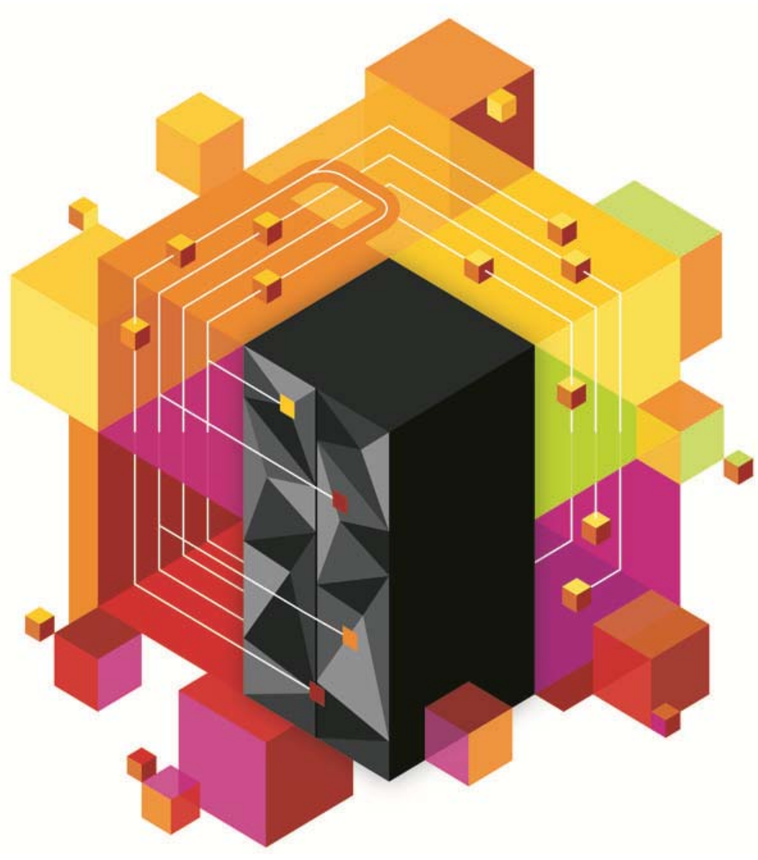
Stored Procedure Monitoring and Analysis



Presenter – Title

Date

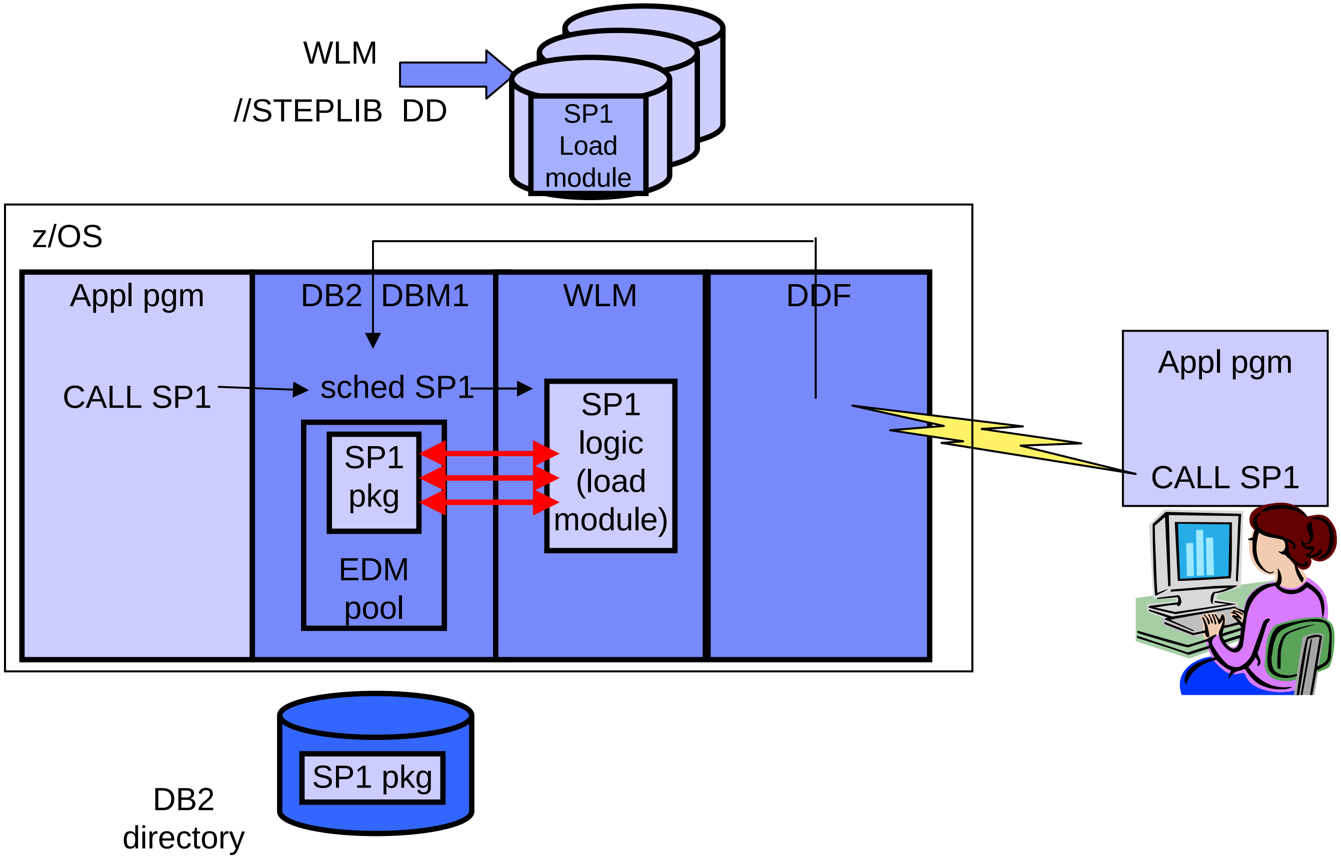
Agenda

- **What are stored procedures?**
- **Benefits of stored procedures**
- **Stored procedure analysis – Issues and solutions**
- **Monitoring stored procedures using OMEGAMON DB2 Performance Expert**
- **Tuning stored procedures and optimizing application configuration**

