



Session: D01

New to DB2 LUW but for experienced z/OS DBAs

Chris Eaton
DB2 Technical Evangelist
IBM Toronto Lab

5 October 2009 • 11:30 - 12:30
Platform: DB2 for Linux, UNIX, Windows

Abstract:

Are you an experienced DBA but you are new to DB2 for LUW? If so this session is for you. Often at a conference you will find a lot of great sessions on advanced performance tuning, monitoring, backup/recovery but sometimes we forget about people that are new to DB2 and need some level setting first. In this session Chris will take you through some of the basics of DB2 from storage management, to backup and recovery manage to the processes model for DB2 and some simple diagnostics. Then you will be able to spend the rest of the week diving deeper with a sound footing.

Agenda



- Introduction to DB2 for Linux, UNIX, Windows
 - Editions
 - Add on features
- A day in the life of a z/OS and LUW DBA
- Understanding processes/threads in DB2 9.5
- Understanding the memory model
- Managing your database
 - Backup/Recovery
 - Creating Tablespaces, Tables, Indexes, etc
- Monitoring your database

Objective1:Introduction to DB2 for LUW

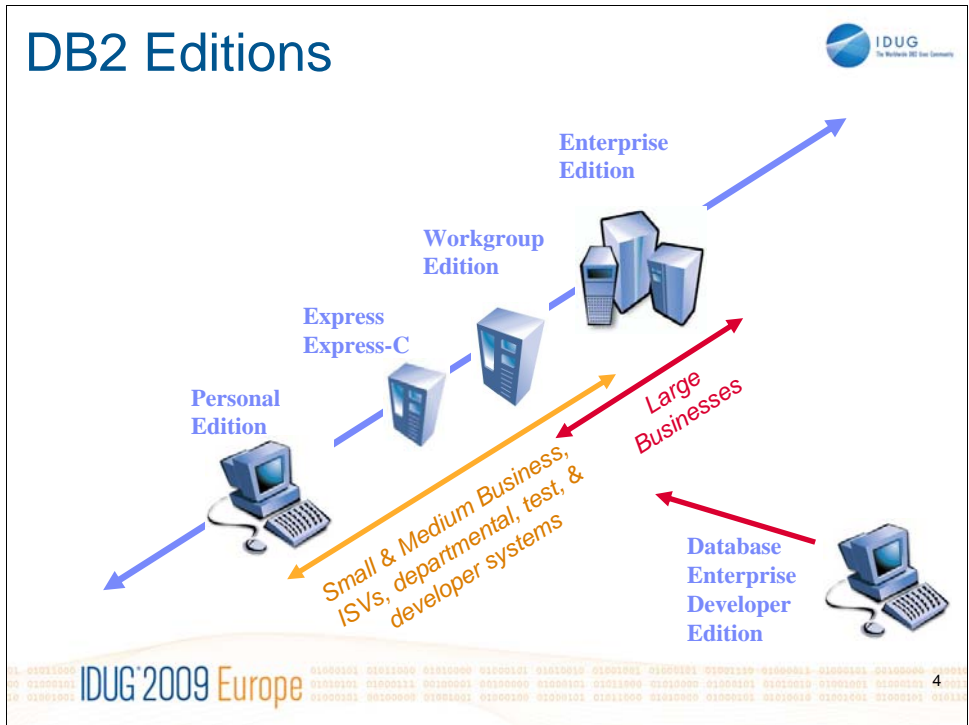
Objective2:Managing the processes in DB2 for LUW (starting/stopping/listing, etc)

Objective3:Storage management in DB2 for LUW

Objective4:Backup and Recovery in DB2 for LUW

Objective5:Simple diagnostics and performance tuning in DB2

Introduction to DB2 for Linux, UNIX, Windows

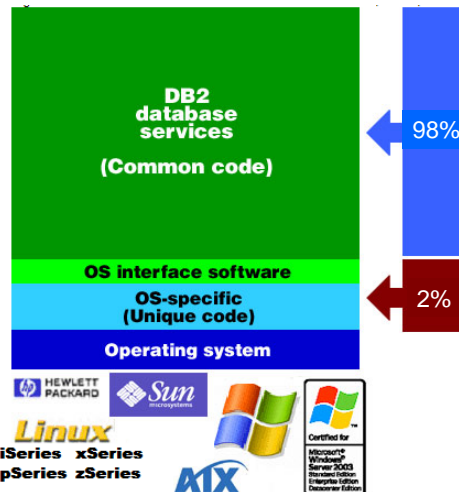


There are different DB2 packages available for different target markets. A small business probably doesn't need all the functions and features that are needed for a large enterprise application. This slide shows the DB2 offerings that are available as of the DB2 9.5 release. At the low end is **DB2 Personal Edition (DB2 Personal)**: a single user version of DB2 and it is licensed as such (per user). As you can see, there are a number of **DB2 Express** editions. These are multi-user versions of DB2, intended for small businesses, developers, and ISVs. There are some limitations on the amount of resources that Express editions will use and Express-C FTL is priced per server for extreme simplicity. **DB2 Workgroup Server Edition** is also intended for small and medium businesses or departmental servers in an enterprise. It can run on servers that are more powerful, as the PVU limits for this edition are higher. On the high end, **DB2 Enterprise Edition** has no PVU restrictions.

