Database Layout and Configuration for SAP NetWeaver based Systems

Helmut Tessarek
DB2 Performance, IBM Toronto Lab
Trademarks

- The following terms are registered trademarks of International Business Machines Corporation in the United States and/or other countries: IBM, AIX, DB2, DB2 Universal Database, Informix, IDS
- SAP, mySAP, SAP NetWeaver, SAP Business Information Warehouse, SAP BW, SAP BI and other SAP products and services mentioned herein are trademarks or registered trademarks of SAP AG in Germany and in several other countries.
- Microsoft, Windows and Windows NT 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
- Oracle is a registered trademark of Oracle Corporation in the United States and other countries
- LINUX is a registered trademark of Linus Torvalds.
- Other company, product or service names may be trademarks or service marks of others.
Agenda

Part 1: DB2 Database Layout for SAP NetWeaver based Systems

Part 2: DB2 Database Layout for SAP Business Warehouse
Part 1

DB2 Database Layout for SAP NetWeaver

Overview:
- SAP NetWeaver usage types
- SAP installation options for DB2
- File system- and table space container layout
- Table space attributes
- Recommendations to improve I/O performance
A SAP NetWeaver system is configured for a specific purpose, as indicated by the usage type for example:

- Application Server ABAP (AS ABAP)
- Application Server Java (AS Java)
- Business Intelligence (BI)
- Enterprise Portal (EP)

The usage type affects the DB2 database layout:

- **Depending on the usage type the ABAP and/or Java runtime environment is installed. Each environment requires different DB2 table spaces.**

- **SAP Business Warehouse may be installed on multiple DB2 database partitions.** This may require to adapt the initial database layout manually during the installation.
SAP DB2 Tablespaces (NW04s ABAP+Java)

Uniform pagesize of 16K with Basis 6.40
⇒ Only one bufferpool required!
⇒ Increased table space size limits

Tablespaces (D/I) Usage
- SYSCATSPACE: DB2 data dictionary
- SAPSID#TEMP: Sort, temp tables, reorg
- SAPSID#USER1: Default TS
- SAPSID#EL<Rel.>: Dev. Environment Loads
- SAPSID#ES<Rel.>: Dev. Environment Sources
- SAPSID#LOAD: Screen and Report Loads
- SAPSID#SOURCE: Screen and Report Sources
- SAPSID#DDIC: ABAP Dictionray
- SAPSID#PROT: Log-like tables (e.g. Spool)
- SAPSID#CLU: Cluster tables
- SAPSID#POOL: Pool tables (e.g. ATAB)
- SAPSID#STAB: Master data
- SAPSID#BTAB: Transaction data
- SAPSID#ODS: ODS, PSA tables
- SAPSID#DIM: Dimension tables
- SAPSID#FACT: InfoCube and Aggregate fact tables
- SAPSID#DB: Tablespace for Java Stack
SAPinst Storage Management Options for DB2

1. **DMS File table spaces in autoresize mode** (SAPinst default with DB2 V8)
   - Tablespace are automatically enlarged and can be shrunk manually.
   - SAPinst uses the NO FILESYSTEM CACHING option by default.
   - You may want to use DMS File containers with filesystem cache enabled to buffer Lobs (Lobs are not buffered in the bufferpool).

2. **Automatic Storage Management** (SAPinst default with DB2 9)
   - Table spaces are automatically managed by DB2.
   - With version 9.5 data table spaces can also be shrunk manually.
   - Also supported for multi-partition databases since version 9.1.

3. **Other table space types** (e.g. DMS on raw devices)
   - Must be defined *manually* in the DDL-file which is generated by SAPinst.
   - Almost no performance difference between raw devices and DMS on file system with FS caching disabled.
SAPinst Storage Management Options for DB2

AutoStorage Option

Should NOT be marked, if you want to use SMS or DMS Raw Devices.
⇒ SAPinst generates a DDL-File that you can edit and execute manually to create tablespaces.
SAPinst generated DDL-Statements

Default DDL-Statements:

```sql
create tablespace SID#STABD in nodegroup SAPNODEGRP_SID pagesize 16k managed by database
  using ( FILE '/db2/SID/sapdata1/NODE0000/SID#STABD.container000' 503 M )
  using ( FILE '/db2/SID/sapdata2/NODE0000/SID#STABD.container000' 503 M )
  using ( FILE '/db2/SID/sapdata3/NODE0000/SID#STABD.container000' 503 M )
  using ( FILE '/db2/SID/sapdata4/NODE0000/SID#STABD.container000' 503 M ) on node ( 0 )
  extentsize 2  prefetchsize automatic  dropped table recovery off
  autoresize yes  maxsize none  no filesystem caching;
```