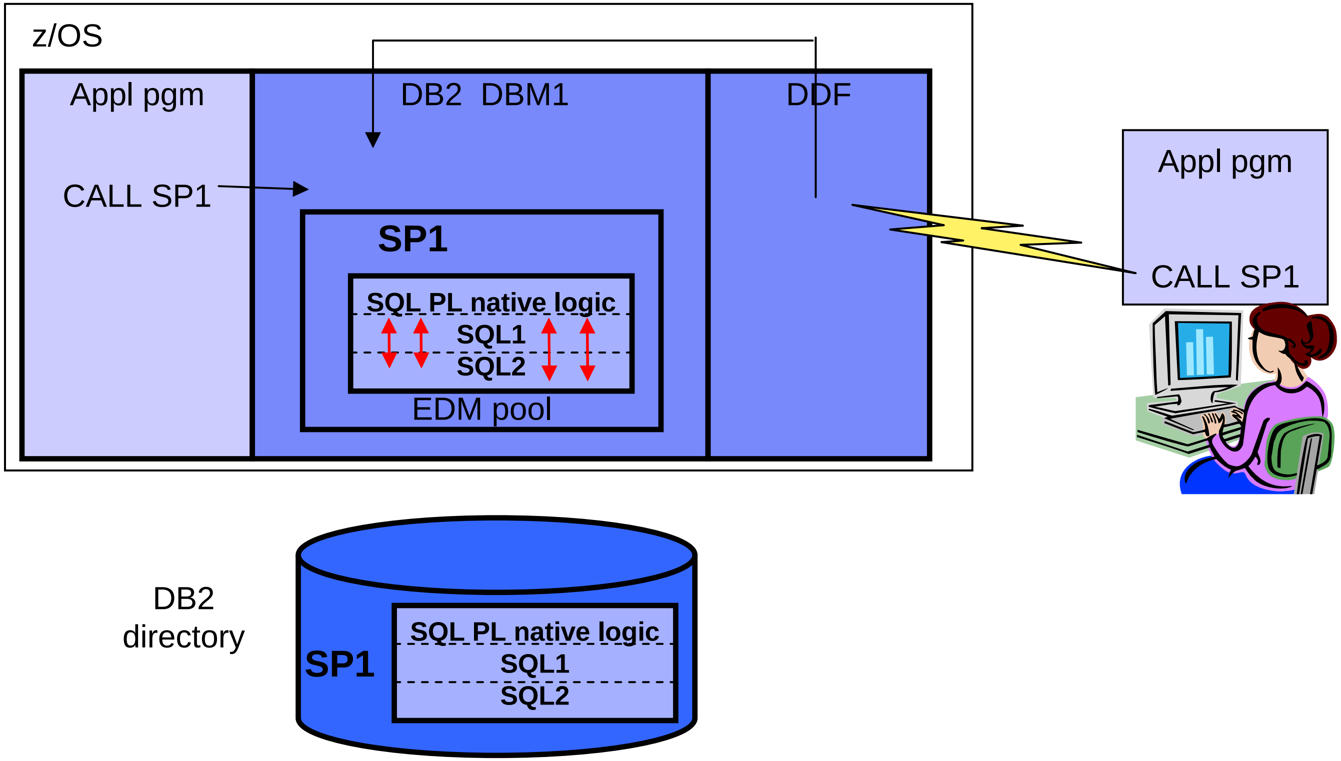
What are Stored Procedures?

- **A stored procedure is a user-written program that can be called by an application with an SQL CALL statement.**
- **It is a compiled program that is stored at a DB2 server**
- **It can execute business logic and SQL statements**
- **Stored procedure types**
  - External high level language procedures COBOL, PL/I, C, C++, Assembler, REXX, and Java
  - External SQL procedures
  - Native SQL procedures introduced by DB2 9 for z/OS

DB2 z/OS Stored Procedure Processing (External)



Native SQL Procedure Processing (Internal)



Programming Benefits of Stored Procedures

- **Modularity in application development**
- **Data will be processed always in a consistent way according to the rules defined in the stored procedure**
- **Enforcement of business rules**
  - You can use stored procedures to define business rules that are common to several applications.
  - can be an alternative to using constraints and triggers.
- **Improved application security**
  - Sensitive business logic runs on the DB2 server
  - End users are authorized to execute a stored procedure,
    they do not need table privilege -> similar to static authorization model
- **Application integration solutions**
  - can access non-DB2 resources
    e.g. VSAM files, MQ queues, IMS or CICS transactions
  - Stored procedures can have access to commands that run only on the server.

Total Cost of Ownership Benefits of Stored Procedures

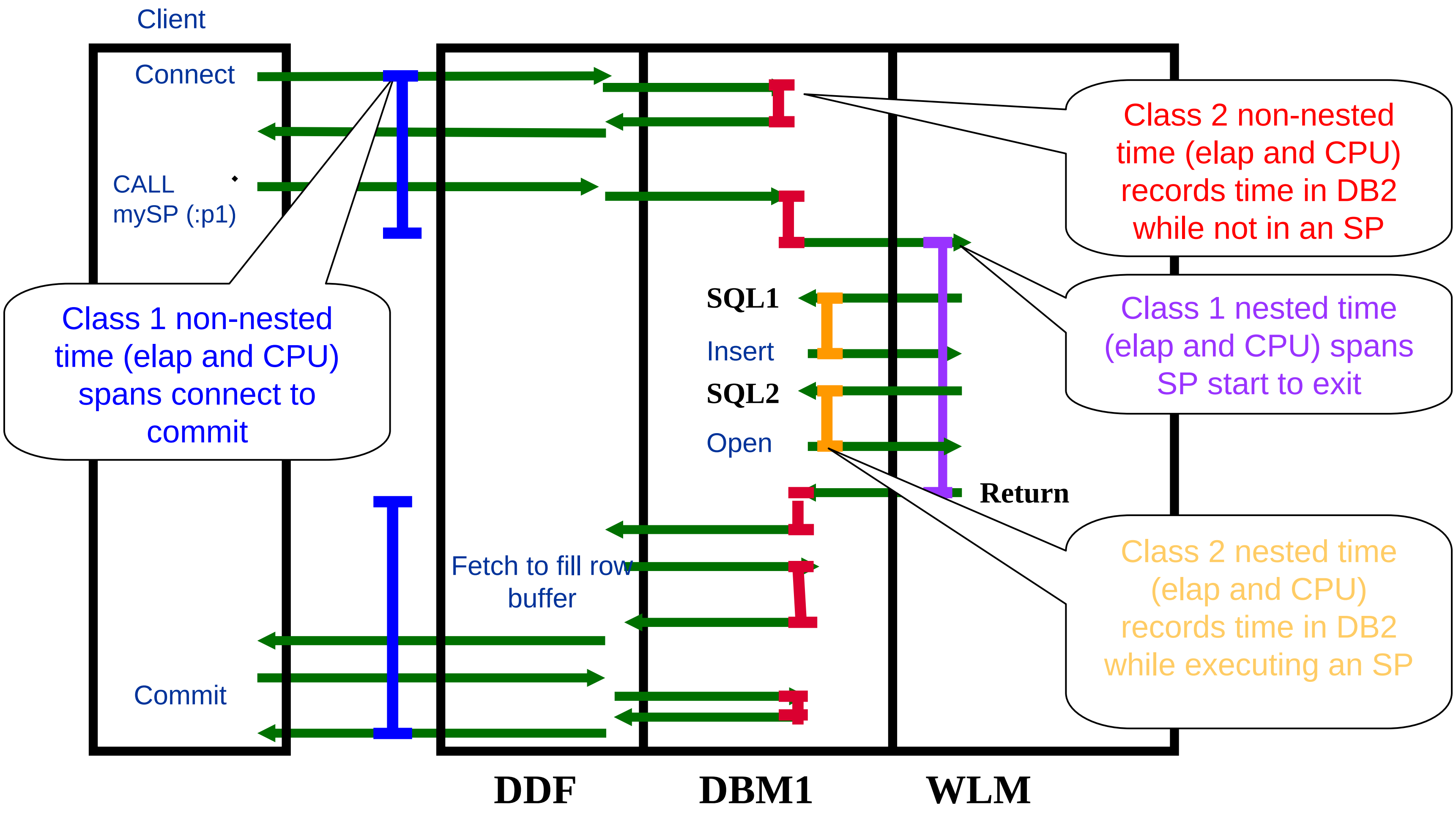
- **Reduced network traffic for distributed applications**
  - Grouping SQL statements into a stored procedure results in two trips across the network for each group of statement, resulting in better performance for applications
- **Cost of ownership reduction**
  - If stored procedure is called from distributed client via DRDA, a portion is eligible for zIIP redirect.
    - Including: Call statement processing; Result set processing; Commit processing
  - Stored procedures written in Java can take advantage of zAAP engines
  - Native SQL procedures run as enclave SRB in DBM1 address space and the SP execution itself is zIIP off-loadable with DB2 9 for z/OS.
  - For WLM managed stored procedures:
    - SQL processing runs under a TCB hence not eligible for zIIP redirect
- **As of now, there is NO performance benefit for calling a SP from a local application**

