IBM Software Group

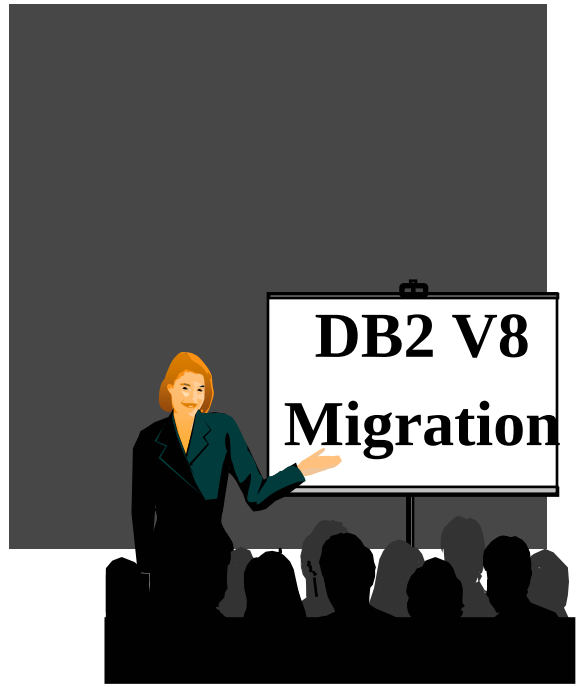
Migrate and Optimize DB2 UDB for z/OS Version 8

DB2 Information Management Software

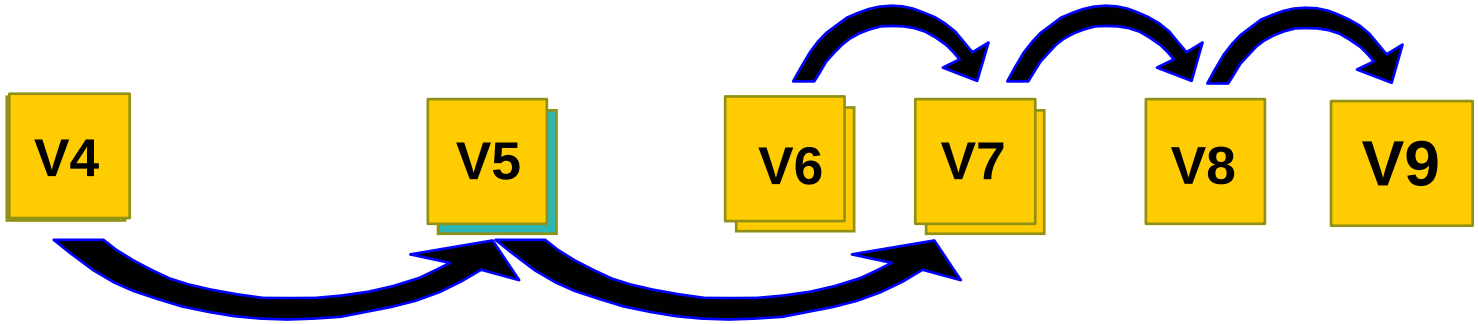
ON DEMAND BUSINESS

Topics

- Migration
  - Migration Paths and Steps
  - Compatibility mode
  - New Function mode
- Optimize
  - DB2 UDB for z/OS Version 8
  - Resource Utilization
    - Performance
    - Availability
- Reference material



Migration Paths



Customers migrate directly to V7 now, then to V8

- Includes fallback & Data Sharing Coexistence
- V8 SPE PQ48486 required PK11129 for V9
- What functions do you need? Where are you now?

When do you need it?

Moving To Compatibility Mode

- Run clist in migrate mode.
- Subsystem (all members in the group) must have been started with migration & fallback SPE APAR PQ48486 PTF UQ81009 just before this.
- Start subsystem at V8 level
- Run DSNTIJTC (CATMAINT).
- Run Version 7 sample jobs.

Running In Compatibility Mode

Minimal new functions are available but…

- Unicode is being used for parsing SQL, conversion, ...
- Some new optimization, new catalog objects
- 64-bit virtual addressing mode
  - All your bufferpools will now be private virtual pools – no more data space backed VP's or hiperpools.
- Consider PGFIX option of ALTER BUFFERPOOL
- Most new utility function is provided.
- Fallback to V7 is fully supported.
- Use precompiler option NEWFUN=NO
- Data sharing members can be running either V7 or V8 in CM, but all must have been run with V7 SPE.
  - Guidance: Keep coexistence short (weekend)

Preparing For Enable New Function Mode (ENFM)

- Define BP8K0, BP16K0 and BP32K. Data sharing users note: ENFM is group scope, so all members & GBPs are required too.
- No going back. – Restore of entire subsystem to point in time is only option. Did you back up?
- All data sharing members must be running V8 CM.
  - ENFM is a group wide event
  - Guidance: Keep ENFM short (under 1 hour typical)

Preparing For Enable New Function Mode (ENFM) ...

- The catalog is getting a little bigger (under 10%)
  - Rebinds will increase catalog size more
  - Look at dataset sizes you will allocate as the shadow datasets as part of ENFM.
  - Reorg the catalog and directory first, if you can.
    - Check timing, check data in catalog
- Look for affected catalog queries (Unicode)
- Simpler to delete user-defined catalog indexes
- Deleted table spaces SYSPROCEDURES & SYSLINKS

Enable New Function Mode

- What is it?
  - The process to move from CM to NFM
  - A fully restartable job DSNTIJNE which converts catalog & directory to long names and Unicode
    - Reorganizes 18 table spaces (read only)
    - Marks its current progress in the catalog.
      - CATENFM START – On the initial run.
      - CATENFM CONVERT INPUT tsname – To mark the currently converting tablespace.
  - Can be stopped
  - Does not allow new function

Enable New Function Mode

- When DSNTIJNE is finished, then
  - Run DSNTIJNF - CATENFM COMPLETE
  - Run job DSNTIJNG to rebuild DSNHDECP
    - DSNHDECM specify NEWFUN=YES
  - All V8 new functions are available.
  - Create user-defined indexes again.
- Can I go back to ENFM? Yes
  - Run job DSNTIJEN - CATENFM ENFMON
  - Run job DSNTIJNG to rebuild DSNHDECP
    - DSNHDECM specify NEWFUN=NO

Running In New-Function Mode

- What are the first things you should you do?
  - Run DSNJCNVB BSDS conversion/expansion utility:
    - Up to 93 active logs and 10,000 archive log datasets.
  - Run the V8 sample jobs.
  - Test some of your existing function again.
- Quiesce data sharing group for new locking
- Then start to look at the new functions….
- To check which mode you are running

  -display group detail

Optimize

DB2 for z/OS V8: The most function-rich release ever!

