"DB2 for LUW" Konzepte zur Datenspeicherung und deren Einsatz bei komplexen Geschäftsanwendungen, wie z. B. SAP

Karl Fleckenstein, Senior IT Architect,
( karl.fleckenstein@de.ibm.com )
SAP DB2 LUW Development Team,
IBM Development Lab Böblingen, Germany
Important Disclaimer

THE INFORMATION CONTAINED IN THIS PRESENTATION IS PROVIDED FOR INFORMATIONAL PURPOSES ONLY.

WHILE EFFORTS WERE MADE TO VERIFY THE COMPLETENESS AND ACCURACY OF THE INFORMATION CONTAINED IN THIS PRESENTATION, IT IS PROVIDED “AS IS”, WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED.

IN ADDITION, THIS INFORMATION IS BASED ON IBM’S CURRENT PRODUCT PLANS AND STRATEGY, WHICH ARE SUBJECT TO CHANGE BY IBM WITHOUT NOTICE.

IBM SHALL NOT BE RESPONSIBLE FOR ANY DAMAGES ARISING OUT OF THE USE OF, OR OTHERWISE RELATED TO, THIS PRESENTATION OR ANY OTHER DOCUMENTATION.

NOTHING CONTAINED IN THIS PRESENTATION IS INTENDED TO, OR SHALL HAVE THE EFFECT OF:

- CREATING ANY WARRANTY OR REPRESENTATION FROM IBM (OR ITS AFFILIATES OR ITS OR THEIR SUPPLIERS AND/OR LICENSORS); OR

- ALTERING THE TERMS AND CONDITIONS OF THE APPLICABLE LICENSE AGREEMENT GOVERNING THE USE OF IBM SOFTWARE.
Trademarks

IBM, the IBM logo, ibm.com, AIX and DB2 are trademarks of International Business Machines Corp., registered in many jurisdictions worldwide. Other product and service names might be trademarks of IBM or other companies. A current list of IBM trademarks is available on the Web at www.ibm.com/legal/copytrade.shtml.

Linux is a registered trademark of Linus Torvalds in the United States, other countries, or both.

Windows is a trademark 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.

SAP, SAP NetWeaver, SAP Business Information Warehouse, SAP BW, SAP NetWeaver BW and other SAP products and services mentioned herein are trademarks or registered trademarks of SAP AG in Germany and in several other countries.

Other company, product and service names may be trademarks or service marks of others.
Agenda

- DB2 Optimized for SAP
- Storage Architecture
  - DB2 Architektur
  - DB2 Tablespaces (SMS, DMS)
  - Automatic Storage
  - Tablespace with Reclaimable Storage

- Data Compression
  - Row Compression
  - Index Compression
  - Temp Table Compression
  - LOB Inlining

- Database Partitioning Feature (DPF)
- Range Partitioning
- Multi-Dimensional Clustering
- DBF + Range Partitioning + MDC
IBM Information Management

DB2 Optimized for SAP

2Q/08 NW 7.0 SR3
3Q/08 NW 7.0 EhP 1
- Turn-key HA solution
- Perf & TCO differentiators (MDC)
- Turn-key compression
- Performance Warehouse
- Integrated Workload Management
- Integrated MDC advisor
- Deferred Table Creation

1Q/07 NW 7.0
- Embedded database
- Unlimited scalability
- Minimal admin
- TCO (reduced storage costs, self tuning)

1Q/10 NW 7.0 EhP 2 SP02
4Q/09 NW 7.20 SP01
- Integrated Nearline Storage
- Top-Down, Revamped Monitoring
- Significant BI query performance boost
- Improved MDC Advisory

2Q/05 NW 2004
- Streamlined install
- Streamlined admin
- Initial BI MDC expl.
- Auto storage EE

Version 9.1
- Storage limits removed
- Near-0 admin
  - Memory
  - Storage
- Selected Autonomic / TCO features
- Compression

Version 9.5
- Integrated & automatic HADR
- Integrated Flash Copy
- TCO improvements
  - DPF
  - Memory Management
  - Statistics Management
  - Compression
- Perf improvements (eg MDC)
- Threaded Architecture
- DPF Scaling Improvements

Version 8.2.2
- Automatic storage admin
- Deployment optimized for SAP
- Tailored concurrency & perf improvements

