IBM Systems and Technology Group

DB2 9 for z/OS Ingredients for SAP

Johannes Schuetzner

IBM Boeblingen Lab

IBM System z™ Software Teleconference
September 4, 2007

DB2 V9 – Numerous SAP Relevant Features

- SHRLEVEL(REFERENCE) for REORG of LOB tablespaces
- Online RENAME COLUMN
- Online RENAME INDEX
- Online CHECK DATA and CHECK LOB
- Online REBUILD INDEX
- Online ALTER COLUMN DEFAULT
- More online REORG by eliminating BUILD2 phase
- Faster REORG by intra-REORG parallelism
- Renaming SCHEMA, VCAT, OWNER, CREATOR
- LOB Locks reduction
- Skipping locked rows option
- Tape support for BACKUP and RESTORE SYSTEM utilities
- Recovery of individual tablespaces and indexes from volume-level backups
- Enhanced STOGROUP definition
- Conditional restart enhancements
- Histogram Statistics collection and exploitation
- WS II OmniFind based text search
- DB2 Trace enhancements
- WLM-assisted Buffer Pools management
- . . .

- Global query optimization
- Generalizing sparse index and in-memory data caching method
- Optimization Service Center
- Autonomic reoptimization
- Logging enhancements
- LOBs network flow optimization
- Faster operations for variable-length rows
- NOT LOGGED tablespaces
- Index on expressions
- Universal Tablespaces
- Partition-by-growth tablespaces
- APPEND option at insert
- Autonomic index page split
- Different index page sizes
- Support for optimistic locking
- Faster and more automatic DB2 restart
- RLF improvements for remote application servers such as SAP
- Preserving consistency when recovering individual objects to a prior point in time
- CLONE Table: fast replacement of one table with another
- Index compression
- . . .

- DECIMAL FLOAT
- BIGINT
- VARBINARY, BINARY
- TRUNCATE TABLE statement
- MERGE statement
- FETCH CONTINUE
- ORDER BY and FETCH FIRST n ROWS in sub-select and full-select
- ORDER OF extension to ORDER BY
- INTERSECT and EXCEPT Set Operations
- Instead of triggers
- Various scalar and built-in functions
- Cultural sort
- LOB File Reference support
- XML support in DB2 engine
- Enhancements to SQL Stored Procedures
- SELECT FROM UPDATE/DELETE/MERGE
- Enhanced CURRENT SCHEMA
- IP V6 support
- Unified Debugger
- Trusted Context
- Database ROLEs
- Automatic creation of database objects
- Temporary space consolidation

These are most important V9 features and most of them apply to SAP

Many of them were explicitly requested by SAP development or SAP customers

DB2 9 for z/OS with SAP

- **"IBM and SAP have cooperated very closely on DB2 9 for z/OS and we look forward to supporting our customers with these new capabilities."**
  Torsten Wittkugel, Vice President of Database and Operating System Platform Development at SAP
  - biz.yahoo.com/iw/070306/0223133.html

- **DB2 9 for z/OS certified by SAP in July 2007**
  - For all current SAP releases (4.6 and later)
  - See SAP Note 1043951
  - Whitepaper „SAP on IBM DB2 for z/OS: Best Practice for Installing or Migrating to DB2 9“
    - service.sap.com/solutionmanagerbp

Universal Tablespaces

Problem

A tablespace needs both partitioned and segmented organization:

- it's larger than 64GB
- inter-partition parallelism or independent processing is needed
- partition scope operations (ADD,ROTATE) apply
- rows are variable in length and a fast insert is required
- mass delete operations should be fast

Solution

- A hybrid between partitioned and segmented organization
- One table per UTS only
- Incompatible with MEMBER CLUSTER
- Two types:
  1. Partitioned-by-growth
     - always UTS
     - described on the next page
  2. Partitioned-by-range
     - traditional partitioned tablespaces
     - optionally UTS

```
CREATE TABLESPACE …
  SEGSIZE    integer
  NUMPARTS integer
```

Partition By Growth

Problem

A table's growth is unpredictable (it could exceed 64GB) and there is no convenient key for range partitioning.

