

2-6 October, 2006

Hilton Vienna

Vienna, Austria

IDUG® 2006

Europe

D09 Repeated in D15

DB2 Autonomics: Implementation and Exploitation

Philip K. Gunning
Gunning Technology Solutions, LLC

4 Oct 2006 • 11:00 a.m. – 12:00 p.m.

Repeated 5 Oct 2006, 10:30 a.m. – 11:30 a.m.

Platform: Linux, UNIX and Windows

GoFurther



Disclaimer

- DB2® is a registered trademark of IBM Corporation
- DB2 for Linux, Unix and Windows® is a registered trademark of IBM Corporation
- All other products mentioned herein are trademarks or copyrights of their respective companies

Outline

- DB2 Autonomics – Background and History
- Autonomics Overview – DB2 V8.1 - V9.1
- DB2 V9.1 – Significant Increase in Autonomic Capabilities
 - Automatic Storage by default
 - Automatic Configuration of Pre-fetchers and Page Cleaners
 - Automatic Table and Index REORG Policy Options
- Self Tuning Memory Manager (STMM)

Background

- Autonomics trend began in the mid to late 90's and DOT.COM bust brought it to the forefront
 - Bust forced companies to become more efficient and to do more with less
- Autonomics formalized and introduced in DB2 V7.2
 - **Not enabled by default**
- *Implementation and Exploitation* began in DB2 V8.2
- In V9.1 **MOST** important parameters and features now enabled and exploited by default


Present Day

- **DB2 V9.1 Autonomic Implementation and Exploitation**
 - **Self-tuning Memory Manager (STMM)**
 - **Automatic storage**
 - Automatic Table Maintenance
 - Automatic REORG
 - Automatic RUNSTATS
 - **Automatic key DB CFG Parameters**
 - **More autonomic support for DPF**
- ***MOST*** settings now enabled by **DEFAULT**

DB2 V8.2 – DB2 V9.1 Autonomic Parameter Comparison

Parameter	V8.2 Default Value	V9.1 Default Value
auto_maint	OFF	ON
auto_runstats	OFF	ON
auto_tbl_maint	OFF	ON
self_tun_mem	NA	ON
avg_appls	1	AUTOMATIC
database_memory	AUTOMATIC	AIX and Windows AUTOMATIC Linux, Solaris COMPUTED
locklist	100,50,50,25	AUTOMATIC
maxlocks	10, 22	AUTOMATIC
num_iocleaners	1	AUTOMATIC
num_ioservers	3	AUTOMATIC
pckcachesz	-1	AUTOMATIC

DB2 V8.2 – DB2 V9.1 Autonomic Parameter Comparison

Parameter	V8.2 Default Value	V9.1 Default Value
sheapthres	20000,10,000	0
sheapthres_shr	sheapthres	<i>AUTOMATIC</i>
sortheap	256	<i>AUTOMATIC</i>
auto_reorg	OFF	OFF 

DB2 V9.1 Autonomic Parameter Defaults

Output from “db2 get db cfg for sample”

Database Configuration for Database sample

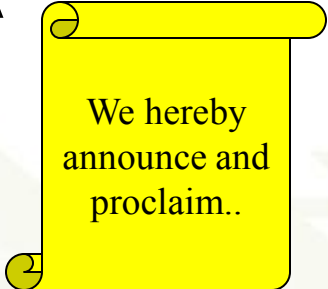
Self tuning memory (SELF_TUNING_MEM) = **ON**
 Size of database shared memory (4KB) (DATABASE_MEMORY) = AUTOMATIC
 Max storage for lock list (4KB) (LOCKLIST) = **AUTOMATIC**
 Percent. of lock lists per application (MAXLOCKS) = **AUTOMATIC**
 Package cache size (4KB) (PCKCACHESZ) = **AUTOMATIC**
 Sort heap thres for shared sorts (4KB) (SHEAPTHRES_SHR) = **AUTOMATIC**
 Sort list heap (4KB) (SORTHEAP) = **AUTOMATIC**
 Number of asynchronous page cleaners (NUM_IOCLEANERS) = AUTOMATIC
 Number of I/O servers (NUM_IOSERVERS) = AUTOMATIC
 Default prefetch size (pages) (DFT_PREFETCH_SZ) = AUTOMATIC
 Max number of active applications (MAXAPPLS) = AUTOMATIC
 Average number of active applications (AVG_APPLS) = AUTOMATIC
 Automatic maintenance (AUTO_MAINT) = ON
 Automatic database backup (AUTO_DB_BACKUP) = OFF
 Automatic table maintenance (AUTO_TBL_MAINT) = ON
 Automatic runstats (AUTO_RUNSTATS) = ON
 Automatic statistics profiling (AUTO_STATS_PROF) = OFF
 Automatic profile updates (AUTO_PROF_UPD) = OFF
 Automatic reorganization (AUTO_REORG) = **OFF**