© 2009 IBM Corporation
**DB2 Features for SAP**

<table>
<thead>
<tr>
<th>Deployment, Configuration, Administration</th>
<th>Version 8.2.2</th>
<th>Version 9.1</th>
<th>Version 9.5</th>
<th>Version 9.7</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>TCO reduction</strong></td>
<td>- Silent Install (Windows)</td>
<td>- Silent install (Unix)</td>
<td>- Easy fixpack deployment</td>
<td>- Online REORG improvements</td>
</tr>
<tr>
<td><strong>BI &amp; Query Performance</strong></td>
<td>- SAP tuner (DB2_WORKLOAD = SAP)</td>
<td>- Automatic configuration: IO cleaner, server</td>
<td>- Simplified configuration by multi-threaded architecture</td>
<td>- TEMP table compression</td>
</tr>
<tr>
<td><strong>Supportability, Backup &amp; Recovery, HA</strong></td>
<td>- Automatic storage (single node)</td>
<td>- Self tuning database global memory</td>
<td>- Advanced Memory Management: Automation for all memory areas</td>
<td>- Index compression</td>
</tr>
<tr>
<td></td>
<td>- Auto extend DMS tablespaces</td>
<td>- Deep compression</td>
<td>- Automatic dictionary creation for deep compression</td>
<td>- Migration to Automatic Storage</td>
</tr>
<tr>
<td></td>
<td>- Uniform page-size</td>
<td>- Automatic storage for DPF</td>
<td>- Simpler space reclaim for automatic storage and regular DMS</td>
<td>- Automatic MDC space reclaim</td>
</tr>
<tr>
<td></td>
<td></td>
<td>- SAP BI Optimizer enhancements</td>
<td>- Real Time Statistics</td>
<td>- LOB inlining (with compression)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>- Sort capacity cap</td>
<td>- Improved MDC roll out for fast deletion</td>
<td>- Support for LONG-&gt;LOB transition</td>
</tr>
<tr>
<td></td>
<td></td>
<td>- MDC roll out – stage 1</td>
<td>- Advanced DPF Monitoring</td>
<td>- Starjoin query improvements</td>
</tr>
<tr>
<td></td>
<td></td>
<td>- MDC roll in</td>
<td>- Parallel index creation</td>
<td>- DPF costing improvements</td>
</tr>
<tr>
<td></td>
<td></td>
<td>- Index heuristics</td>
<td>- HA Cluster Manager Integration</td>
<td>- Scan Sharing</td>
</tr>
<tr>
<td></td>
<td></td>
<td>- Concurrency enhancements</td>
<td>- Automatic Backup/Log Retention Management</td>
<td>- Enhanced Monitoring</td>
</tr>
<tr>
<td></td>
<td></td>
<td>- Uniform page-size</td>
<td>- Single system view backup for DPF databases</td>
<td>- Diagnostics improvements</td>
</tr>
<tr>
<td></td>
<td></td>
<td>- Complete deadlock analysis</td>
<td>- Enhanced Backup using Flash Copy</td>
<td>- HA Cluster Manager Integration</td>
</tr>
<tr>
<td></td>
<td></td>
<td>- One-shot optimizer service tool</td>
<td>- Tablespace space reclaim below HWM</td>
<td>- Automatic Backup/Log Retention Management</td>
</tr>
<tr>
<td></td>
<td></td>
<td>- Optimization guidelines</td>
<td>- Online Table Move</td>
<td>- Single system view backup for DPF databases</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>- Tablespace capacity increase</td>
<td>- Enhanced Backup using Flash Copy</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>- Compression support for replicated tables</td>
<td>- Enhanced Monitoring</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>- Tablespace space reclaim below HWM</td>
<td>- Diagnostics improvements</td>
</tr>
</tbody>
</table>

- **Version 8.2.2**
  - Silent Install (Windows)
- **Version 9.1**
  - Silent install (Unix)
  - Multiple DB2 deployments on the same machine
  - One DB2 image serves all: Install, upgrade, patch
  - Installation free client
  - Larger index key parts
  - Larger index names
- **Version 9.5**
  - Easy fixpack deployment
  - SA MP bundled and installed with DB2 on AIX and LINUX
  - Simplified configuration by multi-threaded architecture
  - Advanced Memory Management: Automation for all memory areas
  - Automatic dictionary creation for deep compression
  - Simpler space reclaim for automatic storage and regular DMS
- **Version 9.7**
  - Online REORG improvements
  - Online Table Move
  - Tablespace capacity increase
  - Compression support for replicated tables
  - Tablespace space reclaim below HWM
  - TEMP table compression
  - Index compression
  - Migration to Automatic Storage
  - Automatic MDC space reclaim
  - LOB inlining (with compression)
  - Support for LONG->LOB transition
  - Starjoin query improvements
  - DPF costing improvements
  - Scan Sharing
  - Enhanced Monitoring
  - Diagnostics improvements
Agenda

- DB2 Optimized for SAP
- Storage Architecture
  - DB2 Architektur
  - DB2 Tablespaces (SMS, DMS)
  - Automatic Storage
  - Tablespace with Reclaimable Storage
- Data Compression
  - Row Compression
  - Index Compression
  - Temp Table Compression
  - LOB Inlining
- Database Partitioning Feature (DPF)
- Range Partitioning
- Multi-Dimensional Clustering
- DBF + Range Partitioning + MDC
DB2 Architektur

- **Parallelism**
  - SQL
  - Utilities
  - Dynamic throttling based on load

- **Full Resource Exploitation**
  - Large/complex queries executed on multiple CPUs
  - Data automatically striped across multiple disks
  - Internal algorithms designed to scale effectively to TBs of main memory and beyond

- **Advanced I/O Subsystem**
  - Asynchronous, Parallel I/O
  - Automatic Intelligent Data Striping
  - Big block I/O
System Managed (SMS) Tablespace

Datenspeicherung bei SMS

– Datenspeicherung in OS Dateisystemen
– Daten werden in Extent-Größen (Anzahl aufeinanderfolgender Pages) „Round Robin“ auf die Dateisysteme verteilt
– Jedes DB2 Objekt hat seinen eigenen Namen, der in allen Containern gleich ist
– Beispiel:
  – db2 create tablespace TS1 managed by system using ( ‘/mydir1‘, ‘/mydir2‘ ) extentsize 4
  – db2 create table T1 (c1 int ... ) in TS1
  – db2 create table T2 (c1 int ... ) in TS1
Database Managed (DMS) Tablespace

