DB2 Data Management Software

IBM

DB2 for z/OS Version 8 Technical Overview DB2® Usergroup Finland

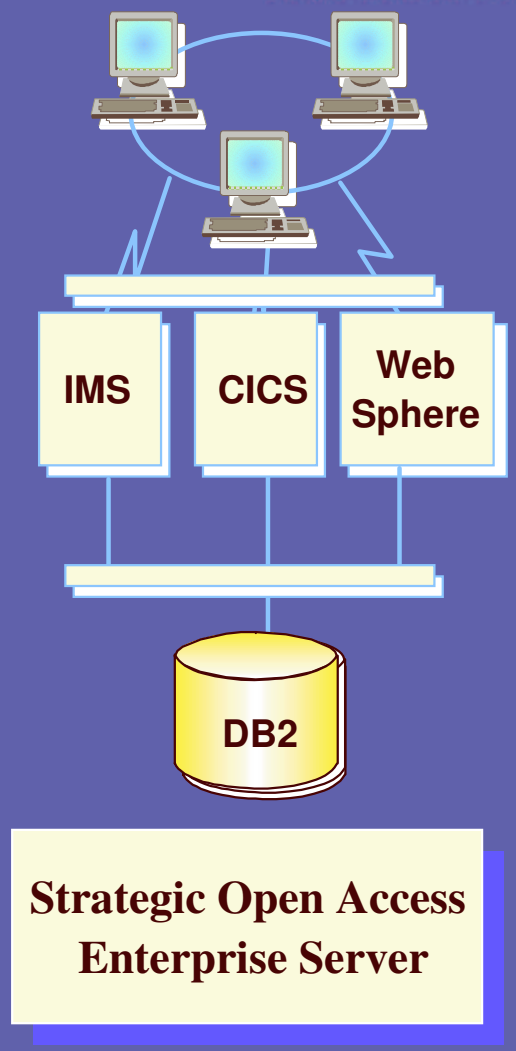
Themes and Highlights

Carsten Rasmussen
DB2 for z/OS
IBM Denmark
clr@dk.ibm.com

Important Redbook

DB2 UDB for z/OS Version 8:
Everything You Ever Wanted to Know ,
... and More
SG24-6079-00
Number of Pages: 1090

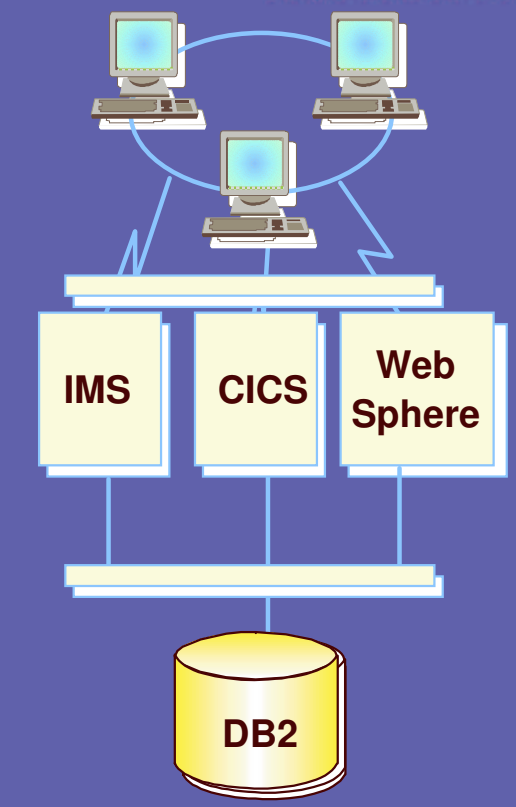
DB2 for z/OS Version 8



Themes:

- Reengineering & Renaissance
- Breaking the Limits
- Performance Enhancements
- Continuous Availability
- Indexing Improvements
- Very Large Database
- WebSphere & Java
- SQL Function & DB2 family
- SAP, PeopleSoft & Siebel
- Migration
- Tools

DB2 for z/OS Version 8



Strategic Open Access Enterprise Server

Key Enhancements:

- Schema evolution
- 64 bit virtual storage
- Longer names and statements
- Improve optimization & parallelism
- Materialized Query Tables
- WebSphere integration
- Dynamic Cursor Scrolling
- Unicode
- Improved indexes & VLDB
- Enhanced utilities

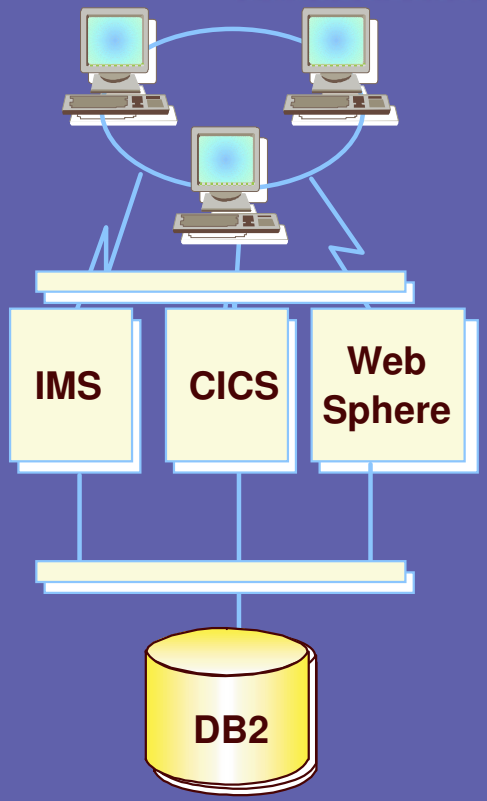
Reengineering & Renaissance: History & Future

- Indexes
- Data Sharing
- Stored Procedures
- Object Relational
- Storage & Space
- SQL & Optimization
- Unicode
- DRDA & Combined Client

- ✓ Scalability
- ✓ Availability
- ✓ Productivity

- ✓ Total Cost to Operate
- ✓ Synergy: DB2 Family & zSeries

Limits: DB2 for z/OS V8

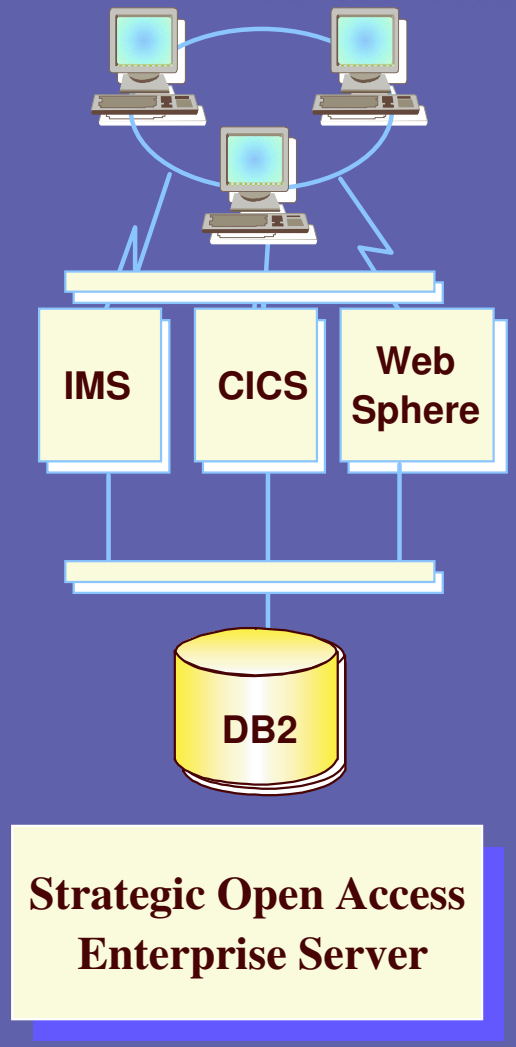


Strategic Open Access Enterprise Server

Breaking the limits

- Virtual Storage 2 GB 2^{31} to 2^{64}
- Table name sizes 18 to 128
- Column name sizes 18 to 30
- Partitions 254 to 4096
- SQL statement length 32K to 2 MB
- Index key size 255 to 2000
- Literals 255 to 32704
- Short strings 255 to 32704
- Predicates 255 to 32704
- ...