SP Language / API CPU Cost comparison - Update

- **IRWW workload (OLTP workload consisting of 7 transactions)**
- **Called from distributed JCC type 4 client**

Language/API	Base CPU/Tran Cost	Billable CPU/Tran Cost after zIIP and/or zAAP redirect
COBOL Stored Proc	**1X (BASE)**	0.80x (Some zIIP)
C Stored Proc	1.02x	0.82x (Some zIIP)
SQLJ Stored Proc	2.01x	1.11x (zAAP+ some zIIP)
JDBC Stored Proc	2.97x	1.84x (zAAP+ some zIIP)
Native SQL Stored Proc	**1.09x**	**0.59x (Significant zIIP)**

Performance Reporting – External Stored Procedure



Class 2 non-nested time (elap and CPU) records time in DB2 while not in an SP

Class 1 nested time (elap and CPU) spans SP start to exit

Class 1 non-nested time (elap and CPU) spans connect to commit

Class 2 nested time (elap and CPU) records time in DB2 while executing an SP

External SP Performance Summary - Plan-Level

- DB2 Accounting class 1 and 2 needed (3 is recommended)

```
AVERAGE       APPL(CL.1)  DB2 (CL.2)
------------  ----------  ----------
ELAPSED TIME    0.003212    0.002575
 NONNESTED      0.000714    0.000694
 STORED PROC    0.002498    0.001881
 UDF            0.000000    0.000000
 TRIGGER        0.000000    0.000000

CP CPU TIME     0.000715    0.000654
 AGENT          0.000715    0.000654
  NONNESTED     0.000149    0.000129
  STORED PRC    0.000567    0.000525
  UDF           0.000000    0.000000
  TRIGGER       0.000000    0.000000
 PAR.TASKS      0.000000    0.000000
```

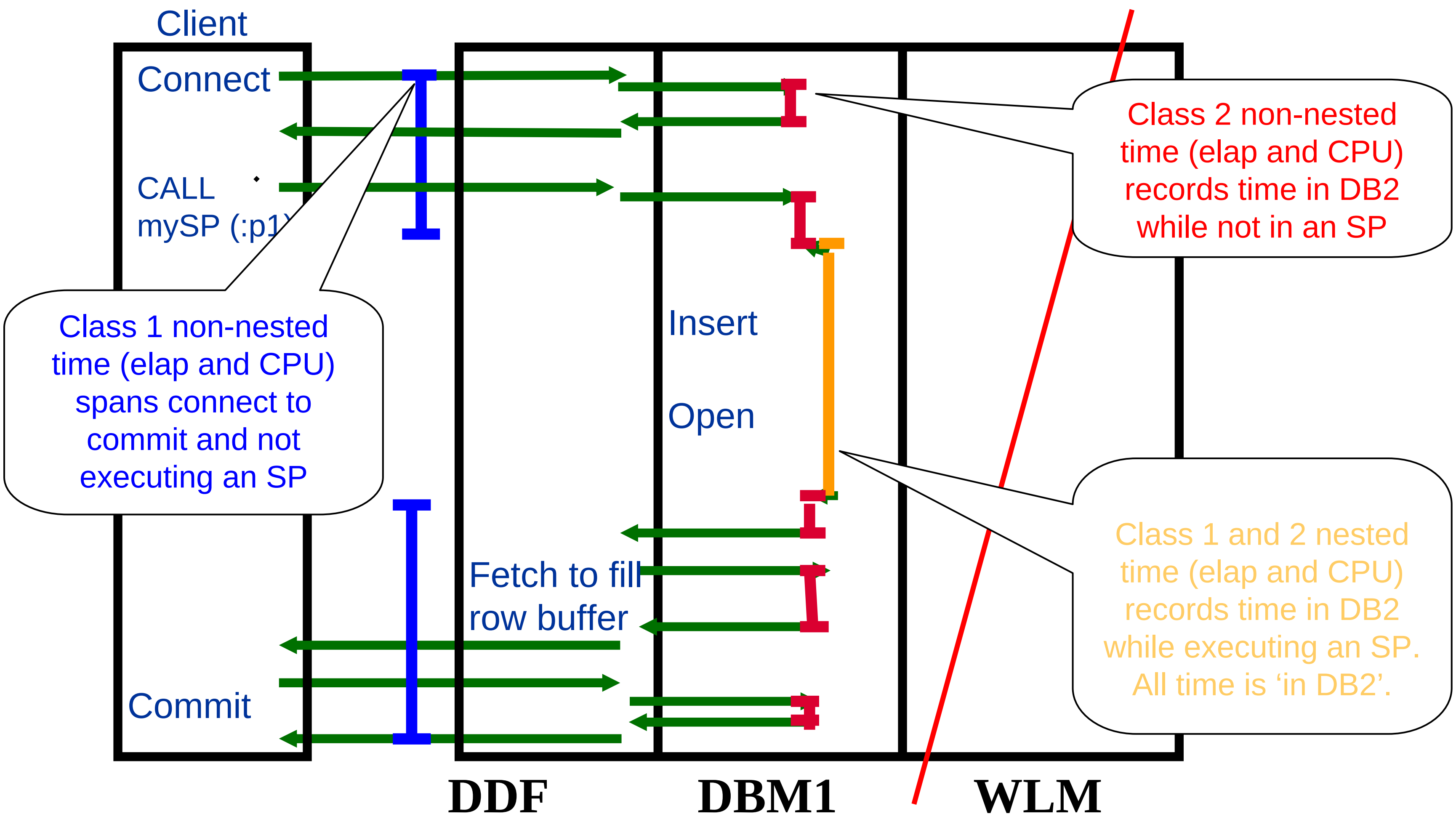
Class 1 non-nested time (ET & CPU)

Class 2 non-nested time (ET & CPU)

Class 1 nested time (ET & CPU)

Class 2 nested time (ET & CPU)

Performance Reporting – Native SQL Stored Procedure



Class 1 non-nested time (elap and CPU) spans connect to commit and not executing an SP

Class 2 non-nested time (elap and CPU) records time in DB2 while not in an SP

Class 1 and 2 nested time (elap and CPU) records time in DB2 while executing an SP. All time is ‘in DB2’.