- Datenspeicherung bei DMS Tablespaces
  - Datenspeicherung in Dateien (Container Files) oder OS Devices (Container Devices)
  - Mehrere DB2 Objekte können in einem Tablespace, der aus mehreren Containern bestehen kann, gespeichert werden
  - Daten werden geblockt in Extent-Größen gespeichert
  - Tablespace hat definierte logische Struktur, die auf die physischen Container abgebildet wird.
  - Beispiel:
    - `db2 create tablespace TS2 managed by
database
ing using ( file '/myfile'
1024,
    device ,/dev/rhd7'
2048 )
extentsize 4 prefetchsize 8`
    - `db2 create table T1 (c1 int ...) in TS1`
    - `db2 create table T2 (c1 int ...) in TS1`
Wichtige I/O relevante Parameter: EXTENTSIZE

- **Empfehlung:**
  - EXTENTSIZE = mehrfache Größe der RAID Stripe Größe (ohne Parity)

- **Beispiel**
  - RAID 4 + P, RAID strip size = 64 KB, page size = 16 KB
  - EXTENTSIZE = 4 * 64 KB = 256 KB = 16 pages
Wichtige I/O relevante Parameter: PREFETCHSIZE

- **Empfehlung:**
  - Von allen LUNs sollte bei einem Single Prefetch gelesen werden
  - Allgemeine Empfehlung: PREFETCHSIZE = **AUTOMATIC**

- **Beispiel (berechnete Größe)**
  - PREFETCHSIZE = RAID Stripe Size (ohne Parity) * # Containers
    \[ = 256 \text{ KB} \times 2 = 512 = 32 \text{ pages} \]
Wichtige I/O relevante Parameter: NUM_IOSERVERS

- Datenbank Konfigurations Parameter: Max. Anzahl der DB2 Agents, die das „Prefetching“ durchführen
- I/O Größe pro Prefetch = EXTENTSIZE
- Empfehlung:
  - NUM_IOSERVERS = Anzahl Container
  - Allgemeine Empfehlung: NUM_IOSERVERS = AUTOMATIC
- Beispiel mit NUM_IOSERVERS = 2

Parallel I/Os from separate I/O Servers

PREFETCHSIZE=32 pages

LUNs

/data1fs

/data2fs
Wichtige I/O relevante Parameter: DB2_PARALLEL_IO

- Registry Parameter: Bestimmt Parallelität des Zugriffs auf einen Tablespace-Container
- Default: DB2 nimmt an, dass pro Container eine physische Platte existiert.
Wichtige I/O relevante Parameter: DB2_PARALLEL_IO (2)

- With db2set DB2_PARALLELIO=*\,4
  - DB2 weiss, pro Container von 4 Platten parallel gelesen werden kann.
  - Bemerkung: NUM_IOSERVERS sollte in diesem Beispiel >= 8 sein
    Wird berücksichtigt, falls NUM_IOSERVERS = AUTOMIC gesetzt ist

![Diagramm mit LUNs, PREFETCHSIZE und Datenstrukturen]
SMS or DMS?

- **SMS**
  - Akzeptable Performance mit wenig Administration
    - Speicher Allokierung „On Demand“
    - Prefetching des File Systems wird ausgenutzt.

- **DMS**
  - Gute Performance, jedoch mit „etwas“ Administrationsauf
    - Performance „Fein-Tuning“ ist möglich: Verschiedene DB2 Objecte (Tabellen, Indizes, Large Objects (LOBs) können in getrennten Tablespaces gespeichert werden, denen eigene Bufferpool-Bereiche zugeordnet werden können.
    - Perfromance-Vorteile gegenüber SMS im Bereich 5% – 15%
DMS – Vereinfachung der Administration mit DB2 9.5

- **Auto-Resize**
  - Automatisches Wachsen des Tablespaces

- **Automatic Storage**
  - Vereinfachte Administration der Speicherzugriffspfade
DMS – Auto-Resize DMS Tablespaces (1)

- Tablespace wächst, wenn er voll ist und mehr Speicher benötigt wird.
- Beispiel:
  - `db2 create tablespace MYTS managed by database using
    (file '/data1fs/mytsC1' 10000,
    file '/data1fs/mytsC2' 10000 )
    AOTORESIZE YES INCREASESIZE 1G MAXSIZE 100G`

- Lediglich die Container wachsen, die im letzten Tablespace Range enthalten sind, um Rebalancing zu vermeiden.
- Automatisches Anwachsen stoppt, falls
  - MAXSIZE erreicht ist   ODER
  - Einer der der Container im letzten Tablespace-Range nicht weiter anwachsen kann (siehe nächste Seite)
DMS – Automatic Storage (1)

- Neue DB2 Speicherverwaltung, welche die Vorteile von SMS und DMS verbindet
  - Automatic Tablespace Wachstum
  - Performance-Vorteile von DMS

- Automatic Storage ist ein „Single Point“ der Speicherverwaltung
  - Tablesaces speichern ihre Daten in denselben Speicherpfaden
  ➔ Vorteil: Weniger Speicherpfade müssen verwaltet werden
  - Automatisches Tablespace-Wachstum ist mit Automatic Storage spezifiziert