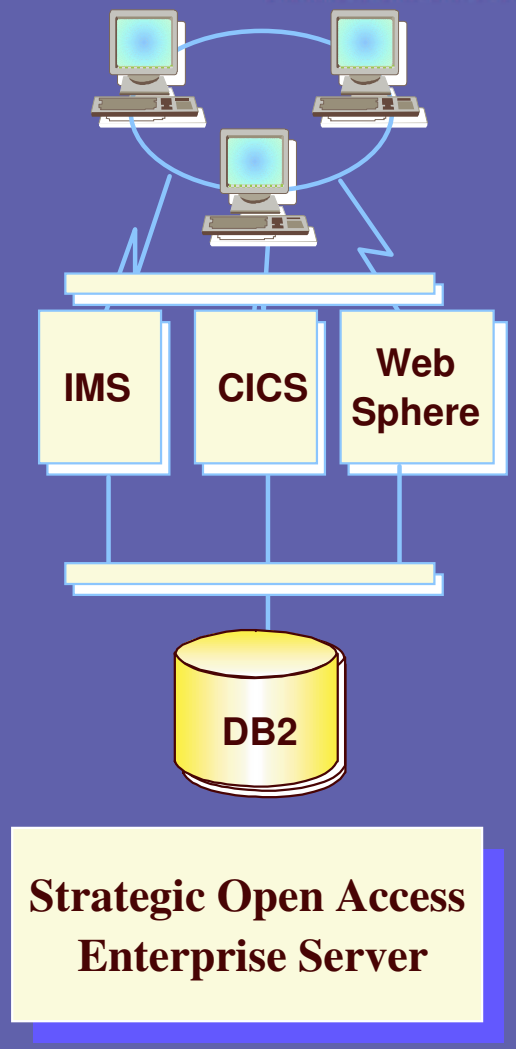
Limits: DB2 for z/OS V8



Breaking the limits ...

- Current optimization hint 8 to 128
- CURRENT PACKAGESET 18 to 128
- CURRENT PATH 254 to 2048
- CURRENT SQLID 8 to 128
- USER 8 to 128
- VIEW 18 to 128
- ALIAS 18 to 128
- ...

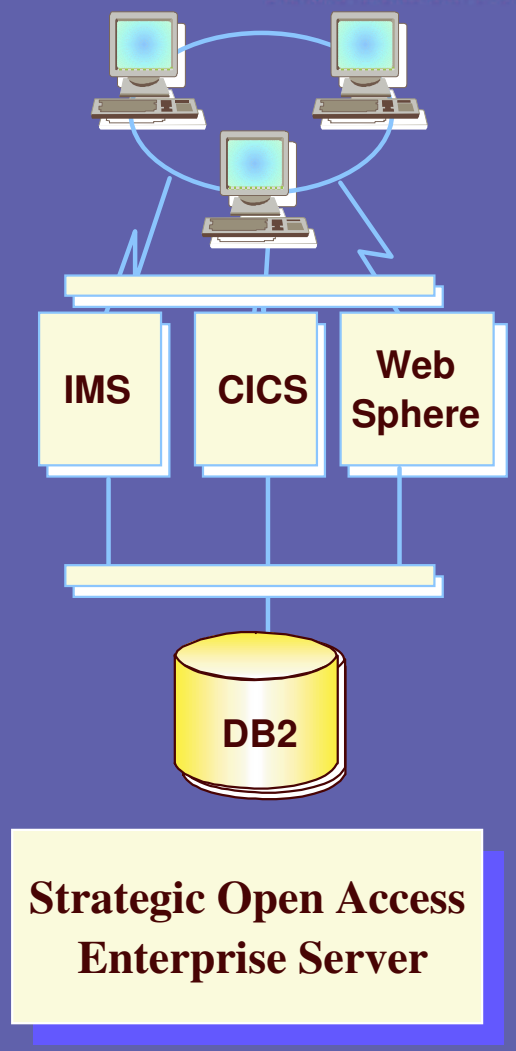
Performance: DB2 for z/OS V8



Performance Enhancements

- Ability to use indexes more often
  - More matching in predicates
  - Indexes bi-directional, varying length
  - Distribution statistics on non-index columns
- Multi-row fetch & insert
- Materialized Query Tables
- More Parallel Sorting

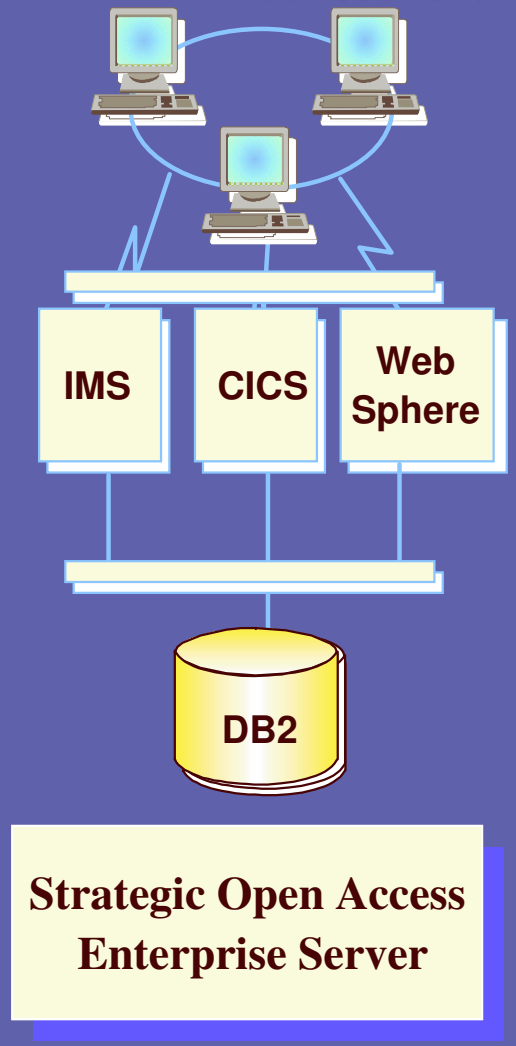
Availability: DB2 for z/OS V8



Continuous Availability

- Schema Evolution
- Data Partitioned Secondary Indexes
- System Level Point-in-Time Recovery
- Online REORG Enhancements
- Changing partition boundaries
- Add a column to an index
- Improved LPL Recovery
- Additional online zparms

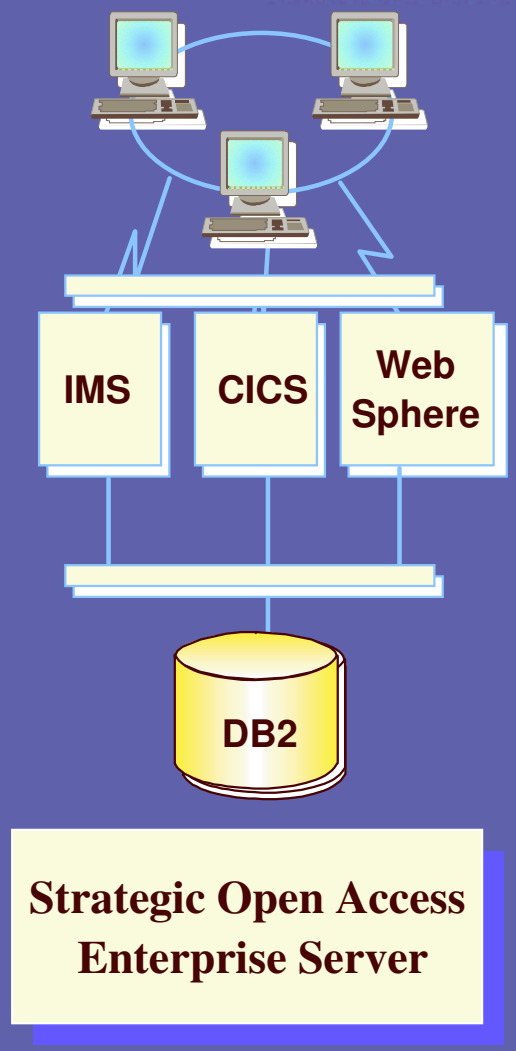
Index: DB2 for z/OS V8



Index Improvements

- Variable length index keys
- Index-only access for varchar data
- Maximum index key 2000 bytes
- Predicates indexable for unlike types
- Backward Index Scan
- Partitioning separate from clustering
- Data-partitioned secondary indexes