V9.1 Default Automatic RUNSTATS

- V8.1 and 8.2 did not enable automatic RUNSTATS by default
 - Enabled by default in V9.1
- Exploit through Statistics Profiling where needed
 - Low cardinality
 - Column correlation

V9.1 Self-Tuning Memory Manager (STMM)

- New feature in DB2 V9.1 that automates memory management and tuning without any DBA intervention
- Tunes `DATABASE_MEMORY` and eligible heaps depending on the setting of the **NEW** `SELF_TUNING_MEM` parameter and settings of `DATABASE_MEMORY` parameter and other eligible heap parameter settings!




We hereby
announce and
proclaim..

V9.1 Self-Tuning Memory Manager (STMM)

- Memory that can be enabled for self-tuning:
 - **BUFFER POOLS**
 - Controlled by CREATE or ALTER BUFFERPOOL
 - **PACKAGE CACHE** (*pckcachesz*)
 - **LOCK LIST** (*locklist and maxlocks*)
 - **SHARED SORTS** (*sheapthres_shr and sortheap*)
 - **DATABASE_MEMORY**
 - Database Shared Memory

V9.1 Self-Tuning Memory Manager (STMM)

- Controlled by new DB CFG Parameter 
 - *self_tuning_mem, Enabled by default*
 - Configurable online, immediate
- **Disabled** by default in V9.1 for single partition databases
 - Set to OFF by default for multi-partition databases
- Is set to **OFF** for databases **migrated** to DB2 V9.1
- In an HADR configuration, can be **enabled** on the primary and **disabled** on the standby
 - Upon takeover, switches to active on the former standby and back to inactive on the former primary

