

Achieving Superior Performance with Speedgain for DB2

See in the dark

Gunning Technology Solutions

ITgain

September 26, 2006



Advanced DB2 performance monitoring for UNIX
and Windows

September, 26th

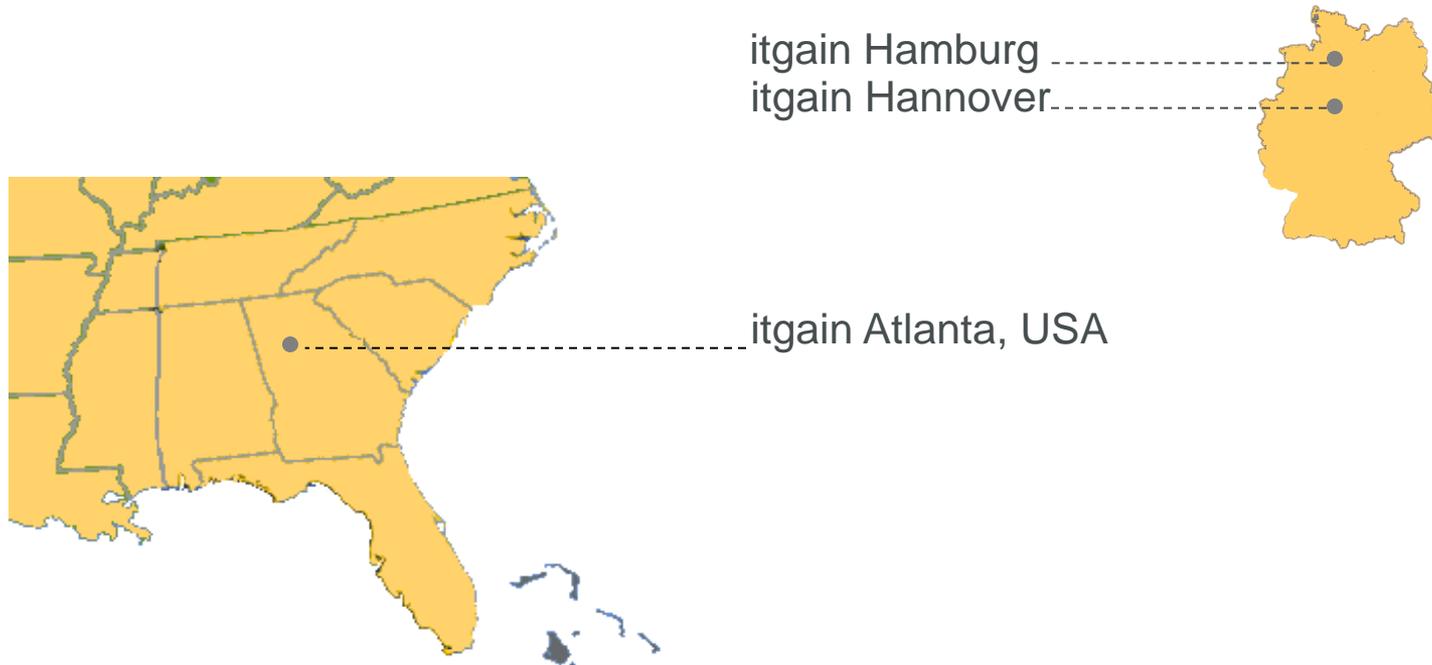
who we are – the company

- DB2 specialists founded in 2000 as a spin off of a global consulting company
- revenue growing annually
- core business DB2 consulting and software development
- Development of Speedgain for DB2 started in 2001
- as a consulting firm we advise and support leading businesses

who we are - employees

- **40 employees (2006/01/01)**
 - every employee an IBM certified DB2 User
 - 50 % IBM certified DB2 Administrators
 - Certified eBusiness Solution Advisor
 - IBM Instructor for DB2 OS/390
 - Instructors for DB2 UDB LUW
 - experienced DBAs
 - Certified Java and C++ Developers

who we are - locations



worldwide Sales Partnerships with our Distributors

who we are – Relationships



IBM Advanced Business Partner

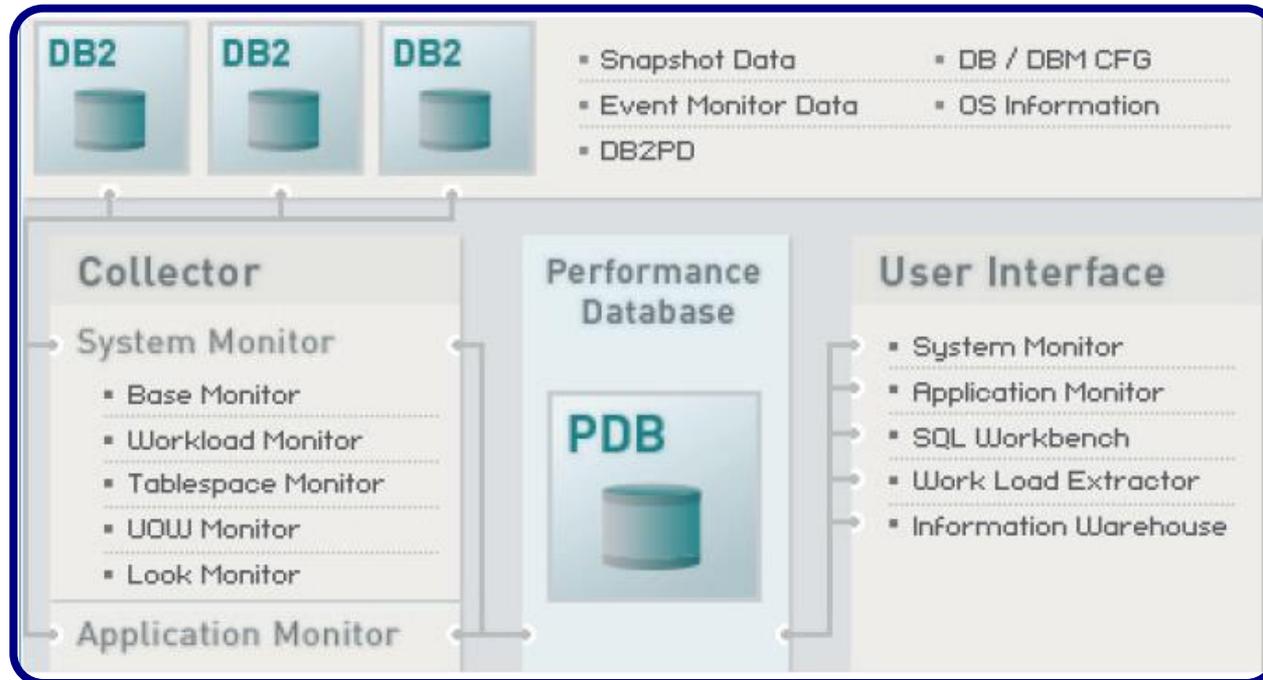
IDUG

Member of IDUG



Member of Guide Share Europe

The monitoring system – Speedgain for DB2



further information

<http://www.it-gain.com>

Florian Boldt

Product Manager Speedgain for DB2

florian.boldt@itgain.de

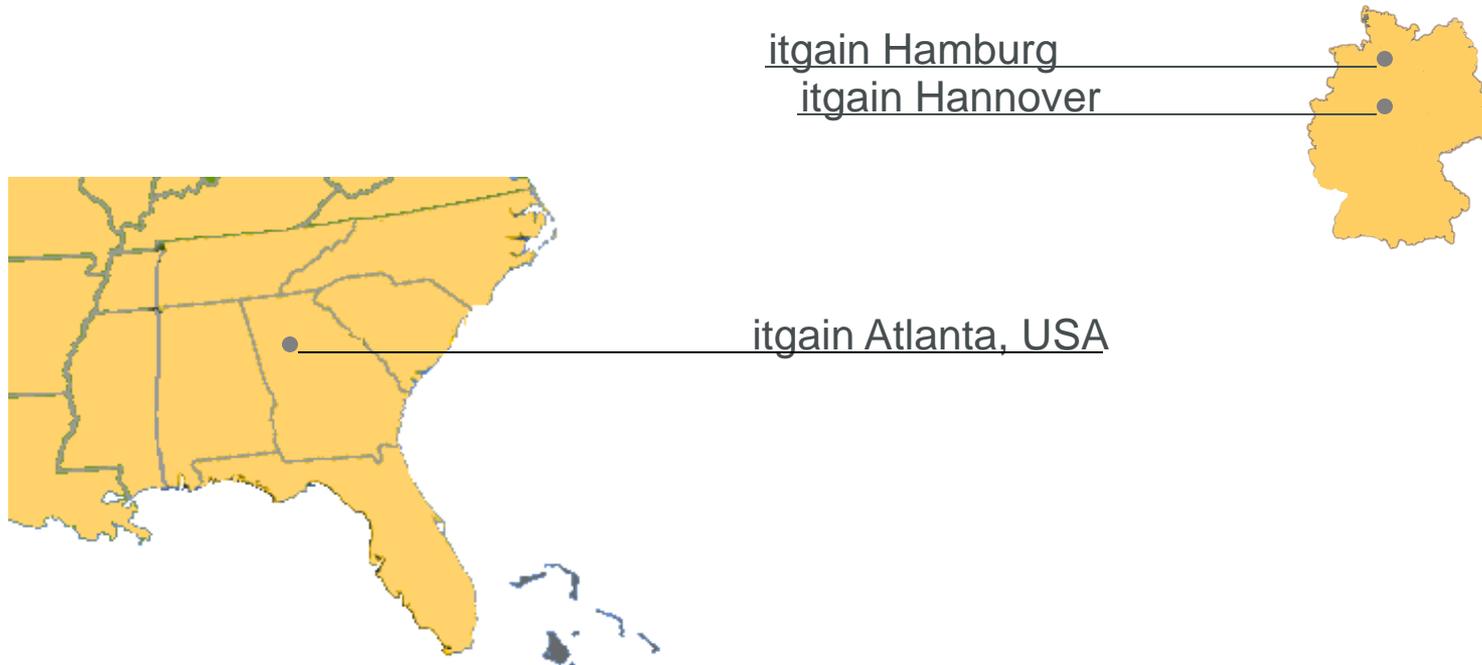
Who we are – the company

- DB2 specialists founded in 2001 as a spin off of a global consulting company
- Revenue growing annually
- Core business DB2 consulting and software development
- Development of Speedgain for DB2 started in 2002
- U.S. and European software customer base
- As a consulting firm we advise and support leading businesses