DDL-Statements with AutoStorage Option:

```sql
create database SID automatic storage yes
  on /db2/SID/sapdata1, /db2/SID/sapdata2, /db2/SID/sapdata3, /db2/SID/sapdata4
dbpath on /db2/SID
  ...
  pagesize 16 k  dft_extent_sz 2
catalog tablespace managed by automatic storage
  ...
create tablespace SID#STABD in nodegroup SAPNODEGRP_SID
  extentsize 2  prefetchsize automatic  dropped table recovery off
  no filesystem caching;
```
Network Based Storage Concepts

- In a Storage Area Network (SAN) storage appears to hosts as local **SCSI ‘disks’ (LUNs)**. A LUN can be mapped to anything within a SAN storage server:
  - A single spindle
  - A portion of a spindle
  - One or more RAID arrays
  - A “meta” of multiple RAID arrays

On the host machine, file systems can be created on LUNs.

- Network Attached Storage (NAS) appears to hosts as **file systems (FS)**. Hosts access NAS storage via file based protocols (for example NFS).
Mapping of Tablespace Containers to Disks

Example of a SAN-Storage and file system configuration:
- Each LUN is mapped to a dedicated RAID Array
- Each file system is created on a dedicated LUN

Performance recommendations:
- Spread containers of each tablespace over all available spindels.
- Use 15 – 20 dedicated spindels per CPU core.
- Avoid too many levels of striping:
  - DB2 is striping across containers
  - Storage controllers provide RAID striping
  - OS level striping, e.g. LVM striping should NOT be used.
- Put each container of a table space on a separate file system to maximize I/O parallelism (DB2 striping)
- Enable DB2_PARALLEL_IO for table spaces with one container per RAID device or stripe set
- For partitioned databases: Use dedicated LUNs/filesystems for each partition for easy problem determination.
PREFETCH SIZE and DB2_PARALLEL_IO

SAP recommends **automatic** prefetch size.
How does DB2 calculate the prefetch size in this case?

Prefetch size = (extent size) * (number of containers) * (number of physical disks per container)

Example:
- Extentsize = 2
- Number of containers = 4
- Number of disks per container = 1
  ⇒ **Prefetch size = 8 pages**
  ⇒ All disks are busy during for example a table scan.

How does DB2 determine the **number of physical disks per container**?
- If DB2_PARALLEL_IO is not specified then **number of physical disks per container** defaults to 1.
- If DB2_PARALLEL_IO=* then **number of physical disks per container** defaults to 6. This is the number of disks that is frequently used in RAID5 devices.
- You may also explicitly specify the number of disks.
  For example DB2_PARALLEL_IO=*:3,1:4

All table spaces use 3 as the number of disks per container
Table space 1 uses 4 as the number of disks per container.
DB2 File Systems for SAP NetWeaver

Database Server

Database Partition 0

Base table spaces (ABAP+Java)

BI Tablespace

Temporary table spaces

Online log files

Offline log files

Archived log files

DB2 diagnostic files

DB2 instance home dir

Local DB directory

Performance recommendations:

- Use separate disks for data and logging (e.g. RAID 5 for data and RAID1 for logging)
- Do not configure operating system I/O (e.g. swap space) on disks that are used for DB2 data or logging.
- Use SMS for temporary table spaces. SMS table spaces can shrink automatically.
Page Size

- With Basis 6.40 all table spaces are installed with uniform page size 16k
  ⇒ Reduces required number of bufferpools. More efficient usage of memory.
- Page size determines table space size limit (DB2 V8):
  - 256Gb for 16k pages
  - Size limit is per partition
  - Put large fast growing tables into their own table spaces!
- DB2 9 provides large tablespace support (used by default with DB2 9):
  - DB2 9 uses 6 byte RID: 4 byte page number, 2 byte slot number.
  - SAP Business Warehouse indexes with large RIDs consume 10-15% more space on disk and in bufferpool compared to regular table spaces.
Extentsize

- With SAP Basis 7.00 uniform extentsize of 2 is used (with Basis < 7.00 different extentsizes were used for different table spaces)

- Small extentsize reduces unused disk space for empty or very small tables.