Partitioning by a ROWID column introduces additional tablespace administration overhead:

- estimating optimal number of partitions
- ADDing partitions if necessary
- less then optimal space utilization

Solution

CREATE TABLESPACE … *explicit specification*
  MAXPARTITIONS integer

CREATE TABLE … *implicit specification*
  PARTITIONED BY SIZE EVERY integer G

Associated SYSTABLESPACES columns

MAXPARTITIONS	= max number of partitions
PARTITIONS	= actual number of partitions
TYPE	= G

- Only single-table tablespace
- Universal Tablespace organization: although the table space is partitioned, the data within each partition is organized according to segmented architecture
- Incompatible with MEMBER CLUSTER, ADD PARTITION, ROTATE PARTITION

→ **Automated operations**

Partition By Growth

- **MAXPARTITIONS can be ALTERed but observe the limits that are dependent on page size and DSSIZE**

DSSIZE \ Page Size	4K	8K	16K	32K
1–4 GB	4096	4096	4096	4096
8 GB	2048	4096	4096	4096
16 GB	1024	2048	4096	4096
32 GB	512	1024	2048	4096
64 GB	256	512	1024	2048

- **REORG can add new partitions (except if there are LOB or XML columns)**
- **REORG will not remove empty partitions, but it can shrink them to contain a header and space map page, again, subject to absence of LOB and XML columns**

Automatic Creation of Database Objects

Problem

- Database objects that have different meanings for different database platforms must be transparent to database-platform agnostic applications, e.g. database and tablespace
- Some of the DB2 database objects must be manually created although all their attributes are implicitly known to DB2, e.g. indexes that enforce primary and unique keys

Solution

Unless explicitly specified, DB2 will implicitly create the following database objects:

- Database
- Tablespace
- Index that enforces primary key
- Index that enforces unique key
- ROWID index
  - For ROWID columns defined as GENERATED BY DEFAULT
- LOB tablespace
- LOB auxiliary table
- LOB auxiliary index

→ **SAP DDIC relies on it (SAP 7.1)**

Automatic Creation of *Database*

- Automatically (implicitly) created databases are used when IN clause is omitted at CREATE TABLE. There are two possible outcomes:
  1. A new database is created
     - Namespace DSN00001 – DSN60000
     - Schema: SYSIBM
  2. An existing implicitly created database is used
     - Start reusing existing databases after reaching 60000
     - Wrap-around fashion
     - Therefore, implicitly created databases can contain multiple tablespaces
- Explicitly created tables and tablespaces in implicitly created databases are not allowed
- Implicitly created databases can be explicitly dropped

Automatic Creation of *Tablespace*

If CREATE TABLE does not include explicit tablespace specification, a tablespace is automatically (implicitly) created with the following attributes:

- Partition-by-growth universal tablespace
  - SEGSIZE=4
  - DSSIZE=4G
  - MAXPARTITIONS=256
- Compression is controlled by zparm IMPTSCMP
  - Default is COMPRESS=NO
- Dataset creation is controlled by zparm IMPDSDEF
  - Default is DEFINE=YES
- Sliding scale for optimizing secondary extent allocation is used by default (MGEXTSZ=YES)
- The default buffer pool is determined based on the ZPARM settings
  - Buffer pool page size is derived automatically using larger page size if max record size reaches 90% of capacity of the needed size
  - LOB tablespaces can be assigned to separate buffer pools

BACKUP and RESTORE SYSTEM Enhancements

Problem

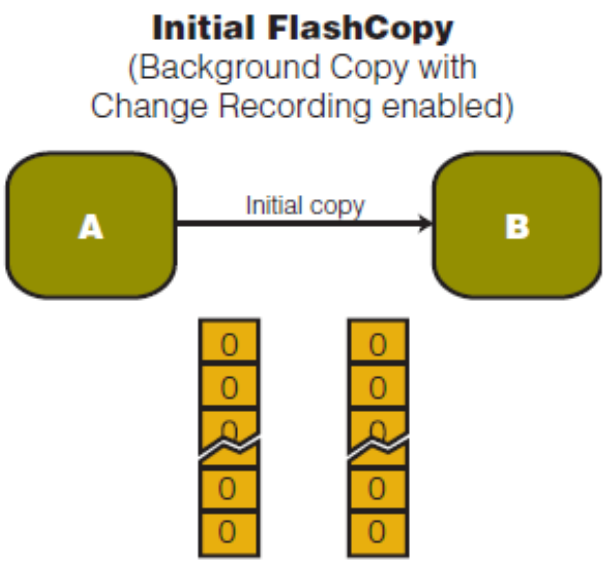
- Off-loading system-level backups to tape and restoring it from tape is a manual process that is not driven by the BACKUP and RESTORE SYSTEM utilities
- Despite availability of system-level backup many sites produce tablespace and index image copies for simpler recoveries of the individual objects