- Beim Anlegen von Automatic Storage Tablesaces müssen keine Speicherpfade definiert werden.
  ➔ Vereinfacht das Anlegen von Tablesaces

Ohne Automatic Storage
Speicherzuordnung auf Tablespace-Level

Mit Automatic Storage
Speicherzuordnung auf Datenbank-Level
DMS – Automatic Storage (2)

Example

- **Datenbank anlegen**
  
  ```sql
  create database on /data1fs, /data2fs DBPATH on /logfs
  
  - Default Tablespaces (SYSCATSPACE, TEMPSPACE1 & USERSPACE1) werden mit zwei Containern angelegt, einer in /data1fs der andere in /datafs
  - Transaction Log und andere Meta-Daten der Datenbank werden unter /logfs angelegt.
  - Mit 9.7 können separate Storage Pfade für Catalog, User and Temp Tablespaces separat angegeben werden.
  ```

- **Tablespace anlegen**
  
  ```sql
  create tablespace MYTS1
  create tablespace MYTS2 INITIALSIZE 500M INCREASESIZE 50M MAXSIZE 100G
  
  - MYTS1, MYTS2 und alle anderen AS Tablespaces werden mit 2 Container (/data1fs, /data2fs) angelegt.
  ```
DMS – Automatic Storage Usage (3)

- Neue Storage Pfade hinzufügen mit ALTER DATABASE
- Daten-’Rebalance’ ist erforderlich, damit neue Storage Pfade verwendet werden
- Speicherpfade können nur für Automatic Storage Datenbanken hinzugefügt werden (< V9.7)
- Beispiel

  1) `db2 create database mydb on '/a/b', '/c/d', '/e/f', '/g/h'`
  2) `{ db2 create tablespace TS3
      db2 create tablespace TS4
    }`
  3) `{ db2 create table T1 (c1 int ...) in TS3
      db2 load insert into T1
      db2 create table T2 (c1 int ...) in TS4
    }`
  4) `db2 load insert into T2`
  5) `db2 alter database add storage on '/i/j', '/k/l', '/m/n', '/o/p'`

Table wird auf neuen Storage Pfaden angelegt, falls bei alten Storage Pfaden kein Platz mehr frei ist.
### Tablespace Bewertung

<table>
<thead>
<tr>
<th>Tablespace Option</th>
<th>Single Point of Storage Mgt</th>
<th>Auto Container Growth</th>
<th>Auto Stripe Set Addition</th>
<th>File System Buffering</th>
<th>Allows Different Tablespaces &amp; Bufferpools for a Table’s Data, Index, &amp; Long Data</th>
</tr>
</thead>
<tbody>
<tr>
<td>SMS</td>
<td>No</td>
<td>Yes</td>
<td>No</td>
<td>Selectable (DIO/CIO)</td>
<td>No</td>
</tr>
<tr>
<td>DMS</td>
<td>No</td>
<td>No</td>
<td>No</td>
<td>Selectable (DIO/CIO)</td>
<td>Yes</td>
</tr>
<tr>
<td>RAW</td>
<td>No</td>
<td>No</td>
<td>No</td>
<td>No</td>
<td>Yes</td>
</tr>
<tr>
<td>Automatic Storage</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
<td>Selectable (DIO/CIO)</td>
<td>Yes</td>
</tr>
</tbody>
</table>

**Empfehlung: Automatic Storage oder DMS Auto-Resize**
DB2 Automatic Storage (AS)

- Fully integrated into SAP installation and DBA Cockpit
- Greatly reduce the effort of database administration
- More flexibility with DB2 9.7

DB2 9.7 New Features

- Add new AS paths to non AS Databases
  
  ```sql
  ALTER DATABASE <dbname> ADD STORAGE ON <path1>, <path2> …
  ```

- Easy drop of existing AS paths, which are not used any more for future database growing
  
  ```sql
  ALTER DATABASE <dbname> DROP STORAGE ON <path1>, <path2> …
  ```

  - State of storage path changed: “In Use” ➔ “Drop Pending”
  - Future growth on dropped paths will not occur
  - Each affected tablespace must be rebalanced before dropped path can be removed.

- Start data rebalancing (online) to move existing data to the new AS paths
  
  ```sql
  ALTER TABLESPACE <tsname> REBALANCE
  ```
Migration DMS Tablespaces to Automatic Storage (AS)

Enable AS for the database, if it is a non AS database
   ALTER DATABASE <dbname> ADD STORAGE ON <path1>, <path2> …

Alternative 1: Use „Redirect Restore“ (offline operation)
   • RESTORE DB <dbname> REDIRECT
   • For all tablespaces, which should be migrated to AS:
     SET TABLESPACE CONTAINERS FOR <tsnameID> USING AUTOMATIC STORAGE
     • RESTORE <dbname> CONTINUE

Alternative 2: Assign existing DMS tablespace to AS (online operation)
   • ALTER TABLESPACE <tsname> MANAGED BY AUTOMATIC STORAGE
   • ALTER TABLESPACE <tsname> REBALANCE
   • Fully integrated into SAP’s DBA Cockpit
Normal DMS and AS Tablespaces
• RIDs are pointing directly to data pages
• Moving of data pages online within tablespace is impossible

Reclaimable DMS and AS Tablespaces
• Object relative RIDs are pointing to EMP pages. Only EMP pages contain the physical address of data pages.
• Moving of data pages online within a tablespace is now possible.
• Fully supported by SAP’s DBA cockpit
• Reclaimable tablespaces are default*