VLDB: DB2 for z/OS V8

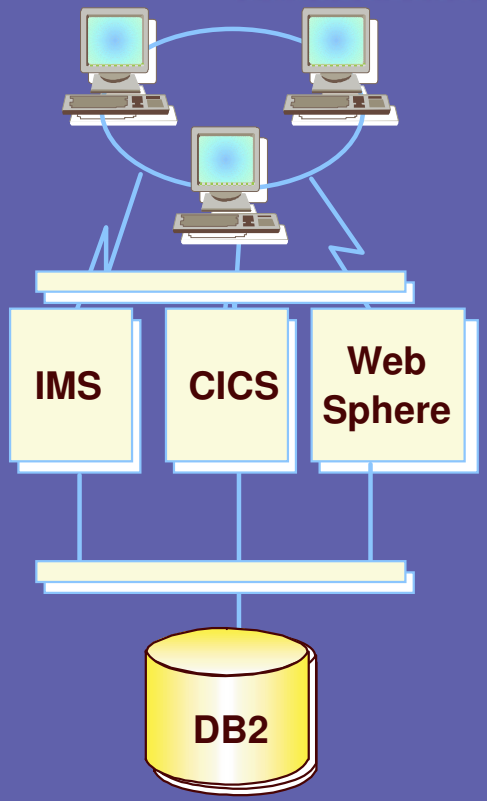


Very Large Database Improvements

- Separate partitioning from clustering
- Partition without an index
- Data-partitioned secondary indexes
- 4096 Partitions
- Add partitions
- Rotate partitions
- Adjust partition boundaries without interruption
- Add a column to the index

- Max number of parts raised from 254 to 4096
  - Table spaces and indexes
  - Table space must have LARGE or DSSIZE to go beyond 254 parts
- ALTER TABLE ... ADD PART adds partitions to the end
- Max table size remains 16TB for 4 KB pages
- 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

WebSphere & Java: DB2 for z/OS V8

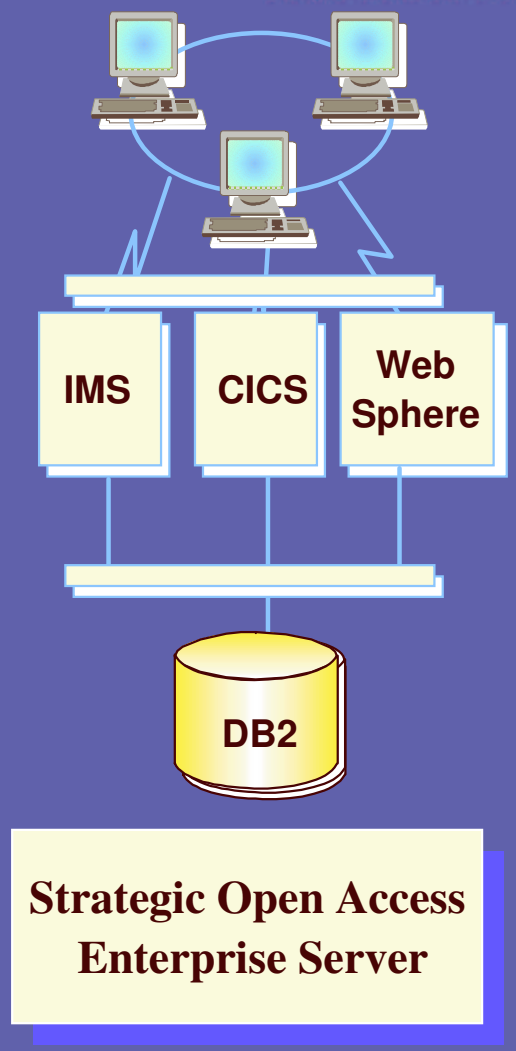


Strategic Open Access Enterprise Server

Web Improvements

- DRDA improvements
  - Encryption of data flowing on the wire
  - CDB improvements for data sharing member routing
  - LOCATION aliases
- WebSphere and SQL
  - Cursor replicates
- PACKAGE PATH
- XML "in the engine"

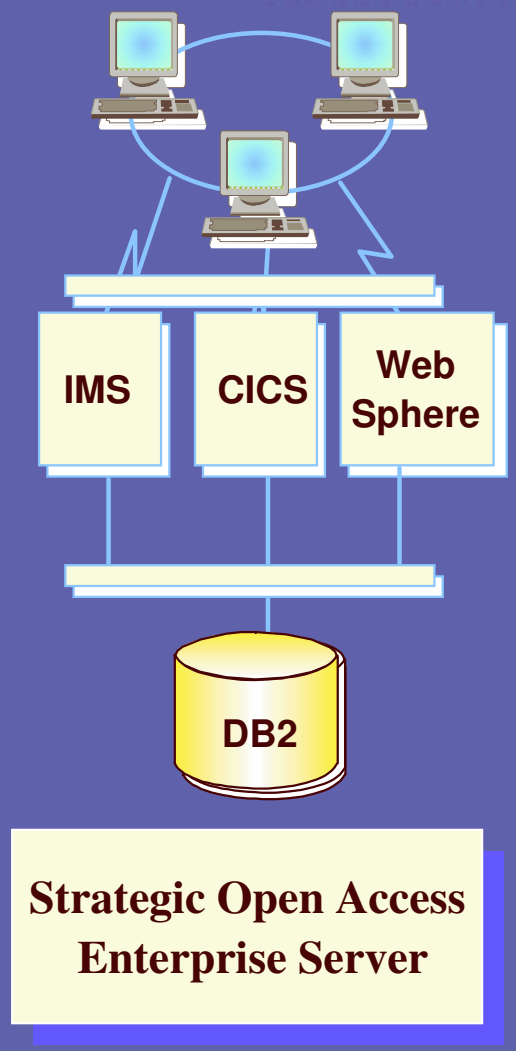
SQL: DB2 for z/OS V8



SQL function & DB2 family

- Scalar Fullselect
- Materialized Query Tables
- Multiple DISTINCT
- INSERT within SELECT Statement
- Dynamic Scrollable Cursors
- Sequences
- Identity column enhancements
- Long names
- Long statements (2 MB)

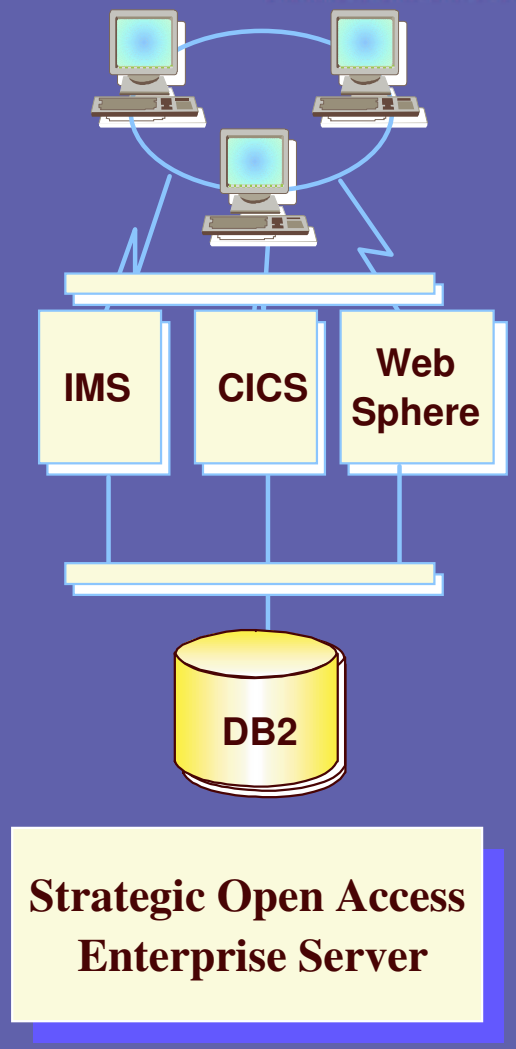
SQL: DB2 for z/OS V8



SQL function & DB2 family ...

- Long strings
- Long predicates
- Multi-row FETCH & INSERT
- Additional Built-in Functions
- Unicode
- Stored Procedure & UDF enhancements

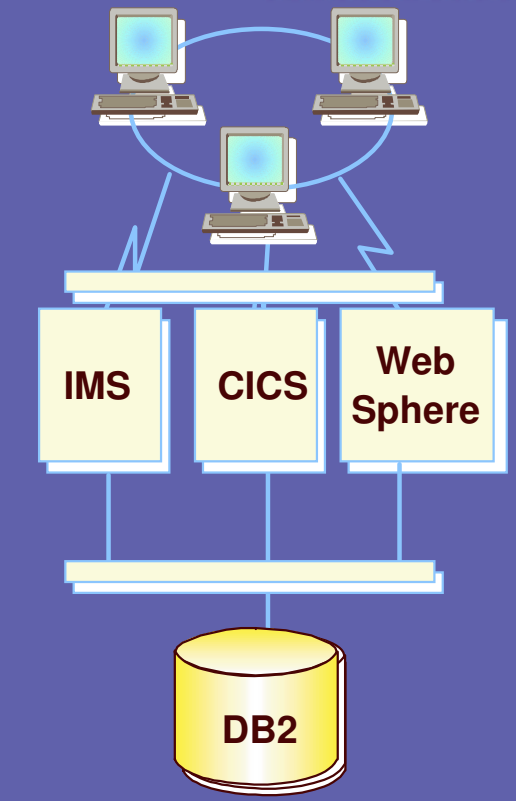
Family: DB2 for z/OS V8



Other DB2 Family Improvements

- Control Center
- Development Center
- Warehouse Manager
- Replication & Replication Center
- DB2 Connect