Architected for Portability on Linux, UNIX and Windows



- Deep Exploitation of OS
- Port Once – Continue Exploitation
- DB2 compiled on **ALL** operating systems each night
- DB2 released on **ALL** platforms simultaneously



IDUG 2009 Europe 5

Over 98% of the codebase for DB2 on Linux, UNIX and Windows is common across all of these platforms. The small percentage of code is used to abstract the operating system out so that most of the developers don't code to a given OS. Instead they code to this abstraction layer and the OSS layer is then responsible for interactions (both deep exploitations and performance capabilities) with that platform.

This allows DB2 to not only be build and tested on each platform every night but it also means that DB2 is available on all platforms and in all languages simultaneously (both for fixes and for generally available releases).

DB2 Family SQL Reference



The SQL Reference for Cross Platform Development

DB2 9 for z/OS, DB2 9.5 for LUW, DB2 for i V6.1

SQL Reference book for DB2 family

<http://www7b.software.ibm.com/dmdd/library/techarticle/0206sqlref/0206sqlref.html>



Version 3.1

IDUG'2009 Europe 6

Do you want to design or write applications for the entire DB2 family, rather than for just one of the platforms? If so, you need the IBM DB2 SQL Reference for Cross-Platform Development.

Version 3.1 covers these product versions:

DB2 9.5 for Linux, UNIX & Windows

DB2 9 for z/OS

DB2 for iSeries V6.1

SQL consistency across the DB2 family has improved substantially in the past few versions, while significant new common function has been added. SQL that is common to the DB2 UDB relational database products and the SQL 1999 Core standard is much more comprehensive. The 934 page book describes the rules and limits for preparing portable programs for these versions. More is coming.

www7b.software.ibm.com/dmdd/library/techarticle/0206sqlref/0206sqlref.html

DB2 Family SQL Reference



Less than 17% of the pages highlight a difference between platforms

ABS

→ ABS(*numeric-expression*) →

The ABS function returns the absolute value of a number.

The argument must be an expression that returns a value of any built-in numeric data type.

G
G
G
G

The data type and length attribute of the result are the same as the data type and length attribute of the argument value. In DB2 UDB for iSeries the result is an INTEGER if the argument value is a small integer and the result is double-precision floating point if the argument value is single-precision floating point.

If the argument can be null, the result can be null; if the argument is null, the result is the null value.

Note

Syntax alternatives: ABSVAL can be specified in place of ABS. It is supported only for compatibility with previous DB2 releases.

Example

- Assume the host variable PROFIT is a large integer with a value of -50000.

```
SELECT ABS(:PROFIT)
FROM SYSIBM.SYSDUMMY1
```

Returns the value 50000.



Here is an example of what you might find in the documentation. That is, a SQL Reference guide that highlights any minor differences between platforms.

A Day in the Life



	DB2 for z/OS DBA	DB2 for LUW DBA
8:00 am	Read email and answer everyone's questions	Read email and answer everyone's questions
9:00 am	Have your first of 5 coffees	Have your first of 5 coffees
10:00 am	Review status of overnight jobs	Review status of overnight jobs
11:00 am	Review the health of my system	Review the health of my system
12:00 pm	Eat lunch while I fix something that's broken (that I didn't break ☺)	Eat lunch while I fix something that's broken (that I didn't break ☺)
1:00 pm	All afternoon work to fix application developer "issues", explain to people why our existing change controls are useful , drink remaining 4 cups of coffee,	All afternoon work to fix application developer "issues", explain to people why we should implement change controls , drink remaining 4 cups of coffee,
5:00 pm	Go home on time as planned ... not!	Go home on time as planned ... not!

