

A series of thin, horizontal grey lines that are staggered and recede into the distance, creating a sense of depth and movement from the center towards the right edge of the page.

SPEEDGAIN & DB2PD

- new features powered by a new informant -

A large, solid blue shape that curves from the bottom left towards the right, ending in a sharp, curved edge that follows the bottom of the page.

What's new in Speedgain LUW (DB2PD)

The new version of Speedgain LUW is now providing access to new important performance data for better analysis.

With the integration of **db2pd** data Speedgain's capability to pinpoint resource bottlenecks and other reasons for a poor performance is even more enhanced.

The new Version is gathering the information from db2pd and stores it into the Performance Database (PDB).

The information from **db2pd** and the information from the snapshot API are merged together. In many cases Speedgain uses the information from db2pd to complete the existing information from the snapshot API and vice versa.

What is important about this, is that db2pd does not use any DB2 engine resources. Although snapshots can be considered low overhead, they still incur 3-10% overhead in the DB2 engine depending on the number of snapshots being taken and the frequency of snapshots.

Speedgain for DB2 is the first commercial product to take effect of db2pd lightweight monitoring and offers tight db2pd integration with the Speedgain for DB2 graphical user interface (GUI).

Itgain is excited about this new capability and will continue to enhance Itgain products to integrate with db2pd as IBM adds new functionality to db2pd

With the **db2pd** information Speedgain can now show the relationships between the monitored objects. The usual way to find the performance problems is to drill down from one information deeper to the next information.

To show the bottleneck of performance a good monitoring tool has to answer the following questions:

- Which application has used a specific SQL Statement ?
- Which SQL-Statements were used by a specific application ?
- Which applications have used a specific table
- And so on

An Overview about the new features:

1) Performance values about the index resources.

- ➔ No other tool shows information concerning indexes (the most important db2 performance object)

2) Relationships between the dynamic statements from the package cache and the applications, which are using these statements.

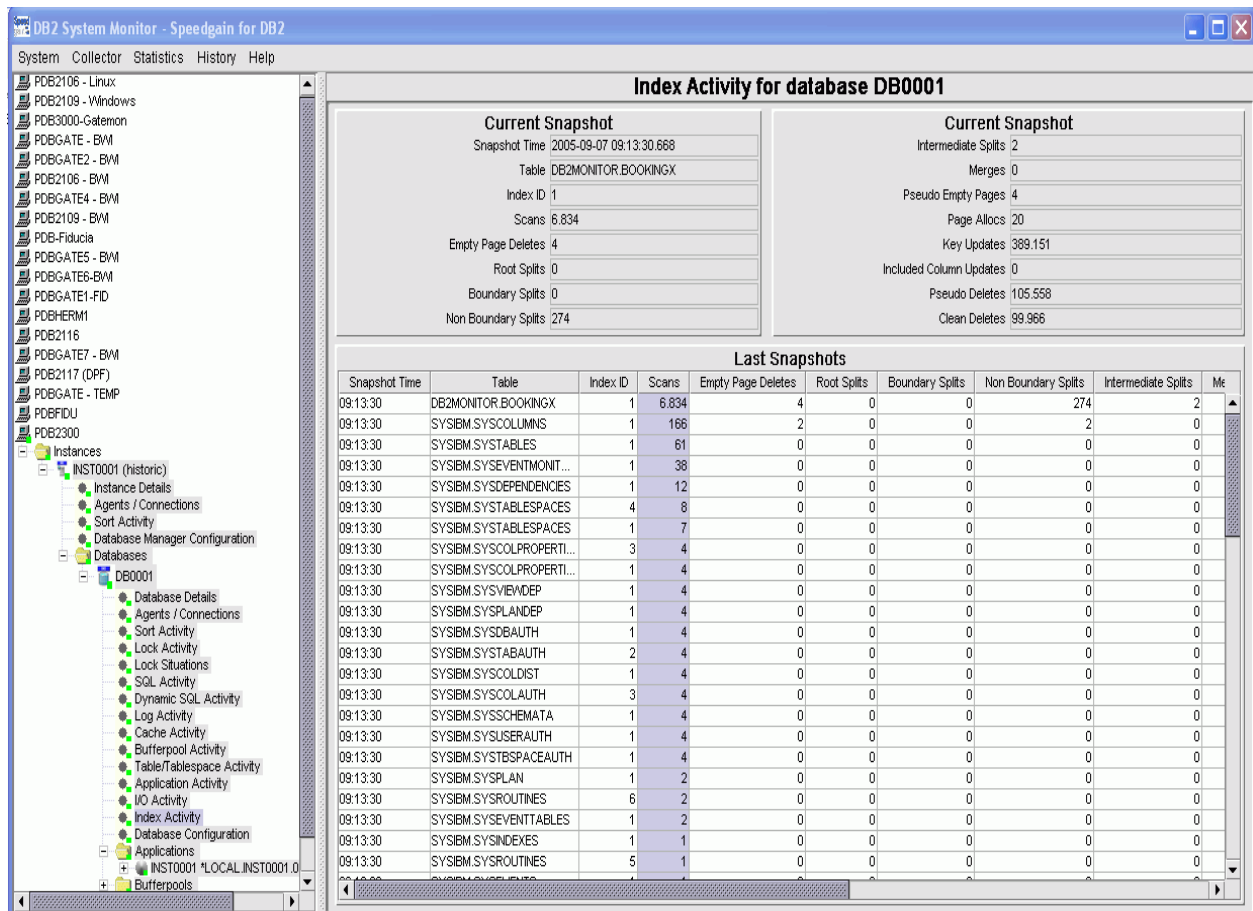
- ➔ Now, with the new version of Speedgain we have the possibility to show almost the whole SQL Workload of an application without starting expensive application traces.