Who we are - employees

- **35 employees (2005/01/01)**
 - Every employee an IBM certified DB2 User
 - 50 % IBM certified DB2 Administrators
 - Certified eBusiness Solution Advisor
 - IBM Instructor for DB2 OS/390
 - Instructors for DB2 UDB LUW
 - Experienced DBAs
 - Certified Java and C++ Developers

Who we are - locations



worldwide Sales Partnerships with our Distributors

Who we are – Relationships



IBM Advanced Business Partner

IDUG

Member of IDUG



Member of Guide Share Europe

Who we are – Relationships



Major ITgain Consulting partner in North America



Our Products

- **Speedgain for LUW**
 - System Monitor
 - Application Monitor (Event Traces)
 - SQL Workbench
 - Workloadextractor
- **Speedgain Gateway Monitor**
- **Upcomming product enhancements**
 - 3. Qtr. 2006 Information Warehouse for long term analysis

Agenda

- **Need for Monitoring**
- **Routine Monitoring**
- **Exception-based Monitoring**

Key areas for monitoring

- Sort Usage
 - Sort overflows
 - Post threshold sorts
 - Sort memory
- Lock Contention
- Application Resource consumption
- Suboptimal SQL
 - Hash join loops and overflows
- Buffer pool performance
- Overall Database Memory Management

Agenda

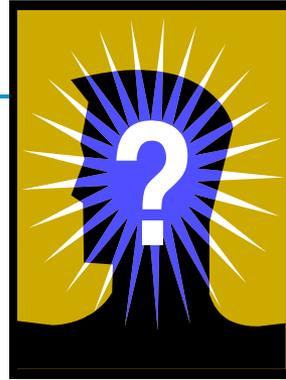
- Speedgain for DB2
 - Architecture
 - Features
- **Conclusion**

Routine Monitoring

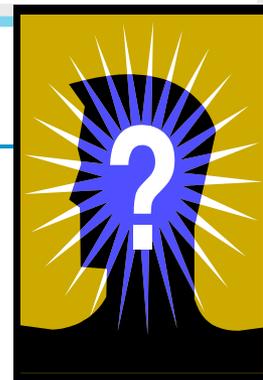
- **Consists of online real-time monitoring**
 - Classic snapshot functions
 - Insert to table SQL snapshot functions
 - New SYSCATV82 release specific views
 - db2pd command line tool
- **Create snapshot repository for realtime, this typical problem determination analysis**

db2pd Background

- **Developed based on OnStat utility from Informix**
- **Problem determination and monitoring tool**
- **First integrated into DB2 V8.2 (Stinger)**
 - Does not yet contain all options from OnStat
 - Documented in the Command Reference
- **Low monitoring overhead (latch free), preferred where possible over snapshots or event monitors**
- **Command line driven which makes it excellent for use with scripts**
- **Not dependent on monitor switches**



db2pd Background



- db2pd returns information without acquiring any locks or use of DB2 engine resources
- Since no locks are obtained, data returned by db2pd may not be completely current or accurate
 - Zero use of database engine resources however is a good trade-off
- It is important to become familiar with db2pd as there is much information provided by db2pd that cannot be obtained through snapshot or event monitoring
- db2pd contains 34 options
- We will cover these new monitoring and problem determination information elements in this presentation

Routine Monitoring

- **Automatic online real-time monitoring with automated analysis of key performance metrics**
 - Both ad hoc and scripted and collected and computed every 30 minutes, every hour, for every day
- **OS level monitoring at same interval**
 - IOSTAT
 - VMSTAT, PERFORM
 - TOP
 - SAR

Exception-based Monitoring

- **Consists of unattended agent-based monitoring in the background**
 - Monitors pre-defined threshold breaches
 - Alerts DBA's/Operations staff via email, text message, page, etc
 - Can take corrective action via script
- **After DBA's are alerted, they use online real-time (point based) monitoring to drill down to the problem**
- **Exception based monitoring runs 24x7 without regard to database environment**

Essential Snapshots

- Database Manager
- Database
- Bufferpool*
- Tablespace
- Application*
- Dynamic SQL
- Table
- Locks*

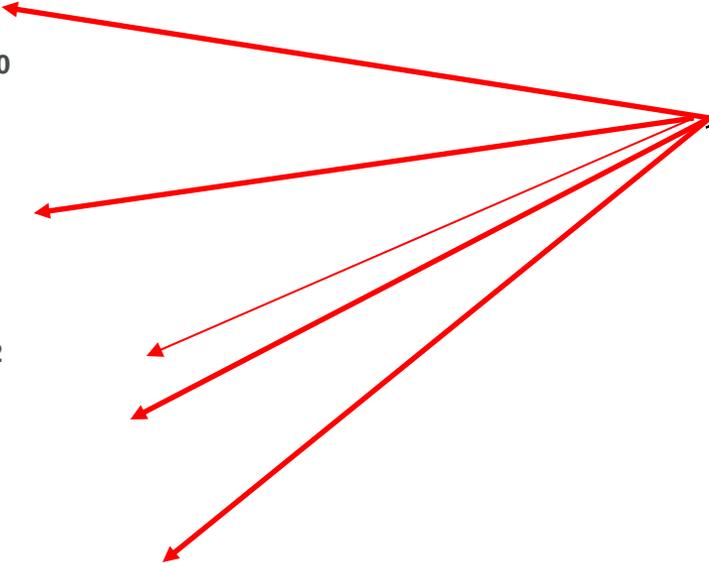
Key Database Manager Snapshot Elements

- **Post threshold sorts**
 - Occur when sheapthres has been reached

- **Agent elements**

| | |
|--|-------|
| Post threshold sorts | = 0 |
| High water mark for agents registered | = 100 |
| High water mark for agents waiting for a token | = 0 |
| Agents registered | = 100 |
| Agents waiting for a token | = 0 |
| Idle agents | = 39 |
| Agents assigned from pool | = 64 |
| Agents created from empty pool | = 103 |
| Agents stolen from another application | = 0 |
| High water mark for coordinating agents | = 62 |
| Max agents overflow | = 0 |
| Hash joins after heap threshold exceeded | = 0 |

Key DBM Snapshot Elements!



Sort Overflows

- Sort overflows occur when sorts cannot complete in sorheap
- Monitored via the following snapshot monitoring elements:
 - Post threshold sorts (DBM)
 - Pipe sorts accepted/rejected (DBM)
 - Sort overflows (DB)
 - Sort time

Eliminate Sort Overflows

- Sort overflows can be eliminated (or at least controlled) through proper index design
- Indexes defined on columns in order by sequence can eliminate sorts
- Indexes defined with “Allow Reverse Scans” specified

Sort Best Practices

- Use explain to evaluate amount of sortheap required and to determine if defined sortheap is adequate
- Only increase sortheap after reviewing sortheap requirements
- **Eliminate sort overflows (OLTP) via proper in index design**
 - Eliminate sort overflows in OLTP/Web environment
 - Keep sort overflows <3% in mixed environments
 - Keep sort overflows < 10-20% in data warehouse environment

Suboptimal SQL

- **Biggest problems are:**
 - Poorly written SQL
 - Vendor packages with poor indexing
 - Lack of understanding of DB2 predicates
 - Improper index design
 - Lack of SQL reviews and use of Explain during development process
- **Problems are pervasive in both large and small companies**

SQL Coding Best Practices

- Use Explain or third party vendor tools to review and tune SQL during development and on an ongoing basis

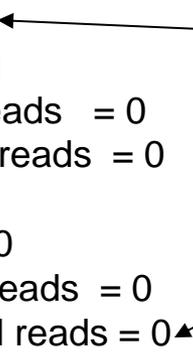
- Understand DB2 predicate rules
 - Use Range Delimiting and Index Sargable predicates whenever possible

- **Take Dynamic SQL snapshots and search for suboptimal SQL indicators**
- **Look for**
 - High CPU usage
 - High rows read vs rows selected
 - Should be 3 to 1 or less for OLTP
 - Sort overflows
 - Sorts
 - Use classic SQL snapshots, SQL snapshot functions or new SYSCATV82 views
- **Use event monitoring if necessary for hard to find suboptimal SQL**

Dynamic SQL Snapshot

| | |
|--|------------|
| Number of executions | = 1 |
| Number of compilations | = 1 |
| Worst preparation time (ms) | = 135 |
| Best preparation time (ms) | = 135 |
| Internal rows deleted | = 0 |
| Internal rows inserted | = 0 |
| Rows read | = 64 |
| Internal rows updated | = 0 |
| Rows written | = 0 |
| Statement sorts | = 1 |
| Statement sort overflows | = 0 |
| Total sort time | = 15 |
| Buffer pool data logical reads | = 4 |
| Buffer pool data physical reads | = 1 |
| Buffer pool temporary data logical reads | = 0 |
| Buffer pool temporary data physical reads | = 0 |
| Buffer pool index logical reads | = 0 |
| Buffer pool index physical reads | = 0 |
| Buffer pool temporary index logical reads | = 0 |
| Buffer pool temporary index physical reads | = 0 |
| Total execution time (sec.ms) | = 0.063596 |
| Total user cpu time (sec.ms) | = 0.000000 |
| Total system cpu time (sec.ms) | = 0.000000 |

Use these elements to compute BP hit ratios for the statement.