See life is not so different ☺

Processes and Threads in DB2 9.5

z/OS Subsystem and LUW Instance



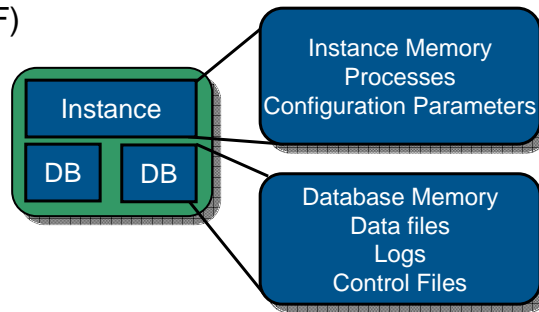
- DB2 subsystem identifies a DB2 data server, which contains its own set of catalogs and is the entity to which you connect
- A DB2 Instance is a copy of the database manager that has its own set of processes, configuration parameters and some memory associated with it
 - Databases are contained in an instance. It is the database that you connect to in LUW
 - An instance is owned by a SYSADM who has the ultimate authority over all databases in the instance

There is no exact equivalent in LUW to a z/OS subsystem. Similarly on LUW there is the concept of an instance which “houses” databases. But each individual database has its own catalog and you connect to each database individually.

DB2 for LUW Instance



- One server can have multiple DB2 instance
- One instance can have multiple databases
- One database is associated to only a single instance (except for DPF)



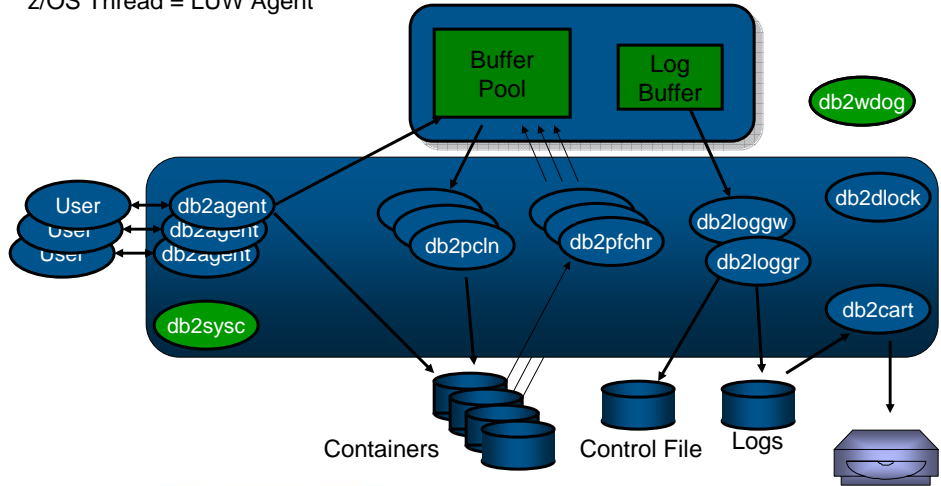
- Start the instance – login as sysadm and run `db2start`

An instance in DB2 for LUW is effectively a copy of the DB2 code running on a server. Databases live within an instance and are managed by that instance.

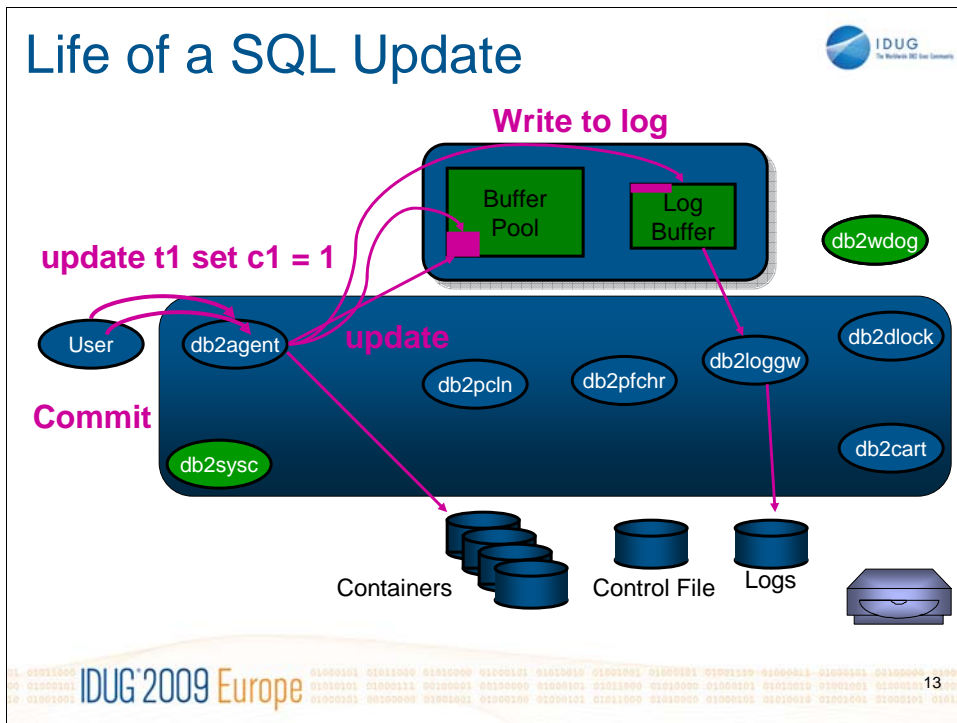
Threaded Engine in DB2 9.5



z/OS Thread = LUW Agent



As of DB2 9.5 there are now a set of operating system threads running in the instance that do all the work.



