to roll in the new release, migrate the first 2 members, wait, then roll through the remaining members quickly.
3. Run through complete monthly processing cycle in production to flush out implementation/regression issues - can fallback/come forward again if necessary
4. After (3) is successful, start to migrate V8 test data sharing group towards V8 NFM mode
5. After (4) is successful, start to migrate V8 production data sharing group towards V8 NFM mode



Migration Plan Recommendations ...

- Entry to CM, ENFM, NFM modes are instantaneous group wide events
- SYSCOPY stays as EBCDIC, not converted to Unicode
- New tablespace called SYSEBCDC contains a single row/byte to support SYSDUMMY
- Run DSNTIJP8 ASAP under V7 to check for rogue conditions in V7 Catalog prior to V8
 - ▶ Mode (CM, ENFM, NFM) is stored away in Catalog and Directory
- DSNTIJNE job (ENFM) can be restarted from the beginning without change
 - ▶ Objects reorg'd in a specific order
- Up to 10% increase in disk space requirement for Cat/Dir
 - ▶ Some objects shrunk by 50% because they had never been reorganized



Migration Plan Recommendations ...

- Cannot start DB2 V8 with default DSNHDECP
 - ▶ DSNHDECP -> SDSNEXIT
 - ▶ DSNHMCID -> SDSNEXIT
- What do you get with V8 CM
 - ▶ 64-bit virtual
 - ▶ Long term page fix
 - ▶ New access paths
 - ▶ New utility function in V8 CM
 - ▶ No new SQL function
- Dataspace bufferpools and Hiperpools converted to primary virtual pools above 2GB bar on first start of member on V8
- Possibility of different query result sets when using range predicates
 - ▶ Can use cast function in NFM mode to get back to EBCDIC
- Issue -ALTER BP with non-zero value prior to V8, and define corresponding GBP structures in CFRM policy
 - ▶ BP8K0 and BP16K0 (and BP32K)



Migration Plan Recommendations ...

- Understand DASD space requirement and measure of elapsed time
 - ▶ Run REORG SHRLEVEL(CHANGE) in CM mode
- Online Schema
 - ▶ Trailing blanks are not removed on ALTER CHAR->VARCHAR
- SYSREC2 dataset added to some Online REORGs jobs as work space because some characters in SYSTRIGGERS and SYSDBASE span multiple rows
- Job DSNTIJTC has to be run even for new installs to fix national characters in CCSID 1047 to Germany (CCSID 273), Norway, etc.
- Feedback in SQLCA following SQL CONNECT without parameters indicates 'DSN' for DB2 and >5 for NFM mode
- User defined indexes on Catalog would swell up in size during ENFM mode
 - ▶ DROP before ENFM mode
 - ▶ Re-CREATE after ENFM as NOT PADDED



Migration Plan Recommendations - COBOL

- COBOL
 - ▶ V8 supported by COBOL V3.2 (EOS Oct 2005), Enterprise COBOL V3.2 or later
 - ▶ Other compilers are no longer supported
 - OS/VS COBOL has been out of support since roughly 1994
 - VS COBOL II went out of support in 2001
 - ▶ COBOL II load module running in a LE environment will still work
 - ▶ Must re-link all COBOL II load modules to run in a LE environment if they currently do not do so
 - ▶ Understand the options
 - See Compiler and Run-time Migration Guide
 - No OS/VS COBOL or VS COBOL II compiles
 - Run old COBOL modules under LE
- Set strategy and plan
 - ▶ Use V7 Precompiler?