Dynamic SQL Snapshot

| | | |
|--------------------------------|--|---|
| Number of executions | = 1 | |
| Number of compilations | = 1 | |
| Worst preparation time (ms) | = 1 | |
| Best preparation time (ms) | = 1 | |
| Internal rows deleted | = 0 | |
| Internal rows inserted | = 0 | |
| Rows read | = 85135 | |
| Internal rows updated | = 0 | |
| Rows written | = 0 | |
| Statement sorts | = 0 | |
| Total execution time (sec.ms) | = 0.048327 | |
| Total user cpu time (sec.ms) | = 0.050000 | ← |
| Total system cpu time (sec.ms) | = 0.000000 | ← |
| Statement text | = SELECT OID_ID FROM T_OID WHERE DATE_CREATED = '2003-04-22-17.46.30.746521' | |

Common Lock-related Problems

- Locktimeout set to -1
- Locklist set too small
- Maxlocks parameter set too high

- Lock full conditions hard to detect and not well understood

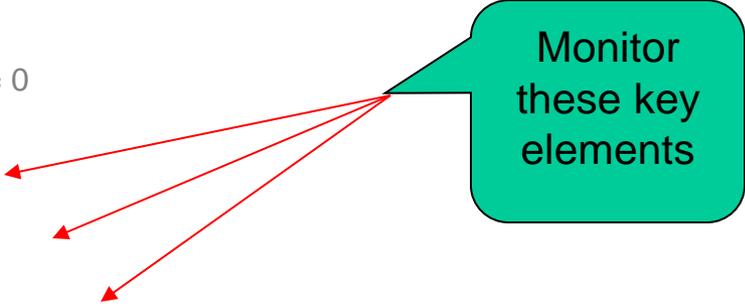
Locking Best Practices

- **Set locktimeout to a value other than -1**
 - For OLTP good starting point is 10 seconds
 - Test applications and adjust but do not set too high
- **Set locklist so that 50% of locklist is unused under normal workloads**
 - This can prevent locklist full conditions from occurring which cause unnecessary lock escalations
- **In DW, when queries will scan most of table, consider using lock table in exclusive mode**

Hash Join Best Practices

- Monitor via database and application snapshots
- Monitor hash join overflows via the following monitor elements:

| | |
|-------------------------------------|-----|
| Number of hash joins | = 0 |
| Number of hash loops | = 0 |
| Number of hash join overflows | = 0 |
| Number of small hash join overflows | = 0 |



Monitor these key elements

Achieving Breakthrough Performance with Speedgain for DB2

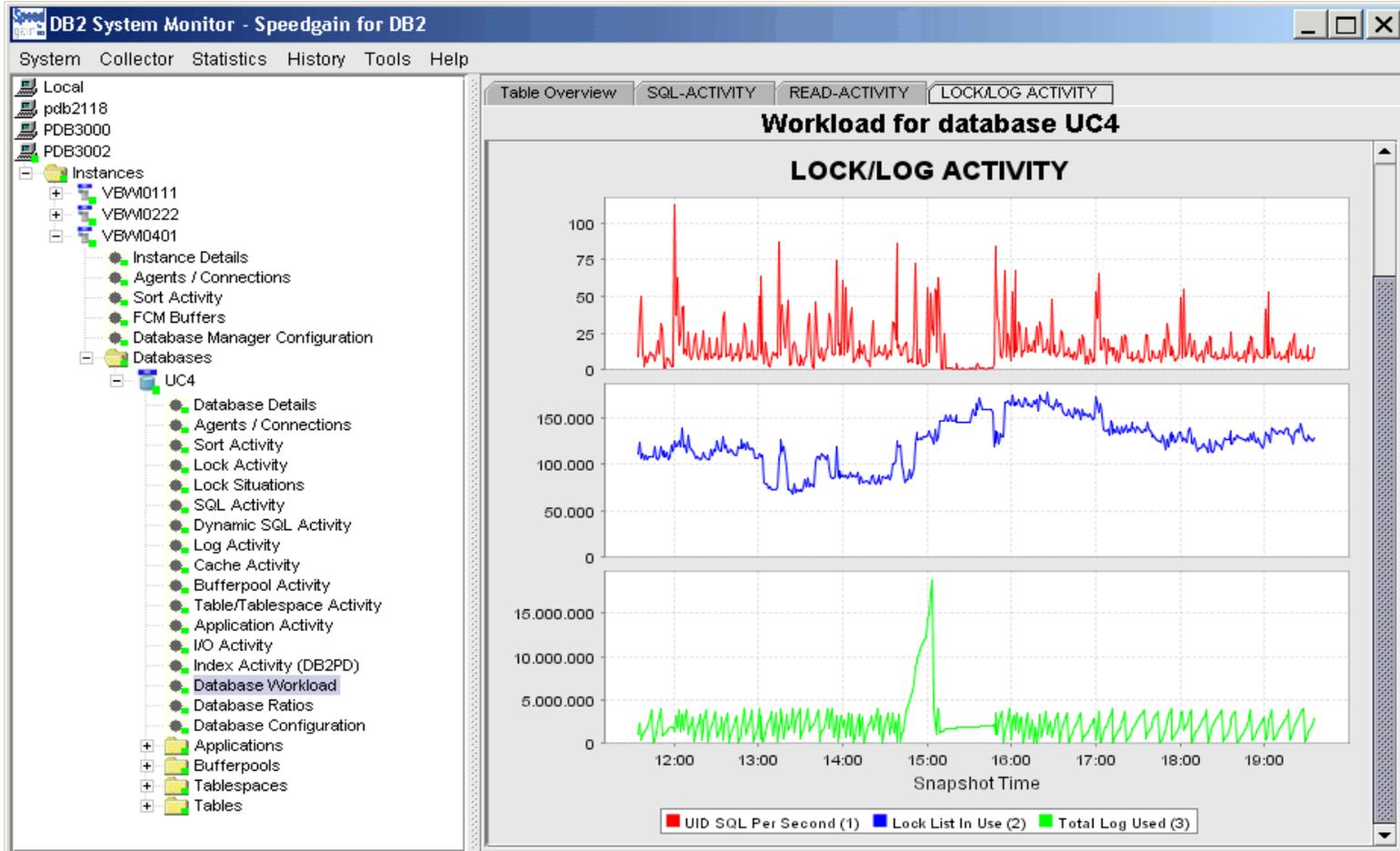
See in the dark

Gunning Technology Solutions

ITgain

September 21, 2006

System Monitor



Application Monitor

DB2 Application Monitor - Speedgain for DB2

System Application-Trace Interval Help

Snapshot: 2006-05-08 -- 18:23:01 Set Previous Next Refresh

Current Interval: 0:01:00 hours Sort 1: User CPU Time [s] Sort 2: Sys CPU Time [s] Sort 3: Elapsed Exec Time [s]

| User CPU Time [s] | Sys CPU Time [s] | Elapsed Exec Time [s] | Appl Info Type | Instance | DB Name | Appl ID | Appl Status | Rows Read | Rows Write |
|-------------------|------------------|-----------------------|----------------|----------|---------|---------------|----------------|-----------|------------|
| 11,75 | 129,2 | 0 | Event + Snap | INST0001 | DB0001 | C0A84401.B... | UOW executi... | 139728158 | |
| 0,01 | 0,77 | 0 | Snapshot | INST0001 | DB0001 | C0A84401.A... | UOW waiting | 26177 | 44 |
| 0 | 0,31 | 0 | Snapshot | INST0001 | DB0001 | C0A84401.A... | UOW waiting | 2 | |

SQL Activity UOW Activity I/O Activity Cache Activity Bufferpool Activity SQL Statements

Application Details Agents / Connections Sort Activity Lock Activity

| Start Time | Stmt Dyn Stat | Elapsed Time [s] | User CPU Time [s] | System CPU Time [s] | Stmt Type | Stmt Operation | Stmt Sqlcode | Stmt Text |
|----------------|---------------|------------------|-------------------|---------------------|-----------|----------------|--------------|---------------------|
| 19.05.13.52607 | 2 | 12,408 | 0,91 | 6,58 | Dynamic | SQL Close | 0 | select a.*, b.* ... |
| 19.05.13.52607 | 2 | 12,404 | 0,91 | 6,58 | Dynamic | SQL Describe | 0 | select a.*, b.* ... |
| 19.04.12.59424 | 2 | 12,28 | 0,52 | 6,57 | Dynamic | SQL Close | 0 | select a.*, b.* ... |
| 19.04.12.59424 | 2 | 12,274 | 0,52 | 6,57 | Dynamic | SQL Describe | 0 | select a.*, b.* ... |
| 19.04.42.77457 | 2 | 11,31 | 0,69 | 6,27 | Dynamic | SQL Close | 0 | select a.*, b.* ... |
| 19.04.42.77457 | 2 | 11,305 | 0,69 | 6,27 | Dynamic | SQL Describe | 0 | select a.*, b.* ... |
| 19.05.03.52530 | 2 | 9,954 | 0,71 | 6,42 | Dynamic | SQL Close | 0 | select a.*, b.* ... |
| 19.05.03.52530 | 2 | 9,95 | 0,71 | 6,42 | Dynamic | SQL Describe | 0 | select a.*, b.* ... |
| 19.04.54.16236 | 2 | 9,325 | 0,59 | 6,72 | Dynamic | SQL Close | 0 | select a.*, b.* ... |

SQL Workbench

The screenshot displays the Speedgain SQL Workbench interface. The main window shows an execution plan for a query. The plan starts with three base operations: **DB2MONITOR.BOOKING** (XBOOKING4, KeyColumns 0), **DB2MONITOR.CUSTOMER** (XCUSTOMER, KeyColumns 1), and **TBSCAN(8)** (DB2MONITOR.ADDRESS). These are joined by **NLJOIN(3)**, which feeds into **NLJOIN(2)**, which finally results in **RETURN(1)**.