ERM: DB2 for z/OS V8

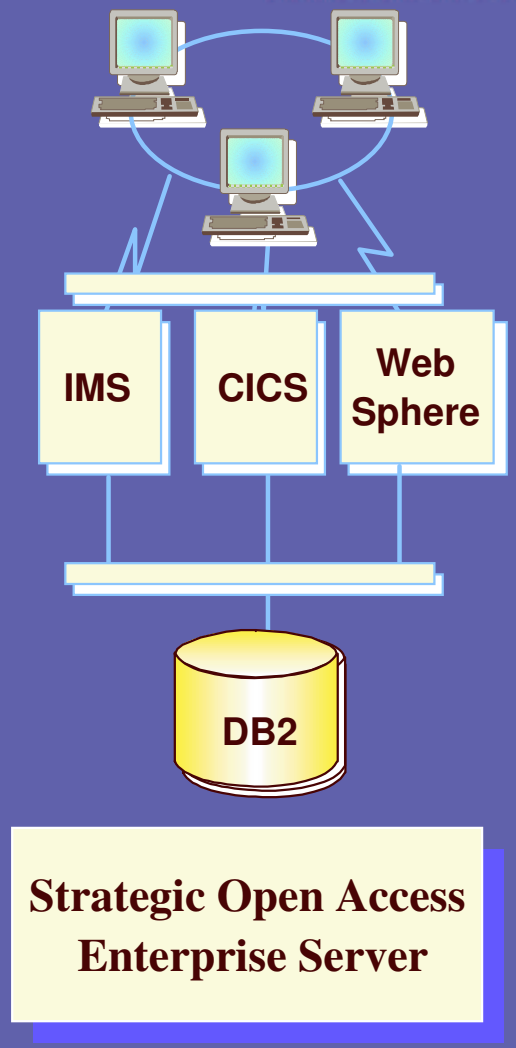


Strategic Open Access Enterprise Server

SAP, PeopleSoft & Siebel

- 64 bit virtual storage
- Unicode
- Prior Point in Time Recovery Automation
- 64 bit DB2 Connect for Linux zSeries
- Array Fetch, array insert
- Multiple DISTINCT Clauses
- Lock Contention on volatile Tables
- Fast Retrieval of Most Recent Value
- DRDA performance improvements

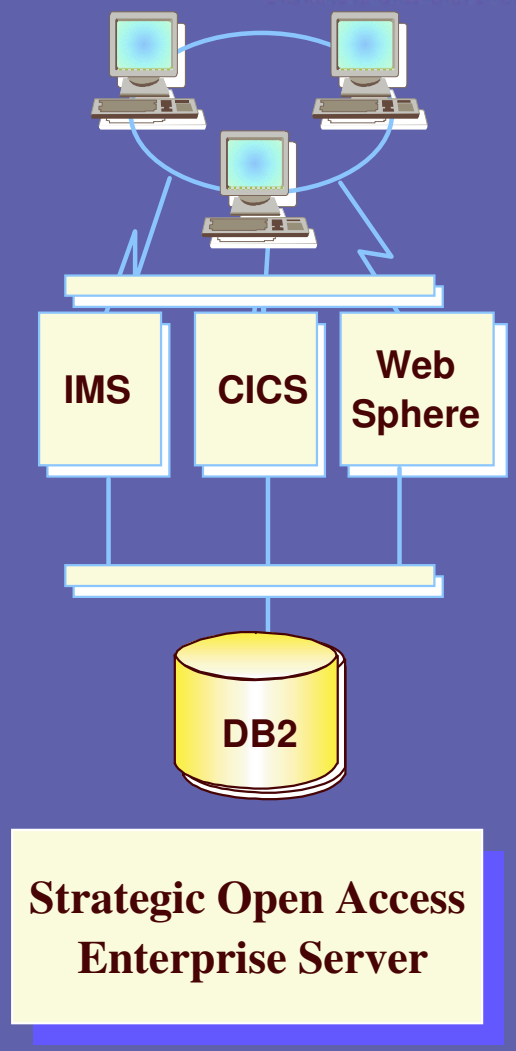
ERM: DB2 for z/OS V8



SAP, PeopleSoft & Siebel ...

- ASCII Compiled C Programs and SQL
- Minimize Impact of Creating Deferred Indexes
- Longer Table Names
- Additional statistics
- Convert Column Type
- Altering CLUSTER Option
- Adding Columns to Index
- Index-only Access Path for VARCHAR

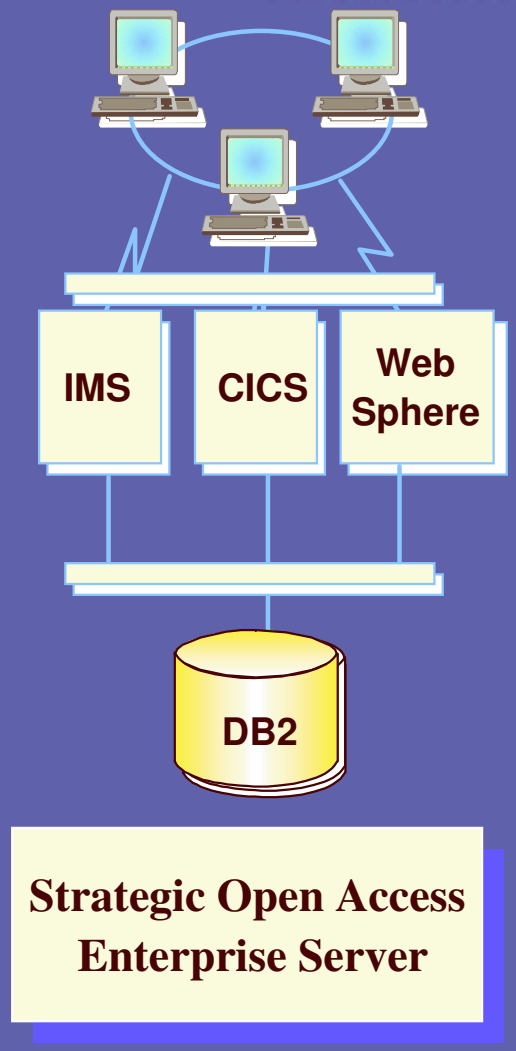
ERM: DB2 for z/OS V8



SAP, PeopleSoft & Siebel ...

- Adding New Partitions
- Separate Clustering from Partitioning
- Control Center Enhancements
- DRDA Performance improvements

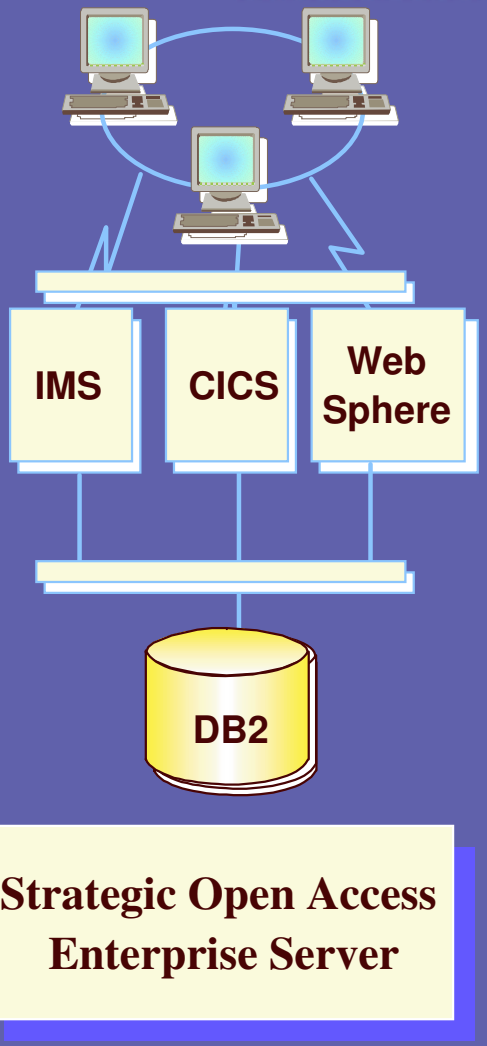
Utility: DB2 for z/OS V8



Utility Enhancements

- Schema Evolution
  - ROTATE partitions
  - REBALANCE
- On-line REORG Enhancements
  - DISCARD
  - Avoid BUILD2 with DPSI
  - Allow long-running UR read only
- LOAD & UNLOAD delimited
- SCOPE PENDING