- Online schema change
- Backup and Recovery enhancements
- Automated space management
- Fast and automatic cached statements invalidation
- Transaction/end-user based accounting and workload management
- Enhancing RUNSTATS with DSTATS
- Long-running, non-committing readers detection
- Lock escalation reporting improvement
- Providing cached statement id in IFCID 124
- Improved LPL recovery
- DRDA and JCC tracing and diagnostics
- . . .

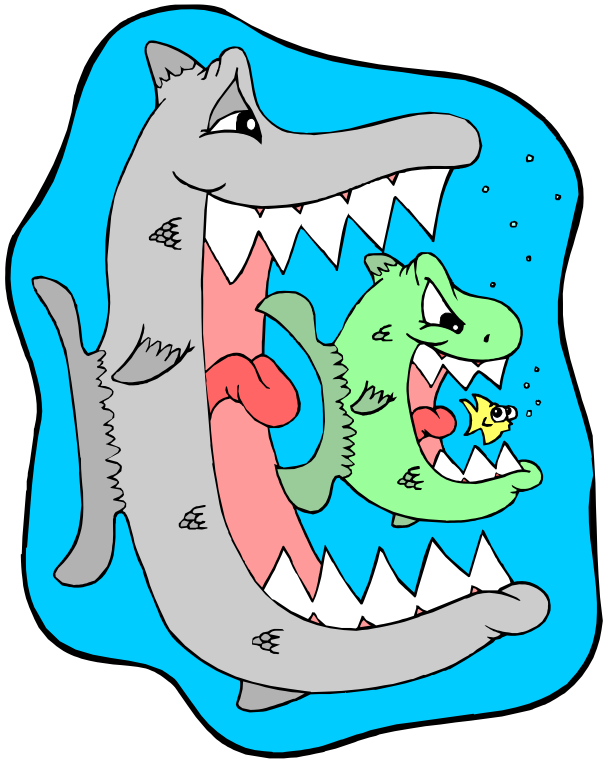
- Lifting virtual storage constraints
- Piece wise LOB insert
- More stage 1 predicates
- Index only access path for VARCHARs
- Fast retrieval of the most recent value
- Eliminating lock contention on special purpose tables
- Option t  ‹s at cursor close
- Allowing updating partitioning key column without partition locks
- Disassociating clustering from partitioning key
- DPSI
- Up to 4096 partitions
- Data sharing improvements
- DRDA performance improvements
- . . .

- Array inserts and fetches
- Sparse indexes
- Reducing negative impact of host variables at access path selection REOPT(ONCE)
- Transparent ROWID for tables containing LOBs
- Full Unicode support
- Lifting database object names length limits
- Up to 255 tables in FROM
- Materialized Query Tables (a.k.a. Automatic Summary Tables)
- Common Table Expressions
- Recursive SQL
- Multiple DISTINCTs
- DB2 Connect 64-bit client for Linux on zSeries
- Allowing comments in dynamic SQL
- . . .

Address Some Myths About V8

- V8 is not a complete re-write of engine, significant re-engineering in select places
- Compatibility Mode is not “No New Function Mode”!
- V8 is not two releases in one: Compatibility Mode (CM), New Function Mode (NFM)
- IBM provides first class support for Compatibility Mode
- Can stay in Compatibility Mode as long as you like
- No need to convert all your application data to Unicode
- DFSORT is really required for V8 Utilities Suite (no cost)
- Large number of secondary extents is not necessarily bad for performance
- 5-10% is typical increase in CPU resource consumption assumes zero exploitation
- Tuning options available to mitigate CPU regression
- Significant opportunity for both performance and scalability improvements
- No need to fear migration with proper planning and testing, follow step by step approach followed by other successful customers