Here is an example of what happens during an update and commit statement. It looks best in screen show mode so my apologies if you are viewing this in PDF but I will try to walk you through it.

- 1) The user executes an update statement like “update t1 set c1 = 1”. This update statement is sent to the coordinating DB2 thread called db2agent which does the necessary compilation and optimization to build an access plan and then begins to execute that access plan.
- 2) The first step is to see if the page needed from t1 is in the bufferpool (for simplicity lets assume that T1 has only one data page and no index). If the page needed is not in the bufferpool then the db2agent will perform an I/O to bring that page into the bufferpool. It will then process that page and update the column C1 = 1 (for all rows on that page). It also will write the log record for this change into the log buffer (in memory).
- 3) At this point the log buffer has the change and the bufferpool has the change but there is nothing yet on disk. If the server were to crash at this point those changes would be lost which is the correct behavior because at this point it is not a committed transaction.
- 4) Next the user sends a COMMIT statement into DB2. At this point, DB2 will instruct the log writer (db2loggw) to flush the log buffer to disk (into the log file).
- 5) Only after the log records are on disk is the user application notified that the commit was successful. Notice that the changed data page is still in the bufferpool. At some point in the future it will be written out to disk when that space in the bufferpool is needed by someone else. If the server were to fail at this point the data files for T1 would not have the update but the change is recorded in the log files on disk so when the server is brought back online the automatic crash recovery would perform the update before opening the database for access.

Memory Model in DB2 9.5

Buffer Pools and z/OS



- DB2 for z/OS
 - Live in the DBM1 address space
 - In v8 can be up to 1TB in size
 - You can have up to 50 buffer pools for 4k page size and 10 for other page sizes.
 - Catalog tables go in BP0
- DB2 for Linux, UNIX, Windows
 - Same as z/OS but live in the Database Shared Memory
 - Can be up to 69TB in size
 - You can have up to 4096 different buffer pools (with different page sizes)
 - Catalog tables can go in any bufferpool (simply assign the syscatspace tablespace to any bufferpool)

In Linux, UNIX, Windows, bufferpools are created using DDL and you can have over 4000 of them if you like (each of different page sizes). A tablespace is then assigned to a given bufferpool.

Where is my EDM, RID, Sort Pool?



- The EDM pool contains:
 - Cursor Tables, Package tables and authorization cache block for each plan
 - **On LUW the first two are stored in the package cache and the authorization information is stored in the catalog cache**
- The EDM DBD cache, which contains database descriptors (DBDs)
 - Describes the structure of the database (tables, indexes, RI, etc)
 - **On LUW this information is stored in the database catalog and cached in the Catalog Cache**
- The EDM statement cache contains the skeletons of dynamic SQL
 - **On LUW dynamic SQL plans are also stored in the package cache**
- The RID pool
 - Used for all record identifier (RID) processing (for sorting RIDs during List prefetch, access via multiple indexes, Hybrid joins)
 - **On LUW RID list processing is done in the Sort Heap**
- Sort Pool
 - **On LUW this is known as the Sort Heap**

Here I have tried to show where you can find different areas of memory that you might be familiar with on z/OS in a Linux, UNIX, Windows environment.