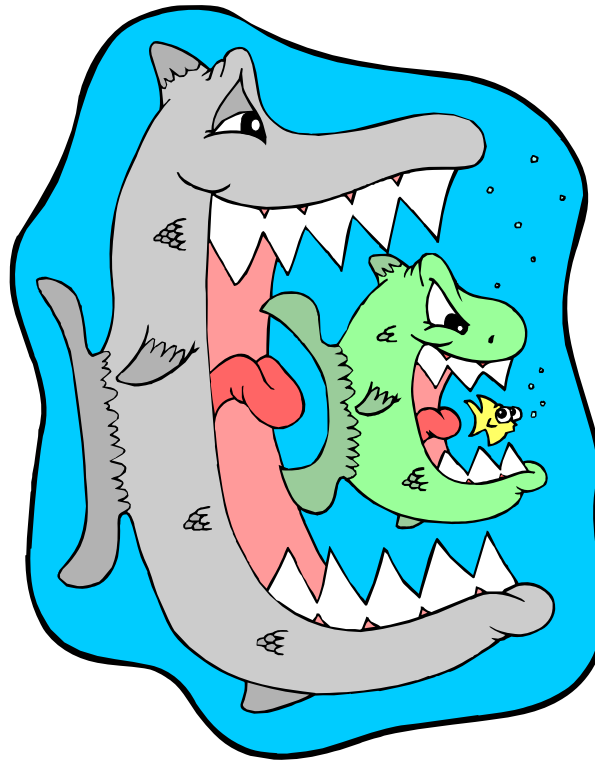


Migration Plan – COBOL Conversion Tool

- Debug Tool and Advanced Functions
 - ▶ Converts older COBOL source to Enterprise COBOL source
 - ▶ “CCCA” - CICS and COBOL Conversion Aid
 - Continue to run your older COBOL under LE and DB2 V8
 - When a source change is needed, use the tool to convert the source and then make the other changes
 - IBM does not have a tool that will disassemble object code and convert that assembler code back to COBOL
- <http://www-306.ibm.com/software/awdtools/deployment/>
- <http://www-306.ibm.com/software/awdtools/debugtool/>



Resource Utilization



How much will you gain from 64-bit virtual ?

- At first sight, re-engineering of DBM1 to exploit 64-bit should make a very significant difference in terms of providing significant VSCR
- Now consider V6 or V7 installation achieving significant VSCR
 - ▶ Maximum use of Dataspace bufferpools (400-800MB)
 - ▶ Using dataspace extension to EDM Pool for GDSC (80-160MB)
 - ▶ Use of zparms CONTSTOR=YES and MINSTOR=YES
 - ▶ Reduced number of persistent threads
- What is the 'net' benefit in V8
 - ▶ Dataspace Lookaside Pool (80-120MB) is eliminated
 - ▶ Buffer Manager control blocks (70-120MB) going above 2GB bar
 - ▶ Other items going above the 2GB bar
 - Compression dictionaries
 - Certain EDM/RID Pool components
 - Thread Sort Pool
 - etc



64-bit and Thread Storage

- Most of the thread storage stays below the 2GB bar
 - ▶ Agent Local, Stack Storage
 - ▶ Local Dynamic Statement Cache (0-400MB)
 - ▶ Expect some regression
- Regression estimates are very preliminary and subject to change as more customer data are obtained
 - ▶ Thread storage: +30 to 40% for static or system, +50 to 100% for dynamic
 - ▶ Stack storage: +50 to 100%
 - ▶ Dynamic statement cache: +100%
 - ▶ EDM pool: roughly the same
 - ▶ RID pool: -90%
 - ▶ Others: -100%