*existing tablespaces cannot be altered to be reclaimable
Automatic Storage Tablespaces
• ALTER TABLESPACE <tsname> REDUCE [ <size>| MAX | STOP ]

DMS Tablespaces
• ALTER TABLESPACE <tsname> LOWER HIGH WATER MARK [ STOP ]
• ALTER TABLESPACE <tsname> REDUCE …

CREATE DATABASE testdb
CREATE TABLESPACE ts MANAGED BY AUTOMATIC STORAGE
CREATE TABLE t1(key int) IN ts
CREATE TABLE t2 LIKE t1 IN ts
CREATE TABLE t3 LIKE t1 IN ts

// Populate tables with records to achieve extent usage

DROP TABLE t1 // extents in FREE state
DROP TABLE t3 // extents in FREE state

ALTER TABLESPACE ts REDUCE MAX

Legend
- Object 1
- Free
- Object 2
- Table Space
- Meta-data
Agenda

- DB2 Optimized for SAP
- Storage Architecture
  - DB2 Architektur
  - DB2 Tablespaces (SMS, DMS)
  - Automatic Storage
  - Tablespace with Reclaimable Storage
- Data Compression
  - Row Compression
  - Index Compression
  - Temp Table Compression
  - LOB Inlining
- Database Partitioning Feature (DPF)
- Range Partitioning
- Multi-Dimensional Clustering
- DPF + Range Partitioning + MDC
Row Compression (1)

- **Dictionary Compression**
  - Dictionary contains repeated information from the rows in the table
  - Common sequences of consecutive bytes in row replaced with 12 bit symbol
  - Compression candidates can be across page and column boundaries or within column
  - Can be enabled at CREATE table time
  - Existing tables can be ALTERed to enable compression. Table reorganization necessary
  - Automatic Dictionary Creation possible with DB2 Version 9.5

---

**Uncompressed Records**

<table>
<thead>
<tr>
<th></th>
<th>Canada</th>
<th>Ontario</th>
<th>Toronto</th>
<th>Smith</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td></td>
<td></td>
<td></td>
<td>Smith</td>
</tr>
<tr>
<td>2</td>
<td>Canada</td>
<td>Ontario</td>
<td>Toronto</td>
<td>Miller</td>
</tr>
</tbody>
</table>

**Compressed Records**

<table>
<thead>
<tr>
<th></th>
<th>x'01C'</th>
<th>Smith</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td></td>
<td>Smith</td>
</tr>
<tr>
<td>2</td>
<td>x'01C'</td>
<td>Miller</td>
</tr>
</tbody>
</table>

**Compression Dictionary**

```
Canada Ontario Toronto
.....    .....    x'01C'
```
Row Compression (2)

- **Significant Lower TCO with DB2 Row Compression**

- **Key Figures**
  - Better bufferpool utilization and less I/O
  - Saving of up to 90% storage for single tables; up to 50% for whole SAP application
  - Dialog response time could be improved by up to 20%
  - Backup and Restore time could be reduced
  - CPU user time went up by about 5%
Row Compression (3)

Monitoring and Administration

- DB2 row compression is integrated part of SAP DBA Cockpit
- Compression adviser exists to identify table candidates for row compression (/ISIS/ZCOMP - SAP note 980067)
- Checks for DB2 row compression are implemented in SAP BW

Installation and Migration

- Automatic Dictionary Creation possible for installation and migration
- Data sampling implemented in R3load for creating optimal compression dictionaries
Index Compression (1)

DB2 Index Compression

- Is a transparent feature of the DB2 engine*
- Reduces index size of a SAP system by 40% and more
- Reduces I/O data transfer and increase buffer pool hit ratio of a SAP system
- Works for unique indexes and secondary indexes as well
- Is fully supported for all SAP releases in standard and extended maintenance (4.6D – 7.20)
- Is default for tables with that are DB2 row compressed*
- Is covered by the SAP OEM license for DB2 at no extra costs

* an index reorganization is required to compress existing indexes
## Index Compression (2)

DB2 Index and Row Compression Savings with Real Customer Data

- SAP R/3 4.6D System

<table>
<thead>
<tr>
<th></th>
<th>Size Before Compression</th>
<th>Size After Row Compression</th>
<th>Size After Index Compression</th>
</tr>
</thead>
<tbody>
<tr>
<td>DATA/LOB/LONG</td>
<td>570 GB</td>
<td>162 GB / 38 %</td>
<td>162 GB / 38 %</td>
</tr>
<tr>
<td>INDEXES</td>
<td>135 GB</td>
<td>135 GB / 100 %</td>
<td>71 GB / 47 %</td>
</tr>
<tr>
<td>Σ Absolute</td>
<td>705 GB</td>
<td>297 GB / 42 %</td>
<td>233 GB / 33 %</td>
</tr>
<tr>
<td>Δ Relative</td>
<td>0 GB</td>
<td>-408 GB / -58 %</td>
<td>-64 GB / -22 %</td>
</tr>
</tbody>
</table>