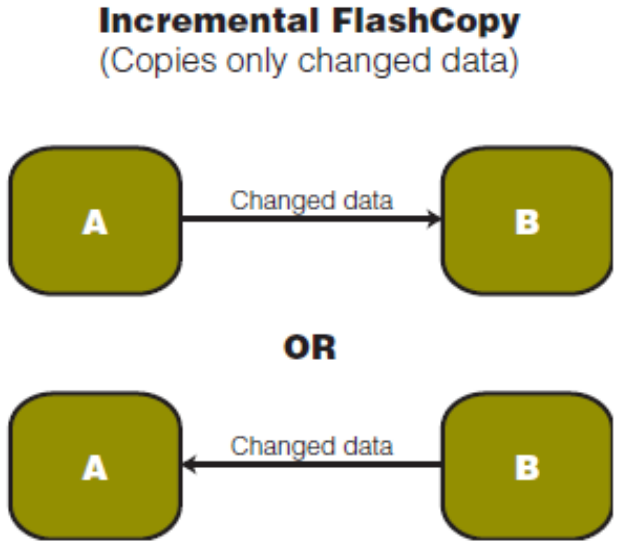
Solution

- Full integration of tape into the BACKUP SYSTEM utility as the target for preserving system-level backups
- Full integration of tape into the RESTORE SYSTEM utility as the source for restoring previously saved system-level backups
- Automatic using of system-level backups as alternative sources for recovering individual tablespaces and indexes
- Enhancing MODIFY RECOVERY to support system-level backups as the primary means of backing up DB2 data

Incremental FlashCopy

- **Introduced by DFSMShsm in z/OS 1.8**
  - Initial incr. FlashCopy creates full base backup
  - Change recording keeps track of changes
  - Subsequent incr. FlashCopies copy changed tracks to backup volumes only (overriding initial backup)
- **Minimizes I/O impact**
- **Considerably reduces elapsed time of physical copy**





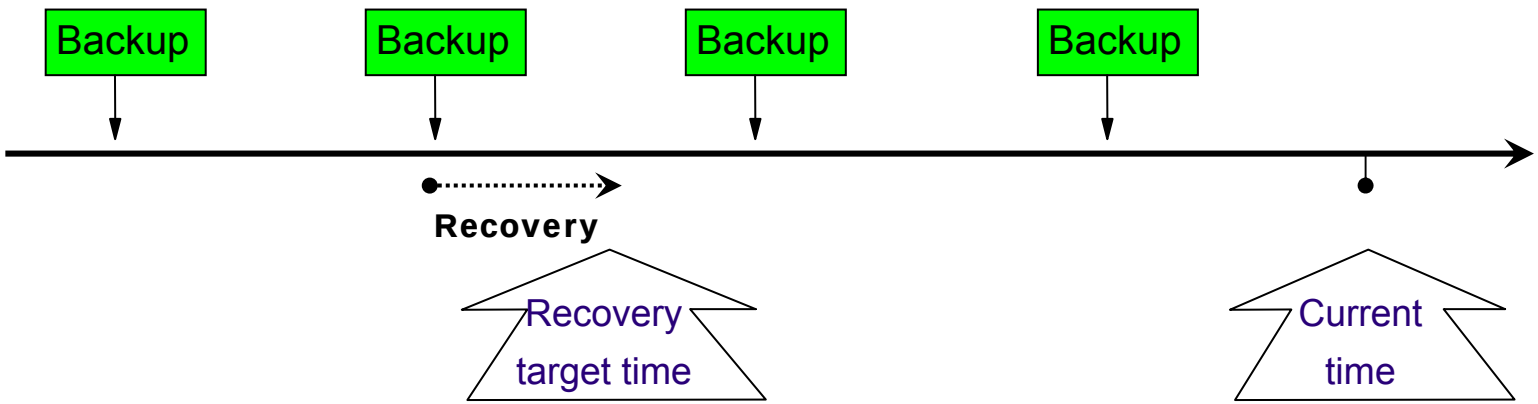
DB2 9: Incremental DB2 system-level backups

- **BACKUP SYSTEM explicitly supports incremental FlashCopy**
- **Via DFSMShsm**
- **New BACKUP SYSTEM options:**
  - **`ESTABLISH FCINCREMENTAL`**
    - Passes **`FCINCREMENTAL`** keyword to DFSMShsm
    - Takes initial full FlashCopy backup and enables change recording
  - **`END FCINCREMENTAL`**
    Takes last incremental Flashcopy and withdraws incr. FlashCopy relationship

Incremental DB2 system-level backups

- **Based on incremental FlashCopy**
  - If used, history of backups on disk gets lost
    - DB2 cannot use them for PIT recovery (always forward recovery)
  - To keep backup history, dump backups to tape
    - Backups on tape always full