3) Relationships between tables and applications.

- ➔ The new version stores the information about the last application which was using the table

The new features in detail

1) Information about the indexes



Index Activity for database DB0001

Current Snapshot

Snapshot Time: 2005-09-07 09:13:30.668

Table: DB2MONITOR.BOOKINGX

Index ID: 1

Scans: 6.834

Empty Page Deletes: 4

Root Splits: 0

Boundary Splits: 0

Non Boundary Splits: 274

Current Snapshot

Intermediate Splits: 2

Merges: 0

Pseudo Empty Pages: 4

Page Allocs: 20

Key Updates: 389.151

Included Column Updates: 0

Pseudo Deletes: 105.558

Clean Deletes: 99.966

Last Snapshots

Snapshot Time	Table	Index ID	Scans	Empty Page Deletes	Root Splits	Boundary Splits	Non Boundary Splits	Intermediate Splits	Me
09:13:30	DB2MONITOR.BOOKINGX	1	6.834	4	0	0	274	2	
09:13:30	SYSIBM.SYSCOLUMNS	1	166	2	0	0	2	0	
09:13:30	SYSIBM.SYSTABLES	1	61	0	0	0	0	0	
09:13:30	SYSIBM.SYSEVENTMONIT...	1	38	0	0	0	0	0	
09:13:30	SYSIBM.SYSDependencies	1	12	0	0	0	0	0	
09:13:30	SYSIBM.SYSTABLESPACES	4	8	0	0	0	0	0	
09:13:30	SYSIBM.SYSTABLESPACES	1	7	0	0	0	0	0	
09:13:30	SYSIBM.SYSCOLPROPERTI...	3	4	0	0	0	0	0	
09:13:30	SYSIBM.SYSCOLPROPERTI...	1	4	0	0	0	0	0	
09:13:30	SYSIBM.SYSVIEWDEP	1	4	0	0	0	0	0	
09:13:30	SYSIBM.SYSPLANDEP	1	4	0	0	0	0	0	
09:13:30	SYSIBM.SYSDBAUTH	1	4	0	0	0	0	0	
09:13:30	SYSIBM.SYSTABAUTH	2	4	0	0	0	0	0	
09:13:30	SYSIBM.SYSCOLDIST	1	4	0	0	0	0	0	
09:13:30	SYSIBM.SYSCOLAUTH	3	4	0	0	0	0	0	
09:13:30	SYSIBM.SYSSCHEMATA	1	4	0	0	0	0	0	
09:13:30	SYSIBM.SYSUSERAUTH	1	4	0	0	0	0	0	
09:13:30	SYSIBM.SYSTBSPACEAUTH	1	4	0	0	0	0	0	
09:13:30	SYSIBM.SYSPLAN	1	2	0	0	0	0	0	
09:13:30	SYSIBM.SYSROUTINES	6	2	0	0	0	0	0	
09:13:30	SYSIBM.SYSEVENTTABLES	1	2	0	0	0	0	0	
09:13:30	SYSIBM.SYSINDEXES	1	1	0	0	0	0	0	
09:13:30	SYSIBM.SYSROUTINES	5	1	0	0	0	0	0	

- Scans

This value shows the number of usage of an index. There is no difference between matching and non-matching access. If a index is used twice in one sql statement there are also two scans counted. This is a good way of identifying your most active and used indexes

- Empty Page Deletes

The number of empty leaf nodes that were deleted. This is an indicator that there are deletes and/or updates of the key columns

- Root Splits

The number of key insert or update operations that caused the index tree depth to increase. If the index level are increasing the search to locate one data row needs one more getpage request. Root splits are especially costly as the entire index RID structure has to be rebuilt. They should be avoided.