Native SQL SP Performance Summary - Plan-Level

- DB2 Accounting class 1 and 2 needed (3 is recommended)

```
AVERAGE          APPL(CL.1)  DB2 (CL.2)
------------     ----------  ----------
ELAPSED TIME       0.004834    0.002789
 NONNESTED         0.002819    0.000774
 STORED PROC       0.002015    0.002015
 UDF               0.000000    0.000000
 TRIGGER           0.000000    0.000000

CP CPU TIME        0.000963    0.000909
 AGENT             0.000963    0.000909
  NONNESTED        0.000198    0.000143
  STORED PRC       0.000765    0.000765
  UDF              0.000000    0.000000
  TRIGGER          0.000000    0.000000
 PAR.TASKS         0.000000    0.000000
```

CL1 and CL2 will always be equal!

CL1 and CL2 will always be equal!

SP Detail Reporting - Package level Reporting

- **Accounting class 7 and/or 8 needed**

- SYSSTAT package contains time for CALL statement, result set processing, SET special registers, and VALUES statements for LOB handling

```
SYSSTAT              VALUE                 SYSSTAT                     TIMES
-------------------  -------------------   ------------------   ------------
TYPE                 PACKAGE               ELAP-CL7 TIME-AVG        0.000387
                                           CP CPU TIME              0.000072
LOCATION             DSND91B                 AGENT                  0.000072
COLLECTION ID        NULLID                  PAR.TASKS              0.000000
PROGRAM NAME         SYSSTAT               SE CPU TIME              0.000000

NSQLNEW              VALUE                 NSQLNEW                     TIMES
-------------------  -------------------   ------------------   ------------
TYPE                 PACKAGE               ELAP-CL7 TIME-AVG        0.004751
                                           CP CPU TIME              0.001667
LOCATION             DSND91B                 AGENT                  0.001667
COLLECTION ID        USRT001                 PAR.TASKS              0.000000
PROGRAM NAME         NSQLNEW               SE CPU TIME              0.000000
```

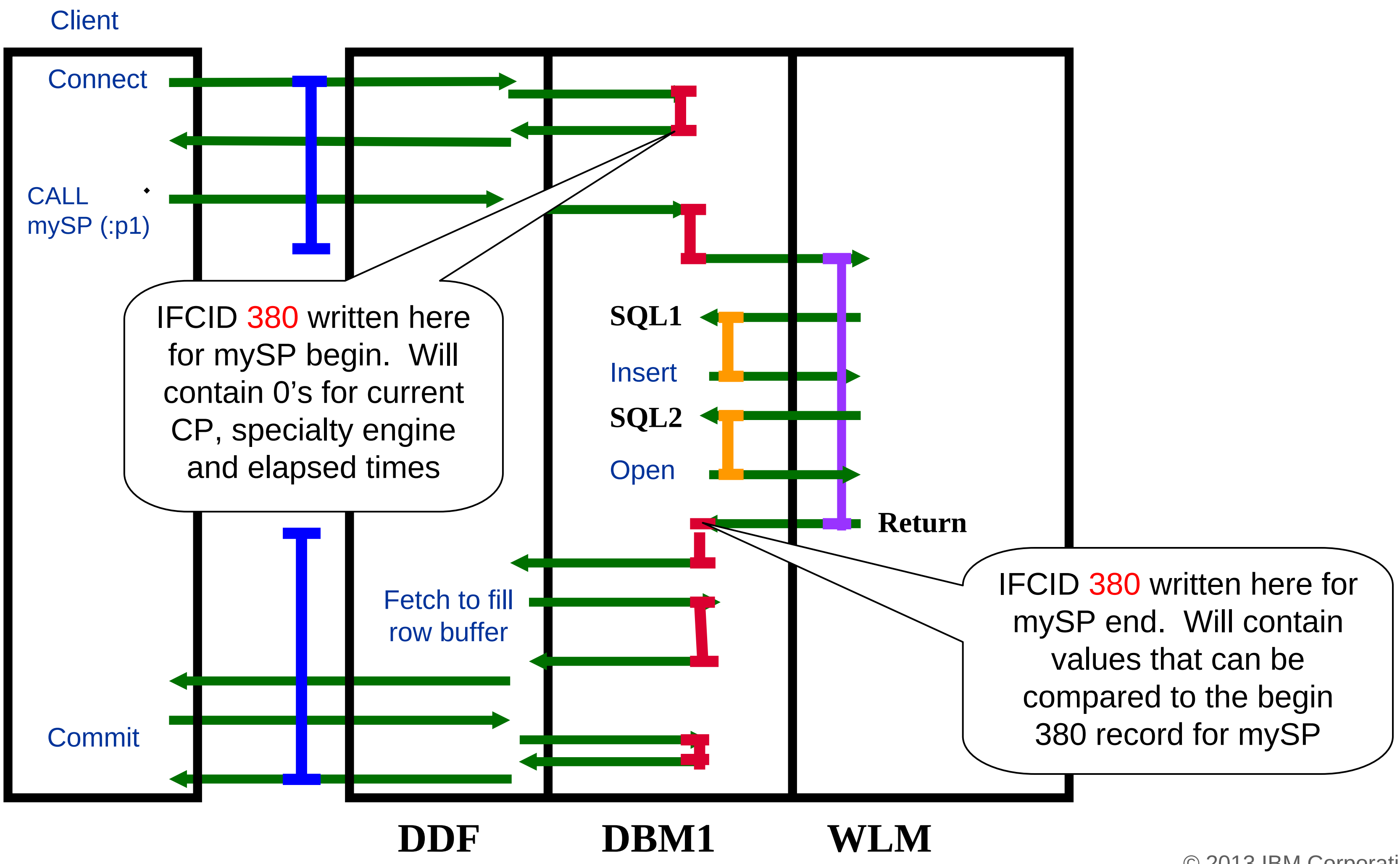
Issues with Plan and Package Level SP Analysis

- Multiple SPs called in a transaction are summed at the plan level. By definition this affects the analysis of nested SPs.
- Package level analysis can be difficult if an SP execute different paths and SQL based on parameters. How do you differentiate between the invocations?
- Package level analysis does not apply to SPs that do not execute SQL

Enhanced Instrumentation for Stored Procedure Performance Analysis

- **PM53243 (DB2 10) New IFCIDs 380 and 381 are created for Stored Procedure and User-Defined Function detail respectively. These records:**
  - Identify the stored procedure or UDF beginning or ending
  - Include the current CP, specialty engine, and elapsed time details for nested activity
- **These record can be used to determine the CP, specialty engine, and elapsed time for a given SP or UDF invocation**