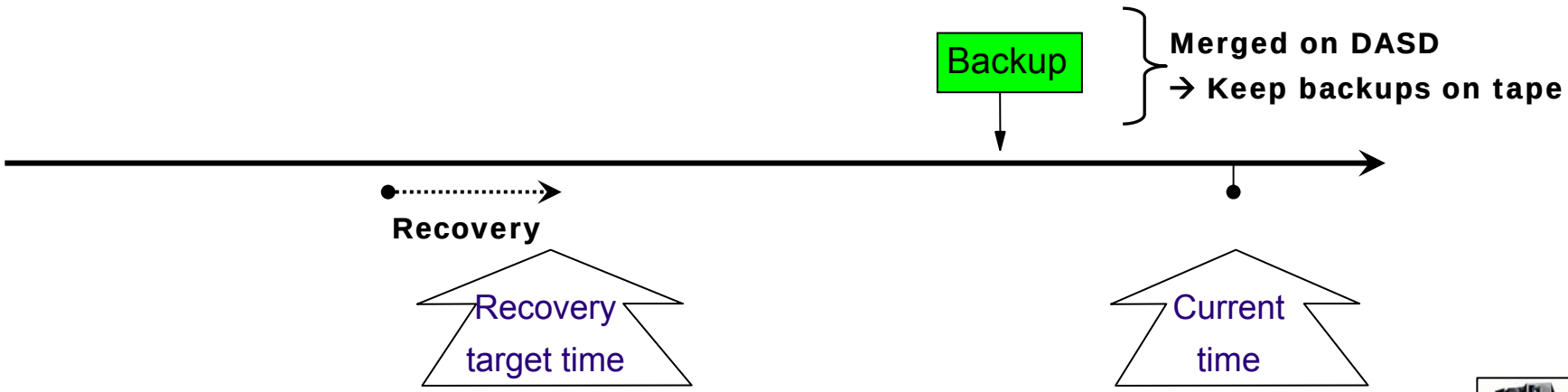
Full FlashCopy backups:



Incremental DB2 system-level backups

- **Based on incremental FlashCopy**
  - If used, history of backups on disk gets lost
    - DB2 cannot use them for PIT recovery (always forward recovery)
  - To keep backup history, dump backups to tape
    - Backups on tape always full

Incr. FlashCopy backups:



Preserving Consistency at Recovery to a Prior Point in Time

Problem

- When recovering a DB2 object to a point in time at which there were uncommitted changes for that object, the recovered object is left in an inconsistent state
- In order to ensure that prior point in time recoveries result in consistent objects, frequent quiesce points need to be established. However:
  - Quiesce can be very disruptive process for concurrent transactions
  - Even when quiesce points exist, they might not coincide with the point to which an object needs to be recovered

Solution

- RECOVER utility enhanced to automatically detect uncommitted transactions running at the recovery point in time and roll them back
- RECOVER TOLOGPOINT and TORBA will always recover with consistency
- RECOVER TOCOPY, TOLASTCOPY and TOLASTFULLCOPY using SHRLEVEL CHANGE copy will continue working as in V8

Online REBUILD INDEX

Problem

- During rebuilding of an index the tablespace is unavailable for all data modifying operations
- This is especially restrictive for very large tables where the rebuild process typically takes considerable amount of time

Solution

```
REBUILD INDEX …
  SHRLEVEL REFERENCE
  DRAIN_WAIT integer
  RETRY integer
  RETRY_DELAY integer
```

```
REBUILD INDEX …
  SHRLEVEL CHANGE
  DRAIN_WAIT integer
  RETRY integer
  RETRY_DELAY integer
  MAXRO integer | DEFER
  LONGLOG CONTINUE | TERM |DRAIN
  DELAY 1200 | integer
```

Online REORG Enhancements

Problem

1. Reorganizing individual partitions causes data unavailability due to BUILD2 phase
   - BUILD2 phase updates logical parts of NPIs
   - In case of parallel REORGs on different partitions BUILD2 causes an eruption of disk I/O requests and often leads to reaching critical bufferpool thresholds
2. Reorganization of large tablespaces takes very long
   - Most of the REORG phases are single threaded
3. Some defaults for Online Reorg are not appropriate

Solution

1. BUILD2 phase eliminated
   - A complete copy of NPIs created, maintained like other objects and switched at the end
   - Concurrent, separate REORGs of different partitions are no longer supported, but different partitions can be reorganized within a single REORG
2. Intra-REORG parallelism
   - Partitions unloaded and reloaded in parallel
   - Parallelism also used in the LOG phase
3. Defaults for following options adjusted for more usability
   - TIMEOUT, RETRY, DRAIN_WAIT, RETRY_DELAY, MAXRO

Renaming VCAT, SCHEMA, OWNER and CREATOR

Problem

- Some administrative operations (such as cloning a DB2 system) often includes the need to change any or all of the following:
  VCAT, SCHEMA, OWNER, CREATOR
- These are error prone, time consuming and risky operations:
  - The changes are spread throughout DB2
  - Preserving coherency between the catalog, the dictionary and actual physical objects is absolutely crucial