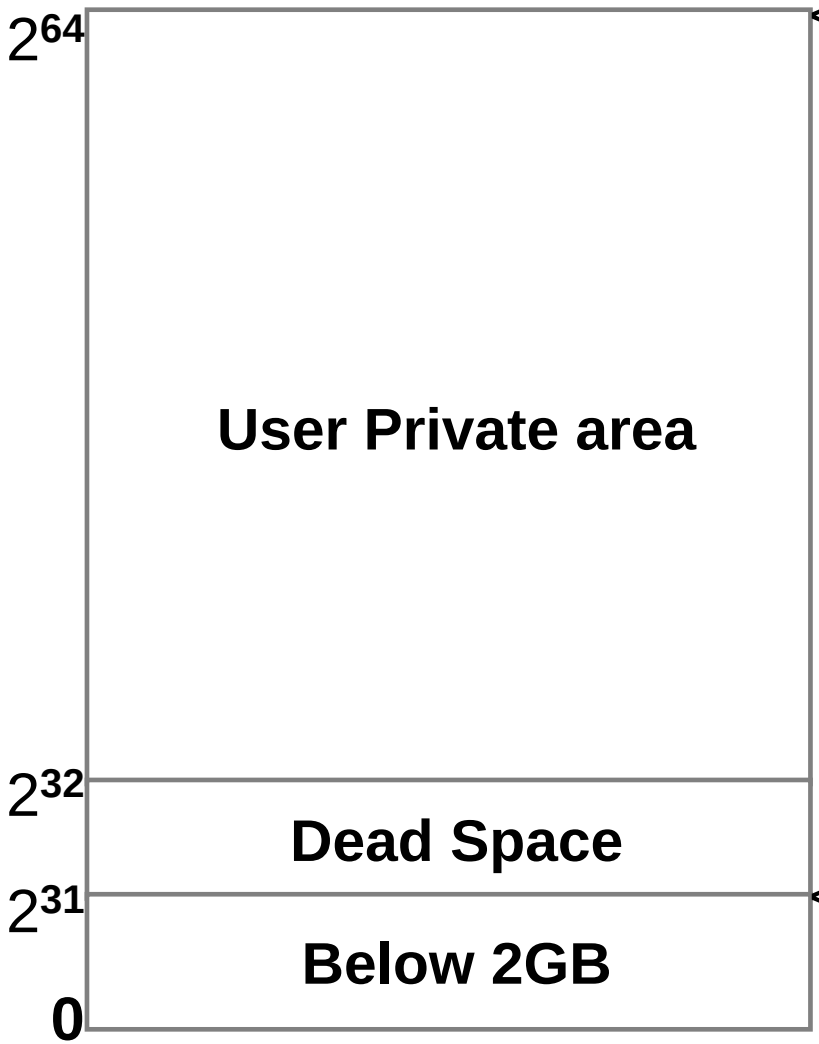
Resource Utilization



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 data space buffer pools (400-800MB)
  - Using data space 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
  - Data space Look-aside 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

DBM1 64 bit Virtual Memory Map (not to scale)



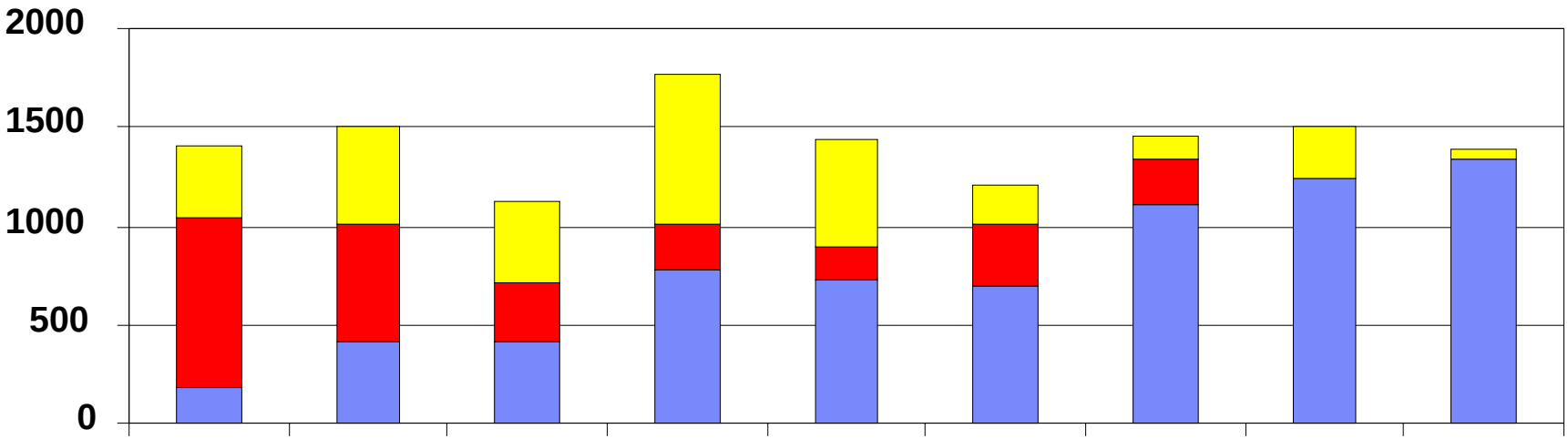
<- 16EB "beam"(?)

- Buffer pools, BM control blocks
- Castout buffers
- RID pool
- EDM DBD cache (OBDs)
- Global dynamic stmt cache
- Sort pool
- Trace tables (Global, Lock, BB)
- Accounting blocks
- Compression dictionaries
- IRLM locks

<- 2GB "bar"

- Most thread storage. Local DSC
- Other EDM and RID component

Potential Value of 64-bit Virtual



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

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 buffer pools 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 around +20-30% for high end production customers
    - More for some customer environments, less for others
  - Many watch virtual, few watch real
- 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

Major Performance Opportunities

- 10 to 100 times improvement possible from
  - Materialized Query Table
  - Stage 1 and indexable predicate for unlike data types
  - Distribution statistics on non-indexed columns
  - Other access path selection enhancements
- 2 to 5 times improvement possible from
  - Multi-row Fetch, cursor Update/Delete, Insert
  - Star join with work file index, in-memory work file, more parallelism
  - DBM1 virtual storage constraint relief
  - Partition Load/Reorg/Rebuild with DPSI

For applications not taking advantage of V8 performance features

- **Some CPU time increase is expected in order to support a dramatic improvement in user productivity, availability, scalability, portability, family consistency,..**
  - **DBM1 virtual storage constraint relief with 64bit instructions**
  - **Long names, long index keys**
  - **Longer and more complex SQL statements**
- **I/O time**
  - **No change for 4K page I/O**
  - **Significant sequential I/O time improvement possible for 8K, 16K, or 32K page because of bigger Vsam Control Interval size**
    - **Up to 70% i/o data rate (MB/sec) improvement**
    - **Also Vsam i/o striping now supported**
  - **V8 PQ99608 2/05 to fix excessive log write i/o's**

CPU Performance Regression - Antidotes

- Some options with significant potential to offset expected increase
  - Rebind of all plans and packages (avoid puffing, re-instate SPROC, new APS?)
  - Long Term Page Fix by Bufferpool
  - If exploiting data sharing coexistence, rebuild GBPs after last V7 member gone
  - Quiesce data sharing group after entry to V8 NFM to switch to Locking Protocol 2
  - 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 with relatively high IO Intensity
  - IO Intensity calculated as (#Pages Read+Written) / #Buffers
- 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

