DB2 for z/OS Data Management Overview

Dr. Jim Teng
IBM Silicon Valley Laboratory
May 9, 2006 - 8:30 am to 9:40 am

Platform: z/OS
Key DB2 Components

- RDS
- Utility
- Data/Index Manager
- Buffer Manager
- Data Space Mgr
- DB2 Data
- Catalog Directory
**DB2 Database Physical Organization**

- DB2 data is stored in one or more databases
- Each database consists one or more tablespaces and indexspaces
- Each tablespace contains one or more tables
- Each indexspace contains one B+ tree index structure
- Each TS/IS is a collection of one or more VSAM data sets
  - LDS (Linear Data Set) or ESDS
  - EF (Extended Format) or EA (Extended Addressability)
    - SMS managed
  - Only EA can exceed 4 GB
- Each TS/IS is formatted into fixed-size units, call "pages"
  - TS Page Size: 4K, 8K, 16K, or 32K
  - IS Page Size: 4K
**Types of Data**

- **System Tables and Indexes**
  - In DB2 catalog and directory databases
  - Tables in directory database not accessible via SQL
  - Datasets are user-managed

- **User Tables and Indexes**
  - Datasets can either be DB2-managed (i.e. stogroup-managed) or user-managed

- **Database descriptor (i.e. DBD)**
  - Internal representation of DDL for objects within a database
  - Stored in the DBD01 tablespace of the directory database
  - Cached in the EDM pool for access or update
**Directory Database - DBID = 1**

- **DBD01 Tablespace**
  - Used to store system and user DBDs, except for the directory database
  - DBD for the directory database is loaded into memory during DB2 startup
- **SCT02 Tablespace**
  - Skeleton Plan Tables - runtime structures for plans
- **SPT01 Tablespace**
  - Skeleton Package Tables - runtime structures for packages
- **SYSUTILX Tablespace**
  - Utility status tables
- **SYSLGRNGX Tablespace**
  - Log ranges for data recovery
Other System Databases

- DSNDB02 - dbid = 2
  - Contains a bit map on assignment/free DBIDs for user created databases
    - No underlying tablespace
    - The bit map is stored in its own DBD
  - TEMP and WORKFILE (except DSNDB07) databases are considered as user created databases (DBIDs > 256)
  - The maximum limited is 64K databases
  - The entire database 2 is locked until CREATE/DROP DATABASE commit
  - No Parallel CREATE/DROP DATABASEs
**Other System Databases ...**

- DSNDB04 - default database, dbid = 4
  - Used for create table/tables pace without specifying a database name
- DSNDB06 - dbid = 6
  - Contains DB2 Catalog Tables/Indexes
- DSNDB07 - DBID = 7
  - Used for WORKFILE database
  - Used by RDS sort/..., Create Global Temp Tables, ...
  - In a data sharing system, only one member can use DSNDB07
Catalog Table Samples

- SYSIBM.SYSDATABASE
  - All databases in the system
- SYSIBM.SYSTABLESPACE
  - All tablespaces
- SYSIBM.SYSTABLES
  - All tables
- SYSIBM.SYSCOLUMNS
  - Column names and attributes
- SYSIBM.SYSINDEX
  - All indexes
- SYSIBM.SYSKEYS
  - Columns comprising an index key
The **tablespace**

<table>
<thead>
<tr>
<th>header page</th>
<th>spacemap page</th>
<th>data page</th>
<th>data page</th>
<th>spacemap page</th>
<th>data page</th>
</tr>
</thead>
</table>

- Each page size is 4K, 8K, 16K or 32K.
- The header page contains some general information on the tablespace.
- The spacemap page
  - The spacemap page contains space information on its range of data pages. There are "fullness bits" for every data page (2 for non-segmented, 4 for segmented) where 00= empty...11=full.
  - The spacemap page also contains 1 bit per data page in its range of data pages which isYES if that page has been modified since the last image copy.
- The data pages contain the user's data.
**Simple Tablespace**

- May contain more than one tables
- Data from multiple tables mixed on the same data page
- Not recommended for multiple tables due to the following concerns:
  - Locking considerations
  - Loss of table clustering
  - Additional page I/Os during scanning
  - No reclamation of free space after dropping a table until REORG
  - Not efficient on a mass delete operation
Simple Tablespace ...

- Space allocation performance might be degraded for variable length records
- Fewer settings are used to represent free space information for each data page
- I/O overhead is greater due to less accurate free space information
- Suitable for multiple tables if they need to be strict clustered (i.e. parent and its descent records are stored in one or more adjacent data pages)
  - Data can only be arranged in strict clustering order by user via the LOAD utility
Partitioned Tablespace

- Can only contain a single table
- Up to 4096 partitions
- Each partition can be placed on a different STOGROUP
- Best usage is with large tables
- Data in each partition dependent on a "key range"

<table>
<thead>
<tr>
<th>PART</th>
<th>SM</th>
<th>D</th>
<th>D</th>
<th>SM</th>
<th>D</th>
<th>SM</th>
<th>D</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>H</td>
<td>SM</td>
<td>D</td>
<td>D</td>
<td>SM</td>
<td>D</td>
<td></td>
</tr>
<tr>
<td>2</td>
<td>H</td>
<td>SM</td>
<td>D</td>
<td>D</td>
<td>SM</td>
<td>D</td>
<td></td>
</tr>
<tr>
<td>3</td>
<td>H</td>
<td>SM</td>
<td>D</td>
<td>D</td>
<td>SM</td>
<td>D</td>
<td></td>
</tr>
<tr>
<td>4</td>
<td>H</td>
<td>SM</td>
<td>D</td>
<td>D</td>
<td>SM</td>
<td>D</td>
<td></td>
</tr>
</tbody>
</table>
**Segmented Tablespace**