Solution

```
CATMAINT UPDATE …
  SCHEMA SWITCH (schema_name) TO (new_schema_name)
  VCAT SWITCH (vcat_name) TO (new_vcat_name)
  OWNER FROM (owner_name) TO ROLE
```

- Performs authorization/semantics checking and serialization
- Updates catalog and directory to reflect the new names
- Invalidates plan, packages and statement cache
- Names to be changed must not have view, function, MQT and trigger dependencies, cannot be ‘SYSIBM’
- Use SCHEMA option to change owner, creator and schema names

Resource Limit Facility Improvements

Problem

The qualifiers used by RLF to identify processes for which CPU time is governed are not specific enough for most 'middleware' servers such as SAP:

- Plan name is fixed to DISTSERV
- CLI and JDBC drivers use a common set of packages
- Authorization ID is typically the same for the entire workload

Solution

- RLF governed processes can be now qualified by typical 'client-side' identifiers:
  - End-user ID
  - Transaction name
  - Workstation name
- Additionally, the process can be qualified by the IP address of the server that initiated request
- This applies to both the predictive and reactive governor
- The existing CLI and JDBC APIs are used to set the client-side identifiers.

→ **Control resources for BW**

DB2 Connect V9: New DB2 Driver for CLI and ODBC

Problem

With DB2 Connect V8, a DB2 Connect server with instance for each SAP instance needs to be deployed on every SAP app server

- Large footprint
- Separate installation
- Complex administration: Fixpaks needs to be applied on each app server

Solution

- DB2 Connect V9 introduces new DB2 Driver for CLI and ODBC
  - Licensing does not change
- ZIP file that contains library
  - No installation necessary
  - Size: 20 to 30 MB
- SAP will ship CLI and JDBC driver on SAP installation DVDs
  - Allows deployment of different versions for rolling maintenance
  - Allows central installation
  - Distributed through SAP marketplace
  - No network statistics
- SAP Note 1031213

Addressing security/compliance pressure

Problem

Regulatory compliance initiatives are impacting IT organizations in most countries/industries and are changing fast

Examples:
- Sarbanes-Oxley
- Basel II

Solution

New security capabilities in DB2 9

- **Database ROLEs**
  - 'Virtual authorization ID'
  - Enables you to temporarily grant privileges to DBA and to audit activities
- **Trusted security context**
  - User id and password only valid when used on specific app server

LOB Performance and Concurrency Enhancements

Problem

Very large number of locks even for UR scanners resulting in a high IRLM storage usage or lock escalations

The reason is LOB locks which are acquired for any data access operation in order to:

- control LOB space usage
- serialize readers and updaters of LOB columns

Solution

- For UPDATE, INSERT and DELETE, LOB lock avoidance will be attempted. If it fails, the resulting X-LOB lock will have *manual* duration only (unlock after the operation completion). Changed LOB data pages and its index pages are flushed out to the GBPs prior to the unlock in order to ensure consistency for UR readers on other data sharing members.
- For non-UR SELECTs, LOB lock will be no longer acquired
- For UR SELECTs, the resulting S-LOB locks will have *autorel* mode (lock is acquired and released immediately)
- LOB locks as means to control space reuse are no longer used. DB2 relies on other technique to achieve that

MERGE Statement

Problem

For a set of input rows update the target table when the key exists and insert the rows for which keys do not exist.
E.g.

- For activities whose description has been changed, update the description in table **archive**.
- For new activities, insert into **archive**.

Prior to V9 this has been coded as a loop over conditional INSERT and UPDATE statements

Solution

```
MERGE INTO archive AR
  USING VALUES (:hv_activity, :hv_description) FOR :hv_nrows ROWS
    AS AC (ACTIVITY, DESCRIPTION)
  ON (AR.ACTIVITY = AC.ACTIVITY)
  WHEN MATCHED THEN UPDATE SET DESCRIPTION = AC.DESCRIPTION
  WHEN NOT MATCHED THEN INSERT (ACTIVITY, DESCRIPTION)
                                VALUES (AC.ACTIVITY, AC.DESCRIPTION)
  NOT ATOMIC CONTINUE ON SQLEXCEPTION
```

→ **Performance boost for ABAP**

APPEND

Problem

All of the following applies:

- Critical, high insert rate workload needs better performance and all the conventional tuning steps have already been applied.
- Clustering is either not beneficial or more frequent reorganizations are acceptable
- Insert algorithm introduced by PQ87381 is still not fast enough or the prerequisites cannot be satisfied:
  - MEMBER CLUSTER
  - FREEPAGE=PCTFREE=0

Solution

CREATE TABLE … APPEND YES | NO

ALTER TABLE … APPEND YES | NO

- The APPEND YES results in a fast insert at the end of the table or appropriate partition at the expense of data organization and rapid table space growth.
- After populating with the APPEND option in effect, clustering can be achieved by running the REORG utility providing a clustering index has been explicitly defined.