Supersize Bufferpools and ESA Compression

- More use of ESA Compression provided fully backed by real storage
  - Less DASD space
  - Faster sequential scan
  - Potential for better bufferpool hit ratio
    - Reduce sync IO
    - Reduce DB2 Activity Time
    - Fewer threads to maintain same throughput (VSCR)
- Super size bufferpools provided fully backed with large real storage
  - Potential for better bufferpool hit ratio
    - Reduce sync IO
    - Reduce DB2 Activity Time
    - Fewer threads to maintain same throughput (VSCR)

Multi-Row SQL Operations

- Avoids CPU overhead of SQL application interface ‘round trip’
- Multi-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)
- Multi-Row FETCH
  - Up to 33% better for 10 rows
  - Automatically enabled for DRDA when
    - Read-only
    - Ambiguous cursor with CD(N)

Positioned to Exploit IBM zIIP

- New specialty engine for the System z9 mainframe designed to help eligible data serving workloads:
  - Improve resource optimization
  - Lower the cost of ownership
- z/OS manages and directs work between the general purpose processor and the zIIP
  - Transparent to DB2 for z/OS V8 applications
  - Number of zIIPs per z9-109 not to exceed number of standard processors
- DB2 for z/OS V8 in any mode will be first exploiter of the zIIP with:
  - System z9 109
  - z/OS 1.6 or later

Types of Workload that may benefit from IBM zIIP

1 -ERP or CRM application serving*

- For applications, running on z/OS, UNIX, Linux, Intel, or Linux on System z, that access DB2 for z/OS V8 on a System z9 109, via DRDA over a TCP/IP connection DB2 gives z/OS the necessary information to have portions of these SQL requests directed to the zIIP

2 - Data warehousing applications*

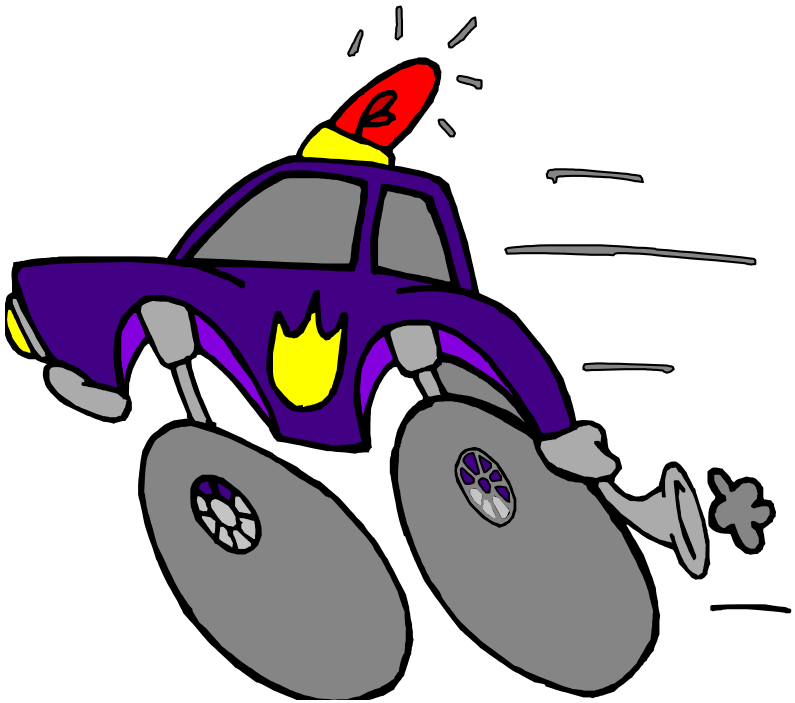
- Requests that utilize DB2 for z/OS V8 star schema parallel queries may have portions of these SQL requests directed to the zIIP when DB2 gives z/OS the necessary information

3 – Some DB2 for z/OS V8 utilities*

- A portion of DB2 utility functions used to maintain index maintenance structures (example LOAD, REORG, and REBUILD INDEX) typically run during batch, can be redirected to zIIP.

* The zIIP is designed so that a program can work with z/OS to have all or a portion of it's Service Request Block (SRB) enclave work directed to the zIIP. The above types of DB2 V8 work are those executing in SRB enclaves, portions of which can be sent to the zIIP.

SQL and Performance



Online Schema Evolution (Dynamic ALTER)

- Extend CHAR(n) column lengths
- Change column data types
  - Must remain within domain type (numeric, char, graphic)
  - OK if referenced by a view
  - OK if indexed
  - Immediate index availability if CHAR, VARCHAR, GRAHPIC, VARGRAPHIC
  - Delayed index availability if numeric
    - Index placed in rebuild pending (RBDP) status
    - Availability to index delayed until index is rebuilt
    - But RBDP index will not block dynamic SQL for:
      - Deletes
      - Updates or inserts for non-unique index
      - Queries (APS won't chose a RBDP index)
- Add column to index
  - Index immediately available if column added to table in same transaction
  - Otherwise index put into RBDP status

Online Schema Evolution (Dynamic ALTER) …

- Add partition to the end
  - Extends the limit value
- Rotate partitions
- Automatic rebalancing of partitions during REORG
- DROP partitioning index, or allow create of a partitioned table without a partitioning index
  - "Table based" partitioning
- Change the clustering index
  - Historically, the partitioning index had to be the clustering index
  - Now an NPI can be the clustering index
  - If no explicit clustering index specified
    - The first index created is used for clustering

Data Partitioned Secondary Index

- Overview
  - DPSI = physically partitioned secondary index
  - #parts(DPSI) = #parts(table)
  - Keys in part 'n' of DPSI refer only to rows in part 'n' of table
- Benefits include:
  - More efficient utility processing
  - Higher availability
  - Streamline partition level operations
  - Potential for lower data sharing overhead
  - Potential impact to query performance
- 3 kinds of indexes now:
  - Partitioning Index (PI).
    - As today, except optional in V8 and may or may not be partitioned
  - Data Partitioned Secondary Index (DPSI) - New in V8
  - Non Partitioned Secondary Index (NPSI) - As today's NPI