- Small extentsize reduces unused disk space for Multi Dimensional Clustering tables (space usage of MDC tables depends on selected MDC dimensions and actual data in the table. Make sure to select appropriate MDC dimensions to keep amount of partially filled extents as low as possible).

For each combination of MDC dimensions values at least one extent is allocated!
The following recommendations apply for database shared memory (bufferpools, locklist, package cache etc.) and sort memory:

Central ABAP-System (DB and App-Server on same machine)
- ~30% of real memory for database shared memory and sort memory.

Distributed ABAP-System (Database on dedicated machine)
- ~60% of real memory for database shared memory and sort memory.

- Follow SAP notes 584952 (DB2 V8), 899322 (DB2 9.1), and 1086130 (DB2 9.5) to initially configure database memory.

- Since DB2 9 you can use the Self Tuning Memory Manager. If STMM is activated, set DATABASE_MEMORY=<fixed value> to define an upper limit for the database memory (for 9.5 set INSTANCE_MEMORY=<fixed value> and DATABASE_MEMORY=AUTOMATIC)

- The STMM should not be used with multi-partition SAP BI/DB2 systems, if the partitions have different memory requirements.
Part 2
DB2 Database Layout for SAP Business Warehouse

Overview:
- SAP BW concepts
- Layout on single-partition systems
- Data Partitioning Feature
- How to determine the correct number of partitions
- Layout on multi-partition systems
Data Structures in SAP Business Warehouse

SAP BW specific Tablespaces

- SID#ODSD
- SID#ODSI
- SID#FACTD
- SID#FACTI
- SID#DIMD
- SID#DIMI

SAP Basis Tablespaces

- SID#STABD
- SID#STABI
SAP BW Queries

Typical SAP BW Query:

How much **Revenue** was generated with **Product X** in **Timeframe Y** for each **Customer**?

```sql
SELECT C.name,
      SUM( Fact.Revenue )
FROM   Fact F, Customer C, Product P, Time T
WHERE  F.key1 = C.key  AND F.key2 = P.key  AND F.key3 = T.key  AND P.key = X AND T.key = Y
      GROUP BY C.name
```

- Fact data (e.g. revenue) is aggregated
- Fact table is joined with each dimension table
- Restrictions are defined on dimension tables
SAP BW Database Layout on single-partition Systems

Assign SAP BW table spaces with small tables to IBMDEFAULTBTP.

Create separate table spaces for large InfoCubes, PSA- and ODS-Objects to simplify storage management.

Assign SAP BW table spaces with large tables (> 1 Mio records) to buffer pool BP_BW_16K.

SAP Basis Tablespace (SID#STABD, ...)

Dimension data
SID#DIMD, SID#DIMI

Small InfoCubes and Aggregates
SID#FACTD, SID#FACTI

PSA and ODS data
SID#ODSD, SID#ODSI

Tablespace for large InfoCubes, Aggregates
PSA- and ODS-Objects

IBMDEFAULTBTP
16 KB page size

BP_BW_16K
16 KB page size

Database Partition 0

Must be created manually after SAPinst has finished!
DB2 multi-partition databases

- Support for multiple database servers.
- Parallel query processing with linear scalability.
- Each database partition uses its own memory areas (buffer pools, sortheap, locklist,...)
- Each database partition uses its own set of DB-Parameters.
- Statistics for partitioned table are only collected on the first partition that contains table data.
The DB2 Data Partitioning Feature (DPF) is currently supported for SAP systems which are based on SAP Business Warehouse.

DB2 Multi-Partitioning concept:
Shared Nothing = Each partition has its own set of discrete data.

Attach DB Servers as needed!

Network
DB2 Hash Partitioning

<table>
<thead>
<tr>
<th>User ID</th>
<th>Name</th>
<th>Street</th>
<th>City</th>
</tr>
</thead>
<tbody>
<tr>
<td>4711</td>
<td>Joe Smith</td>
<td>Hillstreet</td>
<td>London</td>
</tr>
</tbody>
</table>

Partition Key value hashed to value "5"

<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>...</th>
</tr>
</thead>
<tbody>
<tr>
<td>Partition</td>
<td>0</td>
<td>1</td>
<td>2</td>
<td>0</td>
<td>1</td>
<td>2</td>
<td>0</td>
<td>1</td>
<td>2</td>
<td>0</td>
<td>1</td>
<td>2</td>
<td>...</td>
</tr>
</tbody>
</table>

Hash Value "5" assigned to Partition 2

Partition Key

Partition 0

Partition 1

Partition 2
Query Processing with partitioned Tables

A coordinating agent splits the query into sub queries, one for each database partition.