- Makes best use of space.
- Data pages are organized contiguously into segments, each of which holds records (a.k.a. rows) from only one table.

```
+---+   +---+   +---+   +---+   +---+   +---+   +---+
<table>
<thead>
<tr>
<th>H</th>
<th>SM</th>
<th>D</th>
<th>D</th>
<th>D</th>
<th>D</th>
<th>D</th>
<th>D</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>T1</td>
<td>T1</td>
<td>T2</td>
<td>T2</td>
<td>T2</td>
<td>T2</td>
</tr>
</tbody>
</table>
```

*In this example, segsize is 4.*

*The first 4 data pages are used for table T1, the next 4 for table T2.*

Segsize can be any multiple of 4 in 4 to 64 inclusive.

SEGSIZE is a parameter of CREATE TABLESPACE.

A spacemap page for a segmented tablespace also contains information on which table each segment is allocated for.

— Both simple and segmented tablespaces are referred to as linear.
Segment Tablespace ...

- Table scan performance improved
  - Only pages related to selected table are accessed
  - I/O eliminated for empty pages
- Mass delete performance improved
  - Only space map pages are updated and logged
  - Data pages are not accessed
- Space can be reused immediately after DROP TABLE or Mass Delete is committed
Segmente Tablespace ...

- Copy will not copy empty data pages caused by mass deletes or drop tables
- LOCK TABLE will only lock table, not the entire TS
  - Lock escalation is done at the table level
- Improve Insert performance for variable length records
  - More free space settings for each data page
- True clustering for tables
**Member Cluster - attribute for Simple or Partitioned**

- Benefit: Improve insert performance

```sql
INSERT INTO TABLE1 (COL1) VALUES ('ABC')
```

```sql
INSERT INTO TABLE1 (COL1) VALUES ('ABD')
```

With Member Cluster
Header Page

- One Header Page per tables machine or partition
- Contains DBID, PSID, RBRBA, DLD level-ids, 1st dictionary page, page size, ...
- Accessed during open for DLD
- RBRBA is used for LOGONLY recovery
- Update RBRBA and DLD level-ids based on ZPARM DLDFREQ value
  - Default every 5th Checkpoint
  - Update is only done for updated tables machine/partition
Dictionary Pages

- Immediately following the first space map page if exists
- Used to store compression/de-compression dictionary
- Loaded into memory when a tablespace/partition is opened
  - <= 64K Bytes
  - in DBM1 address space
  - V8 - above the 2GB area
Index Structure

Root Page (Non-leaf)

Non-leaf

Leaf Leaf

Non-leaf

Leaf Leaf Leaf

Non-leaf

Leaf Leaf Leaf Leaf
**Index Leaf Page**

- **Index Page 3 Header**

<table>
<thead>
<tr>
<th>Name</th>
<th>RID</th>
</tr>
</thead>
<tbody>
<tr>
<td>Tina</td>
<td>206</td>
</tr>
<tr>
<td>Dave</td>
<td>201</td>
</tr>
<tr>
<td>Rob</td>
<td>203</td>
</tr>
</tbody>
</table>

- **keymap** offsets to key entries maintained in descending collated sequence
**Index NonLeaf Page**

**Index Page 2 Header**

Last Page Ptr = 000006

<table>
<thead>
<tr>
<th>Page #</th>
<th>Key</th>
<th>RID</th>
</tr>
</thead>
<tbody>
<tr>
<td>000005</td>
<td>Tina</td>
<td></td>
</tr>
<tr>
<td>000003</td>
<td>Dave</td>
<td>00000237</td>
</tr>
<tr>
<td>000004</td>
<td>Rob</td>
<td></td>
</tr>
</tbody>
</table>

Off(Tina) Off(Rob) Off(Dave)
Partitioned Table with Indexes

V8

DPSI 1

DPSI 2

............................

DPSI N

V7

NPI

PI 1

PI 2

............................

PI N

Partition 1

Partition 2

............................

Partition N
**Workfiles**

- Workfiles are also known as temporary files
  - non recoverable
  - private to an agent
  - resides in Workfile Tablespace defined in WF Data Base (DSNDB07)

- Uses
- Sort
  - Group by
  - Order by
  - Joins
  - Column functions (DISTINCT...)
  - Utility Create Index

- View Materialization
WF Structure

- A workfile is a temporary table that spans multiple physical WF tablespaces.
  - WF1 spans TBSP A and B
  - WF2 spans TBSP B only

Space management scheme is to allocate blocks of pages within each WF tablespace to the respective WF tables that resides in that tablespace.
Possible Future Enhancements

- Autonomic Index Split
- Table with Append Insert option
- Auto-generated Tablespace/Database
- New Table Space Structures
- Converge TEMP Space
- Large Index Page Size
- Automatic identify unused indexes and plans/packages
- SMS/DB2 Integration
Session Title: DB2 for z/OS Data Management Overview

Session: C04

Dr. Jim Teng
IBM Silicon Valley laboratory
jteng@us.ibm.com