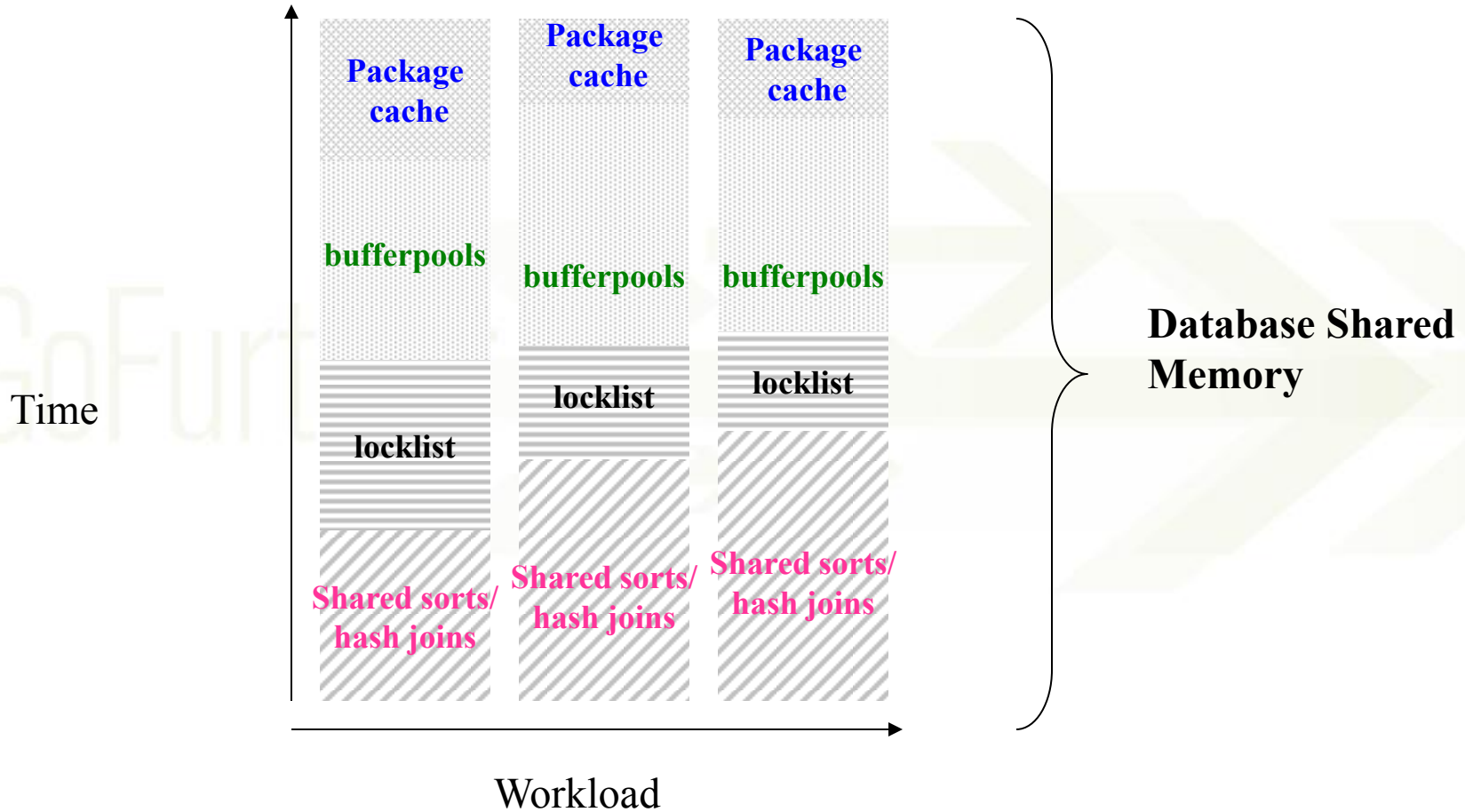
V9.1 Self-Tuning Memory Manager (STMM)

- Two forms of operation
 - **Tuning of Database Memory parameter**
 - **IF DATABASE_MEMORY set to AUTOMATIC**
 - Allocates/De-allocates memory from/to OS as necessary
 - Total amount of memory used by DB2 can grow over time
 - **No Database Memory tuning but tuning between heaps**
 - **Memory used by database is constant**
 - **Dynamically allocates memory from one heap to another as needed**
 - **Requires two heaps to enable tuning**
- Ability to tune multiple databases and instances on same server at the same time
- New Inter-instance communication enables communication and sharing of memory requirements between databases

V9.1 Self-Tuning Memory Manager (STMM)

- In order for STMM to be active, at least two database shared memory areas must be enabled for self tuning
 - Use “db2 get db cfg for <dbname> show detail” command to determine status of memory tuning
- ON (Active) means that STMM is actively tuning memory on this database
- ON (Inactive) means that even though the DB CFG parameter is ON, memory self-tuning is not occurring as there are less than 2 database shared memory areas enabled for self-tuning
- **SORTHEAP** will always be tuned if set to **AUTOMATIC** (which is **not** the default) regardless of setting of **SELF_TUN_MEM**

V9.1 Self-Tuning Memory Manager (STMM)



V9.1 Self-Tuning Memory Manager (STMM)

- STMM algorithm can evaluate and update memory requirements up to 60 times per hour
- Optimize memory based on currently running workload
- Can do in an hour what it would take a DBA weeks to do
- Can optimize memory requirements in just 1 hr
- No DBA interaction once enabled
- Stops tuning automatically when optimal configuration reached

STMM Best Practices

- Enable and Assess in Test/Dev Environment
 - Monitor memory usage and understand how STMM works
- ENABLE and EXPLOIT for production databases after DBA staff is STMM aware and memory monitoring procedures are established
 - db2mtrk
 - db2pd
 - Snapshots
 - OS tools