Up to 4096 Partitions

- Max number of parts raised from 254 to 4096
  - Tablespaces and indexes
  - Tablespace must have LARGE or DSSIZE to go beyond 254 parts
- Max table size remains 16TB
- Dataset naming convention
  - 'Axxx' - partitions 1-999
  - 'Bxxx' - partitions 1000-1999
  - 'Cxxx' - partitions 2000-2999
  - 'Dxxx' - partitions 3000-3999
  - 'Exxx' - partitions 4000-4096
- Max number parts allowed depends on page size and DSSIZE
  - 4K page size, DSSIZE=1GB => 4096 parts allowed, 4TB max table size
  - 4K page size, DSSIZE=64GB => 256 parts allowed, 16TB max table size
- ALTER TS ADD PART adds partitions to the end

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

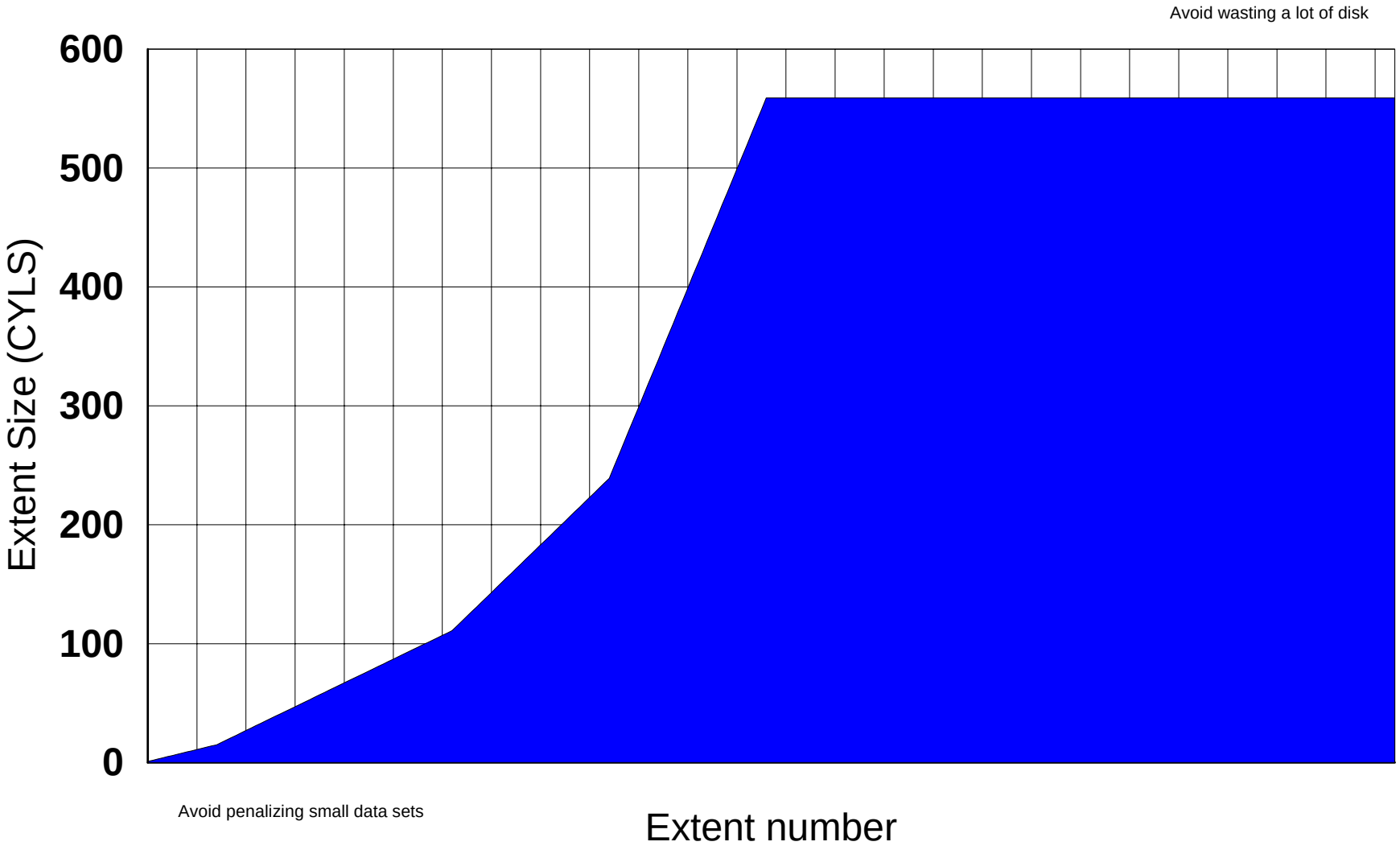
Online Schema Changes …

- 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

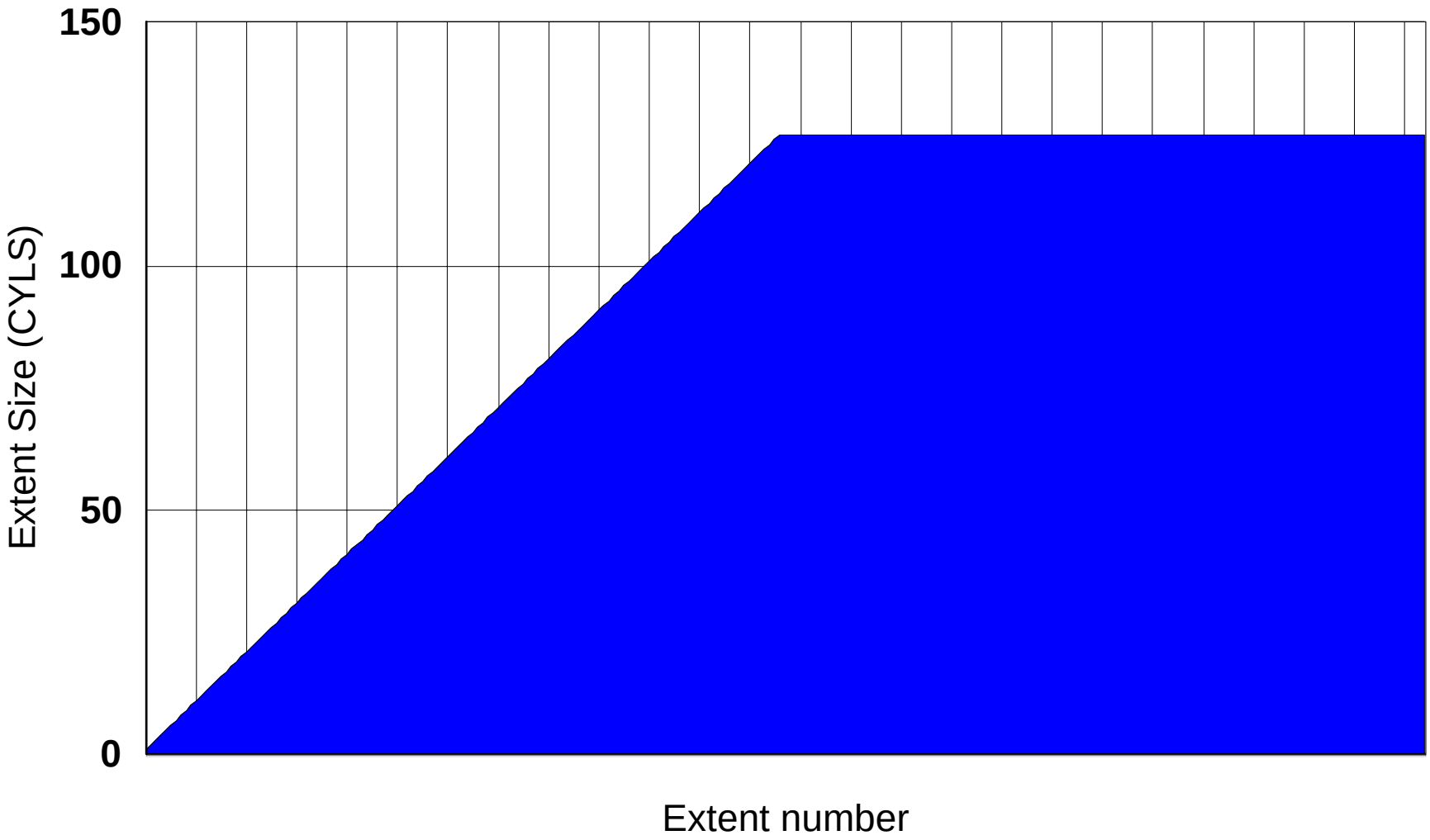
Automatic Space Management

- SMART enhancement: sliding secondary allocation quantity size
- Applies to DB2 managed pagesets only