**With DB2 Compression you can reduce the size of your SAP system by 50% and more**
Temp Table Compression

DB2 Temp Table Compression
• Is a transparent feature of the DB2 engine
• Reduce the amount of temporary disk space required
• Works like DB2 Row Compression for large temporary tables
• Improves query performance with large sort operations:
  • Reduces I/O data transfer for large sorts
  • More sorts can be processed in bufferpool
• Is fully supported for all SAP releases in standard and extended maintenance (4.6D – 7.20)
• Is default for SAP systems
• Is covered by the SAP OEM license for DB2 at no extra costs

Uncompressed Records

<table>
<thead>
<tr>
<th></th>
<th>Canada</th>
<th>Ontario</th>
<th>Toronto</th>
<th>Smith</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Canada</td>
<td>Ontario</td>
<td>Toronto</td>
<td>Miller</td>
</tr>
</tbody>
</table>

Compressed Records

<table>
<thead>
<tr>
<th></th>
<th>x'01C'</th>
<th>Smith</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>x'01C'</td>
<td>Smith</td>
</tr>
<tr>
<td>2</td>
<td>x'01C'</td>
<td>Miller</td>
</tr>
</tbody>
</table>

Compression Dictionary

Canada Ontario Toronto x'01C'

...
Large Object (LOB) Inlining

DB2 LOB Inlining

• Is a transparent feature of the DB2 engine*
• Reduces LOB space in a SAP system
• Allows buffering and compression of LOB fields
• Accelerates access of small LOB fields and improves query performance
• Is fully supported for all SAP releases in standard and extended maintenance (4.6D – 7.20)
• Is default for SAP systems*

* a table reorganization/move is required to inline existing LOB fields
IBM Information Management

Agenda

- **DB2 Optimized for SAP**
- **Storage Architecture**
  - DB2 Architektur
  - DB2 Tablespaces (SMS, DMS)
  - Automatic Storage
  - Tablespace with Reclaimable Storage
- **Data Compression**
  - Row Compression
  - Index Compression
  - Temp Table Compression
  - LOB Inlining
- **Database Partitioning Feature (DPF)**
- **Range Partitioning**
- **Multi-Dimensional Clustering**
- **DPF + Range Partitioning + MDC**
Database Partitioning Feature (DPF) (1)

Goal: Performance improvement with parallel processing
- SQL (Select, Delete, Update), Backup/Restore, Create Index, ...
- One database can reside on several separate computers
- Each partition has its own Buffer Pools, Sort Areas and Logging
- Hash function is used to distribute records horizontally on database partitions
- Shared nothing data distribution. Each partition accesses only its local data
- Several logical partitions can be on the same machine
- Physical or logical partitioning is transparent for the database
- Fast communication is needed (performance improvements with DB2 Version 9.1)
Database Partitioning Feature (DPF) (2)

- Collection of “Partition” put together to form a single database
- Applications see a single database
  - Data loaded/inserted transparently hash-partitioned across the nodes
  - SQL and utilities transparently parallelized across nodes
- Virtually unlimited & super linear scale-up possible/common
  - Each node autonomously performs it’s part of a request; coordinator node collects all node’s responses; and routes back to application
Database Partitioning Feature (DPF) (3)

Data distributed with hash function.

Partitioning Key of a table is used to determine a partition on which the data row is stored. Partitioning key is a key that is part of table definition.

How does Hash Partitioning work?

```
<table>
<thead>
<tr>
<th>Social Insurance Number</th>
<th>Name</th>
<th>Location</th>
</tr>
</thead>
<tbody>
<tr>
<td>123-456-789</td>
<td>JoeBoston</td>
<td>Toronto</td>
</tr>
</tbody>
</table>
```

Partition Key hashed to value: "8"

```
<table>
<thead>
<tr>
<th>Hash value</th>
<th>0</th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
<th>6</th>
<th>7</th>
<th>8</th>
<th>9</th>
<th>10</th>
<th>11</th>
<th>12</th>
<th>...</th>
</tr>
</thead>
<tbody>
<tr>
<td>Partition</td>
<td>1</td>
<td>2</td>
<td>3</td>
<td>1</td>
<td>2</td>
<td>3</td>
<td>1</td>
<td>2</td>
<td>3</td>
<td>3</td>
<td>3</td>
<td>3</td>
<td>1</td>
<td>...</td>
</tr>
</tbody>
</table>
```

Hash value "8" assigned to Partition 3

```
Partition 1
Partition 2
Partition 3
```
Database Partitioning Feature (DPF) (4)

Customer Scenario

- SAP BI query performance test
- Compare DB2 with Non-DB2 database. Same hardware was used
- Eight logical DB2 database partitions were configured on a 8-way database server

Single-user Test

- Set of 27 most important SAP BI queries run in a loop
- At any time one SAP BI query was running on the system

⇒ Result: On average DB2 LUW is 44% faster than the non-DB2 database.