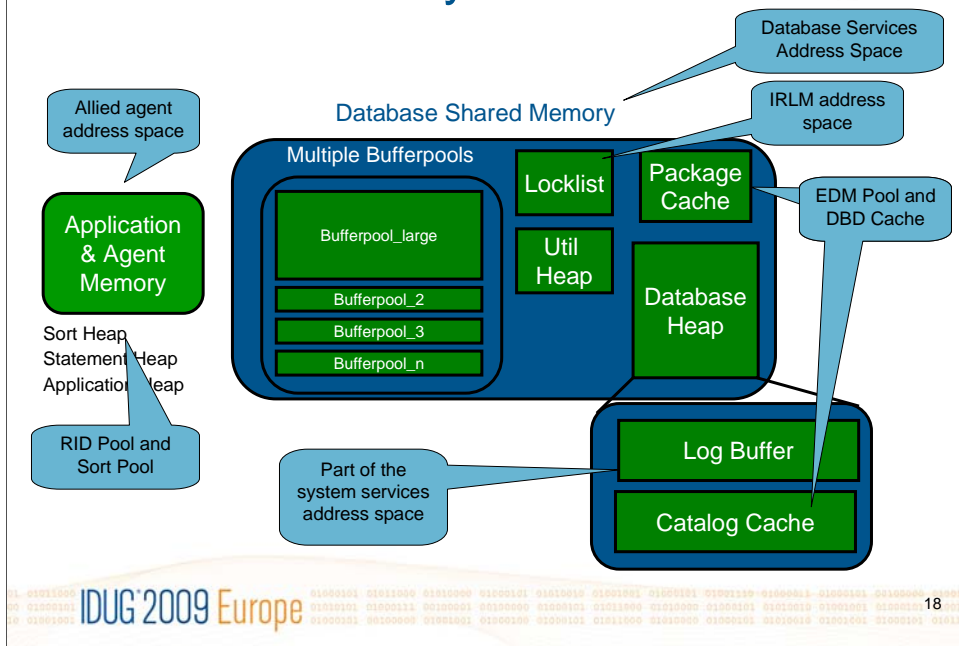
Other z/OS Address Spaces



- Internal Resource Lock Manager Address Space (IRLMPROC)
 - Used for managing locks
 - **On LUW the memory for locks is controlled by the locklist parameter and is allocated out of the database shared memory**
- DDF Address Space (DSN1DIST)
 - Used to query data between different DB2 subsystems
 - **No equivalent memory area on Linux, UNIX, Windows**
- DB2 System Services Address Space (DSN1MSTR)
 - Contains the DB2 log and coordinates the attachment of DB2 to other subsystem
 - **On LUW the log buffer is allocated out of the database heap inside the database shared memory**
- DB2 allied agent address space
 - The users address space which includes resource recovery services attachment facility (RRSAF), TSO attach, IMS, CICS, and batch address spaces.
 - **On LUW this would be known as the agent private memory**
- DB2 stored procedures address spaces (DSN1SPAS)
 - **On LUW the agent shared memory is used for stored procedure processing**

And here you can find the similarities in the various other address spaces which map to memory configuration parameters in DB2 on Linux, UNIX, Windows.

DB2 LUW Memory and z/OS

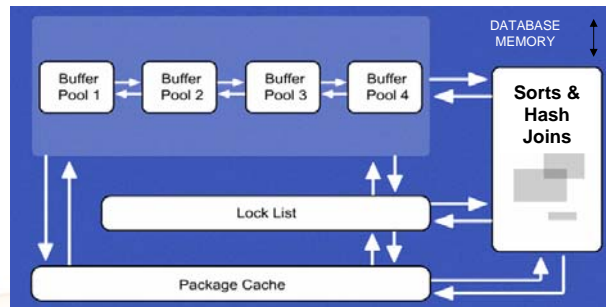


This chart summarized the memory areas in DB2. The green boxes are memory areas in DB2 on the Linux, UNIX, Windows environment. This is the type of diagram you would find in a book describing the various memory areas. The orange callout bubbles show equivalence in z/OS.

DB2 for LUW Self Tuning Memory Manager



- DB2 9 for LUW introduces STMM
 - Now only one parameter to set for tuning memory (DATABASE_MEMORY)
 - Even this parameter can be set to Automatic
 - Adapts to workload shifts that require memory redistribution



Self tuning memory manager helps you with all the database memory structures by automatically allocating memory where it is needed when it is needed.

STMM – Customer Success



- “STMM dramatically increased our bufferpool size and **reports that took 2-3 minutes to extract are now being extracted in less than 10 seconds**. I have just received a call from my client asking me what kind of magic have we done. **The system has almost near real-time responses with simpler queries, and 5 to 10 times faster on more complex reports** (depending on the case). I took all the credits for that, but I guess I just unleashed a very powerful tool that you guys created.”
- “**Initially, no, I did not trust it. I refused to use it.** Then I had a new database to bring on line and did not have time to tune it, so I allowed DB2 9 to tune itself. I set only the initial buffer pool sizes and allowed DB2 to adapt automatically. The idea was, get the database up now and I would tune it later. **4 months later, the database is performing above expectations and I never did tune it myself.** The Next Test: Enabling on our existing highly tuned databases resulted in 3-5% performance improvement after enabling STMM on our existing databases.”