Others: DB2 for z/OS V8



Other Enhancements

- RRSAF compatibility for CAF appls.
- UDT function cardinality
- Improve access for volatile tables

Prerequisites for DB2 V8

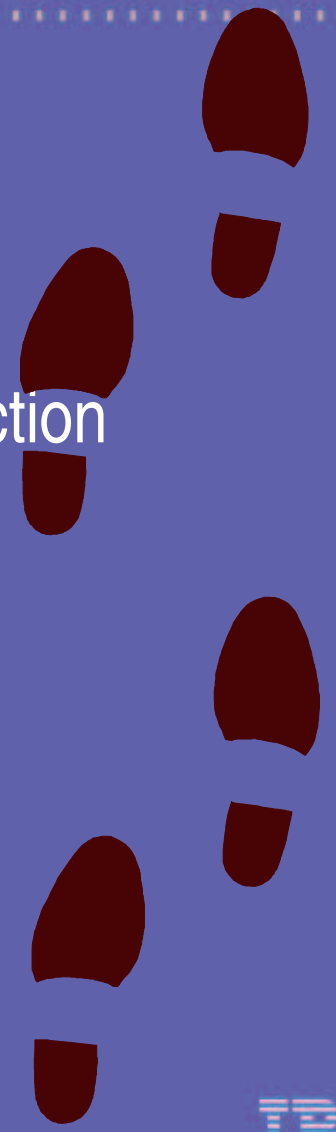
- Base requirements
  - zSeries
  - z/OS V1R3
- Increased release levels
  - WebSphere Application Server V4R2
  - IMS V7
  - CICS TS V2.2
  - JDK 1.3
- Requirements for specific function
  - System Level Point in Time Recovery
    - z/OS V1R5 & DFSMShsm & DFSMSdss
  - CF level 12 batching

System level PIT recovery

- Easier, more efficient, less disruptive
- Two new utilities are introduced
  - BACKUP SYSTEM
  - RESTORE SYSTEM
- New HSM construct called a COPYPOOL
  - Named set of SMS storage groups
  - Each DB2 system defines one COPYPOOL for data, one for logs
  - z/OS 1.5
- New DFSMS construct called "copy target" storage group
  - Storage group reserved to be target of FlashCopy

Four step process for migration

- Test with new install & migrate
- Apply the fallback SPE to all members
  - Start all members at SPE level
- Migrate to new release without new function (Compatibility Mode)
- Enable New Function Mode
  - Then in New Function Mode



Key checks before you migrate

Run job DSNTIJPM (DSNTIJP8 in V7)

Check on CCSIDs: wrong or more than one

Deprecated function: selective partition locking

Data capture on DB2 catalog tables (turn back on after migration)

If DSNWZPR used, switch to WLM-managed DSNWZP

V7 sample programs used in V8 CM

Truncated limit key in indexes

Type 1 indexes (final call)

Primary incompatible changes

- Use valid host variable declaration for PREPARE
- Valid VCAT required no x.y (V5 and up)
- No COMPJAVA, use JIT
- CREATE, ALTER PROCEDURE are WLM only
- Unicode used in DB2 catalog, changes order
  - SQL statements, translation, lengths may change, instrumentation
- DB2 Universal Driver for SQLJ/JDBC
- SQLJ preparation
- Index default is variable length, NOT PADDED
- REORG SORTDATA and SORTKEYS default
- REORG implicit clustering index

Important Information

DB2 UDB for z/OS Version 8:
Everything You Ever Wanted to Know ,
... and More SG24-6079-00

DB2 UDB for z/OS, Version 8 Transition (CG38AFI)

DB2 for z/OS V8 Functions
ITSO Workshop - ITSO8MFR

DB2 V8 Optimizer DBTECHFR (DBTECHDE)

DB2 V8 System Backup SG24-6370

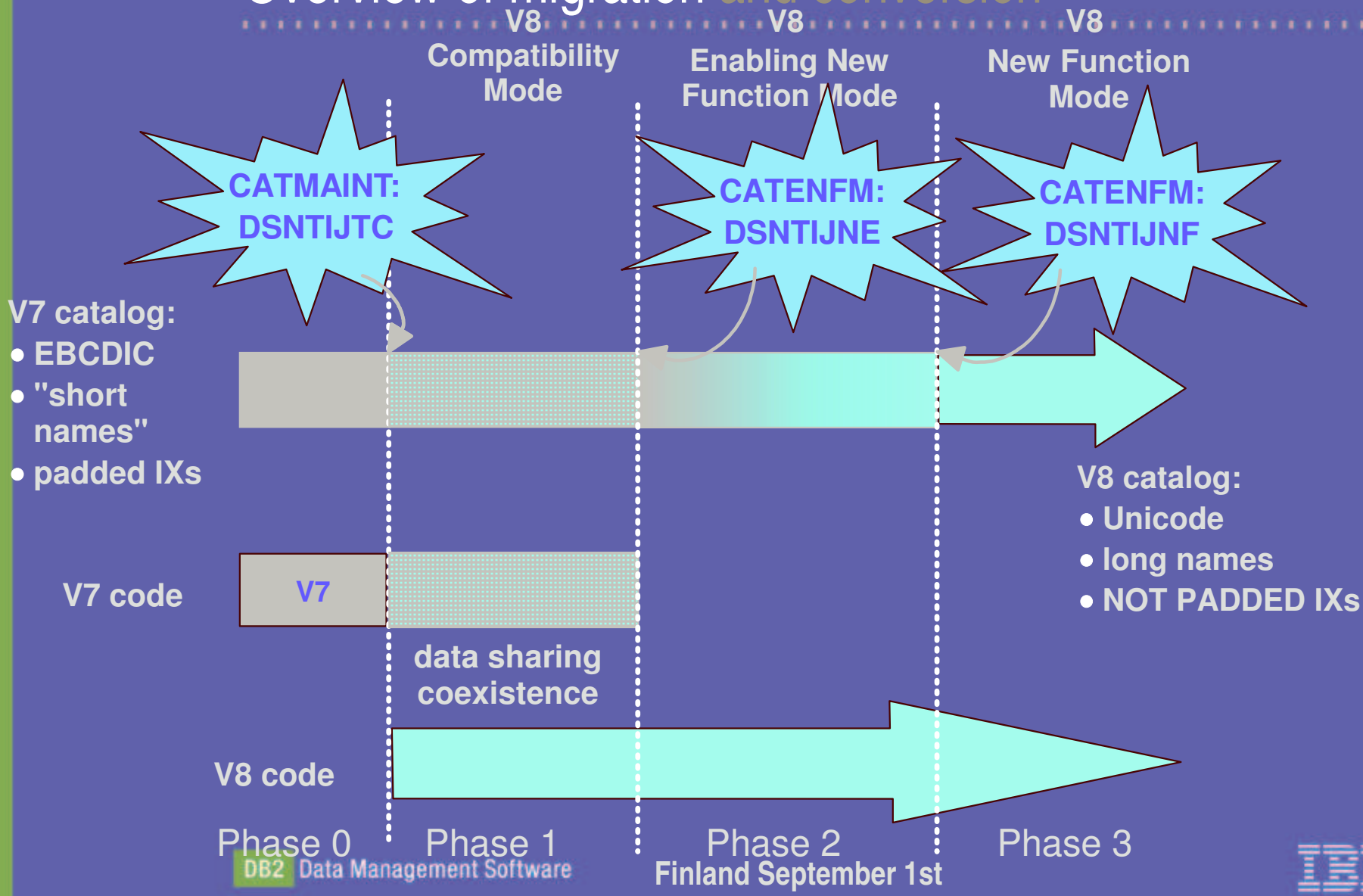
Migration

- Overview of the modes of DB2 Version 8
  - ►Why we introduced the multi-mode migration process
- Migration to DB2 Version 8 Compatability Mode (CM)
- Coexistence and fallback from DB2 Version 8
- Conversion to DB2 Version 8 New Function Mode (NFM)
  - ►The Enable New Function Mode (ENFM) process
- DB2 Version 8 New Function Mode
- Planning considerations