→ **Beneficial for mass input**

Autonomic Index Page Split and New Index Page Sizes

Problem

- Sequential insert pattern in the middle of index causes a symmetrical (50:50) page split resulting in:
  - Increased index latch contention
  - More frequent index page splits
  - Inefficient space usage
- Note that the same pattern at the end of index causes an asymmetrical split leaving all existing keys on the splitting page

Solution

- New, autonomic index page split algorithm
  - asymmetrical splits for sequential insert patterns in the middle of index
  - symmetrical splits for random insert pattern
- Additional (8K, 16K and 32K) index page sizes
  - smaller size BP should be chosen for indexes with random insert patterns
  - larger size BP should be specified for indexes with sequential insert patterns.

CREATE | ALTER INDEX …
BUFFERPOOL bpname

CREATE | ALTER DATABASE …
INDEXBP bpname

Not Logged Tablespaces – Use It for Extreme Cases Only

Problem

Performance degradation for workloads involving very large number of parallel inserts, updates and deletes due to:

- Log write latching
- Log data sets write bottlenecks

Very large volume of log data generated for these workloads (example: SAP Unicode migration)

At the same time **recoverability of data is not required**, e.g. the data can be recreated from its original source rather than from backups and logs

Solution

New tablespace attribute that controls whether Undo and Redo information for that tablespace and associated indexes are logged or not.

CREATE or ALTER TABLESPACE …
LOGGED|NOT LOGGED

Typical process sequence:

1. Take full image copy
2. Turn off logging
3. Make massive changes
4. Ensure that no other concurrent, non-repeatable changes happen
5. Turn on logging
6. Take full image copy

Get familiar with ramifications to rollbacks, restarts, lock contentions, data sharing, long running transactions

Data Warehousing and Optimizer Enhancements

- **Dynamic Index ANDing for Star Schema**
- **Optimization Service Center (OSC)**
  - Stats advisor
- **Global query optimization**
  - Considering effects of one query block on another
  - Considering reordering query blocks
- **Histogram statistics**
  - Quantiles
  - Significant benefit to optimizer's estimate for range predicates
- **Generalizing sparse index and in-memory data caching method**
  - Create in-memory index when adequate index does not exist
- **Parallelism enhancements**
- **INTERSECT and EXCEPT set operations**

Dynamic Index ANDing

- **Enhanced Star Join access method**
- **Better exploitation of SAP BW single column fact table indexes**
- **Consistent parallelism**
  - Independent filtering dimension access in parallel
  - Fact table access (and post fact table) in parallel
- **Adaptive query execution based upon runtime filtering**
  - Less filtering dimensions can be discarded for pre-fact table access
- **RID pool overflow to workfile**
- **Less dependent on perfect statistics**
  - Although optimizer costing is still performed
  - Better tolerance of less than perfect access path choice

→ **Stable access path that considerably reduces tuning efforts for SAP BW**

Dynamic Index ANDing Example …

Filtering may come from multiple dimensions

. . .

AND (DP.C5 = 0)