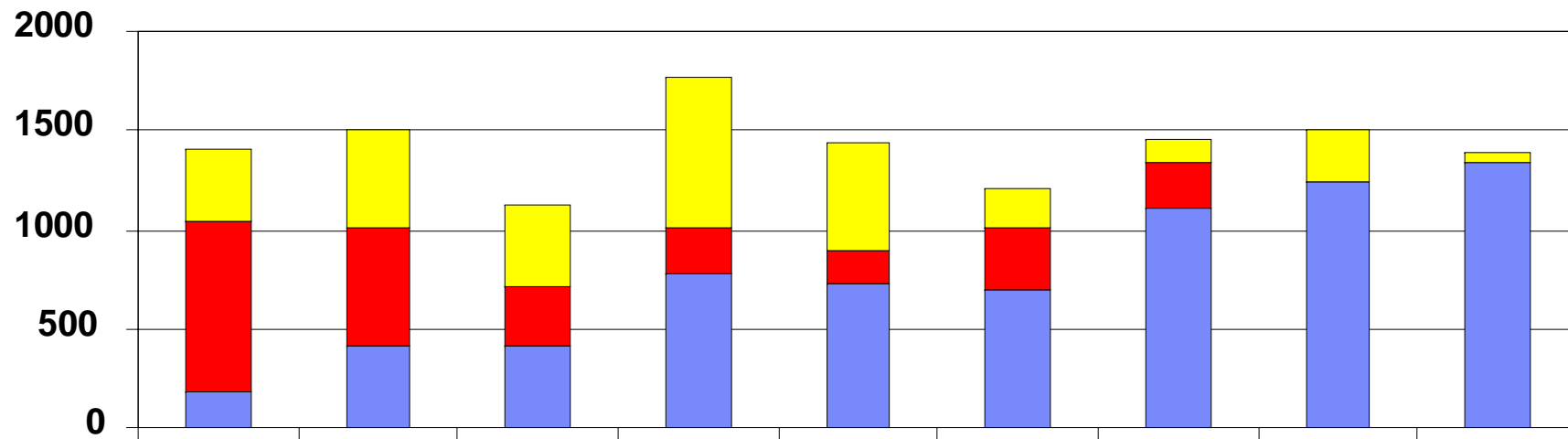


64-bit and Thread Storage ...

- How many additional threads (?) can be supported will depend on 'Thread Footprint' which will vary by workload depending on
 - ▶ Duration of thread
 - ▶ SQL workload
 - ▶ RELEASE parameter setting on plan/package bind
 - ▶ Effectiveness of thread storage contraction (CONTSTOR=YES)
- Net benefit might only work out at 300-500MB for some installations, much less for other installations
- Still need to monitor and track VSTOR usage with IFCID 225
- IFCID 225 now available via READS with additional fields



DBM1 below 2GB available, used by V7, used by V8 High-end peak period data from



64-bit Messages

- Some key messages
 - ▶ 64-bit support will not absolutely eliminate VSC below the 2GB bar in DBM1
 - ▶ Will provide valuable additional relief, but will vary by installation
 - ▶ Will be able to exploit all available processor storage on latest processor models (currently 256GB, current DB2 limit of 1TB)
 - Some additional number of active threads and DBATs
 - May be able to set zparms CONTSTOR=NO and MINSTOR=NO
 - XXL bufferpools to eliminate I/O and speed up remainder
 - Provides a good real memory vs IO trade off
 - Increase exploitation of ESA Compression
 - Larger thread Sort Pool
 - ▶ Must have sufficient real storage to fully back increased usage
 - ▶ Installations must continue to plan for, monitor and tune VSTOR usage below 2GB bar
 - ▶ Get current service for storage, monitor info APAR II10817



64-bit and EDM Pool

- V7 option to have EDM data space for Cached Dynamic Statements
- No such option in V8
- EDM Pool split into three specific pools
 - ▶ EDM Pool below 2GB Bar will now only contain CTs, PTs, SKCTs, SKPTs
 - Might be able to reduce the size of this EDM pool
 - Provide some VSCR for below the 2GB Bar storage
 - Monitor $(FREE+SKPT+SKCT)/TOTAL$ pages
 - ▶ EDM Pool above the 2GB Bar
 - DBDs
 - Cached Dynamic Statements
 - ▶ Use SET SYSPARM LOAD to increase/decrease these pools as needed: EDMPOOL, EDMDBDC, EDMSTMTC



Real Storage Use

- Increase with V8 in real storage usage
 - ▶ Typically in the range of 1% to 10%
 - ▶ Many watch virtual, few watch real, which is a concern
- Important subsystems such as DB2 should not be paging to auxiliary storage
- Strong recommendation
 - ▶ Keep DASD paging rates low (near zero)
 - ▶ Monitor via RMF Mon III



CPU Performance Regression

- Dramatic new function delivered in V8
- Performance objective is less than 10% average CPU regression
- Typical customer workload regression is expected to be 5 to 10% higher on average
- This assumes no application rewrite and no aggressive configuration change to take advantage of V8 significant performance features
 - ▶ 0 to +15% Online Transaction
 - ▶ -5 to +10% Online Transaction in Data Sharing
 - ▶ 0 to +20% Batch
 - -5 to 5% INSERT
 - +5 to +20% SELECT, FETCH, UPDATE
 - ▶ -10 to +15% in Batch Data Sharing
 - ▶ -20 to +15% Batch DRDA
 - ▶ -10 to +10% Utility
 - ▶ -20 to +15% Query