Overview of the various modes of DB2 Version 8

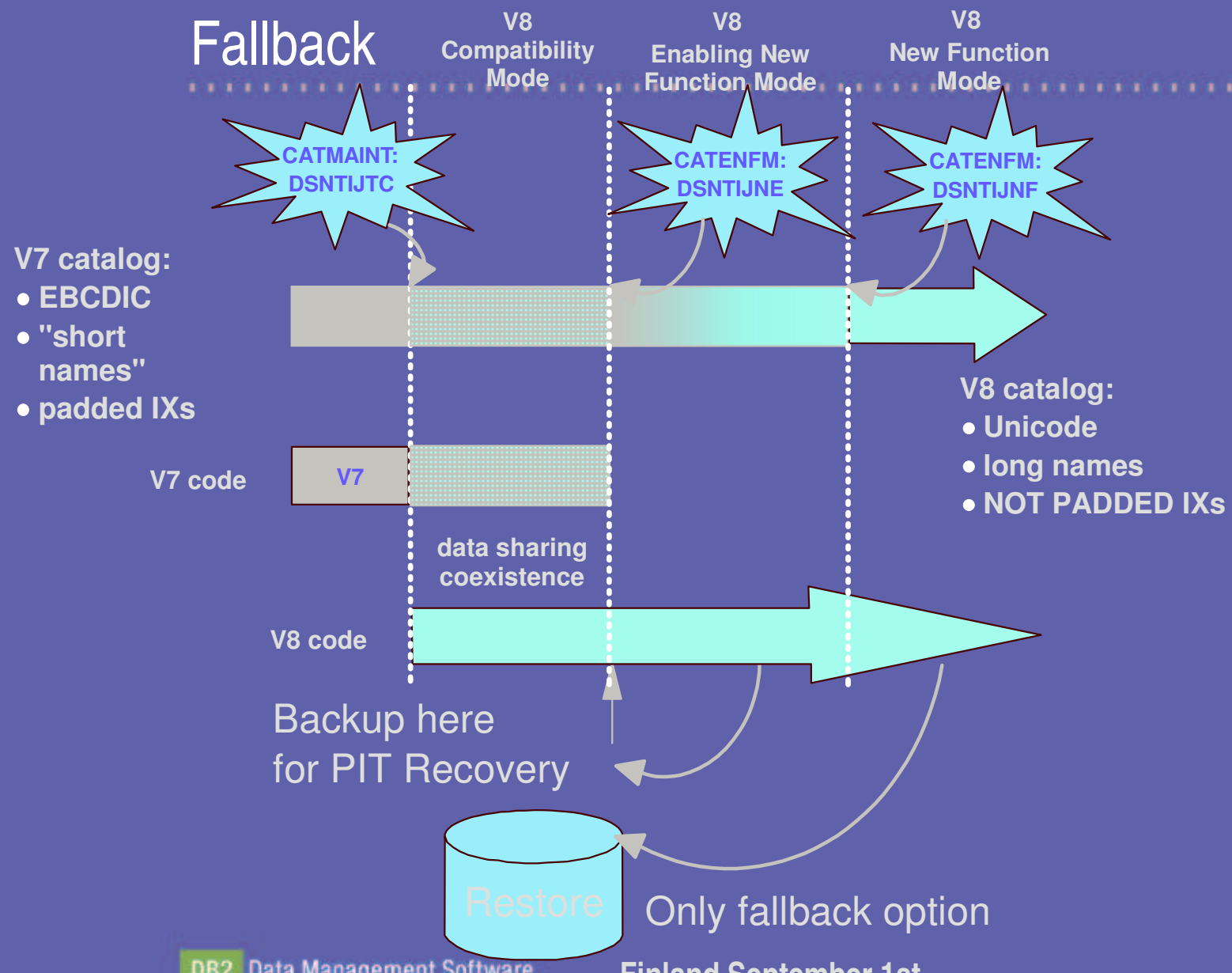
- Two distinct migration phases for a DB2 Version 8 migration
  - A new phased migration protecting data during the migration process
- Phase 1 - new release migration processing
  - 'Normal' release migration process as for previous releases - CATMAINT
  - Completion of this phase places the DB2 subsystem into version 8 **Compatibility Mode**
    - Transitional phase
    - No new external functions such as long names
  - Fallback support to version 7 permitted
  - Ensure client satisfaction with V8 code supporting production
- Phase 2 - enable new function mode processing
  - 'Additional' Version 8 conversion processing - CATENFM
  - Catalog and directory conversion period **Enabling New Function Mode**
  - **NO** fall back to V7 now possible
- Phase 3 - New Function Mode
  - Final phase - enabled by running DSNTIJNF job
  - All new functions now available
  - Can fall back to ENFM by running job

Overview of migration and conversion



DB2 Data Management Software

Fallback



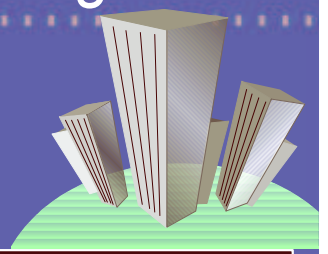
Finland September 1st

Compatibility Mode - What does it mean?

- No new release function can be used
  - Exception is online REORG of entire DB2 catalog
  - 64 bit benefits also available now
- Migrations from V7 to V8 CM will be short - minutes
  - includes a TS scan of SYSDBASE back to look for Type 1 indexes
- DSNTIJTC - CATMAINT is a single step
  - Single commit scope
    - Authorization check
    - Ensures catalog is at correct level
    - DDL processing
    - Additional processing and tailoring
    - Directory header page and BSDS / SCA updates
  - No need to have additional steps to look for unsupported objects
    - This is done instead during the above catmaint step

Migration to V8 CM -- evolution of the DB2 catalog

DB2 catalog continues to grow with every DB2 release.



DB2 Version	Table Spaces	Tables	Indexes	Columns	Table Check Constraints
V1	11	25	27	269	N/A
V3	11	43	44	584	N/A
V4	11	46	54	628	0
V5	12	54	62	731	46
V6	15	65	93	987	59
V7	20	82	119	1206	105
V8	22	83	133	1265	105

Bufferpool and other changes

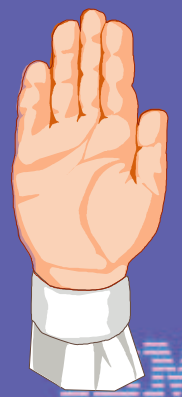
- Max BP size is lifted to lesser of 1TB and twice real storage
  - Max size of single or summation of all
  - Always allocated above 2GB
- Data space pools and Hiperpools are eliminated
- Migration
  - V8_VPSIZE = V7_VPSIZE + HPSIZE
  - Fallback uses V7_VPSIZE and HPSIZE
- Default BP0 size raised from 2000 to 20000
- Default BP32K size raised from 24 to 240
- EDM, RID, Sort pool max sizes increased
- Max number of concurrent threads increased
- Optional CI Size change - refer zparm DSVCI on DSNTIP7
  - CI size matches BP size for tablespaces
  - Implemented by REORG, RECOVER or LOAD REPLACE
  - part of ENFM for catalog

Things to check before proceeding with ENFM

- Run online REORGs (in V8 CM) against the catalog to check:
  - Timings
    - Elapsed times
    - Plan staged execution around outage windows
  - Data set sizings
    - Eliminate space failures during ENFM
    - Consider Increase the size of the catalog table space and index space VSAM data sets, to accommodate longer names
      - Review space for any user defined catalog indexes
  - For declared temporary tables
    - Ensure at least one table space with page size of 8K or greater
  - For data sharing
    - Ensure GBP8K0, GBP16K0, GBP32K is defined

Attention: you cannot fallback from Version 8 NFM

Do NOT convert to NFM until you are certain that you will not need to fallback



Catalog changes during ENFM processing

- Catalog changes during ENFM phase:
  - SYSOBDS catalog table is added
  - SYSDUMMY1 catalog table is moved to the new SYSEBCDC catalog ts
- No more columns added to existing catalog tables
- Columns changed in existing catalog tables
  - Most catalog and one directory tables have columns that are changing:
    - VARCHAR ==> VARCHAR(128 or greater)
    - CHAR ==> VARCHAR(128 or greater)
- Two indexes are created on the SYSOBDS table
- Many catalog indexes changed to NOT PADDED indexes
  - No user indexes on the catalog are changed to NOT PADDED
  - default is different for a migration or install
    - migration default is PADDED but install default is NOT PADDED
    - This can be changed via a zparm
    - Only applicable for indexes with at least one varying length key column.

Enabling New Function Mode Process -- DSNTIJNE

- Process consists of running the **DSNTIJNE** job after taking image copies.

If tailoring the DSNTIJNE job be careful NOT to change the table space processing order

- Changes types and lengths of existing catalog columns

- Converts the catalog data from EBCDIC to Unicode
  - ►Most of the catalog and one directory using online REORG
    - –17 catalog tsps, 1 dir tsp
- Changes buffer pools for some catalog table spaces
  - ►due to page size changes => Need 8K, 16K, and 32K BPs in NFM.