- Boundary Splits

The number of boundary leaf splits that result in an insert operation into either the lowest or the highest key.

- Intermediate Splits

The number of intermediate level splits.

- Page Allocs

The number of page allocs. This is an indicator for a lot of inserts on the table or/and updates on the key columns.

- Key Updates

The number of updates to the key. A lot of updates to the key columns are bad for the performance because:

- root splits
- page allocs
- pseudo empty pages
- pseudo deletes
- boundary splits

can be occurred.

To decrease the impact of the performance an reorg on the index have to be done.

- Pseudo Deletes

The number of keys that are marked as pseudo deleted. The index type 2 are using pseudo deletes to reduce the locking contentions. Pseudo deleted entries are cleaned up in the background during a period of idle activity. However, if there isn't a period of idle activity, then use this metric to know when to schedule the recorchk utility to cleanup pseudo deleted entries.

- Clean Deletes

The number of pseudo deleted keys that have been deleted.

If the difference between Pseudo Deletes and Clean Deletes are too high you have to run the reorg utility with the "cleanup only" option.

2) Relationships between the dynamic statements and applications

For the dynamic statements of the package cache Speedgain can reference the applications.

The screenshot displays the 'Dynamic SQL-Activity for database DB0001' window. The left sidebar shows a tree view with 'Dynamic SQL Activity' selected. The main area contains two tables:

Longest Execution Times

Snapshot Time	Elapsed Stmt Execution Time [s]	Has Applications	Executed Since	SQL Dynamic Stmt Text	Average
13:58:14	3,884145	No	2005-09-07 13:5...	drop table db2monitor.stmt_SPG24	
13:58:14	12,809312	No	2005-09-07 13:5...	SELECT line FROM TABLE (db2monitor.db2pd_read_table (CAST(? AS VA...	
13:58:14	0,33377	No	2005-09-07 13:5...	create event monitor SPG24 for connections where appl_id like "LOCAL.in...	
13:58:14	0,127446	No	2005-09-07 13:5...	drop table db2monitor.conn_SPG24	
13:58:14	0,305284	Yes	2005-09-07 13:5...	select count(*) from db2monitor.TAB0001	
13:58:14	0,063214	No	2005-09-07 13:5...	drop table db2monitor.conn_SPG24	
13:58:14	0,0521	No	2005-09-07 13:5...	drop table db2monitor.conn_SPG24	
13:58:14	0,050353	No	2005-09-07 13:5...	set event monitor SPG24 state U	
13:58:14	0,032041	No	2005-09-07 13:5...	set event monitor SPG24 state U	
13:58:14	0,070566	Yes	2005-09-07 13:5...	select count(*) from db2monitor.TAB0002	
13:58:14	0,600294	No	2005-09-07 13:5...	flush event monitor SPG24	
13:58:14	0,010467	No	2005-09-07 13:5...	drop event monitor SPG24	
13:58:14	0,003048	No	2005-09-07 13:5...	select 1 from system.systablespace where tbspace = 'SPGAPPLTRC'	
13:58:14	0	No	2005-09-07 13:5...	CREATE TABLE "DB2MONITOR"."CONN_SPG24" (ACC_CURS_BLK BIGINT ...	
13:58:14	0	No	2005-09-07 13:5...	CREATE TABLE "DB2MONITOR"."CONTROL_SPG24" (EVENT_MONITOR_N...	