CPU Performance Regression - Antidotes

- Some options with significant potential to offset expected increase
 - ▶ Rebind of all plans and packages
 - ▶ Long Term Page Fix by Bufferpool
 - ▶ Multi-Row FETCH and INSERT
- The following types of workloads tend to have less regression in V8:
 - ▶ Virtual storage constrained (zparms MINSTOR=YES, CONTSTOR=YES)
 - ▶ Dataspace buffer pool or Hiperpool in V7
 - ▶ SMF89 active in V7
 - ▶ IRLM PC=YES in V7
 - ▶ CURRENTDATA NO in V7
 - ▶ BIND RELEASE COMMIT option in V7 Data Sharing
 - ▶ Update intensive batch processing in V7 Data Sharing
 - ▶ I/O intensive workload



Long Term PGFIX

- Strong recommendations
 - ▶ Ensure bufferpool 100% backed by real storage
 - ▶ Use Long Term Page Fix by Bufferpool to help compensate for CPU performance regression
- Typically results in the range of 0 to 8% improvement
- Activated by `-ALTER BPOOL(x) PGFIX(YES)`
- Page fix all buffers just once
 - ▶ Avoids expensive page fix and page free for every IO in 64-bit
- Performance benefit is inversely proportional to the buffer hit ratio
 - ▶ Higher the hit ratio, the lower the benefit
- Apply to small buffer pools with lots of IOs e.g., sequential work
 - ▶ Greatest potential benefit
 - ▶ Minimize potential impact on other running workloads

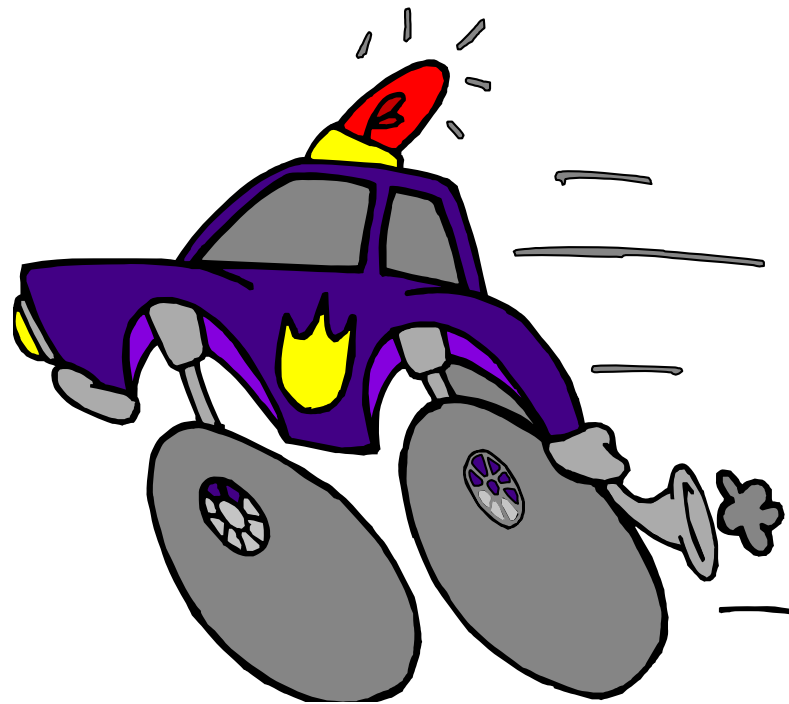


Performance Hits

- Take advantage of very large real storage
 - ▶ Very large bufferpools
 - Must have sufficient real storage to fully back the pools
 - Eliminate sync IO and improve elapsed time performance
 - For example, can improve batch by order of magnitude
- Query performance
 - ▶ All practical reasonable cases can now take advantage of index access
 - ▶ Recommendation to run RUNSTATS with KEYCARD option (not the default)
 - ▶ RUNSTATS collection of Distribution Statistics can help fix many access path selection problems



SQL and Performance



Online Schema Evolution

- SQL ALTER leaves tablespace in AREO
 - ▶ Up to 30% CPU increase until REORG
 - Fast column processing disabled
 - Data conversion
 - ▶ Extra logging
- SQL ALTER can leave index in RBDP
 - ▶ Invalidation of plans and packages
 - ▶ For dynamic queries, indexes that are in RBDP state are ignored
 - If the index is on a partitioned tablespace
 - All the parts (logical or physical) need to be in RBDP
 - If ANY part is available then the index will not be ignored
- Conversion from non-large index-controlled partitioning leaving partition(s) in REORP



Online Schema Changes ...