Each sub query only processes the subset of the table data that is located on a particular database partition.

**Advantages:**
- Fast query response times
- Almost linear scalability
- A single query can be processed concurrently on multiple database servers
- No maintenance overhead (compared to for example range partitioning)

**Disadvantage:** Degree of parallelism can not be changed easily (requires redistribution of tables)
SAP BI/DB2 Scalability Investigation

- Fact table located on 1, 2, or 4 physical partitions (servers)
- Each server with local attached storage
- CPU work load about 100%

Scalability factors for queries with **short or medium** execution times (avg. execution time was 8 seconds on 4 database partitions):
- 1 -> 2 partitions: **1.84**
- 1 -> 4 partitions: **3.58**

Scalability factors for queries with **long** execution times (avg. execution time was 97 seconds on 4 database partitions):
- 1 -> 2 partitions: **2.04**
- 1 -> 4 partitions: **4.11**
How to define the number of database partitions?

Steps to determine number of database partitions:

1. Perform SAP BI sizing to determine required SAPS (unit of measure for computing power).
2. Determine number of CPUs on chosen hardware platform based on required SAPS.
3. Choose number of database partitions based on number of CPUs on each machine:
   - 1 CPUs per partition will cause full utilization of all CPUs for a single BI query.
   - Rule of Thumb: 1 – 4 CPUs per partition.
   - If you plan to extend the DB server (CPU, Memory), you may want to configure more partitions, in order to avoid repartitioning your data later on.
   - Avoid too many partitions for large concurrent workloads (may reduce overall throughput)
SAP BI DB2 Database Layout on multiple Partitions

Database configuration:

- SAP Basis table spaces and dimension table spaces are always on partition 0.
- Large ODS, PSA and Fact data should be distributed over partitions 1 to N.
- Partition 0 should not contain ODS, PSA and Fact data to improve bufferpool hitratio and backup/restore performance.
- SAP BW table spaces with medium sized tables should be distributed over a subset of the database partitions (for example aggregate table spaces).
SAP BW/DB2 Database Layout for BCU Systems

BCU (Balanced Configuration Units) is IBM's lowcost hardware offering for data warehousing:
✓ Many small boxes based on low cost hardware (e.g. Intel CPUs)
✓ „Balanced“ CPU capacity, memory and I/O capacity for each BCU
✓ Many DB2 database partitions

DB Layout Consideration:
Increased workload on DB Server 1 because master- and dimension data must be send to all other partitions for query processing. Statements are prepared on Server 1.

=> Partition 0 should have more CPUs than other partitions.
Backup Foils
Number of I/O Servers (prefetchers)

- **DB2 V8**: The following formula should be used:

  \[
  \text{NUM_IOSERVERS} = \max \text{ over all table spaces ( num_disks_per_container } \times \max\text{num_containers_in_stripe)}
  \]

  Minimum value should be 3

**Example:**
- Each container of a table space on a separate RAID device
- 6 disks per RAID device
- Table space 1 has 4 containers
- Table space 2 has 5 containers
- \( \text{DB2\_PARALLEL\_IO} = 1:6,2:6 \)

  \[
  \Rightarrow \text{NUM_IOSERVERS} = \max (6 \times 4, 6 \times 5) = 30
  \]

- **DB2 9**: SAP recommends to set \text{NUM_IOSERVERS} to AUTOMATIC.
Prefetching

- There are two types of physical reads:
  1. Synchronous Reads are scheduled directly by database agents.
  2. Asynchronous Reads (prefetching) are performed by I/O Servers. Prefetching can be enabled during access plan creation or at runtime. Each prefetch request is broken into multiple I/O requests.

- Prefetch performance is influenced by PREFETCH SIZE, NUM_IOSERVERS and type of buffer pools (e.g. block based).

- SAP recommends to set PREFETCH SIZE and NUM_IOSERVERS to AUTOMATIC. In this case the prefetch size is calculated based on:
  - Extent Size
  - Number of containers in the table space
  - Setting of DB2_PARALLEL_IO