Most Executions

Snapshot Time	Stmt Executions	Has Applications	Executed Since	SQL Dynamic Stmt Text	Elapsed Stmt Execution Time [s]	Average Stmt Execution Time [s]
13:58:14	30	No	2005-09-07 13:5...	flush event monitor SPG24	0,600294	0,020009
13:58:14	10	No	2005-09-07 13:5...	SELECT line FROM TABL...	12,809312	1,280931
13:58:14	3	Yes	2005-09-07 13:5...	select count(*) from db2...	0,305284	0,101761
13:58:14	3	Yes	2005-09-07 13:5...	select count(*) from db2...	0,070566	0,023522
13:58:14	1	No	2005-09-07 13:5...	CREATE TABLE "DB2MO...	0	0
13:58:14	1	No	2005-09-07 13:5...	CREATE TABLE "DB2MO...	0	0
13:58:14	1	No	2005-09-07 13:5...	select 1 from system.sys...	0,003048	0,003048
13:58:14	1	No	2005-09-07 13:5...	CREATE TABLE "DB2MO...	0	0
13:58:14	1	No	2005-09-07 13:5...	CREATE TABLE "DB2MO...	0	0
13:58:14	1	No	2005-09-07 13:5...	CREATE TABLE "DB2MO...	0	0
13:58:14	1	No	2005-09-07 13:5...	create event monitor SP...	0,33377	0,33377
13:58:14	1	No	2005-09-07 13:5...	set event monitor SPG24...	0,032041	0,032041
13:58:14	1	No	2005-09-07 13:5...	drop table db2monitor.st...	3,884145	3,884145
13:58:14	1	No	2005-09-07 13:5...	drop event monitor SPG24	0,010467	0,010467
13:58:14	1	No	2005-09-07 13:5...	drop table db2monitor.co...	0,0521	0,0521
13:58:14	1	No	2005-09-07 13:5...	set event monitor SPG24...	0,050353	0,050353

For the listed SQL Statements Speedgain identifies applications which were using this statement. These most often used statements can then be targeted for analysis and tuning using Speedgain SQL Workbench

SPEEDGAIN & DB2PD

The list of applications using this statement (order by using). This can be used to identify the most often used applications and statements.

Dynamic SQL-Activity for database DB0001

Snapshot Time	Elapsed Stmt Execution Time [s]	Has Applications	Executed Since	SQL Dynamic Stmt Text	Average S
13:58:14	3,884118	No	2005-09-07 13:5...	drop table db2monitor.stmt_SPG24	
13:58:14	12,809312	No	2005-09-07 13:5...	SELECT line FROM TABLE (db2monitor.db2pd_read_table (CAST(?) AS VA...	
13:58:14	0,33377	No	2005-09-07 13:5...	create event monitor SPG24 for connections where appl_id like 'LOCAL.in...	
13:58:14	0,127448	No	2005-09-07 13:5...	drop table db2monitor.conn_SPG24	
13:58:14	0,305284	Yes	2005-09-07 13:5...	select count(*) from db2monitor.TAB0001	
13:58:14	0,063214	No	2005-09-07 13:5...	drop table db2monitor.control_SPG24	

Applications for Dynamic Statement

Snapshot Time	Application Name	Authorization ID	User CPU Time	System CPU Time	Appl ID
13:58:14	akt01a.exe	INST0001	0	0,14 ITGAIN01	8740
13:58:14	akt01a.exe	INST0001	0	0,03 ITGAIN01	742
13:58:14	akt01a.exe	INST0001	0	0,05 ITGAIN01	750
13:57:14	akt01a.exe	INST0001	0	0,14 ITGAIN01	COA84401.HF0D.017F02123740
13:57:14	akt01a.exe	INST0001	0	0,03 ITGAIN01	COA84401.I30D.00D002123742
13:57:14	akt01a.exe	INST0001	0	0,05 ITGAIN01	COA84401.H40D.00D702123750
07:52:37	akt01a.exe	INST0001	0	0,02 ITGAIN01	COA84401.M40C.010141231941
07:52:37	akt01a.exe	INST0001	0,01	0,01 ITGAIN01	COA84401.M50C.0140C1231943
07:51:37	akt01a.exe	INST0001	0	0,02 ITGAIN01	COA84401.M40C.010141231941
07:51:37	akt01a.exe	INST0001	0,01	0,01 ITGAIN01	COA84401.M50C.0140C1231943
07:51:37	akt01a.exe	INST0001	0	0,01 ITGAIN01	COA84401.M60C.014401231945
07:43:37	akt01a.exe	INST0001	0	0,02 ITGAIN01	COA84401.LC0C.010601231632
07:43:37	akt01a.exe	INST0001	0	0,01 ITGAIN01	COA84401.LF0C.013501231638
07:43:37	akt01a.exe	INST0001	0	0,07 ITGAIN01	COA84401.M00C.00FF01231640
07:42:37	akt01a.exe	INST0001	0	0,02 ITGAIN01	COA84401.LC0C.010601231632
07:42:37	akt01a.exe	INST0001	0	0,01 ITGAIN01	COA84401.LF0C.013501231638
07:42:37	akt01a.exe	INST0001	0	0,07 ITGAIN01	COA84401.M00C.00FF01231640
04:15:40	akt01a.exe	INST0001	0	0,01 ITGAIN01	COA84401.KE0C.019281220035
04:15:40	akt01a.exe	INST0001	0	0,02 ITGAIN01	COA84401.L20C.011541220043
04:15:40	akt01a.exe	INST0001	0,01	0,02 ITGAIN01	COA84401.KF0C.013C81220037
04:14:40	akt01a.exe	INST0001	0	0,01 ITGAIN01	COA84401.KE0C.019281220035
04:14:40	akt01a.exe	INST0001	0	0,02 ITGAIN01	COA84401.L20C.011541220043
04:14:40	akt01a.exe	INST0001	0,01	0,02 ITGAIN01	COA84401.KF0C.013C81220037
03:49:41	akt01a.exe	INST0001	0,01	0,02 ITGAIN01	COA84401.KD0C.0165C1213444