- Be aware of standard restrictions which apply related to:
 - ▶ RI, FIELDPROCs, VALIDPROCs, EDITPROCs
 - ▶ IDENTITY columns, MQTs
- Rotate performance can be ugly with many NP(S)Is
 - ▶ Subject partition must be emptied ready for new data
 - Individual row deleted and NPSIs corrected serially
 - Increased logging
 - Logical drain of NP(S)Is
 - Lock on DBD for extended period
 - ▶ PIT recovery not possible for rotated partition
 - ▶ Recommendations
 - Unload or copy data out as an archive
 - LOAD REPLACE PART 1 with dummy input dataset
 - ALTER TABLE ALTER PARTITION ROTATE FIRST TO LAST



Automatic Space Management

- SMART enhancement: sliding secondary allocation quantity size
- Applies to DB2 managed pagesets only
- Tries to avoid VSAM maximum extent limit errors
- Can reach maximum dataset size before running out of extents
- Uses cylinder allocation
 - ▶ Default PRIQTY
 - 1 cylinder for non-LOB tablespaces and indexes
 - 10 cylinders for LOB tablespaces
 - ▶ Improved SQL SELECT and INSERT performance
 - 2x improvement relative to track allocation
- Can be used for
 - ▶ New pagesets: No need for PRIQTY/SECQTY values
 - ▶ Existing pagesets: SQL ALTER PRIQTY/SECQTY values to -1 (minus) plus schedule a REORG