On the right side, the **Catalog** tab is active, showing a table with the following data:

| Tablename | Schema | Type | Rows | Pages | Stat |
|-----------|------------|-------|------|-------|------|
| ADDRESS | DB2MONITOR | Table | 0 | 3424 | 29.0 |
| BOOKING | DB2MONITOR | Table | 0 | 3680 | 03.0 |
| CUSTOMER | DB2MONITOR | Table | 0 | 2656 | 29.0 |

Below this, the **Violated Rules** tab is active, showing a table with the following data:

| Indexname | Schema | Unique rule | Leaf | Levels | Full |
|-----------|------------|-------------|------|--------|------|
| XCUSTOMER | DB2MONITOR | Unique | 402 | 3 | |

At the bottom of the window, the status bar shows **SDB: DB0001 (A)** and **PDB: PDB3005W (A)**.

Workloadextractor (I)

=> Selection of the SQL-Workload

Speedgain WorkloadXtractor

Workload Definition

Start Timestamp: 2006-03-05 00:01:00 Database Name: DB0001
 End Timestamp: 2006-03-05 23:59:00 Default Schema: db2monitor

extract only most expensive statements

extract first: 10 order by: runtime

get available schemata and tables: **Get Tables**

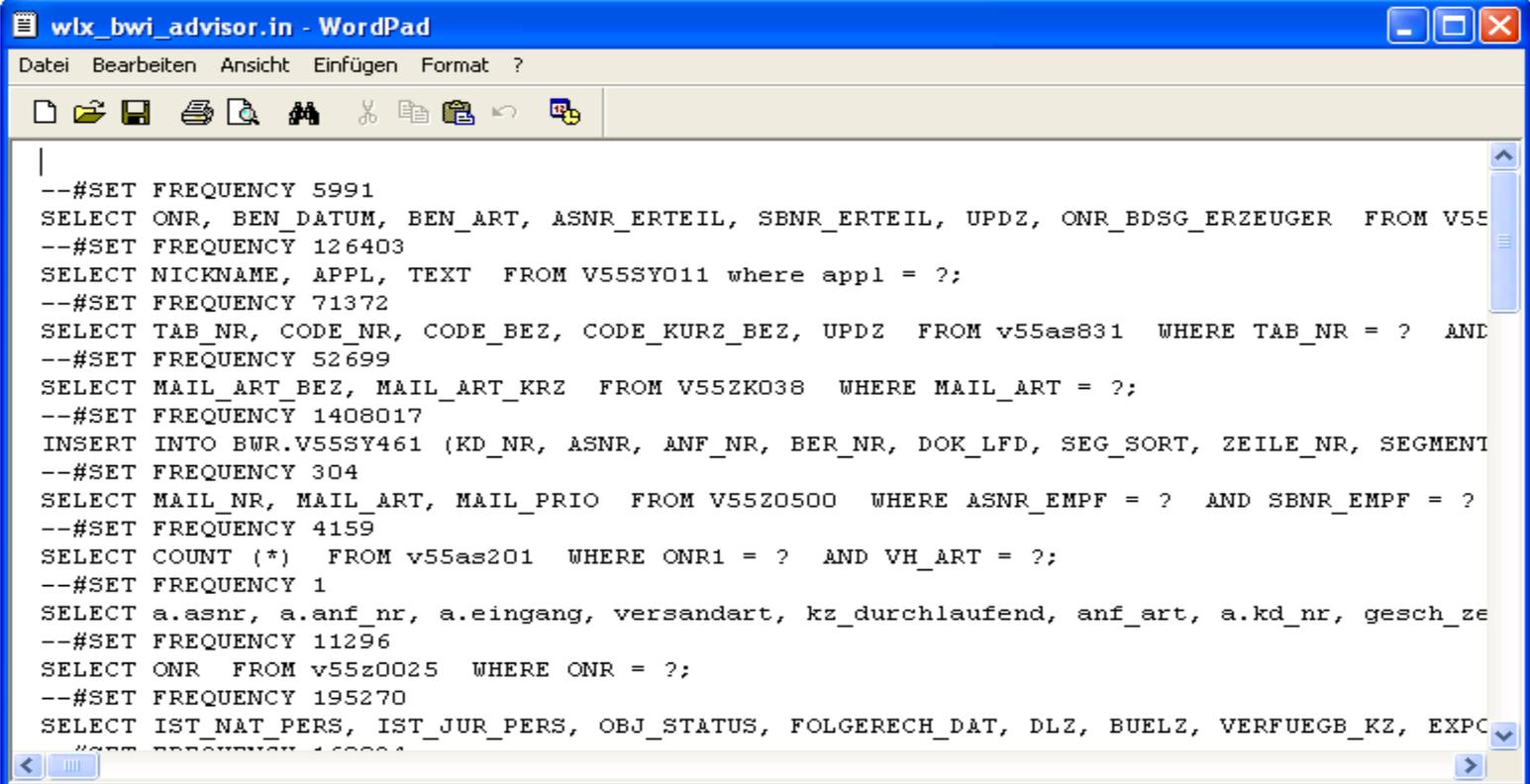
Schema: - none - Table: - none -

 Load settings  reset  Save settings

 Exit  Help  Back  Next  Skip

Workloadextractor (II)

=> The SQL-Workload Result



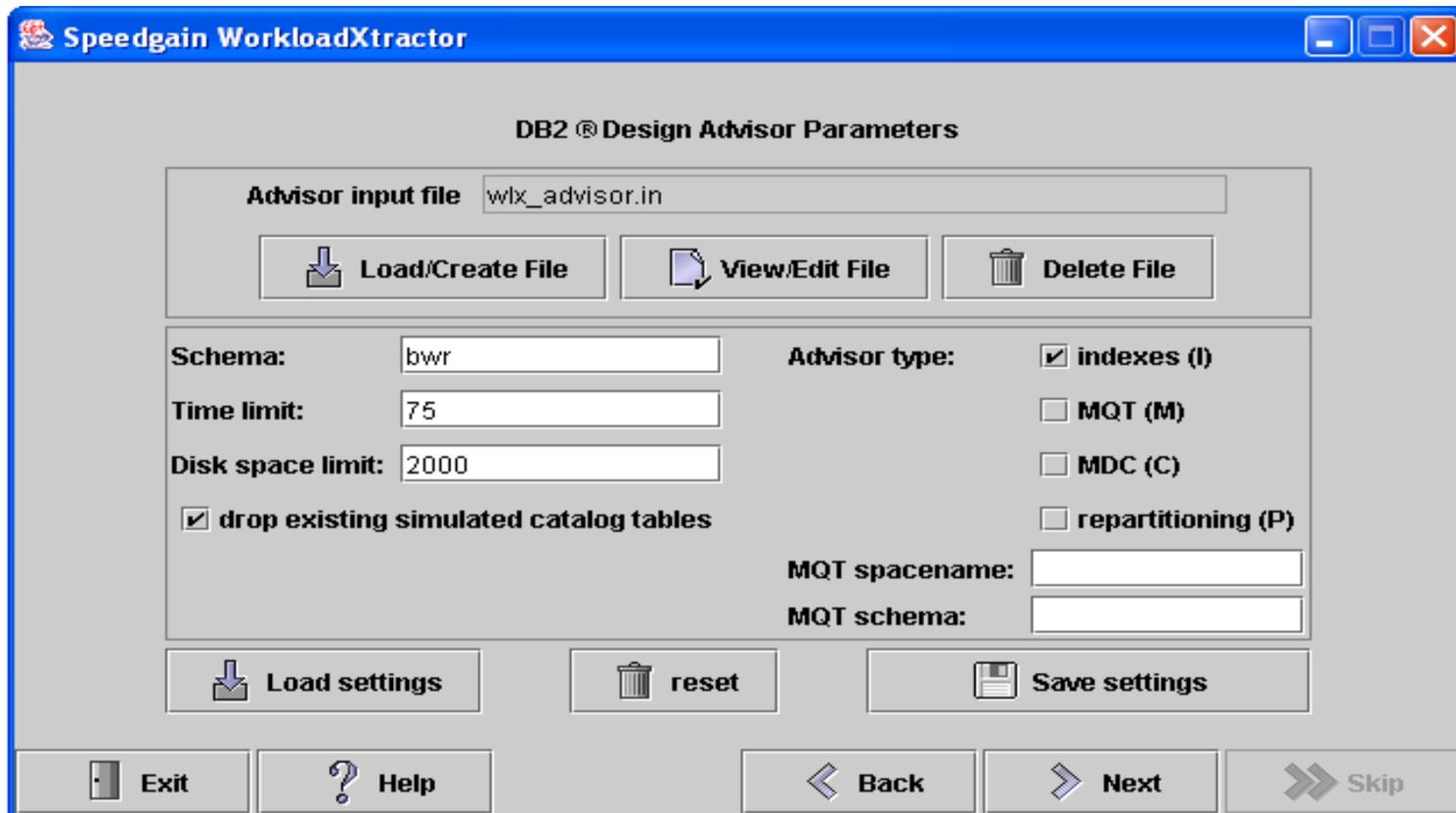
The screenshot shows a WordPad window titled "wlx_bwi_advisor.in - WordPad". The window contains a list of SQL statements, each preceded by a frequency value. The statements are as follows:

```
--#SET FREQUENCY 5991
SELECT ONR, BEN_DATUM, BEN_ART, ASNR_ERTEIL, SBNR_ERTEIL, UPDZ, ONR_BDSG_ERZEUGER FROM V55
--#SET FREQUENCY 126403
SELECT NICKNAME, APPL, TEXT FROM V55SY011 where appl = ?;
--#SET FREQUENCY 71372
SELECT TAB_NR, CODE_NR, CODE_BEZ, CODE_KURZ_BEZ, UPDZ FROM v55as831 WHERE TAB_NR = ? AND
--#SET FREQUENCY 52699
SELECT MAIL_ART_BEZ, MAIL_ART_KRZ FROM V55ZK038 WHERE MAIL_ART = ?;
--#SET FREQUENCY 1408017
INSERT INTO BWR.V55SY461 (KD_NR, ASNR, ANF_NR, BER_NR, DOK_LFD, SEG_SORT, ZEILE_NR, SEGMENT
--#SET FREQUENCY 304
SELECT MAIL_NR, MAIL_ART, MAIL_PRIO FROM V55Z0500 WHERE ASNR_EMPF = ? AND SBNR_EMPF = ?
--#SET FREQUENCY 4159
SELECT COUNT (*) FROM v55as201 WHERE ONR1 = ? AND VH_ART = ?;
--#SET FREQUENCY 1
SELECT a.asnr, a.anf_nr, a.eingang, versandart, kz_durchlaufend, anf_art, a.kd_nr, gesch_ze
--#SET FREQUENCY 11296
SELECT ONR FROM v55z0025 WHERE ONR = ?;
--#SET FREQUENCY 195270
SELECT IST_NAT_PERS, IST_JUR_PERS, OBJ_STATUS, FOLGERECH_DAT, DLZ, BUELZ, VERFUEGB_KZ, EXPC
```