ENFM -- when should I run it?

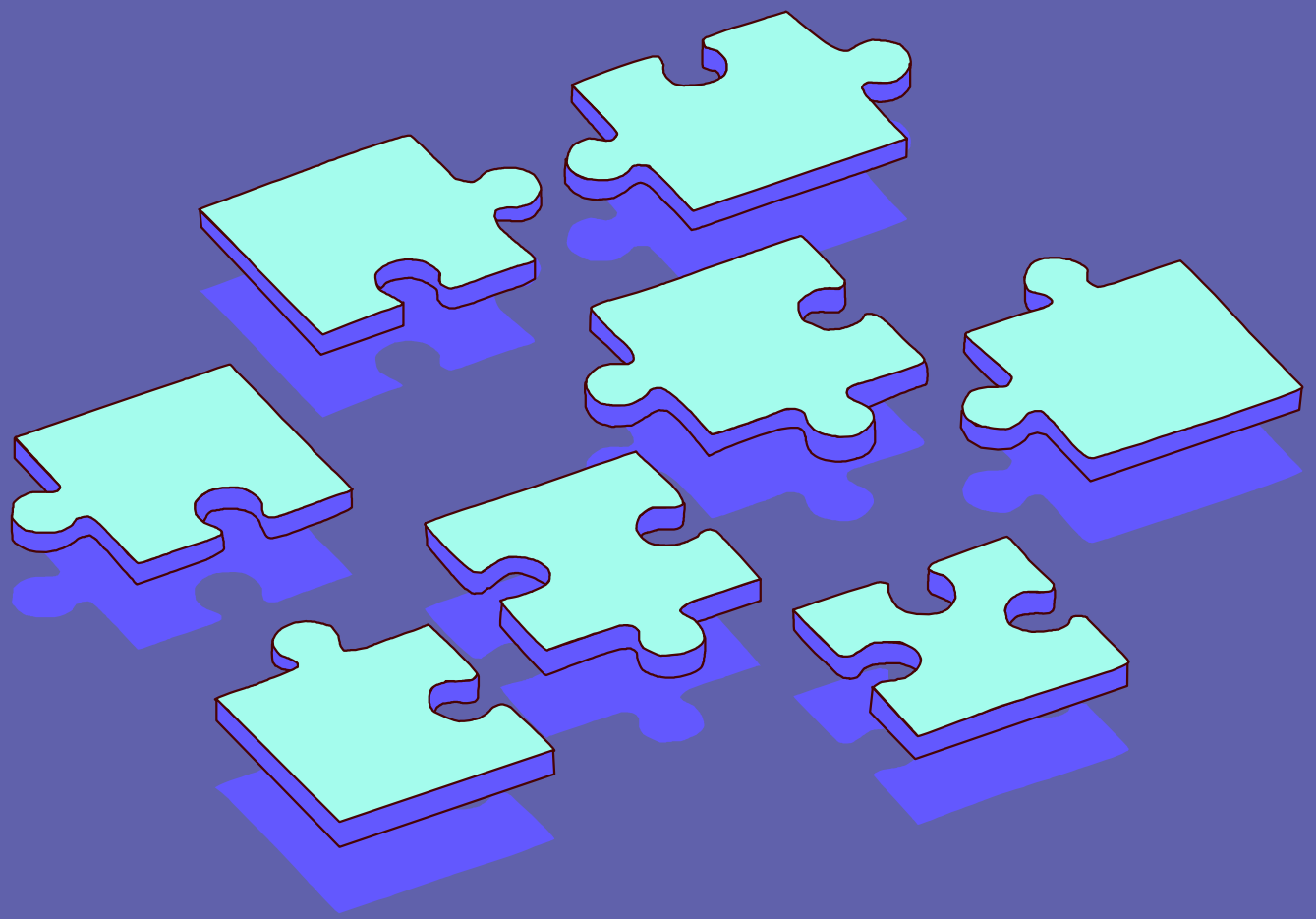
- Enabling process **CAN** span maintenance windows
  - ►You have the flexibility to plan the schedule
  - ►Can be stopped after the REORG of any table space
    - –(or long name DDL steps. These are the ENFMxxx0 steps)
  - ►Can be restarted without modification by submitting the DSNTIJNE job again
    - –Skips already processed table spaces
    - –Resumes processing at first table space not successfully converted
  - ►Recommendation is **NOT** to stay in ENFM phase any longer than necessary

- What happens if I have a space problem during ENFM process?
  - ►Succeeding steps are skipped
  - ►-TERM UTIL issued to make table space available
  - ►Can change space parameters and restart
    - –Skips already processed table spaces
    - –Resumes processing at first table space not successfully converted

New Function Mode

- You can now exploit the new DB2 Version 8 functionality
- Catalog is now mostly Unicode -- for 18 out of 22 catalog table spaces, and 1 directory table space
  - Note that the SYSALTER TS is created in V8 CM as a Unicode TS and this is why 18 of 22 are unicode...even though only 17 catalog TS's are processed during enfm
- No fallback or coexistence with V7
- No going back to Version 8 compatibility mode
- You can return to enabling new function mode
  - DSNTIJEN
  - Prevents users from exploiting new functions
  - DSNHDECP change for NEWFUN NO

Migration question time



TOOLS

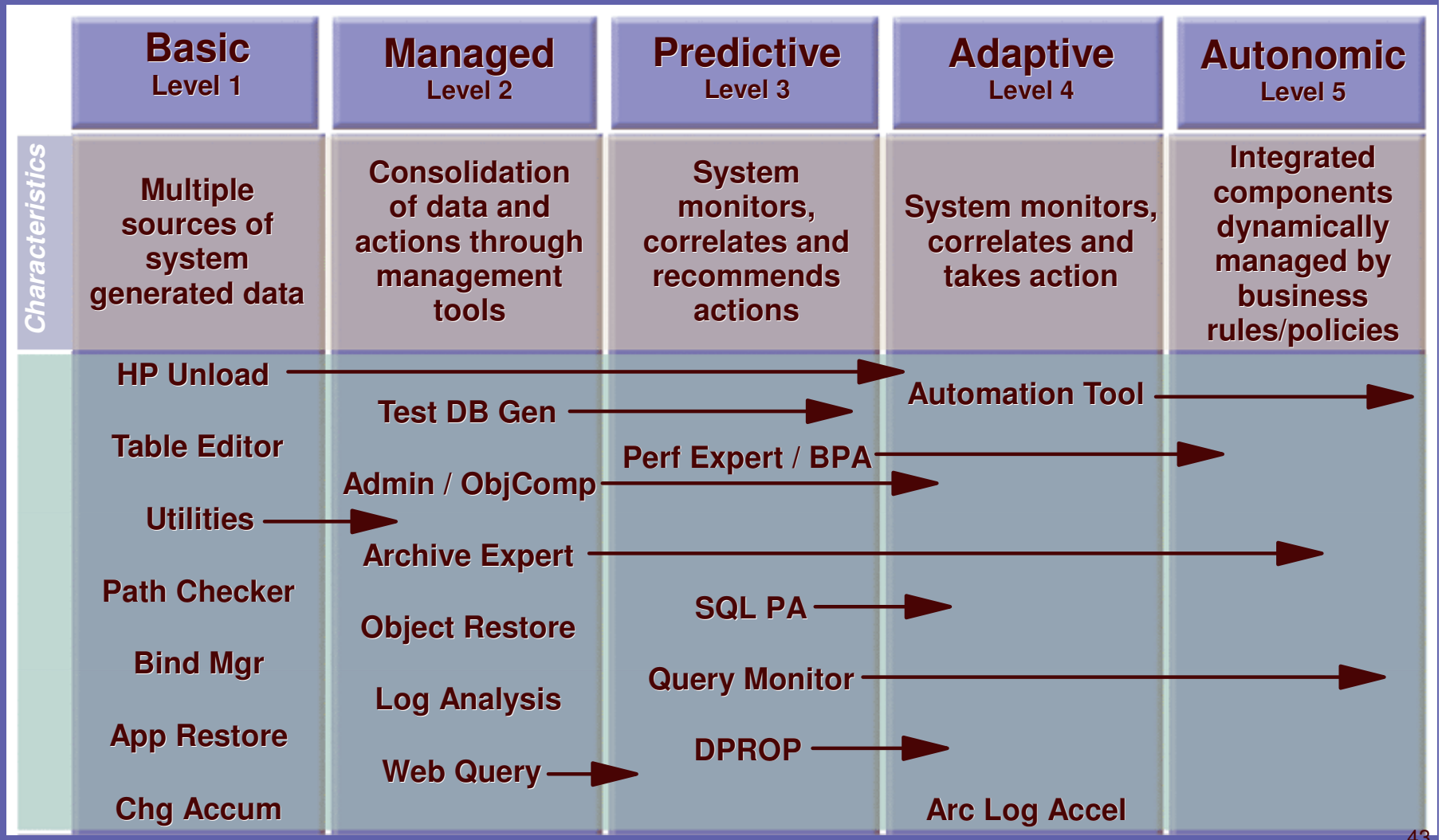
- On Demand
- Autonomic
- DB2 UDB for z/OS Version 8
- System wide Backup and Recovery
- Utilities
- DB2 Tools
  - Administration
  - Performance
  - Archiving
  - Test Database Generator
  - DB2 Grouper
- Summary