Enhanced Instrumentation for Stored Procedure Performance Analysis



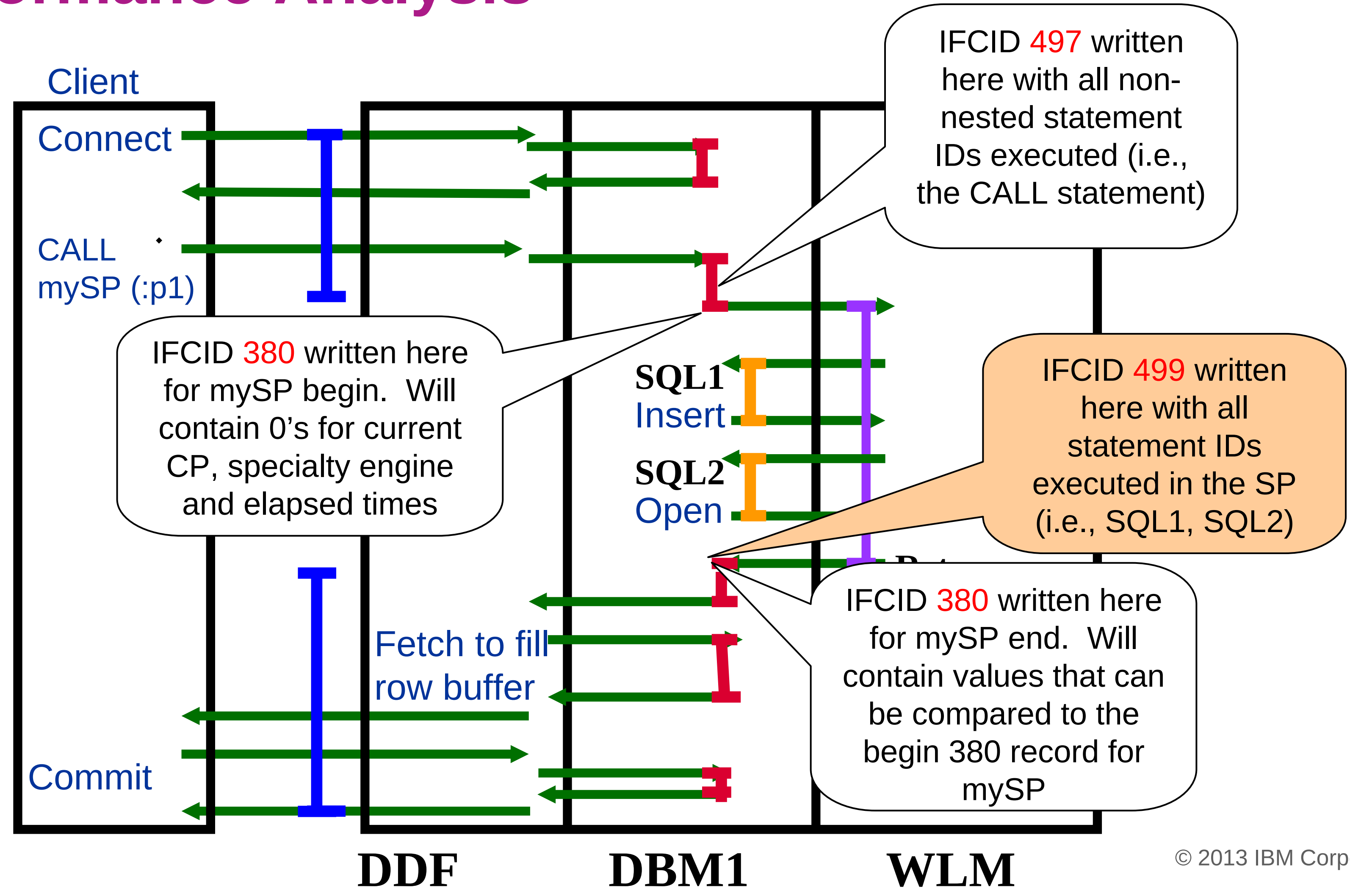
IFCID 380 written here for mySP begin. Will contain 0's for current CP, specialty engine and elapsed times

IFCID 380 written here for mySP end. Will contain values that can be compared to the begin 380 record for mySP

Enhanced Instrumentation for Stored Procedure Performance Analysis

- **Additionally PM53243 (DB2 10) added IFCID 497, 498, 499 for SQL drill down analysis. These records contain the dynamic or static statement IDs for non-nested, UDF, and SP work respectively.**
- **The statement IDs can be correlated to IFCID 316 dynamic statement or IFCID 401 static statement cache data.**

Enhanced Instrumentation for Stored Procedure Performance Analysis



IFCID 497 written here with all non-nested statement IDs executed (i.e., the CALL statement)

IFCID 380 written here for mySP begin. Will contain 0’s for current CP, specialty engine and elapsed times

IFCID 499 written here with all statement IDs executed in the SP (i.e., SQL1, SQL2)

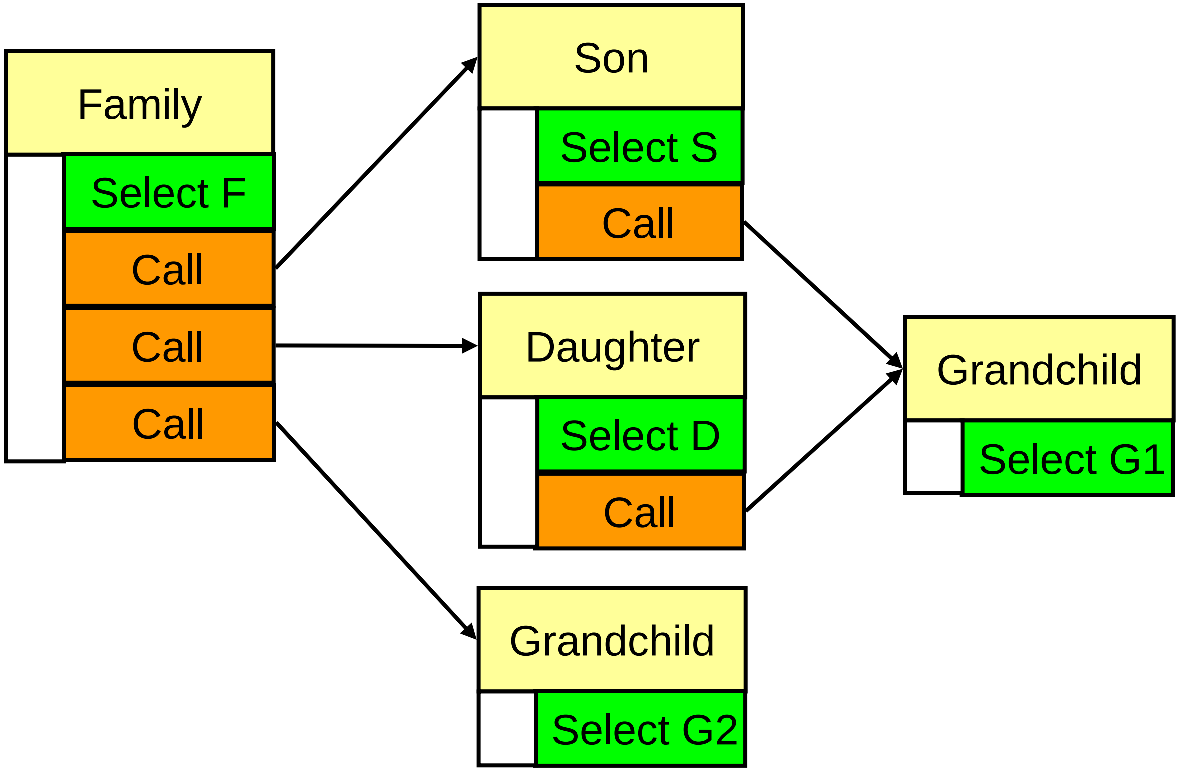
IFCID 380 written here for mySP end. Will contain values that can be compared to the begin 380 record for mySP