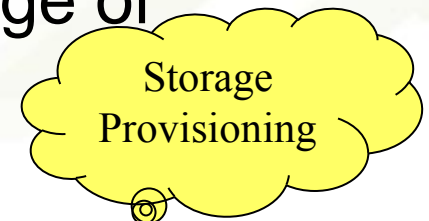
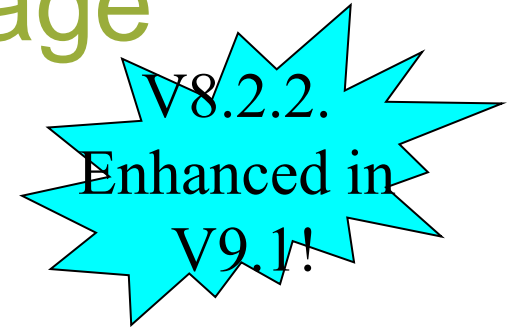
V9.1 Automatic Storage

- Prior to V9.1, only enabled if AUTOMATIC STORAGE set to YES on CREATE DATABASE
- In V9.1 enabled by **DEFAULT** on database creation
 - To disable, AUTOMATIC STORAGE must be set to NO
- **NEW SUPPORT** for multi-partition databases




V9.1 Automatic Storage

- Two enhanced storage paradigms
- Automatic Database Storage
 - Enables a database for automatic storage management via **Storage Paths**
 - Enables automatic storage management of table spaces via internal knowledge of storage paths
- Automatic Storage Table spaces



V9.1 Automatic Database Storage

- In V8.2.2 enabled by specifying AUTOMATIC STORAGE YES option on CREATE DATABASE
- V9.1 enabled by **DEFAULT** ✓
- **Single point of storage management!** 
 - Only available for databases created in DB2 V8.2.2 or higher
 - Cannot Alter and once created cannot be changed back to non-AUTOMATIC STORAGE

Automatic Storage Paths

- Example command to create a database using AUTOMATIC STORAGE
- db2 create database gttst3 on C:\path1,C:\path2, C:\path3 DBPATH on C:
DB20000I The CREATE DATABASE command completed successfully.
- db2 create tablespace ts1 (no need to specify automatic storage!)
DB20000I The command completed successfully.
- DB2 automatically created 3 containers over the defined paths: path1, path2, path3
- **By default DB2 will balance containers across defined paths**



New in
V8.2.2

Automatic Storage Paths V8.2.2

V8.2.2

Output of: db2 list tablespaces show detail

Tablespaces for Current Database

Tablespace ID	= 3
Name	= TS1
Type	= Database managed space
Contents	= Any data
State	= 0x0000

Detailed explanation:

Normal	
Total pages	= 8160
Useable pages	= 8064
Used pages	= 96
Free pages	= 7968
High water mark (pages)	= 96
Page size (bytes)	= 4096
Extent size (pages)	= 32
Prefetch size (pages)	= 96
Number of containers	= 3

Three containers
automatically
created by DB2
based on number of
defined storage
PATHS

Automatic Storage Paths V8.2.2

V8.2.2

Output of: db2 list tablespace containers for 3

Tablespace Containers for Tablespace 3

Container ID	= 0
Name	= C:\path3\DB2\NODE0000\GTSTST3\T0000003\C0000000.USR
Type	= File
Container ID	= 1
Name	= C:\path2\DB2\NODE0000\GTSTST3\T0000003\C0000001.USR
Type	= File
Container ID	= 2
Name	= C:\path1\DB2\NODE0000\GTSTST3\T0000003\C0000002.USR
Type	= File
Container ID	= 3

V9.1 Automatic Database Storage

- Add storage paths as data grows or disk layout or architecture changes
 - **ALTER DATABASE ADD STORAGE**
 - **New STORAGE PATHS available for immediate use**
- Provides single point of control for managing database/table space storage
- Enables single point of storage management!
- Frees-up DBAs for other important tasks while they can still be confident their “STORAGE” is under DB2 Control



Automatic Storage Table spaces

- Automatic Storage
 - Associates table space with one or more storage paths
 - Containers are not explicitly defined
 - File System or Directory
- Database must have been created with the AUTOMATIC STORAGE YES option on the CREATE DATABASE command
- V9.1 ENABLED by DEFAULT ✓



V8.2.2



DB2 V9.1 Automatic Storage **DATABASE** and **STORAGE PATH** Examples



CREATE DATABASE Command	Database Path	STORAGE PATH
CREATE DATABASE GTSTST4	dftdbpath	dftdbpath
CREATE DATABASE GTSTST5 ON L:	L:	L:
CREATE DATABASE GTSTST6 ON /tbs1/path1, /tbs2/path2	/tbs1/path1 (first path)	/tbs1/path1, tbs2/path2
CREATE DATABASE GTSPROD ON /prdtbs1/path1 DBPATH ON /gtsprd	/gtsprd	/prdtbs1/path1