Evolving to Autonomic Computing

	Basic Level 1	Managed Level 2	Predictive Level 3	Adaptive Level 4	Autonomic Level 5
Characteristics	Multiple sources of system generated data	Consolidation of data and actions through management tools	System monitors, correlates and recommends actions	System monitors, correlates and takes action	Integrated components dynamically managed by business rules/policies
Skills	Requires *extensive, highly skilled* IT staff	IT staff *analyzes and takes actions*	IT staff *approves and initiates actions*	IT staff *manages performance* against SLAs	IT staff *focuses* on enabling business needs
Benefits		Greater system awareness Improved productivity	Reduced dependency on deep skills Faster/better decision making	Balanced human/system interaction IT agility and resiliency	Business policy drives IT management Business agility and resiliency

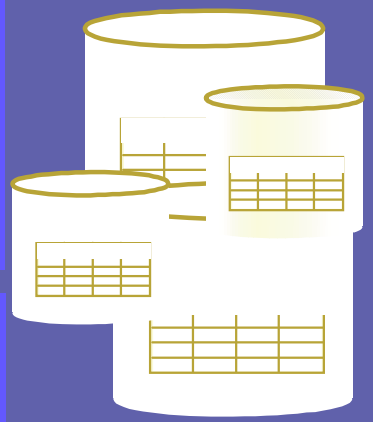
Manual → Autonomic

Evolving DB2 for z/OS Tools to Autonomic Computing



DB2 for z/OS Tools Product Portfolio

- DB2 Administration Tool
- DB2 Object Comparison Tool
- DB2 Automation Tool
- DB2 Utilities Suite
- DB2 HighPerf Unload



- DB2 Performance Expert
  - DB2 Performance Monitor
  - DB2 Bufferpool Analyzer
- DB2 Query Monitor
- DB2 SQL Performance Analyzer

- DB2 DataPropagator
- DB2 Archive Log Accelerator Tool
- DB2 Change Accumulation Tool
- DB2 Log Analysis Tool
- DB2 Object Restore Tool
- DB2 Recovery Expert

- DB2 Data Archive Expert
- DB2 Bind Manager
- DB2 Test Database Generator
- DB2 Path Checker
- DB2 Table Editor
- DB2 Web Query Tool

available on zSeries and multiplatforms

currently only available on multiplatforms

DB2 Data Management Software

Finland September 1st

IBM

End of Presentation

Backup

Data sharing enhancements

- Batching of GBP writes and castouts
  - Write / castout multiple pages in a single CF operation
  - Improved data sharing performance, especially for batch updates
  - Requires z/OS R4, CFLEVEL=11
- Reduced global lock contention for tablespace L-locks
  - IX / IX and IX / IS TS locks no longer hit XES-level contention across members
  - Improved data sharing performance, especially for OLTP
  - Reduced weight behind commendation for RELEASE(DEALLOCATE)

Data sharing enhancements ...

- Changed pages written to GBP at Phase 1 instead of Phase 2
  - Some transaction managers spawn other transactions at SYNCPOINT
  - Spawned transactions can encounter "record not found" if it tries to read originating transaction's update from another member
  - Rare, but a few customers have reported it
  - Moving writes up to phase1 by default; removes need to monitor for this and to set IMMEDWRITE PH1 DSNZPARM or BIND option if needed
  - Equivalent performance for phase 1 vs. phase 2 writes
- Automatic LPL recovery
- Restart light to resolve in-doubt threads
- More efficient index leaf page split processing

Multi-row INSERT

- Inserts multiple rows on one API call
- Can be ATOMIC or NOT ATOMIC
- Can be static or dynamic SQL
- Significant performance boost

```
INSERT INTO T1 FOR :hv ROWS
    VALUES( :ARRAY1, :ARRAY2) ATOMIC;
```

Multi-row INSERT ...

```
STMT = 'INSERT INTO T1
          VALUES( ?, ?)
          FOR MULTIPLE ROWS ATOMIC';

PREPARE S1 FROM STMT;

EXECUTE S1 FOR :hv ROWS
             USING :ARRAY1, :ARRAY2;
```

Multi-row FETCH

- Returns multiple rows on one API crossing
- "Wide" cursor with locks on multiple rows
- Supports scrollable and non-scrollable
- Supports static and dynamic SQL

```
DECLARE C1 CURSOR
    WITH ROWSET POSITIONING
    FOR SELECT COL1, COL2 FROM T1;
OPEN C1;
FETCH FROM C1
    FOR :hv ROWS  INTO  :ARRAY1, :ARRAY2;
```

Positioned UPDATE/DELETE of multi-row FETCH

- Allows positioned UPDATE or DELETE to be used on a "wide" cursor

```
UPDATE T1 SET COL1='ABC'
   FOR CURSOR C1
   FOR ROW  :hv  OF ROWSET
```

GET DIAGNOSTICS

- Returns SQL error information
  - for overall statement
  - for each condition (when multiple errors occur)
- Supports SQL error message tokens greater than 70

```
INSERT INTO T1 FOR 5 ROWS VALUES(:ARRAY);
GET DIAGNOSTICS :ERR_COUNT = NUMBER;
   DO II = 1 TO ERR_COUNT;
      GET DIAGNOSTICS FOR CONDITION :II
            :RC = RETURNED_SQLSTATE;
   END;
```

INSERT within SELECT

- Elegant technique for retrieving values created or modified by DB2 during INSERT
  - Identity columns, sequence values
  - User-defined defaults, expressions
  - Columns modified by triggers
  - ROWIDs

EXAMPLE:

```
SELECT C1, C2, C3, C4, C5 FROM
  INSERT (C1, C5) INTO T1
    VALUES('ABC', CURRENT DATE);
```

Identity column improvements

- ALTER support for identity columns
  - RESTART WITH value
  - INCREMENT BY
  - MINVALUE
  - MAXVALUE
  - CYCLE/NO CYCLE
  - CACHE/NO CACHE

Sequences

- Useful for porting Oracle applications
- New SQL support:
  - CREATE SEQUENCE
  - ALTER SEQUENCE
    - RESTART WITH value
    - INCREMENT BY
    - MINVALUE
    - MAXVALUE
    - CYCLE/NO CYCLE
    - CACHE/NO CACHE
  - GRANT/REVOKE privileges for the sequence object

Materialized query tables (MQT)

- Previously known as "automatic summary tables"
- Optimizer can now rewrite queries to access MQT instead of the base table / view
  - Significant performance improvement
- Two types of MQTs
  - System-maintained via SQL REFRESH statement
  - User-maintained via triggers, batch updates, etc.

Enhanced Unicode support

- String data in DB2 catalog now stored in Unicode
  - Allows SQL to contain Unicode literals and names
  - Provides better integration with Java and Microsoft technologies
- Important to key ISVs (SAP, PeopleSoft, Siebel, ...)

Multiple CCSIDs per statement

- A single SQL statement can now intermix EBCDIC, ASCII and Unicode tables and host variables
- Unlike host variables are translated to column CCSID
- Unlike column CCSIDs are "promoted" to Unicode
  - May require query to be materialized / sorted

```
SELECT T1.COL_EBCDIC ||
       T2.COL_ASCII  ||
       T3.COL_UNICODE FROM T1,T2,T3;
```

Volatile table support

- Forces index access for tables that have unpredictable cardinality
- Significant performance improvement for some SAP applications

```
CREATE TABLE XYZ ..... VOLATILE
```

DB2 V8 catalog

- Longer names (128 byte varchar Unicode)
  - Table, view & alias
  - Column (30)
  - Schema or authorization id
  - UDF, stored procedures, triggers, package
- Some page sizes greater than 4K
  - 4K, 8K, 16K, 32K pages in DB2 catalog
- Indexes longer than 255 bytes
- Contains Unicode data