Managing Your Database

Backup/Recovery
Creating/Managing Objects
Monitoring the Database

DB2 LUW Backup and Recovery



- Integrated command to perform backups
 - full backup (data pages backed up, not files)
 - incremental (delta and cumulative at the page level)
 - online / offline
 - tablespace level (partition level)
- Recovery
 - tablespace or full, point in time
 - recover from dropped table
 - recover to different physical objects (containers)
- Backup to
 - disk, tape device, named pipe, TSM, XBSA backup
 - native compression supported on backup

DB2 has a number of options for backup including full, incremental, online, offline, full database, tablespace and more.

Backup in DB2 for LUW



Command syntax

```
>>-BACKUP--++DATABASE++database-alias----->
'-DB-----'

>----->
'-USER--username--++
'-USING--password-'

>----->
'-ON--++DBPARTITIONNUM--++| Partition number(s) |----->
| '-DBPARTITIONNUMS-' |
'-ALL DBPARTITIONNUMS--++
'-EXCEPT--++DBPARTITIONNUM--++| Partition number(s) |-'
'-DBPARTITIONNUMS-'

>----->
'-ONLINE-'
'-TABLESPACE--(---tablespace-name---)-'

>----->
'-INCREMENTAL--++
'-DELTA-'

>----->
++USE--++-TSM-++| Open sessions |-----| Options |++
| '-XBSA-' |
| '-SNAPSHOT--++
| '-LIBRARY--library-name-' |
++LOAD--library-name--| Open sessions |--| Options |-----+
| '-XBSA-' |
| '-SNAPSHOT--++
| '-LIBRARY--library-name-' |
'-TO--++dir--++
'-dev--'
```

Turns into image copy

Makes an incremental image copy

Here is the syntax for the backup command. Note that you can specify the tablespaces you want to backup right in the command line which will create an image copy. If you don't specify any tablespaces then DB2 will backup the entire database.

Backup and Recovery



	DB2 for z/OS	DB2 for LUW
Backup/restore everything	Yes	Yes
Tablespace level backup/restore	Yes	Yes
Online and Offline	Yes	Yes
Incremental	Yes	Yes
Point in Time recovery	Yes	Yes
Full system backup command	Backup system	Backup database
Full system recovery command	Restore system	Restore database
Tablespace level backup command	Copy	Backup database
Tablespace level recovery command	Recover	Restore database

For backup and recovery the functions are pretty much the same. The main difference is that on Linux, UNIX, Windows we use the backup and restore command both for full system images as well as tablespace level copies.

DB2 for LUW Table spaces

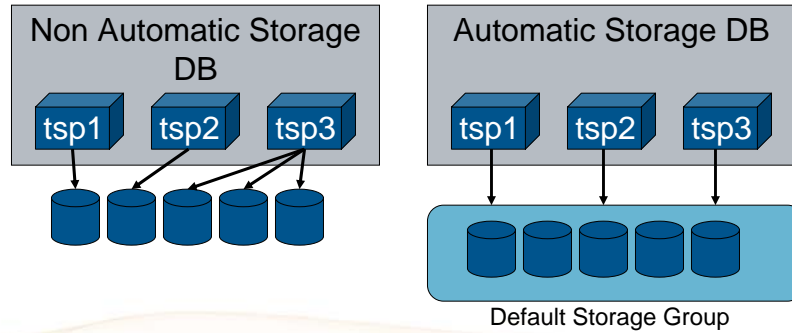


- In DB2 a Tablespace is a set of containers
 - **SMS** is made up of file system directories
 - files inside the directories represent tables and indexes
 - space allocated on demand
 - table and index go in the same tablespace
 - **DMS File** made up of files (datasets) in any directory
 - container size defined at creation time
 - will increase automatically as needed
 - can separate table and index into separate tablespaces
 - **DMS Raw** is a raw logical volume (not recommended)
 - can separate table and index into separate tablespaces
 - logical volumes used as containers to store data