- Tries to avoids 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

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

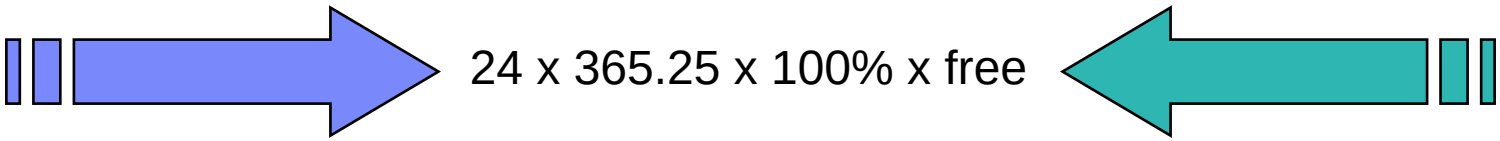
Other Changes

- 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 at next re-allocation – typically 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

What can go wrong in Continuous Availability?

- Users
- Applications
- Operations
- Database Administration
- Systems Administration
- Software bugs

- Data Integrity
- Performance
- Locking
- Continuous Operations



Availability Issues - Manage to Service Level Agreement

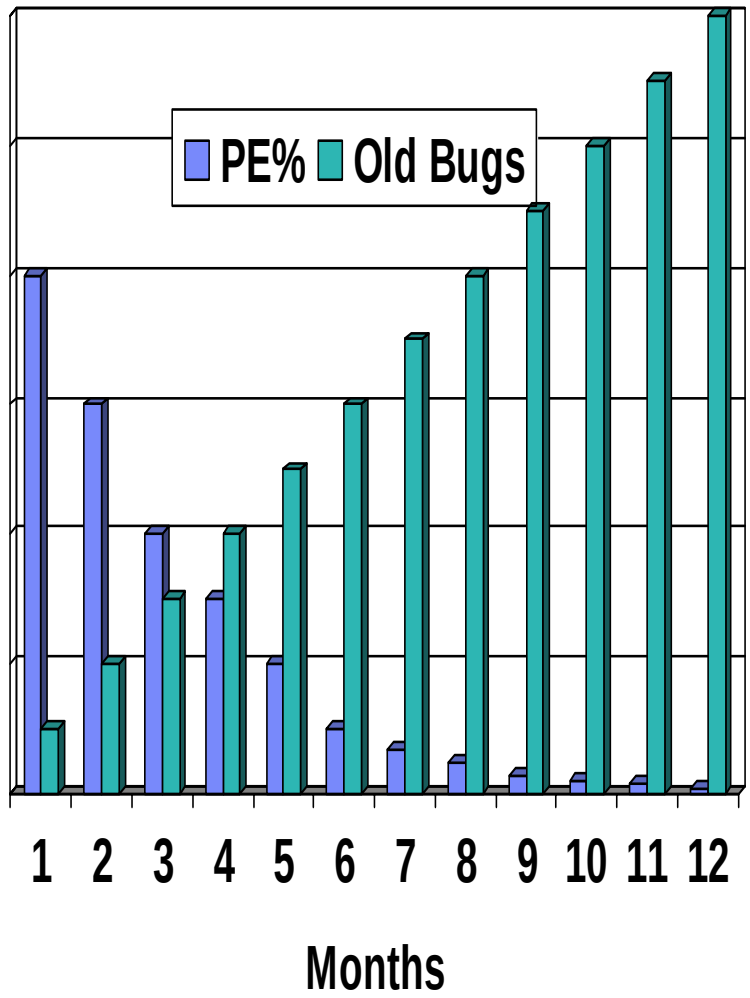
- Manage
  - Set availability criteria - balance needs & costs
  - Use criteria to determine procedures
- Practice operation and recovery
  - If not meeting criteria, then tune
  - Education for skills and time to do the job
- Stay reasonably current with DB2 releases & fixes
- Isolation and controlled replication
- Commit & Copy Frequency & Management