Multi-user Test

- Several user processes run the set of 27 most important SAP BI queries in a loop
- At any time several SAP BI queries were running in parallel on the system
- CPU utilization on the DB server: 90%-100%

⇒ Result: On average DB2 LUW is 23% faster than the non-DB2 database.
Database Partitioning Feature (DPF) (5)

- **Enhancements with DB2 Version 9.1 and 9.5**
  - Fast Communication Manager (FCM) performance improved (DB2 Version 9.1)
  - Faster data redistribution using page level processing with minimal logging (DB2 Version 9.5)
  - Single-system view backup (DB2 Version 9.5)
    - DB2 provides now the capability to create a backup of all partitions with one command
    - SAP Planning Calendar provides backups for DPF systems
  - Single-system database configuration update (DB2 Version 9.5)
    - Updates of database configuration parameters can be distributed to all partitions
DB2 Layoutouot Recommendation for SAP BW

Appl. Server 1  ...  Appl. Server N

DB Server 1
DB Partition 0  DB Part. 1  DB Part. 2

DB Server 2
DB Part. 3  DB Part. 4

Bufferpool IBMDEFAULTBP

Data Transfer during Query Processing

DIMD, DIMI
SAP Basis Tablespace

FACTD, FACTI
Aggregate TS

ODSD, ODSI
Aggregate TS

Fast Communication
Customer PoC: Performance Boost with DPF

Example Production System with 10 logical database partitions
- InfoCube-, Aggregate-, PSA- and DataStore tables are distributed, the number of database partitions depends on the table size.
- Other tables on database partition 0

⇒ Superior Performance and Scalability

![Diagram](image-url)
Customer PoC: Query Results

- Run time of 23 of the most important BI queries are measured
- Single User test; one query runs after the other
- Same hardware was used running queries on 1 and on 10 database partitions

Result: - BI query performance improvements with DB2 DPF
- BI queries run factor 5 faster in average
Agenda

- DB2 Optimized for SAP
- Storage Architecture
  - DB2 Architektur
  - DB2 Tablespaces (SMS, DMS)
  - Automatic Storage
  - Tablespace with Reclaimable Storage
- Data Compression
  - Row Compression
  - Index Compression
  - Temp Table Compression
  - LOB Inlining
- Database Partitioning Feature (DPF)
- Range Partitioning
- Multi-Dimensional Clustering
- DPF + Range Partitioning + MDC
Range Partitioning - Create

• Daten einer Tabelle werden nach Schlüsselbereichen in Partitionen aufgeteilt, die in verschiedenen DB2 „Tablespaces“ gespeichert werden können

• Short and Long Form to specify a range partitioned table

\[
\begin{align*}
\text{Short Form} \\
\text{CREATE TABLE } t1(c1 \text{ INT}) \text{ IN tbsp1, tbsp2, tbsp3} \\
\text{PARTITION BY RANGE(c1)} \\
\text{(STARTING FROM (1) ENDING100) EVERY (33))}
\end{align*}
\]

- or -

\[
\begin{align*}
\text{Long Form} \\
\text{CREATE TABLE } t1(c1 \text{ INT}) \\
\text{PARTITION BY RANGE(a)} \\
\text{(STARTING FROM (1) ENDING(34) IN tbsp1, ENDING(67) IN tbsp2, ENDING(100) IN tbsp3)}
\end{align*}
\]
Range Partitioning - Indizes

- V9.1 + V9.5 Global Indexes
  - Each Index is in a separate storage
  - Can be created in different tablespaces

- V9.7 Local Indexes

```sql
CREATE TABLE t1(c1 INT, c2 INT, ...)
IN tbsp1, tbsp2, tbsp3
INDEX IN tbsp4
PARTITION BY RANGE(a)
(STARTING FROM (1) ENDING100) EVERY (33))

CREATE INDEX i1(c1)
CREATE INDEX i2 (c2) IN tbsp5
```
Range Partitioning – Rolling-Out Data

ALTER TABLE FactTable
DETACH PARTITION Jan
INTO TABLE OldSales

- Requires exclusive lock
- Relatively fast operation; no data movement required
- Index maintenance done later (asynchronously in background)

COMMIT (right after detach)
- Detached data now invisible
- Detached partition ignored in index scans
- Rest of FactTable available

SET INTEGRITY FOR Mqt1, Mqt2
- (Optional) maintains MQTs on FactTable

EXPORT OldSales; DROP OldSales
- (Optional) this becomes an independent table

Asynchronous cleanup of global indexes after Detach
Range Partitioning – Rolling-In Data

LOAD / Insert into NewSales
(Perform ETL on NewSales)

ALTER TABLE FactTable …
  ATTACH PARTITION …
  STARTING ’05/01/2006’
  ENDING ’05/31/2006’
  FROM TABLE NewSales
  • Requires exclusive lock
  • Very fast operation
  • No data movement required
  • Index maintenance done later

COMMIT
  • New data still not visible

SET INTEGRITY FOR FactTable …
  • Potentially long running operation
    • Validates data
    • Maintains global indexes, MQTs
    • Keep logging requirements in mind
  • Existing data available while it runs

COMMIT
  • New data visible
Range Partitioning Summary

- Fast Data Roll-In and Roll-Out
- Reduced Table Reorg
- Query performance improvement through „Partition Elimination“

Missing Features:
- Online Reorg
- Partition Lock
Range Partitioning wird im SAP Dictionary unterstützt

Einsatz für SAP OLTP denkbar
Agenda

- DB2 Optimized for SAP
- Storage Architecture
  - DB2 Architektur
  - DB2 Tablespaces (SMS, DMS)
  - Automatic Storage
  - Tablespace with Reclaimable Storage
- Data Compression
  - Row Compression
  - Index Compression
  - Temp Table Compression
  - LOB Inlining
- Database Partitioning Feature (DPF)
- Range Partitioning
- Multi-Dimensional Clustering
- DBF + Range Partitioning + MDC
Problem with RID Indexes – Not optimized for multiple Key Access