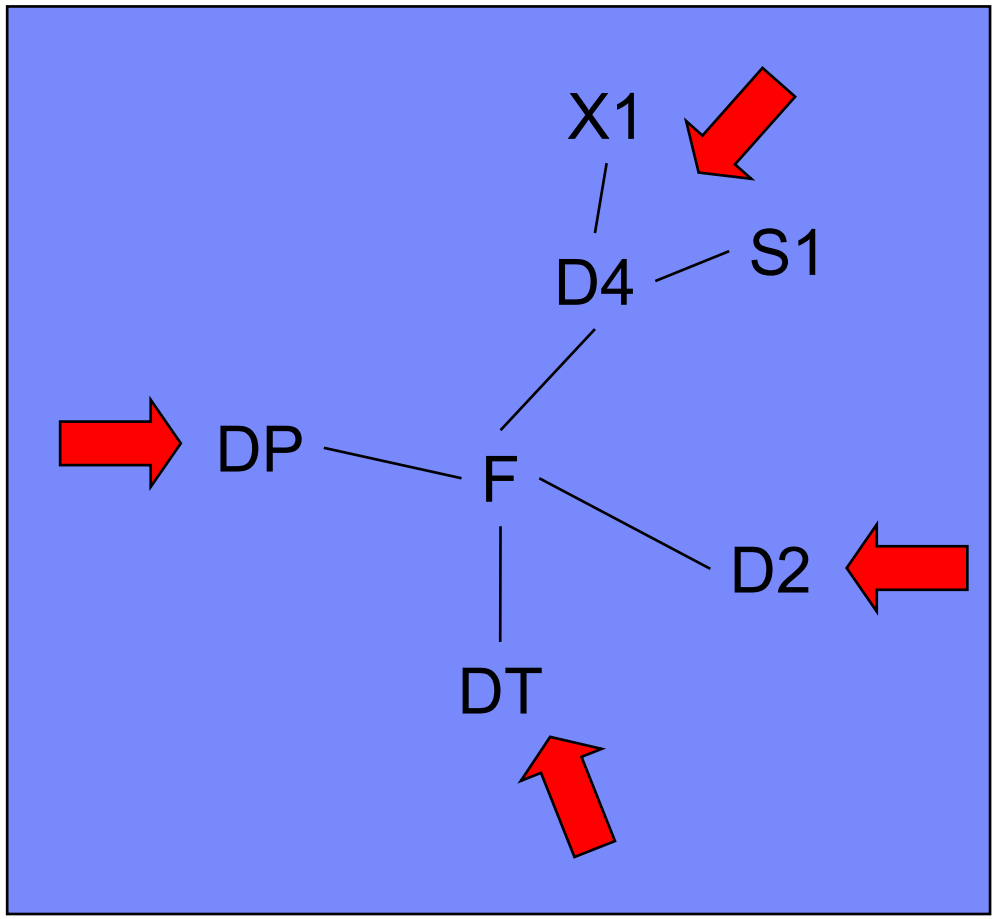
AND (DP.C3 <= 194744)

AND (D2.C1 = 135)

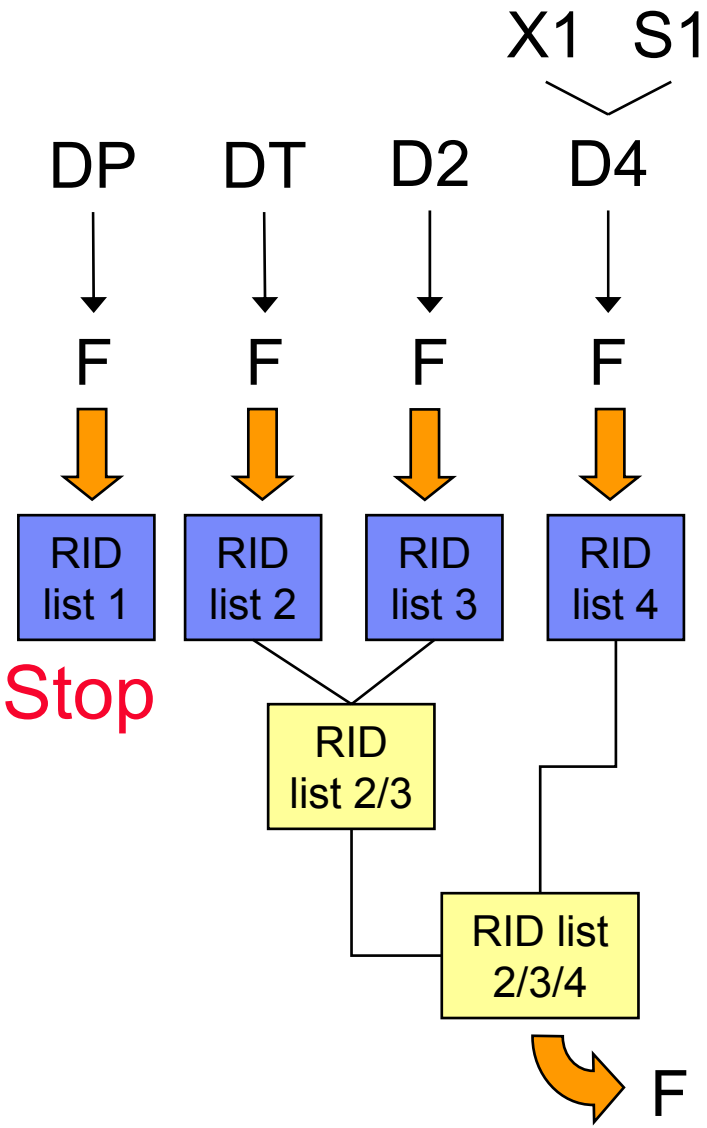
AND (S1.C7 BETWEEN 'H' AND 'N')

AND (DT.C9 BETWEEN 50 AND 60)

Typically, there are multiple single column indexes to support any filtering combination



Dynamic Index ANDing Example



Filtering dimensions accessed in parallel

Join to respective fact table indexes

Build RID lists

As filtering efficiency might not be known until runtime, the runtime optimizer may terminate parallel stream(s) that happen to provide poor filtering