Availability Issues - Reduce planned & unplanned downtime

- Planned downtime
  - Data Sharing allows hardware and software changes to be non-disruptive to applications
- Unplanned downtime
  - Data sharing eliminates a single point of failure for a CEC, a z/OS LPAR, a DB2 subsystem
- Fast, non-disruptive database operations (even without data sharing)
  - Online Copy & Reorg
  - Concurrent Copy – allows you to create a non-fuzzy copy while applications update the data
  - Geographically Dispersed Parallel Sysplex
  - Faster restarts
  - Online parameters
  - Online LOAD resume
  - ...

Availability Issues - Service Strategy

- Stay reasonably current with DB2 releases & fixes
- Balance for severity
  - Potential PTF in Error (PE)
  - Problems encountered instead of avoided
- Work load and windows for installing service
- Recommended process:
  - Consolidated Service Test levels
  - Upgrade service 3-4 times / year
  - Examine hipers and PEs weekly? monthly?
  - Stage through dev to test to prod



Best Practices for Restart

- Restart is the process of starting DB2 on an operating system logical partition (LPAR) after a normal stop, or an unexpected termination. The time required to restart DB2, after a termination, is dependent on a variety of factors:
  - How far back in the DB2 recovery log the last checkpoint was taken
  - The number of log records written from the last checkpoint for outstanding units of recovery (URs)
  - Whether objects are deferred for restart log apply processing
  - Whether postponed abort is used (limit backout)
  - How many datasets need to be opened for restart log apply
- Therefore…
  - Checkpoint frequently (2-5 minutes – the closer to two minutes, the better)
  - Don't have any long running (URs)
  - Consider deferred restart for selected objects or limiting the backout of long-running URs
  - Reduce the number of datasets open for write access
  - Use dual-logging and put each log/BSDS in different ICF catalog

Checkpointing and long running Units of Recovery (URs)

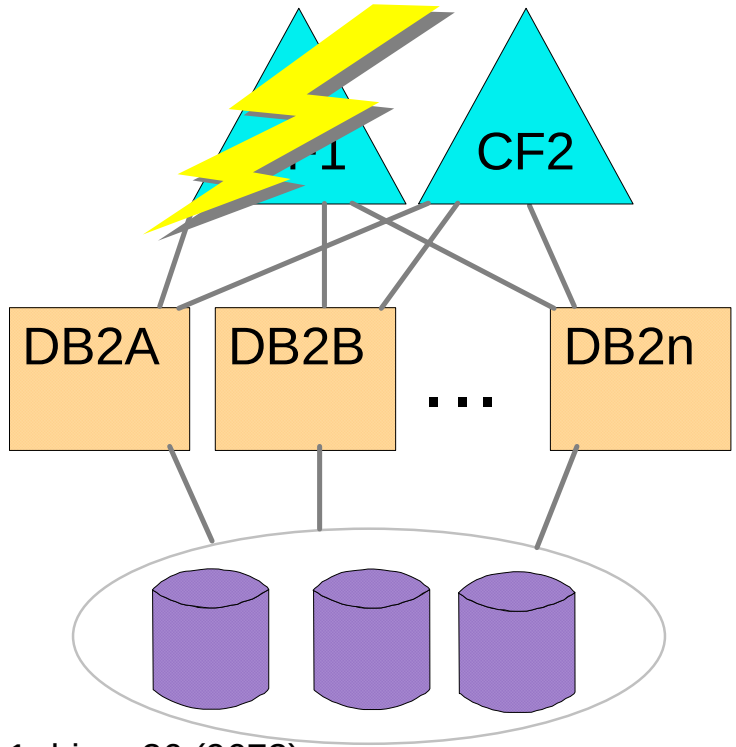
- To checkpoint based on logrecs or time?
  - Checkpointing based on number of log records written will give most consistent restart times
  - Consider checkpointing based on time interval when you have varying logging rates through time and restart time requirements are flexible
- Checkpoint every 2-5 minutes is a good general recommendation
- -SET LOG command can dynamically change checkpoint interval
  - LOGLOAD for # log records
  - CHKTIME for time interval (V7)
  - -DISPLAY LOG shows checkpoint frequency and status
- Last 100 checkpoints are recorded in the BSDS
  - Use DSNJU004 (Print Log Map) to display

Checkpointing and long running Units of Recovery (URs)

- Identifying Long Running URs
  - Problems caused by long running URs:
    - Long restart times (use of Postponed Abort can mitigate this)
    - Long lock hold time can impact concurrency
    - Reduces effectiveness of lock avoidance checking
    - Prevents online Reorg and other utilities from running
  - Zparms control when DB2 issues long-running UR warnings:
    - UR CHECK FREQ , message DSNR035I (V5)
      - number of checkpoints that a UR has not committed
    - UR LOG WRITE CHECK, message DSNJ031I (V7)
      - number of log records that a UR has not committed
      - More granularity than UR CHECK FREQ
  - IFCID 0313 written, if active, when long-runner detected
    - Use IFCID 0313 to keep a history of problematic URs
  - Use UR CHECK FREQ for most accurate warning of restart impact
  - Use UR LOG WRITE CHECK to accurately identify applications that write a lot of log records
    - DB2 Accounting reports can also help identify these
    - V8: LONG-RUNNING READER THRESHOLD Zparm
    - Write IFCID 0313 when long-running reader detected