At the bottom of the window, there is a status bar that reads: "Drücken Sie F1, um die Hilfe aufzurufen."

Workloadextractor (III)

=> The Usage of the SQL-Workload

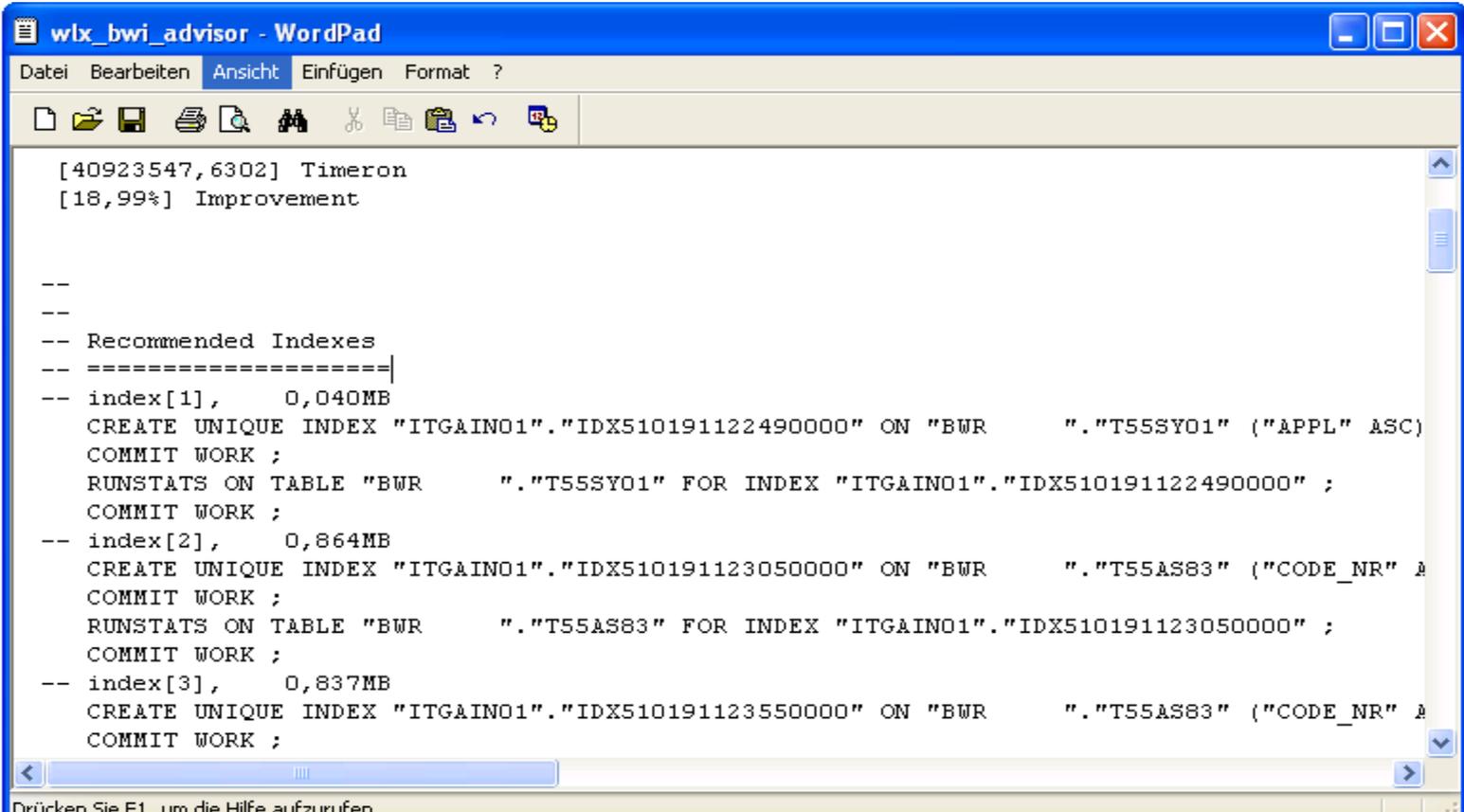


The screenshot shows the 'Speedgain WorkloadXtractor' application window. The title bar reads 'Speedgain WorkloadXtractor'. The main window title is 'DB2 ® Design Advisor Parameters'. The interface includes the following elements:

- Advisor input file:** A text field containing 'wlx_advisor.in'.
- File Management:** Three buttons: 'Load/Create File' (with a download icon), 'View/Edit File' (with a document icon), and 'Delete File' (with a trash icon).
- Schema:** A text field containing 'bwr'.
- Time limit:** A text field containing '75'.
- Disk space limit:** A text field containing '2000'.
- Advisor type:** A list of checkboxes:
 - indexes (I)
 - MQT (M)
 - MDC (C)
 - repartitioning (P)
- Options:**
 - drop existing simulated catalog tables
 - MQT spacename: [Empty text field]
 - MQT schema: [Empty text field]
- Settings Management:** Three buttons: 'Load settings' (with a download icon), 'reset' (with a trash icon), and 'Save settings' (with a floppy disk icon).
- Navigation:** A row of buttons at the bottom: 'Exit' (with a door icon), 'Help' (with a question mark icon), 'Back' (with a left arrow icon), 'Next' (with a right arrow icon), and 'Skip' (with a double right arrow icon).

Workloadextractor (III)

=> Use the SQL-Workload to check the Index Design



```
wlx_bwi_advisor - WordPad
Datei Bearbeiten Ansicht Einfügen Format ?

[40923547,6302] Timeron
[18,99%] Improvement

--
--
-- Recommended Indexes
-- =====|
-- index[1],      0,040MB
CREATE UNIQUE INDEX "ITGAIN01"."IDX510191122490000" ON "BWR"      "."T55SY01" ("APPL" ASC)
COMMIT WORK ;
RUNSTATS ON TABLE "BWR"      "."T55SY01" FOR INDEX "ITGAIN01"."IDX510191122490000" ;
COMMIT WORK ;
-- index[2],      0,864MB
CREATE UNIQUE INDEX "ITGAIN01"."IDX510191123050000" ON "BWR"      "."T55AS83" ("CODE_NR" A
COMMIT WORK ;
RUNSTATS ON TABLE "BWR"      "."T55AS83" FOR INDEX "ITGAIN01"."IDX510191123050000" ;
COMMIT WORK ;
-- index[3],      0,837MB
CREATE UNIQUE INDEX "ITGAIN01"."IDX510191123550000" ON "BWR"      "."T55AS83" ("CODE_NR" A
COMMIT WORK ;

Drücken Sie F1, um die Hilfe aufzurufen.
```

DB2 Monitors

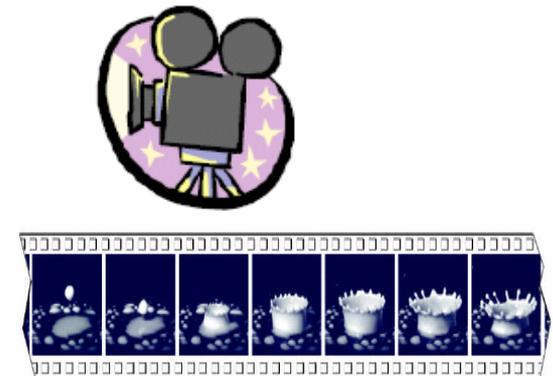
- Snapshot Monitors

- A picture of performance at that moment
- Most common DB2 performance diagnostic