Monitoring Stored Procedures with OMPE

- The new DB2 instrumentation records for Stored Procedures are ingested by the OMPE Collector, aggregated on a system level and returned to the (OPM) Repository Server.
- The OMPE Collector processing includes the sequencing logic and the calculation of elapsed times for the different accounting class times written in the IFI records as timestamps, considering nesting as well.
- In parallel the IFCID 316/401 data for the Statement Caches is collected and a correlation to the executed stored procedure statements via IFCID 499 is made.
- Full RECTRACE support for all new IFCIDs is provided

Using the OMPE Web Console to analyze Stored Procedures – sample scenario

- **Workload:**

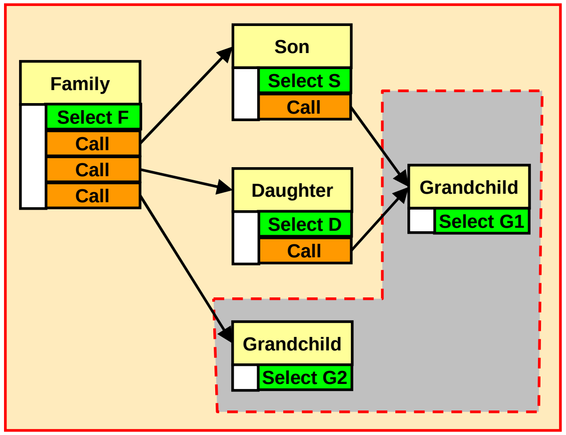


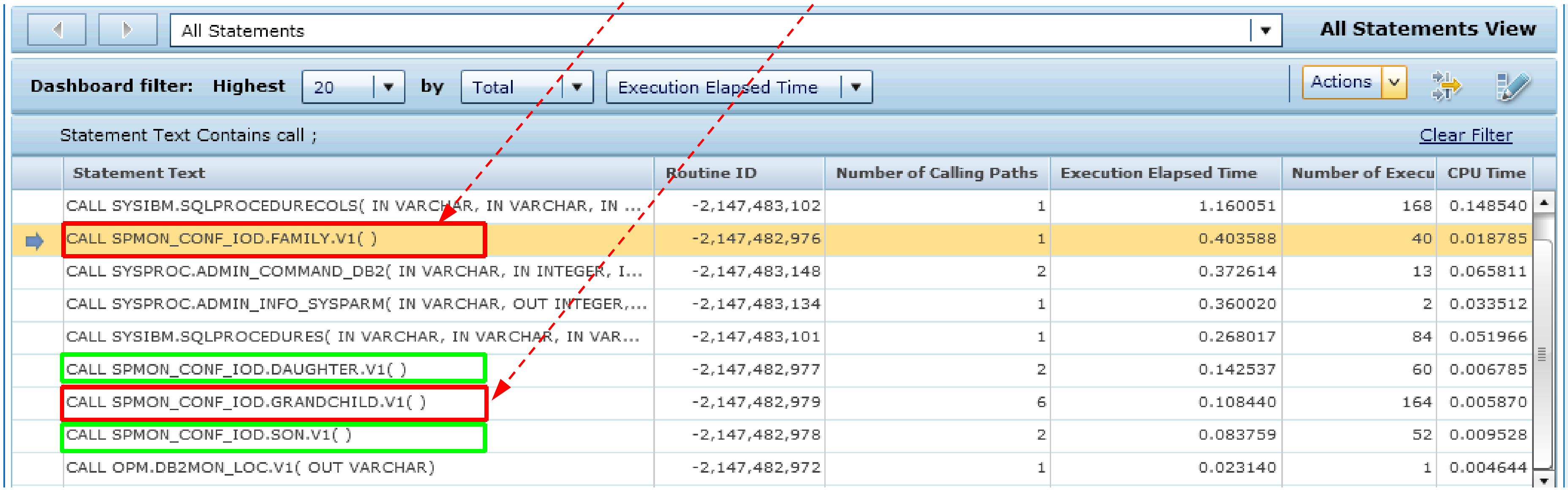
SQL Dashboard – aggregation by ROUTINEID

- Workload at SQL dashboard ("All statements" view) executed in the selected time period (time slider), valid for all subsequent views