CF Duplexing

- GBP duplexing available since V5.
  - Recommendation: use it.
- Duplexing for Lock and SCA available in V7
  - Uses z/OS "system managed" CF duplexing which became GA in Feb '04
  - Useful to remove Internal Coupling Facility (ICF) as a single point of failure
  - Performance impact will depend mostly on intensity of CF lock requests for your application and distance between CFs
  - Pre-req's:
    - z/OS 1.2 or above plus maintenance
    - See CFDUPLEXING PSP Bucket
    - CFCC level 12 driver 3G (zSeries), or CFCC level 11 driver 26 (9672)
    - CF-to-CF links
  - http://w3-1.ibm.com/support/techdocs/atsmastr.nsf/WebIndex/FLASH10272



Best Practices for Recovery

- Recovery is running RECOVER utility to recover one or more objects
  - The restore phase copies the data from the appropriate backup. If there are incremental backups, then they are also processed during this phase. The time for this phase is dependent upon:
    - Size of the objects (if flash copies cannot be used)
    - Media of the backups (tape, slow disk, fast disk, flashed copies)
  - The log apply phase positions the log to the point at which the backup was taken, and using a range index on the log (SYSLGRNX), scans the portions of the log when the objects were open for write access. The time for this phase is dependent upon:
    - The amount of log data that has to be scanned for applicable log records
    - The number of log records that need to be applied
    - Whether archive logs are needed, and whether archive logs reside on disk or tape
- Therefore…
  - Keep recent backups on fast disk (VTS considerations)
  - Take backups frequently enough to meet your recovery time objective
  - Keep 24-48 hours of log on disk
  - Use fast log apply

Data Recovery

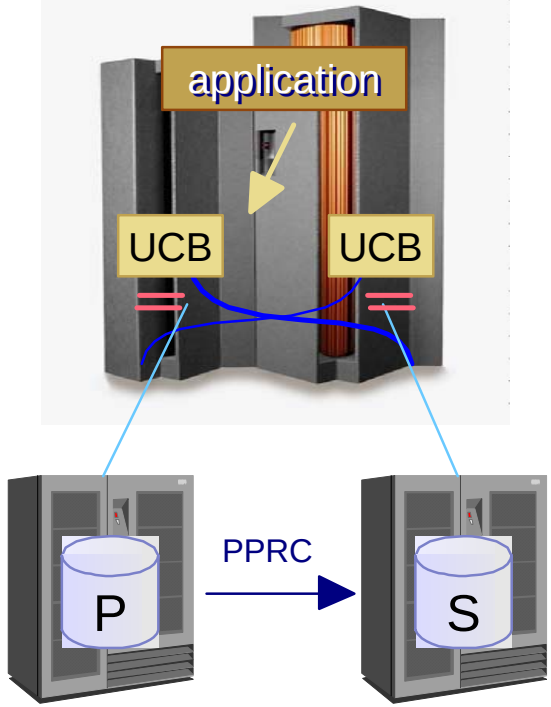
- RECOVER utility time = restore time + log scan time + log apply time
- Restore time:
  - Number of pages, number of objects?
  - ICs on tape or DASD?
  - Degree of parallelism?
- Log scan time:
  - Image copy frequency
  - Archive logs needed to recover?
  - Log read from archive is not as efficient as from active (~25%)
  - Archive logs on tape or DASD?
  - Reads from DASD are faster
- Log apply time:
  - Update frequency and update patterns
  - Maximal fast log apply (4-10X difference)

Data Recovery - Recommendations for faster data recovery

- Reduce COPY cycle time
- For tape, take dual image copies to avoid image copy fallback
- Consider incremental COPY
  - Perform regular MERGECOPY of incremental COPYs
- COPY indexes
- Use DASD to write image copies and manage by DFSMS
- Keep at least 24-48 hours of recovery log on DASD
- Large, dual active logs
- Use VSAM striping for active logs
- Avoid access to archive log datasets
- Write archive log to DASD and manage by DFSMS
- IBM DB2 Archive Log Accelerator for z/OS can speed up arch log access
- Reorganize SYSLGRNX
- Large BP0 (at least 10,000 buffers) for recovering the cat/dir
- Use ESA Compression - be careful before V8 because of virtual storage constraint

GDPS PPRC HyperSwap

- Site failover
- DB2 data sharing group must still be recycled to recover CF structures
- New technology under development to avoid having to recycle DB2 members
- DASD failover
- New GDPS HyperSwap Manager announced 2/15/05
- Entry level GDPS offering
- Keep data available to end user apps during disk subsystem maintenance or failure
- DASD failure is transparent to DB2
- GDPS HyperSwap prerequisites
  - Disk subsystems that support PPRC level 3 (extended query)
  - IBM Tivoli System Automation for GDPS/PPRC HyperSwap Manager
- http://bvrgsa.ibm.com/projects/g/gdpsweb/hyperswap_manager.html



Reference

- V8 manuals, especially Performance Monitoring and Tuning section of Administration Guide

- Redbooks at www.redbooks.ibm.com
  - DB2 UDB for z/OS V8 Technical Review SG24-6871
  - DB2 UDB for z/OS V8 Everything you ever wanted to know… SG24-6079
  - DB2 UDB for z/OS V8 Performance Topics SG24-6465, available on website April 11, 2005

- More DB2 for z/OS information at www.ibm.com/software/db2zos
  - E-support (presentations and papers) at www.ibm.com/software/db2zos/support.html