Automatic Storage Table spaces

Intro
V8.2.2

- Enable AUTOMATIC STOREGE table space by specifying the MANAGED BY AUTOMATIC STORAGE clause on the CREATE TABLESPACE statement
 - No longer necessary as this is the default for databases and table spaces created under V9.1
- **AUTOMATIC STORAGE** table spaces are created as follows:
 - **USER** or **SYSTEM** Temporary table spaces as **SMS** directory
 - **Regular** or **Large** as **DMS** file containers
- **No need** to specify list of containers



Automatic Storage Best Practices

Intro
V8.2.2

- As a rule of thumb, **provide** enough physical disks/arrays per **storage path** to achieve desired I/O rates
- 6-10 physical disks per CPU in OLTP environment
- 10-20 physical disks in per CPU in DW/BI environment
- Place indexes for tables away from tables on different arrays
- Use DMS for XML storage



Auto-Resize Table Spaces

Intro
V8.2.2

- Enables DMS table spaces to be automatically extended or containers added when table spaces becomes full
 - Tablespace must use DMS File Containers
 - Extended or added such that a **rebalance does not** occur
- Enabled by specifying AUTORESIZE YES on CREATE or ALTER TABLESPACE statement
- **V9.1** AUTORESIZE ENABLED (YES) by **DEFAULT**



Auto-Resize Table Spaces

- **INCREASESIZE** clause on **CREATE TABLESPACE**
 - Defines the amount of space used to increase the table space when there are no free extents
- Example:

```
CREATE TABLESPACE TSJW1 MAXSIZE  
150M INCREASESIZE 15 M
```

Auto-Resize Table Spaces

- **MAXSIZE** clause on **CREATE TABLESPACE**
 - Specifies the maximum size for the table space
- **MAXSIZE 150M** specifies that the table space can grow to 150 megabytes (per partition if multiple partitions)
- **MAXSIZE NONE** specifies that there is no maximum limit for the table space
 - Can grow until a file system limit or DB2 table space limit has been reached

Auto-Resize Table Spaces

- Existing DMS table spaces can be altered to use new capability regardless of release of DB2 that they were created with
- Auto-resize can be enabled or disabled by altering the table space option `AUTORESIZE` to YES or NO
- Table spaces can grow automatically with no DBA intervention!



Automatic Storage Summary

- Automates and Simplifies Storage Management for all!
- Significantly reduces DBA Workload!
- If thought out well can provide performance boost!
 - Still important to supply adequate number of disks or disk arrays per path
- Enables EASIER REDIRECTED RESTORE!
- Enables Future Storage Management
Exploitation

Best in CLASS '06

V9.1 AUTO_MAINT



- AUTO_MAINT now defaults to YES
- Parent of all other automatic maintenance database configuration parameters
 - auto_db_backup, auto_tbl_maint, auto_runstats, auto_stats_prof, auto_prof_upd, auto_reorg
- When disabled, all of its children parameters are disabled but settings in DB CFG do not change

V9.1 AUTO_TBL_MAINT

- AUTO_TBL_MAINT is the parent of all table maintenance parameters
 - auto_runstats, auto_stats_prof, auto_prof_upd, auto_reorg
- When disabled, all of its children parameters are disabled but settings in DB CFG do not change
- When ENABLED, values for child settings take effect

V9.1 AUTO_MAINT and AUTO_TBL_MAINT Best Practices

- USE default settings to automate database and table maintenance actions
- Consider using automatic reorgs
 - Define proper maintenance window
 - Several clients have used all of above successfully with little training or knowledge of DB2

Summary

- Investigate new autonomic features
- Understand new autonomic features
- Test in your TEST/DEV environment and understand how settings work
- Implement and Exploit
- Monitor and Adjust
- Stay Tuned for future enhancements

Session D09-D15
DB2 Autonomics:
Implementation and
Exploitation

Philip K. Gunning
Gunning Technology Solutions, LLC
pgunning@gunningts.com