Σ of Family

Σ of Grandchild



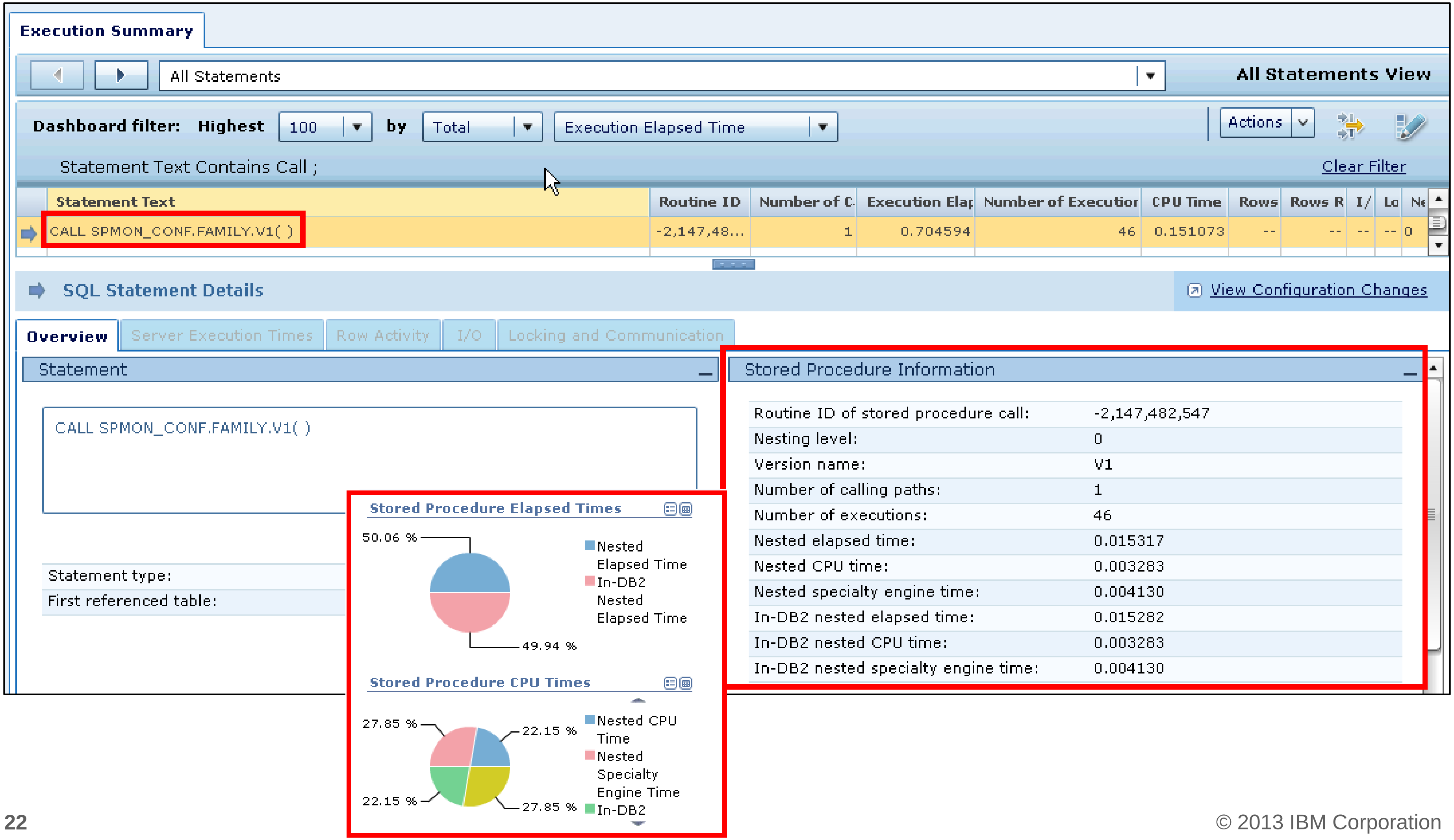


Dashboard filter: Highest 20 by Total Execution Elapsed Time

Statement Text Contains call ;

Statement Text	Routine ID	Number of Calling Paths	Execution Elapsed Time	Number of Execu	CPU Time
CALL SYSIBM.SQLPROCEDURECOLS(IN VARCHAR, IN VARCHAR, IN ...	-2,147,483,102	1	1.160051	168	0.148540
CALL SPMON_CONF_IOD.FAMILY.V1( )	-2,147,482,976	1	0.403588	40	0.018785
CALL SYSPROC.ADMIN_COMMAND_DB2(IN VARCHAR, IN INTEGER, I...	-2,147,483,148	2	0.372614	13	0.065811
CALL SYSPROC.ADMIN_INFO_SYSPARM(IN VARCHAR, OUT INTEGER,...	-2,147,483,134	1	0.360020	2	0.033512
CALL SYSIBM.SQLPROCEDURES(IN VARCHAR, IN VARCHAR, IN VAR...	-2,147,483,101	1	0.268017	84	0.051966
CALL SPMON_CONF_IOD.DAUGHTER.V1( )	-2,147,482,977	2	0.142537	60	0.006785
CALL SPMON_CONF_IOD.GRANDCHILD.V1( )	-2,147,482,979	6	0.108440	164	0.005870
CALL SPMON_CONF_IOD.SON.V1( )	-2,147,482,978	2	0.083759	52	0.009528
CALL OPM.DB2MON_LOC.V1(OUT VARCHAR)	-2,147,482,972	1	0.023140	1	0.004644

Showing SP Details



Execution Summary

◀ ▶ All Statements ▾ **All Statements View**

Dashboard filter: Highest 100 ▾ **by** Total ▾ Execution Elapsed Time ▾ Actions ▾

Statement Text Contains Call ; Clear Filter

Statement Text	Routine ID	Number of C	Execution Elap	Number of Executior	CPU Time	Rows	Rows R	I/	Lo	Ne
CALL SPMON_CONF.FAMILY.V1( )	-2,147,48...	1	0.704594	46	0.151073	--	--	--	--	0

SQL Statement Details

View Configuration Changes

Overview | Server Execution Times | Row Activity | I/O | Locking and Communication

Statement

CALL SPMON_CONF.FAMILY.V1( )

Statement type:

First referenced table:

Stored Procedure Information

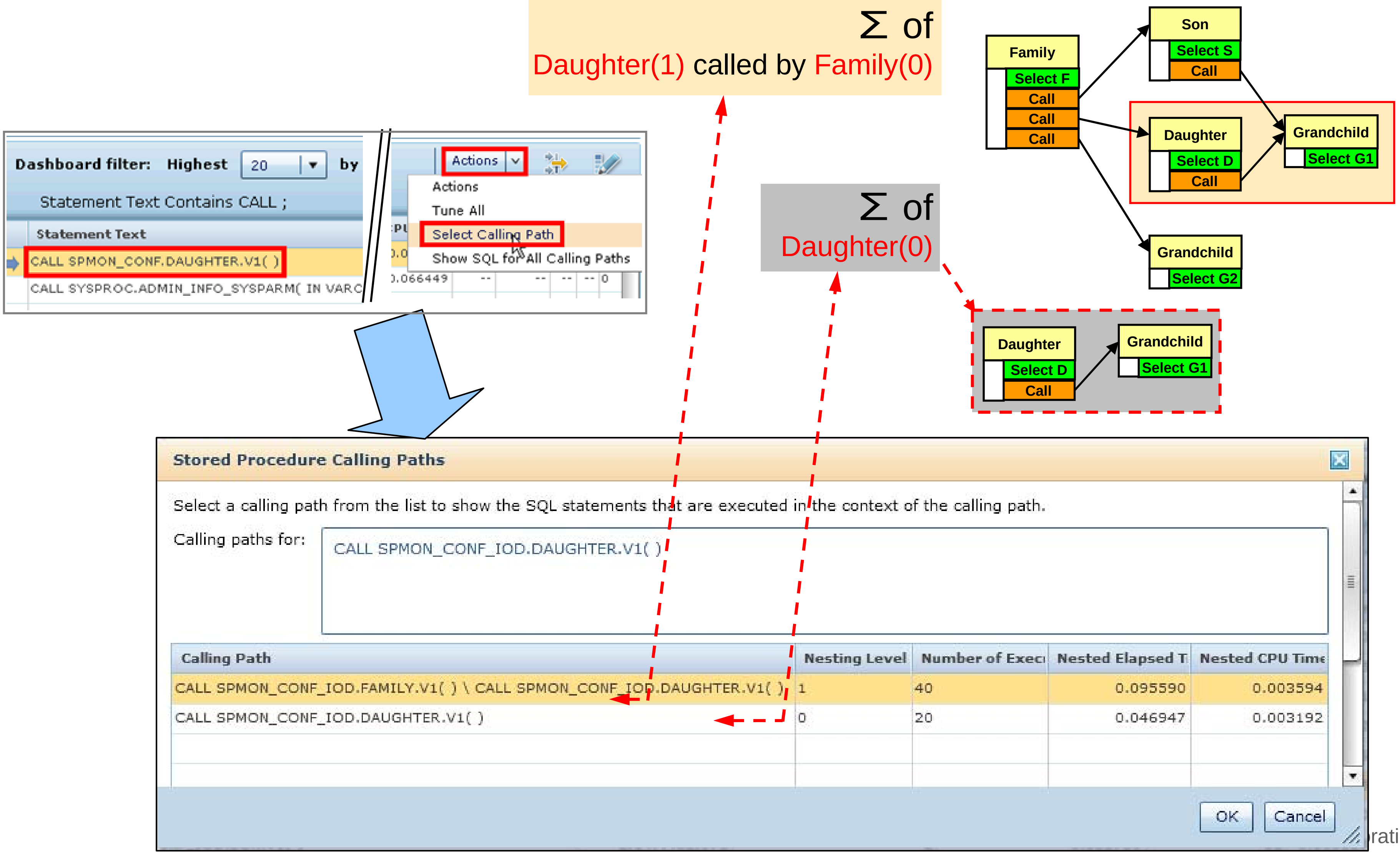
Routine ID of stored procedure call:	-2,147,482,547
Nesting level:	0
Version name:	V1
Number of calling paths:	1
Number of executions:	46
Nested elapsed time:	0.015317
Nested CPU time:	0.003283
Nested specialty engine time:	0.004130
In-DB2 nested elapsed time:	0.015282
In-DB2 nested CPU time:	0.003283
In-DB2 nested specialty engine time:	0.004130