Automatic Storage Table spaces

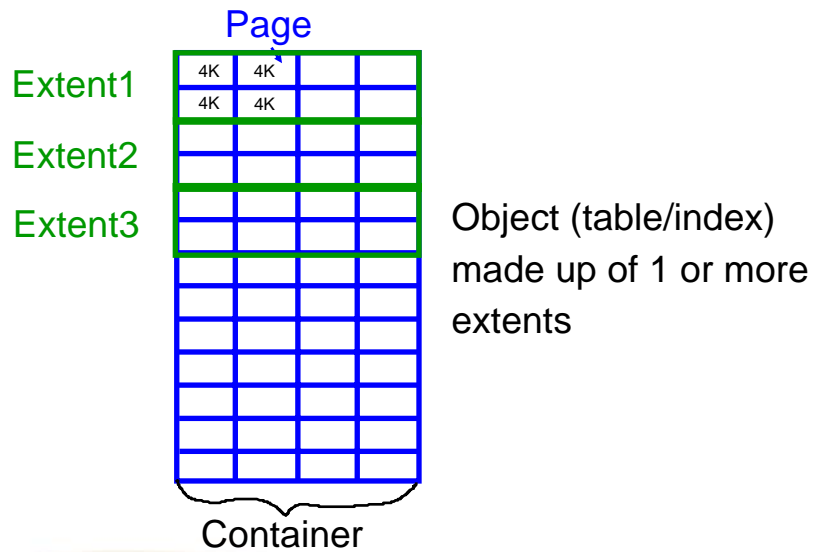


- Default for new databases created in DB2 9
- Adding more storage is simplified
 - **ALTER DATABASE ADD STORAGE ON 'F:'**



Automatic storage is now the default behavior in DB2 9. That is, newly created databases will have a storage pool associated with them and tablespaces created without container paths specified will automatically use this storage pool.

DB2 Data Structures

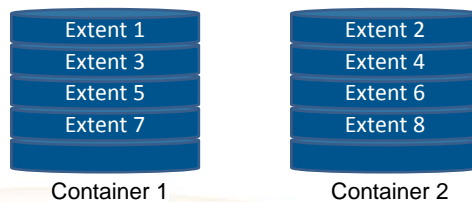


A container is a unit of storage that is broken up into pages (of size 4k, 8k, 16k or 32k). These pages are then allocated to tables or indexes in units we call an extent (which is a configurable number of pages). Each extent is assigned to one database object.

DB2 for LUW Extents



- In DB2 an extent is a set of contiguous pages assigned to a database object (table or index)
- When a tablespace is made up of multiple containers an extent is the unit of space that DB2 will use to write to one container before moving to the next container.
- For optimal I/O parallelism, DB2 maintains extent striping even when new containers are added (default)



Extents are then allocated in a round robin fashion across all of the containers in a tablespace to try to maximize parallelism when do large sequential I/O operations.

Work File Database (DSNDB07)



- Used as storage for SQL statements that require working storage (for example if a sort or join operation needs to spill to disk)
- On LUW this is known as the **system temporary tablespace**
 - By default, one system temp tablespace is created per database called TEMPSPACE1
 - More can be created if needed using the CREATE TABLESPACE command

DB2 on Linux, UNIX, Windows uses what is known as a temporary tablespace for storing temporary tables created during joins and sorts. There is one default TEMPSPACE created when a database is created and you can create more if needed.

TEMP Database



- Used to store Global Temporary Tables
- On LUW this is known as the **User Temporary Tablespace**
 - By default, there are no user temp tablespaces created at database create time
 - User temp tablespaces can be created using the CREATE TABLESPACE command

For global temporary tables, you can create what is known as a USER TEMPORARY TABLESPACE in Linux, UNIX, Windows. There is NOT one created when a database is created so if you intend on using declared global temp tables then you must first create a user temp tablespace.

Tables



- Pretty much the same everywhere
- Some unique features on DB2 for LUW
 - Multi Dimensional Clustered tables
 - Tables in DPF hashed across servers
- Some slight differences
 - Partitioned tables implementations are slightly different
- Indexes are mostly the same
 - Implementations vary slightly

A table is a table is a table...no matter the DBMS.

Runstats



- Virtually the same everywhere
 - Allows for share level reference and share level change
 - Gathers table, column and/or index statistics
 - Difference is that on Linux, UNIX, Windows you specify the table (and/or index) to gather statistics on rather than the tablespace in z/OS

Collecting statistics is virtually identical on both z/OS and Linux, UNIX, Windows.

Reorganization



- Virtually the same everywhere
- All allow for reorganization of table and/or indexes
- All allow online or offline
- The DB2 for LUW online reorg runs INPLACE so there is no shadow table and no switch.
- Syntax for Linux, UNIX, Windows is
 - REORG TABLE ...

As is reorganization. The one difference is that on Linux, UNIX, Windows, an online reorg is performed “In Place” with the real data so there is no shadow copy of the table created and therefore no switchover is required.