Detail Information for the application

Details for application ITGAIN01.COA84401.HF0D.017F02123740

Current Snapshot

Snapshot Time: 2005-09-07 13:58:14.279
 Application Handle: 74
 Application Status: UOW waiting
 ID Of Code Page Used By Appl: 850
 Number Of Associated Agents: 1
 Coordinating Node: 0
 Authorization Level: 18,292,030
 Client Process ID: 3944
 Coordinator Agent: 26352
 Application Status Change Time: 2005-09-07 13:56:35.0
 Client Operating Platform: NT
 Client Communication Protocol: TCP/IP
 Database Country Code: 49
 Application Name: akt01a.exe
 Application ID: COA84401.HF0D.017F02123740
 Sequence Number: 0002
 Authorization ID: INST0001

Current Snapshot

Configuration Name Of Client: ITGAINSPED02
 Client Product Version Of: SQL08020
 Input Database Alias: DB0001
 Database Alias Used By Appl: DB0001
 Database Name: DB0001
 Database Path: /ib2/inst0001/NODE0000/SQL0000
 User Login ID: ITGAIN01
 DRDA Correlation Token: COA84401.HF0D.017F021237
 TP Mon Client User ID:
 TP Mon Client Workstation Name:
 TP Mon Client Application Name:
 TP Mon Client Account String:
 Time Waited For Prefetch [ms]: 0
 Application Idle Time [s]: 0
 Last Reset Timestamp: 1970-01-01 02:00:00.0
 Statement Exec Elapsed Time: 1970-01-01 02:00:00.0
 Inbound Communication Address: 192.168.68.1.7949

Last Snapshots

Snapshot Time	Application Handle	Application Status	ID Of Code Page Used By Appl	Number Of Associated Agents	Coordinating Node	Authorization Level	Cle
13:58:14	74	UOW waiting	850	1	0	18,292,030	3944
13:57:14	74	UOW waiting	850	1	0	18,292,030	3944

Relationship between Application and Statement (Last Statement execution)

SQL-Activity for application ITGAIN01.C0A84401.HF0D.017F02123740

Current Snapshot
 Snapshot Time: 2005-09-07 13:57:14.258
 Statement (V 8.2): select count(*) from db2monito

Current Snapshot
 Internal Rollbacks Deadlock: 0
 Rows Read: 43
 Rows Written: 0
 Rows Deleted: 0
 Rows Inserted: 0
 Rows Updated: 0
 Rows Selected: 1
 Rows Selected Per SQL: 1
 Rows Read Per Rows Selected: 43
 Blinds Precompiles Attempted: 0
 Open Local Cursors: 0
 Open Local Cursors Blocking: 0
 Perc Local Cursors Blocking: 0
 Open Remote Cursors: 0
 Open Remote Cursors Blocking: 0
 Perc Remote Cursors Blocking: 0
 Block Cursor Requests: 1
 Accepted Block Cursor Reqs: 1
 Rejected Block Cursor Reqs: 0
 Perc Rej Block Cursor Reqs: 0

Last Snapshots

Snapshot Time	Statement (V 8.2)	Total SQL Strmts	Dynamic SQL Strmts Attempted	Static SQL Strmts Attempted	Failed Strmts Operations	Selec
13:58:14		7	6	1	0	
13:57:14	select count(*) from db2monitor.TAB0001	7	6	1	0	

Further information about Speedgain

Europe

phone +49 511 9666 817
fax +49 511 9666 701
http www.itgain.de/en/speedgain.html
eMail speedgain@itgain.de

United States

phone 1-800 618 1686
fax (404) 586-6820
http www.itgain.de/en/speedgain.html
eMail speedgain@itgain.de

itgain Integration Solution GmbH
Vahrenwalder Str. 269A
30179 Hannover
Telefon +49 511 9666 - 817
Telefax +49 511 9666 - 701
eMail info@itgain.de