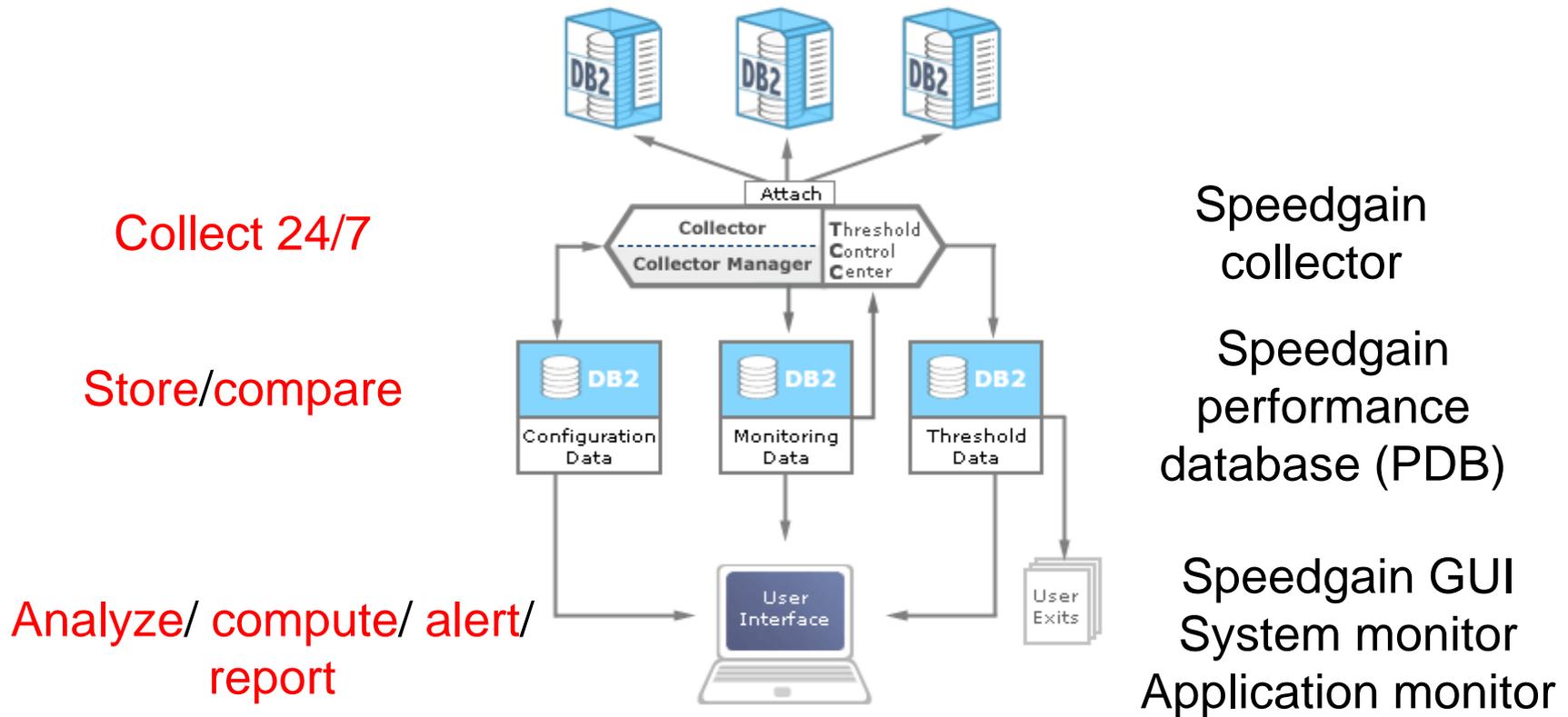
- Event Monitors (Traces)

- Often produces a huge volume of data!
- Generally used for problem diagnosis vs. monitoring

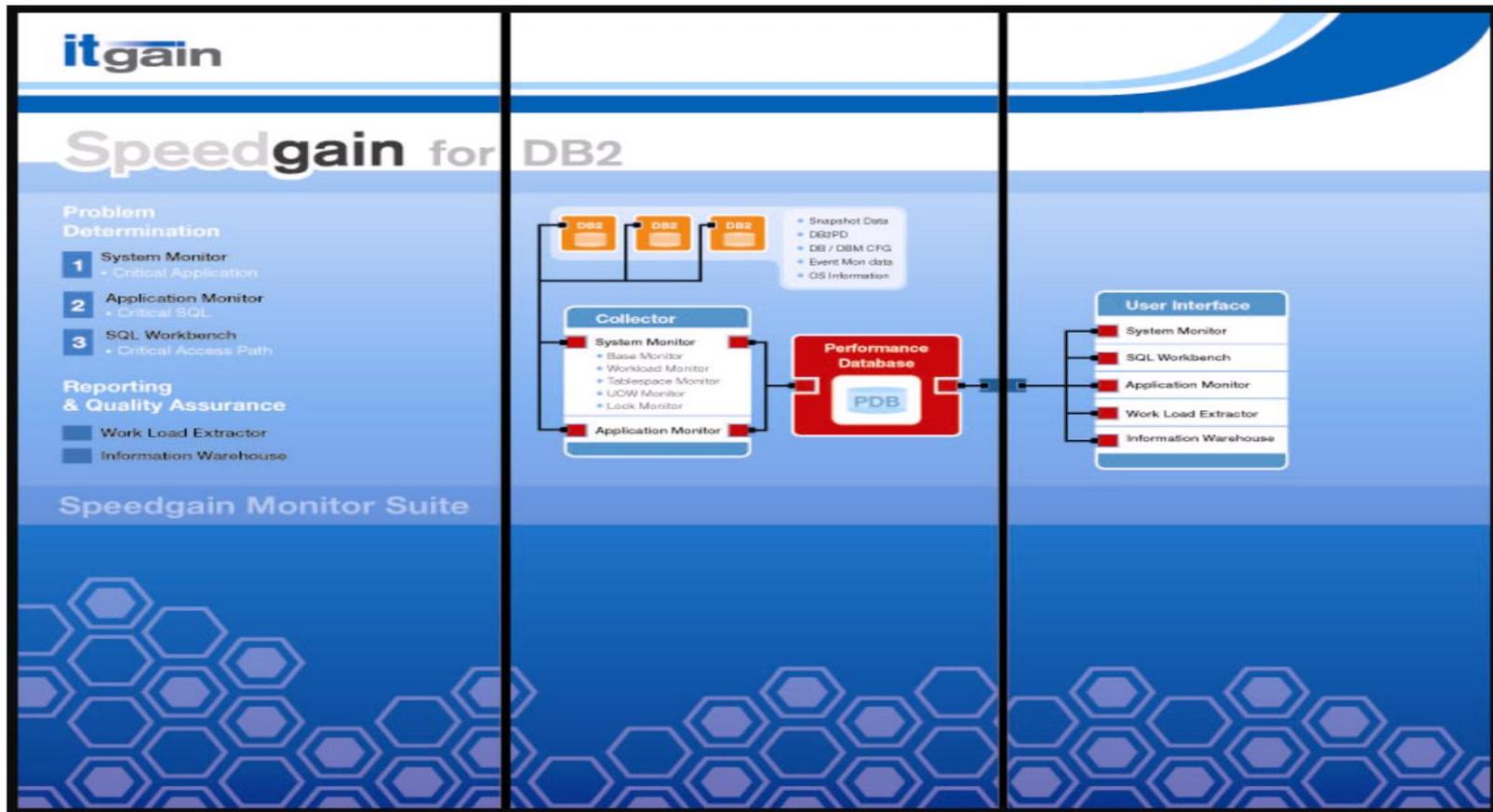


Speedgain can automatically trigger an Application Monitor via threshold violations

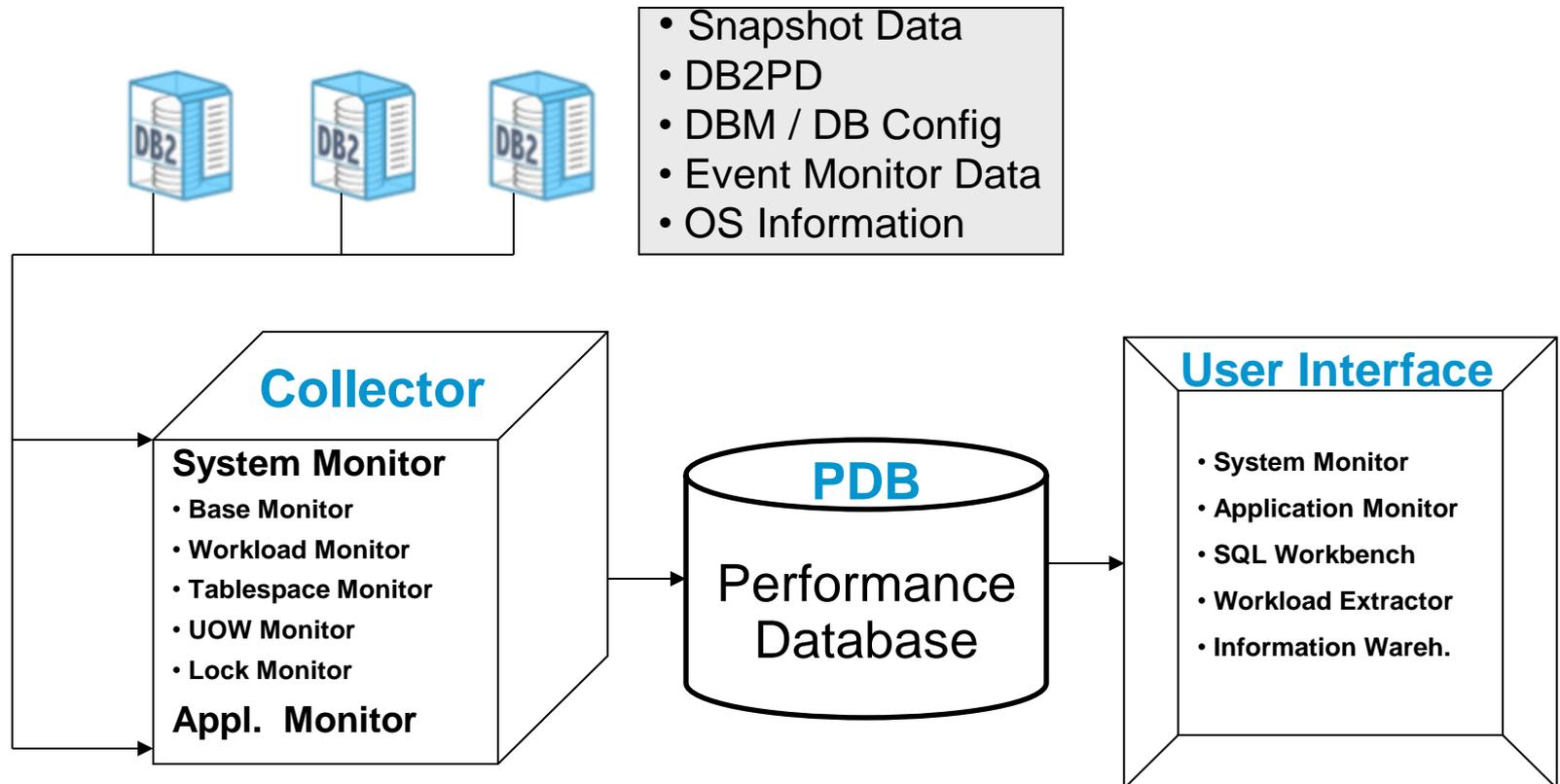
The monitoring system – Speedgain for DB2 (I)