- Database systems try to store the records of a table in a particular order
  - Called „data clustering“
  - Improves query performance for a single key column
  - Query performance for other keys could degrade

- Example

SELECT * FROM Sales WHERE Region = WEST
  - Usually do not require a page I/O when reading the next record (because it’s usually on the same page as the previous record)
  - The page I/Os that are required, are sequential (efficient)

SELECT * FROM Sales WHERE Year = 2005
  - Usually do require a page I/O when reading the next record (because it’s usually on a different page than the previous record)
  - Each of these page I/Os is random (inefficient)
**Solution: Multi-Dimensional Clustering (MDC)**

- MDC organizes table data along one or more columns defined as MDC dimensions
  - Records with the same values in one or more columns are stored physically together in blocks of pages
  - Blocks correspond to tablespace extents
  - MDC block indexes point to blocks instead of single rows
  - Improves query performance for more than one key column (MDC dimensions)

- Example

```sql
SELECT * FROM Sales WHERE Region = WEST
```
- 2 big block I/Os to retrieve pages containing region WEST
- All sequential I/O *(efficient)*

```sql
SELECT * FROM Sales WHERE Year = 2005
```
- 2 big block I/Os to retrieve pages containing year 2005
- All sequential I/O *(efficient)*
CREATE TABLE MDCTAB
(YEAR CHAR(4),
REGION VARCHAR(5), ...)
ORGANIZE BY DIMENSIONS(YEAR, REGION)

Fact table

<table>
<thead>
<tr>
<th>Year</th>
<th>Region</th>
<th>Customer</th>
<th>Revenue</th>
<th>Overhead</th>
</tr>
</thead>
<tbody>
<tr>
<td>2004</td>
<td>West</td>
<td>001532</td>
<td>500,000 $</td>
<td>280,000 $</td>
</tr>
<tr>
<td>2005</td>
<td>West</td>
<td>002047</td>
<td>710,000 $</td>
<td>60,000 $</td>
</tr>
<tr>
<td>2004</td>
<td>East</td>
<td>013901</td>
<td>250,000 $</td>
<td>100,000 $</td>
</tr>
<tr>
<td>2005</td>
<td>North</td>
<td>009954</td>
<td>330,000 $</td>
<td>10,000 $</td>
</tr>
</tbody>
</table>

System generated indexes

<table>
<thead>
<tr>
<th>Block Index: Year</th>
<th>2004</th>
<th>2005</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>1</td>
<td>2</td>
</tr>
<tr>
<td></td>
<td>3</td>
<td>4</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Block Index: Region</th>
<th>West</th>
<th>East</th>
<th>North</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>1</td>
<td>2</td>
<td>4</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>1</td>
<td>3</td>
<td>2</td>
<td>4</td>
</tr>
</tbody>
</table>
MDC Advantages

- **MDC fast insert: Reduced Locking**
  - Enabled by setting LOCKSIZE parameter of an MDC table to BLOCKINSERT (during table creation or by ALTER TABLE)
  - Locks MDC blocks instead of single rows

- **MDC fast delete**
  - Mark pages as deleted instead of single rows
  - Synchronous cleanup of additional secondary indexes in the standard way in DB2 9
  - **New**: Asynchronous index cleanup in DB2 V9.5 (like for range-partitioning) -> large delete operations return much faster

- **Less index maintenance on MDC Block Indexes**

- **V9.7: MDC Space Reclaim**
  - `Reorg TABLE <mdc-table-name> RECLAIM EXTENTS ONLY`
  - Easy space reclaim after data roll-out in MDC tables
  - Can be part of automated maintenance policy
  - Fully integrated into SAP's DBA cockpit
MDC Query Performance

Queries with strong restrictions on Dim1
MDC rollout performance data

11 million rows (134260 pages), 16K page, 16 extent size, 4 nodes, 8 RID indexes
MDC Advantages and Considerations

- **Advantages**
  - Clustering is always guaranteed, no table reorganization necessary to preserve clustering order
  - Significant performance improvement for queries with restrictions on the MDC dimensions
  - Reduced maintenance overhead: block indexes are smaller and easier to maintain than row indexes
  - Support for fast data insertion and deletion (roll-in and roll-out)

- **Considerations**
  - Finding the right MDC dimensions if a variety of queries with different restrictions exist
  - Risk of high disk space consumption if wrong MDC dimensions are chosen
Agenda

- DB2 Optimized for SAP
- Storage Architecture
  - DB2 Architektur
  - DB2 Tablespaces (SMS, DMS)
  - Automatic Storage
  - Tablespace with Reclaimable Storage
- Data Compression
  - Row Compression
  - Index Compression
  - Temp Table Compression
  - LOB Inlining
- Database Partitioning Feature (DPF)
- Range Partitioning
- Multi-Dimensional Clustering
- DPF + Range Partitioning + MDC
CREATE TABLE myTable (INT part_no, DATE sale_date, CHAR region, ...)
DISTRIBUTE BY HASH (part_no)
PARTITION BY RANGE (sale_date) (STARTING FROM ... ENDING ... EVERY 1 MONTHS)
ORGANIZE BY DIMENSIONS (region)

• SAP OLTP: Range Partitioning denkbar
• SAP OLAP: DPF + MDC im Einsatz
Vielen Dank für Ihre Aufmerksamkeit