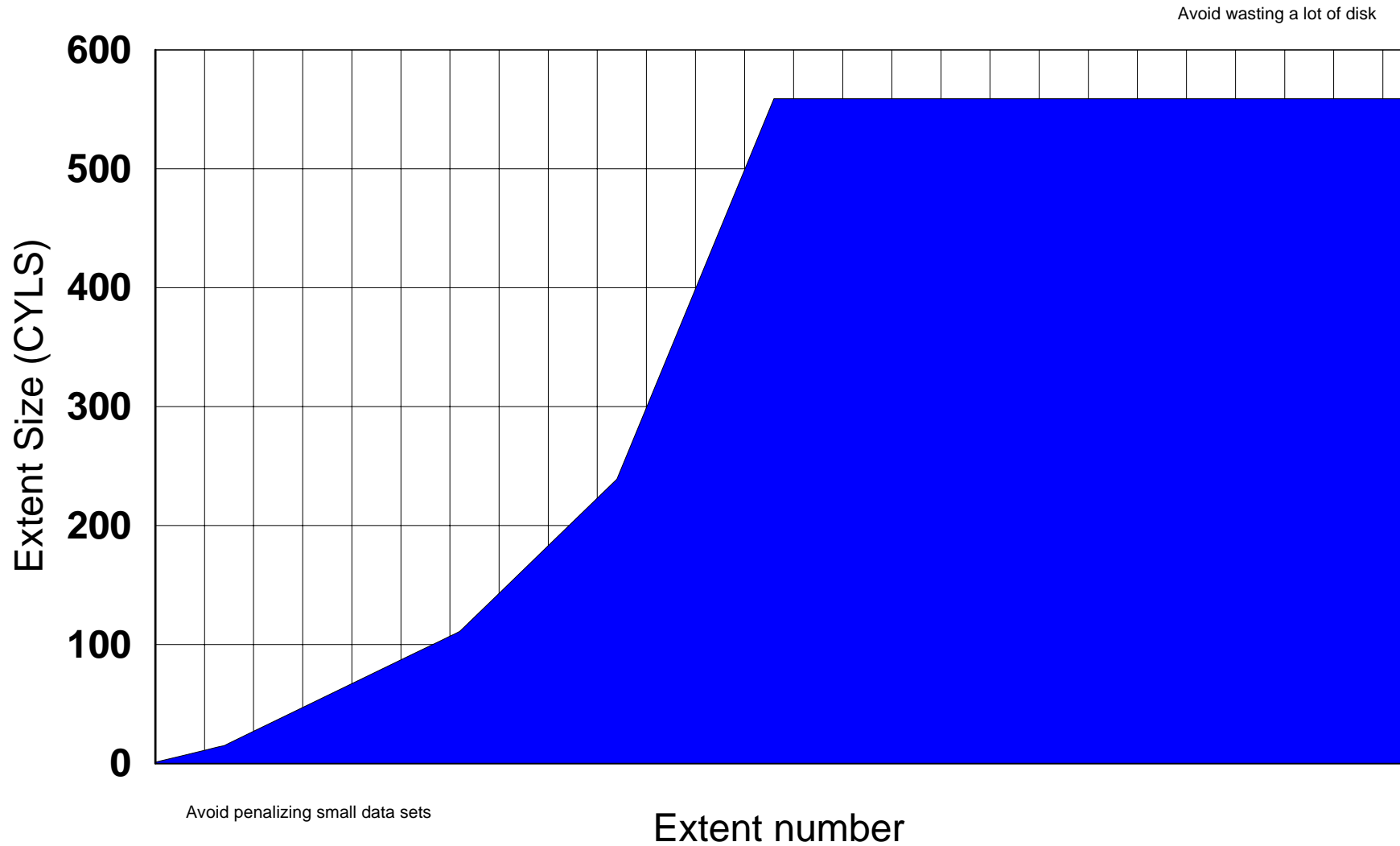


Automatic Space Management ...

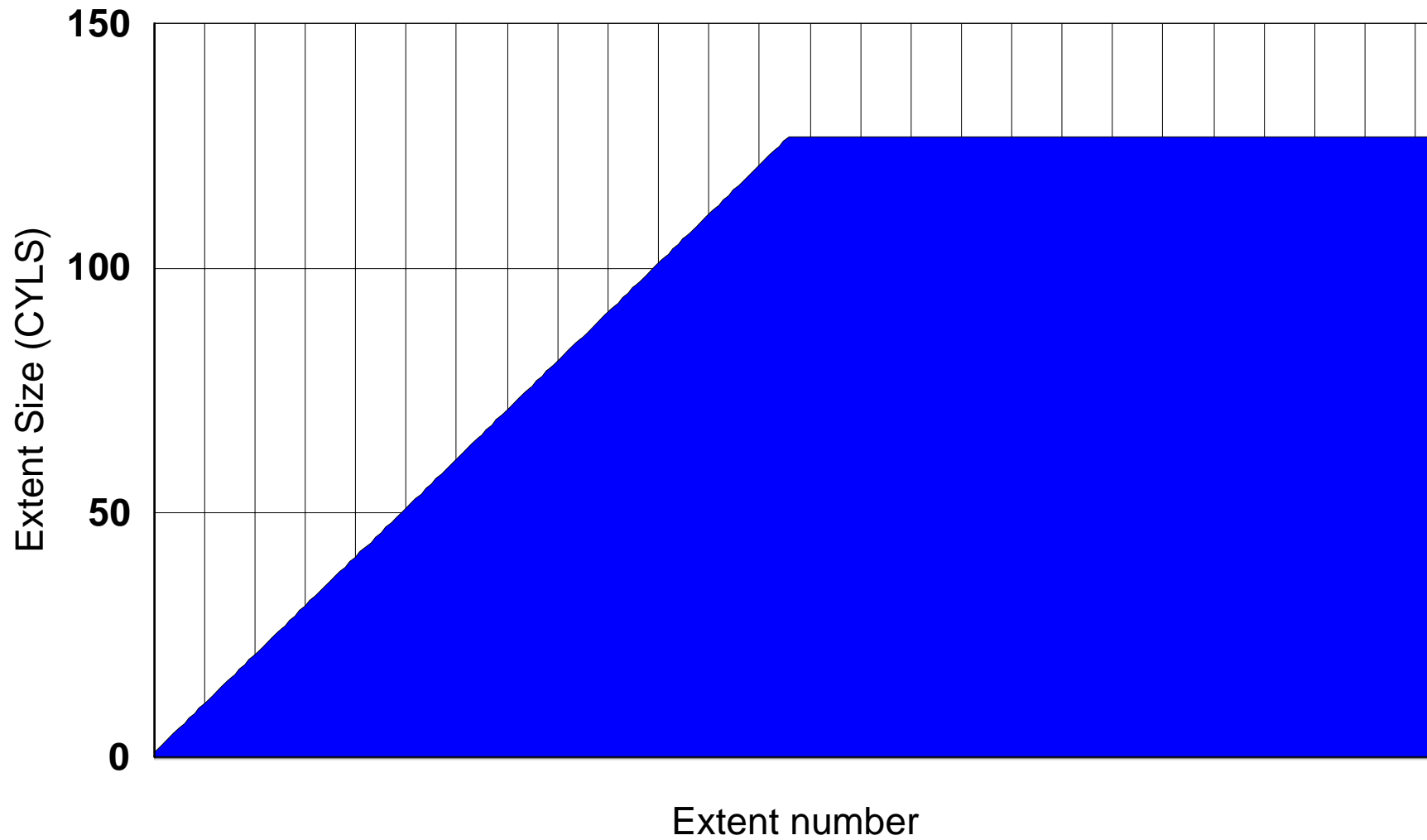
- Two sliding scales will be used depending on maximum dataset size
 - ▶ 32GB and 64GB data sets
 - ▶ Less than 1GB, 1GB, 2GB, 4GB, 8GB, 16GB datasets
- Maximum dataset size determined based on DSSIZE, LARGE and PIECESIZE
- Both sliding scales will allocate
 - ▶ Increasing secondary quantity size up to 127 extents
 - ▶ Constant number there after
 - 559 cylinders for the 32GB and 64GB datasets
 - 127 cylinders for the rest
- Sliding scale minimizes potential for wasted space by
 - ▶ Increasing the secondary extent size slowly at first
 - ▶ Avoiding very large secondary allocations from extents 125-255
- Particularly helpful for users of ERP/CRM vendor applications