The monitoring system – Speedgain for DB2 (II)



The monitoring system – Speedgain for DB2 (III)



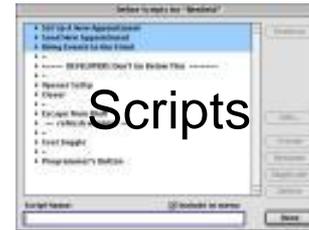
Problem Determination

The **System** Monitor looks for **Applications** with a lot of System Resource Consumption

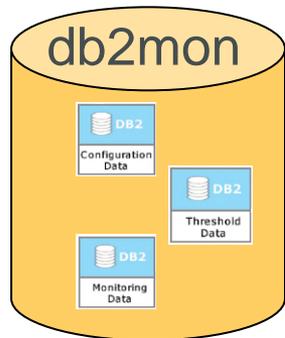
The **Application** Monitor looks for **SQL** Statements with a lot of I/O- and/or CPU-Consumption

The **SQL-Workbench** looks for critical **ACCESS PATHS** of the SQL Statements

Architecture – Collector & System Integration



Performance Database (PDB)!



- Provides an open SQL interface with detailed and aggregated performance data
- 'Everything' can be accessed and queried by SQL
- History performance data allows monitoring in the past and reporting

Architecture – Speedgain performance database

- Performance Data stored in Tables (extract)

| | | | |
|------------------|------------|---|------------|
| AGENT | DB2MONITOR | T | USERSPACE1 |
| APPL | DB2MONITOR | T | USERSPACE1 |
| APPL_INFO | DB2MONITOR | T | USERSPACE1 |
| BUFFERPOOL | DB2MONITOR | T | USERSPACE1 |
| COLL_INSTANCE | DB2MONITOR | T | USERSPACE1 |
| COLL_INSTANCE_DB | DB2MONITOR | T | USERSPACE1 |
| COLL_MANAGER | DB2MONITOR | T | USERSPACE1 |
| COLLECTOR | DB2MONITOR | T | USERSPACE1 |
| DATABASE | DB2MONITOR | T | USERSPACE1 |
| DB2 | DB2MONITOR | T | USERSPACE1 |
| DBASE | DB2MONITOR | T | USERSPACE1 |
| DBCCONFIG | DB2MONITOR | T | USERSPACE1 |
| DBMCONFIG | DB2MONITOR | T | USERSPACE1 |
| DYNSQL | DB2MONITOR | T | USERSPACE1 |
| DYNSQL_LIST | DB2MONITOR | T | USERSPACE1 |
| LOCK_WAIT | DB2MONITOR | T | USERSPACE1 |

Speedgain and DB2PD

What is DB2PD ?

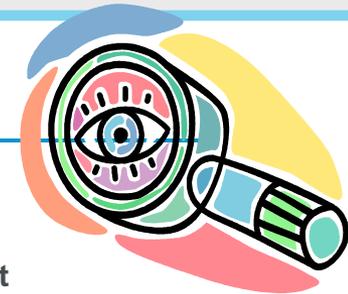
- Similar to the onstat utility of Informix
- It is a DB2 utility, which retrieves information from the DB2 UDB memory sets.
- This utility is for troubleshooting, problem determination, database monitoring and performance tuning.
- It is available since DB2 V8.2
- DB2PD can only be used on the Database Server

<http://www-128.ibm.com/developerworks/db2/library/techarticle/dm-0504poon2/>

How DB2PD works ?

- DB2PD grabs performance data from the internal DB2 OS Systemresources
- Run STRACE to find out in detail how DB2PD works

```
open("/opt/IBM/db2/V8.FP10/lib/libcxa.so.3", O_RDONLY) = 3
semget(2364797044, 1, 0) = 160989251
semop(160989251, 0x441f422c, 1) = 0
shmget(2364797044, 0, 0) = 11206673
shmat(11206673, 0, SHM_RND) = 0x44357000
semop(160989251, 0x441f4234, 1) = 0
open("/home/inst0004/sqlllib/db2system", O_RDONLY|O_SYNC|O_LARGEFILE) = 3
```

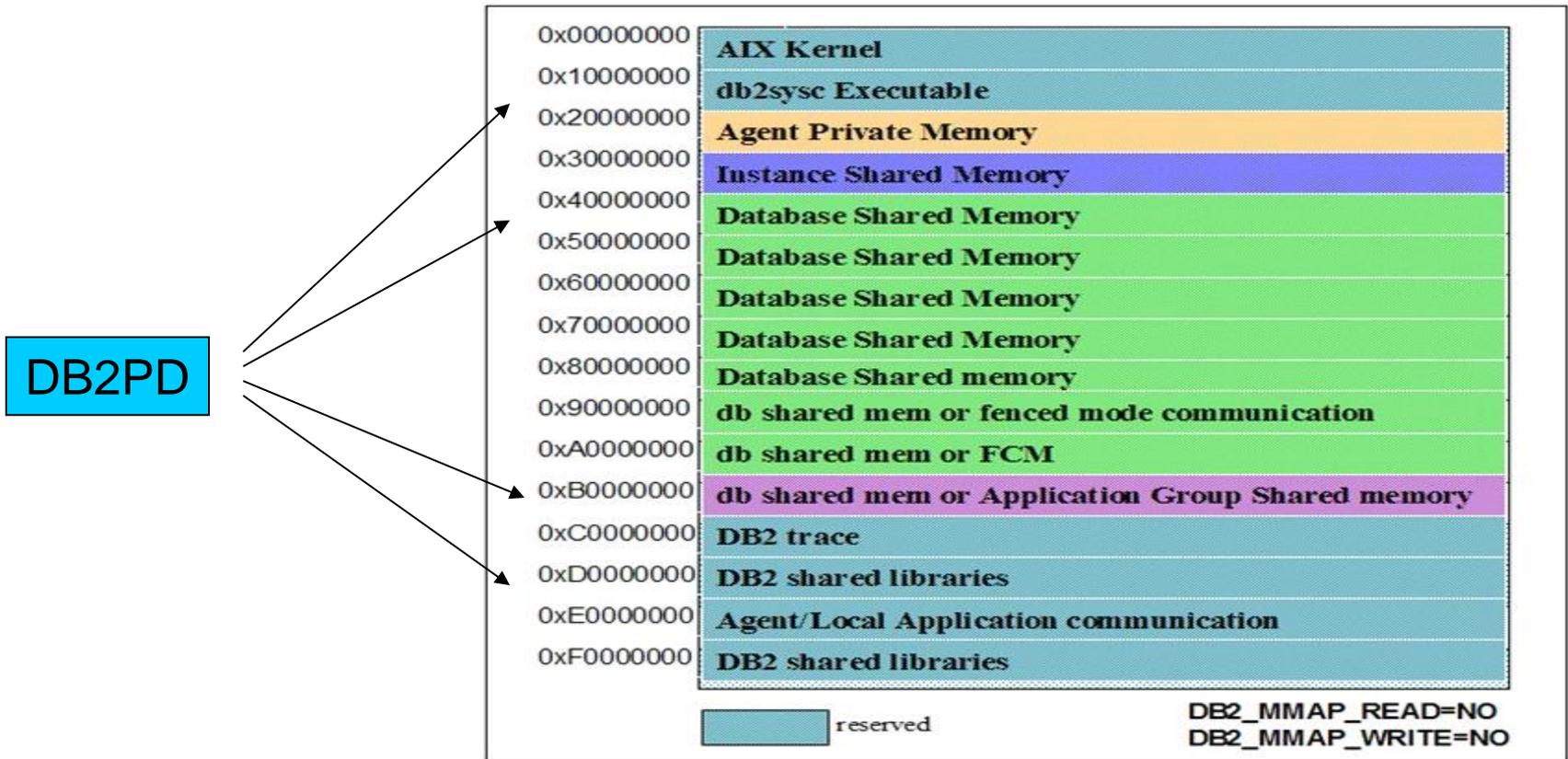


-locks showlocks option

Locks:

| Address | TranHdl | Lockname | Type | Mode | Sts | Owner | Dur | HldCnt | Att |
|--------------------------|---------|--|----------|------|-----|-------|-----|--------|--------|
| 0x0459C510 0x40000000 | 2 | 53514C4332453036BD4A32C841 Pkg UniqueID 434c5153 | Internal | P | ..S | G 2 | 1 | 0 | 0x0000 |
| Loading = 0 | | | | | | | | | |
| 0x0459CA10 0x40000000 | 3 | 53514C4332453036BD4A32C841 Pkg UniqueID 434c5153 | Internal | P | ..S | G 3 | 1 | 0 | 0x0000 |
| Loading = 0 | | | | | | | | | |
| 0x0459CA60 0x40000000 | 3 | 010000000100000001007B0056 Anchor 123 Stmt 1 Env 1 Var 1 | Internal | V | ..S | G 3 | 1 | 0 | 0x0000 |
| Loading 0 | | | | | | | | | |
| 0x0459C9E8 0x40000000 | 3 | 53514C4445464C5428DD630641 Pkg UniqueID 444c5153 | Internal | P | ..S | G 3 | 1 | 0 | 0x0000 |
| Loading = 0 | | | | | | | | | |
| 0x0459EF90 0x40000002 | 2 | 02000300270000000000000052 TbpaceID 2 TableID 3 RecordID 0x27 | Row | | ..X | G 2 | 1 | 0 | 0x0008 |
| 0x0459CAB0 0x00000001 | 3 | 02000300270000000000000052 TbpaceID 2 TableID 3 RecordID 0x27 | Row | | .NS | W 2 | 1 | 0 | 0x0000 |
| 0x0459C8F8 0x40000002 | 2 | 02000300000000000000000054 TbpaceID 2 TableID 3 | Table | | .IX | G 2 | 1 | 0 | 0x0000 |
| 0x0459CA88 0x00000001 | 3 | 02000300000000000000000054 TbpaceID 2 TableID 3 | Table | | .IS | G 3 | 1 | 0 | 0x0000 |

How DB2PD works ? (II)



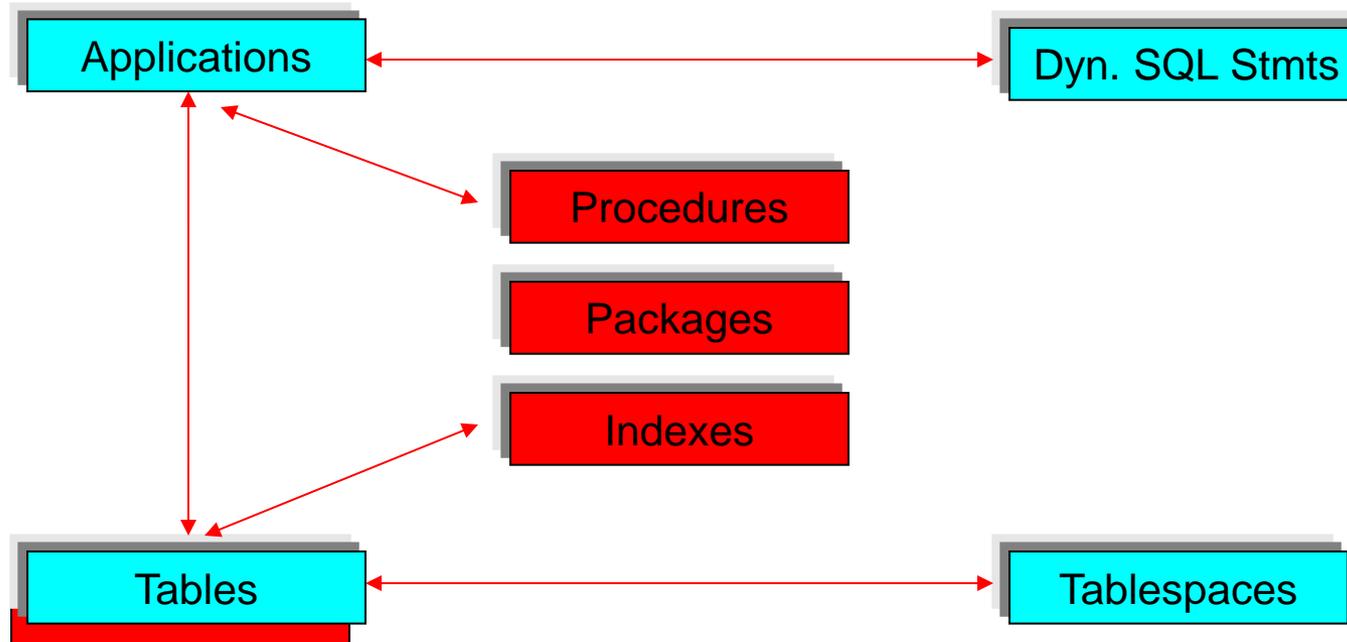
Why does Speedgain use DB2PD ?

- DB2PD provides excellent performance data
- DB2PD does not use any DB2 engine resources
- The new DB2PD performance data and the performance data from the snapshot API are merging together.
- The DB2PD performance data completes the existing performance data from the snapshot API and vice versa

Why does Speedgain use DB2PD ?

- Only DB2PD gathers performance values about index resources (the most important DB2 performance object).
- DB2PD provides the relationship between the monitor objects.
 - Applications <-> Dyn. SQL Stmt
 - Table <-> Applications
- DB2PD delivers detailed Information about Static SQL

SNAPSHOT Information + DB2PD Information



Speedgain & DB2PD

DB2 System Monitor - Speedgain for DB2

System Collector Statistics History Help

Dynamic SQL-Activity for database DB0001

Longest Execution Times (select first 30 rows)

| Snapshot Time | Average Stmt Execution Time | SQL Dynamic Stmt Text | Number Of ... |
|---------------|-----------------------------|--|---------------|
| 10:21:23 | 0,142351 | insert into db2monitor.tab_booking select * | 1 |
| 10:21:23 | 0,019622 | select * from | 0 |
| 10:21:23 | 0,018537 | update TAB0 | 0 |
| 10:21:23 | 0,004542 | set current s | 1 |
| 10:21:23 | 0,003948 | insert into db2monitor.tab0001 (sales_date, s... | 0 |
| 10:21:23 | 0,002525 | select count(*) from db2monitor.TAB0002 | 0 |
| 10:21:23 | 0,002391 | delete from db2monitor.tab0001 where sales... | 0 |
| 10:21:23 | 0,00234 | select count(*) from db2monitor.TAB0001 | 0 |
| 10:21:23 | 0,001111 | select count(*) from db2monitor.tab0001 | 1 |
| 10:21:23 | 0 | SET CURRENT LOCALE LC_CTYPE = 'de_DE' | 0 |
| 10:21:23 | 0 | select * from TAB0003 where sales_date = '... | 1 |

Most Executions (select first 30 rows)

Applications

Applications for Dynamic Statement

| Snapshot Time | Application Name | Authorization ID | User CPU Time | System CPU Time | Appl ID |
|---------------|------------------|------------------|---------------|-----------------|-------------------------------------|
| 10:21:23 | db2bp.exe | INST0001 | 0,09 | 0,52 | ITGAIN01 C0A84401.KA0E.00F9C1223912 |
| 10:10:38 | db2bp.exe | INST0001 | 0,21 | 1,49 | ITGAIN01 C0A84401.JC0E.00B241222204 |
| 09:58:22 | db2bp.exe | INST0001 | 0,2 | 1,27 | ITGAIN01 C0A84401.J50E.011CC1221405 |

Speedgain & DB2PD

Applications for Dynamic Statement

| Snapshot Time | Application Name | Authorization ID | User CPU Time | System CPU Time | Appl ID |
|---------------|------------------|------------------|---------------|-----------------|-------------------------------|
| 10:21:23 | db2bp.exe | INST0001 | 0,09 | 0,52 | ITGAIN01 C0A84401.KA0E.00F9C |
| 10:10:38 | db2bp.exe | INST0001 | 0,21 | 1,49 | ITGAIN01 C0A84401.JC0E.00B241 |
| 10:58:22 | db2bp.exe | INST0001 | 0,2 | 1,27 | ITGAIN01 C0A84401.J50E.011CC1 |

| | | | |
|-------------------------|-----------|-----------------|---|
| Rows Written | 52.000 | Sort Overflows | 0 |
| Rows Written Per Second | 14.049,73 | Total Sort Time | 0 |
| Stmt Sorts | 0 | | |

Last Snapshots

| Snapshot Time | SQL Dynamic Stmt Text | Last Executed | Average Stmt Execution Time [s] |
|---------------|---|-------------------|---------------------------------|
| 10:21:23 | insert into db2monitor.temp_booking ... | 2005-10-21 10:... | 0,142351 |
| 10:21:08 | insert into db2monitor.temp_booking ... | 2005-10-21 10:... | 0,13034 |
| 10:20:53 | insert into db2monitor.temp_booking ... | 2005-10-21 10:... | 0,13771 |
| 10:20:38 | insert into db2monitor.temp_booking ... | 2005-10-21 10:... | 0,154244 |
| 10:20:23 | insert into db2monitor.temp_booking ... | 2005-10-21 10:... | 0,159366 |
| 10:20:08 | insert into db2monitor.temp_booking ... | 2005-10-21 10:... | 0,133854 |

Conclusion

- **Successful system tuning requires knowledge of:**
 - DB2 processing
 - Available monitoring facilities
 - Instance Configuration Parameters
 - Database Configuration Parameters
 - Cause and Effect of parameters to processing
- **Speedgain for DB2 provides you with requisite knowledge to make informed tuning decisions**

Questions



Further information

U.S. Product Information and Sales

Stephen Rubin stephen.rubin@it-gain.com

978-774-8376 / 1-866-4ITGAIN

www.it-gain.com

Gunning Technology Solutions

Phil Gunning

610-451-5801

info@gunningts.com

www.gunningts.com/db2zone.htm