Monitoring Your Database

DB2 LUW Monitoring



- What is Snapshot monitoring?
 - A “picture” of the state of the DB2 system at a point in time
 - A report on a set of counters (mostly) stored inside DB2
 - Just like a camera, a snapshot is initiated by a human
- What is an Event monitor?
 - A similar set of information (counters mostly) triggered by a defined event
 - For example, information about what an application did when it disconnects from the database
 - We won't discuss Event Monitoring in this session

Snapshot monitoring gives you a view into what is happening with the database at an instant in time. Just like a photograph captures a speeding car driving down the road, so does the snapshot capture constantly changing information that is being tracked by DB2. By the time you look at the snapshot, the values will likely have changed. The way a snapshot is triggered is by external interfaces. So as an administrator, you need to run a SQL statement or call an application programming interface (API) to see the snapshot information.

An event monitor on the other hand shows you what is happening when a given event occurs. The collection of this information is triggered by an internal event. For example, you can set an event monitor to trigger whenever an application disconnects or when a deadlock occurs. Most of the elements gathered are the same for both snapshot and event monitors.

In the rest of this presentation we will focus only on snapshot monitoring.

DB2 9 Makes Your Life Simpler – Administrative Views



- All views are in the SYSIBMADM schema
- Convert coded values to text strings
- Allows you to query the snapshot monitors through SQL Statements
 - See session D17 for more details

By using the new administrative views for all the snapshot table functions, your SQL can become a bit easier to read and write.

To select all the columns out of the snapshot_database udf, you need to run

```
select * from table(snapshot_database('',-1)) as sntable
```

However, if you use the SYSIBMADM.SNAPDB view the above select statement becomes

```
select * from sysibmadm.snapdb
```

So much less typing.

Reference / Appendix

vi – the editor you will hate



- When you start using vi on UNIX you will hate it for about 2 years
 - Then you will wonder how you ever lived without it
 - Get a quick reference guide

http://www.adminschoice.com/docs/vi_editor_ref.htm

- Some basics

My Favorite VI Command
:1,\$s/Oracle/DB2/g

Move left	l	Delete a character	x	Save	:w
Move right	h	Delete a line	dd	Save and Quit	:wq
Move up	k	Delete a word	dw	Exit without saving	:q!
Move down	j	Copy a line	yy	Search for "text"	/text
Undo edit	U	Paste above line	P	Replace Oracle w/ DB2	:s/Oracle/DB2
Go to the bottom of file	G	Paste below line	p	Replace Oracle with DB2 everywhere in the file	:1,\$s/Oracle/DB2/g
Go to the start of file	1G	Open new line	o		
Go to line 10 of file	10G	Get out of edit mode	Esc Esc	Repeat the previous cmd	.

Here is a short primer on using vi. It is a very powerful editor but it takes a long time to get familiar with it because all the commands are individual letters or even worse a cryptic combination of characters. Take the time to get familiar with VI because if you are a DBA on DB2 on UNIX you will want to have a good UNIX editor and VI is available on every UNIX platform.

Shell Scripting



- Now that you have an editor to use, you will need to learn how to write shell scripts
- Useful resource
 - <http://www.well.ox.ac.uk/~johnb/comp/unix/ksh.html>
 - http://www.dartmouth.edu/~rc/classes/ksh/print_pages.shtml
- If you want to get really advanced learn to use awk
 - <http://www.grymoire.com/Unix/Awk.html>

Command Reference



DB2 for z/OS	DB2 for LUW
ALTER BUFFERPOOL	ALTER BUFFERPOOL
ARCHIVE LOG	ARCHIVE LOG
CANCEL THREAD	FORCE APPLICATION
DISPLAY BUFFERPOOL	SYSIBMADM.SNAPBP VIEW
DISPLAY DATABASE	VARIOUS SYSIBMADM VIEWS
DISPLAY LOG	SYSIBMADM.SNAPDETAILLOG VIEW
DISPLAY THREAD	LIST APPLICATIONS
DISPLAY UTILITY	LIST UTILITIES
RECOVER INDOUBT	LIST INDOUBT TRANSACTIONS
START DATABASE	ACTIVATE DATABASE
START DB2	DB2START
STOP DATABASE	DEACTIVATE DATABASE
STOP DB2	DB2STOP

Session D01



New to DB2 LUW but for experienced z/OS DBAs

Chris Eaton

IBM Toronto Lab
ceaton@ca.ibm.com

IDUG 2009 Europe 43

Chris Eaton is Senior Product Manager for DB2 primarily focused on planning and strategy for DB2. Chris has been working with DB2 on the Linux, UNIX, Windows platform for over 16 years. From customer support to development manager, to Externals Architect and now as Product Manager for DB2, Chris has spent his career listening to customers and working to make DB2 a better product. Chris is the author of “IBM DB2 9 New Features” and “The High Availability Guide for DB2” and has one of the most popular blogs about DB2 on IT Toolbox at <http://it.toolbox.com/blogs/db2luw>