Remaining RID lists are intersected

Final RID list used for parallel fact table (and remaining dimension) access

Miscellaneous

- **Adding SHRLEVEL(REFERENCE) for REORG of LOB tablespaces**
  - Unlike SHRLEVEL(NONE) which is in-place reorganization, it uses the standard DB2 shadow data sets reorganization
  - Enables reclaiming of physical space
- **ALTER TABLE RENAME COLUMN source-column-name TO target-column-name**
  - Online operation, no moving of data
  - Not allowed if column referenced in a view or has a trigger defined on it
- **RENAME INDEX old-index-name TO new-index-name**
  - Online operation, no moving of data
  - Invalidates cached statements that depend on the source index
- **Online CHECK DATA and CHECK LOB**
- **Online ALTER TABLE ALTER COLUMN SET DEFAULT / DROP DEFAULT**
- **Utilities CPU reduction**
- **DB2 trace filtering**

Miscellaneous

- **Conditional restart enhancement**
  - Allow timestamp specification at conditional restart (CRESTART statement) and system recovery (SYSPITRT keyword)
  - Tolerate missing checkpoint in BSDS by DSNJU003 when creating conditional restart specification (CRESTART)
  - Automatically scan the log backward to find the appropriate checkpoint at subsequent DB2 restart
- **Faster and more automatic DB2 restart in data sharing**
  - Avoiding certain locks for GBP dependant objects and open the objects involved in restart as early as possible
  - Automatic GBP Recovery
  - Support table level retain locks for postponed abort unit of recovery
- **Utility TEMPLATE switching**
  - Most commonly needed for changing the UNIT parameter (e.g. directing small data sets to DASD and large to TAPE)
  - Allows other parameters changes as well

Miscellaneous

- **Logging enhancements**
  - Faster logging in data sharing
  - Striping for archive log data sets
  - Improved archive log read and write performance
- **LOBs network flow optimization**
- **Faster operations for variable-length rows**
  - New, reordered row format
- **Enhanced CURRENT SCHEMA**
  - Removing the V8 restriction that disallows CREATE statements when the value of CURRENT SCHEMA was different from the value in CURRENT SQLID
- **IP V6 support**
  - 128-bit address space, with no practical limit on global addressability
    - 6.6 x 10^{23} addresses per square meter of the planet's surface
  - Colon-hexadecimal display representation
    - e.g. FEDC:BA98:7654:3210:FEDC:BA98:7654:3210

Miscellaneous

- **DECIMAL FLOAT**
- **BIGINT**
- **VARBINARY, BINARY**
- **CLONE TABLE**
  - Creating a table's clone (meta data only, except table name)
  - Fast swap of the contents between a table and its clone (equivalent to instantaneous LOAD REPLACE for both tables)
- **Index compression**
- **LOB File Reference support**
- **64-bit exploitation by DDF**
  - Special "shared private" with DBM1 to eliminate many of the data moves during SQL operations
- **WLM-assisted buffer pools management**

References

- **SAP developer network: SAP on DB2 for z/OS**
  - www.sdn.sap.com/irj/sdn/db2
- **IBM Website: SAP on DB2 UDB for z/OS on IBM System z**
  - www.ibm.com/servers/eserver/zseries/software/sap
- **SAP whitepaper „SAP on IBM DB2 for z/OS: Best Practice for Installing or Migrating to DB2 9“**
  - service.sap.com/solutionmanagerbp
- **DB2 Version 9.1 for z/OS**
  - www.ibm.com/software/data/db2/zos/db2zosv91.html
- **IBM redbook „Enhancing SAP by using DB2 9 for z/OS“**
  - www.redbooks.ibm.com, SG24-7239
- **IBM whitepaper „The ideal platform for SAP NetWeaver Business Intelligence - System z9 and DB2 for z/OS“**
  - www.ibm.com/solutions/sap/doc/content/resource/technical/1996216130.html

Legal information

Future plans articulated via statements of directions may change without notice.

Other company, product or service names may be trademarks or service marks of others such as:

- Java and all Java-based trademarks and logos are trademarks or registered trademarks of Sun Microsystems, Inc. in the United States, other countries, or both.
- Microsoft, Windows, Windows NT, and the Windows logo are trademarks of Microsoft Corporation in the United States, other countries, or both.
- UNIX is a registered trademark of The Open Group in the United States and other countries.

Thank You for Joining Us today!

Go to www.ibm.com/software/systemz to:

- Replay this teleconference
- Replay previously broadcast teleconferences
- Register for upcoming events

Johannes Schuetzner

schuetzner@de.ibm.com