Sliding Scale for 32GB and 64GB Data Sets



Sliding Scale for Other Data Sets



SQL – New and Improved

- MQTs
 - ▶ Query rewrite only for dynamic SQL, static SQL can reference MQTs directly
 - ▶ Use zparm SPRMMQT to prevent unnecessary additional prepare overhead for short running queries
 - ▶ SQL REFRESH
 - Use segmented tablespace for instantaneous mass delete (resets space map)
 - Run RUNSTATS afterwards
- DPSIs
 - ▶ Great benefit for utility performance
 - Partition LOAD / REORG / REBUILD
 - ▶ But can be significant drawback for query processing
 - For example
 - Where no predicate to restrict query to particular partition
 - ORDER BY on DPSI column
 - Sequential detection and Index Lookaside impacted



SQL – New and Improved ...

- Multi-Row INSERT
 - ▶ Avoids SQL application interface 'round trip'
 - Multi-row Insert for 2 rows up to 8% better than single row INSERT
 - Up to 29% better for 10 rows
 - ▶ Dramatic reduction in network traffic
 - Avoids message send/receive pair for each row
 - Potential for huge elapsed time reduction (up to 8x)



SQL – New and Improved ...

- Multi-Row FETCH
 - ▶ Avoids SQL application interface 'round trip'
 - 2 rows is 0.4% worse than single row
 - But 10 rows is 33% better
 - ▶ Dramatic reduction in network traffic
 - Avoids message send/receive pair for each row
 - Potential for huge elapsed time reduction (up to 8x)
 - ▶ Automatically enabled for DRDA when
 - Read-only
 - Ambiguous cursor with CD(N)
 - ▶ Sample programs upgraded to exploit
 - DSNTEP4 (identical function to DSNTEP2)
 - DSNTIAUL



Other Changes

- CI Size = Page Size (Page Size > 4K)
 - ▶ Eliminates partial write problem where 32K page crosses device boundary (16K block size used)
 - ▶ Applies to DB2 managed objects after entry to NFM mode
 - ▶ Implemented via REORG
 - ▶ Significantly higher data rate for page size > 4K
 - ▶ VSAM striping now possible for page size > 4K
- Page Distance Limit for prefetch and castout operations
 - ▶ 180 page distance limit applied to single IO is removed
 - ▶ 3-4% CPU reduction for OLTP workloads
 - ▶ Does not apply to deferred write



Summary

- Quick Hits
- Code Page Issues
- Unicode Catalog
- Migration Plans
- Resource Utilization
- SQL and Performance