Stored Procedure Elapsed Times

Stored Procedure CPU Times

Showing the calling paths of SPs (1/2)

- Select Calling Path for Daughter



Σ of Daughter(1) called by Family(0)

Σ of Daughter(0)

Dashboard filter: Highest 20 by

Statement Text Contains CALL ;

Statement Text
CALL SPMON_CONF.DAUGHTER.V1( )
CALL SYSPROC.ADMIN_INFO_SYSPARM(IN VARC

Stored Procedure Calling Paths

Select a calling path from the list to show the SQL statements that are executed in the context of the calling path.

Calling paths for: CALL SPMON_CONF_IOD.DAUGHTER.V1( )

Calling Path	Nesting Level	Number of Execu	Nested Elapsed T	Nested CPU Time
CALL SPMON_CONF_IOD.FAMILY.V1( ) \ CALL SPMON_CONF_IOD.DAUGHTER.V1( )	1	40	0.095590	0.003594
CALL SPMON_CONF_IOD.DAUGHTER.V1( )	0	20	0.046947	0.003192

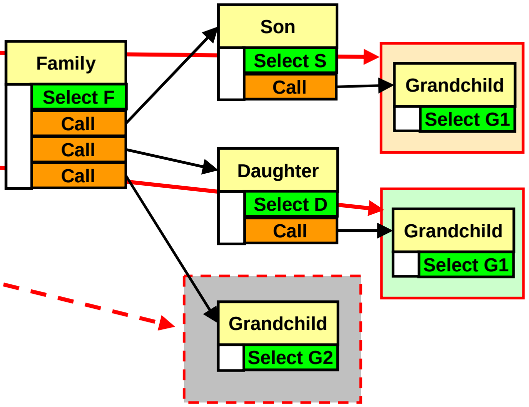
Showing the calling paths of SPs (2/2)

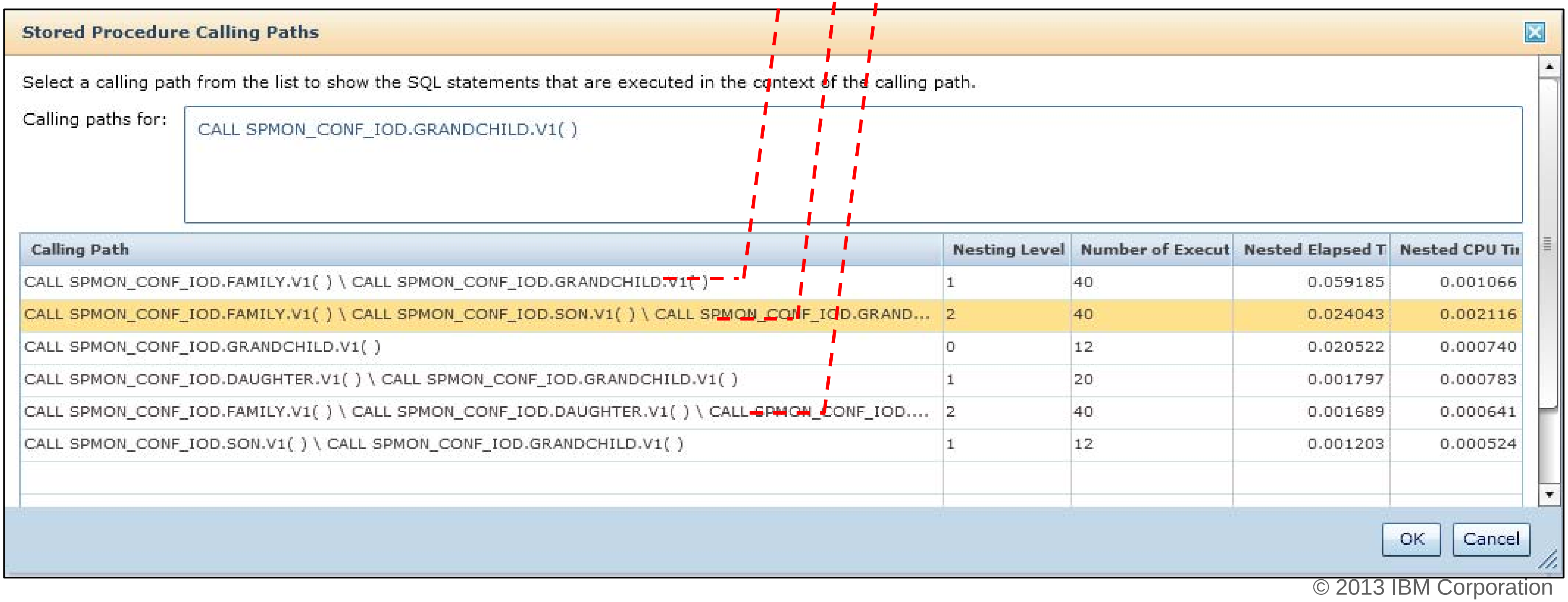
- Select Calling Path for Grandchild

Σ of Grandchild(2) called by Family(0)+Son(1)

Σ of Granchild(2) called by Family (0)+Daughter(1)

Σ of Grandchild(1) called by Family (0)





Stored Procedure Calling Paths

Select a calling path from the list to show the SQL statements that are executed in the context of the calling path.

Calling paths for: CALL SPMON_CONF_IOD.GRANDCHILD.V1( )

Calling Path	Nesting Level	Number of Execut	Nested Elapsed T	Nested CPU Ti
CALL SPMON_CONF_IOD.FAMILY.V1( ) \ CALL SPMON_CONF_IOD.GRANDCHILD.V1( )	1	40	0.059185	0.001066
CALL SPMON_CONF_IOD.FAMILY.V1( ) \ CALL SPMON_CONF_IOD.SON.V1( ) \ CALL SPMON_CONF_IOD.GRAND...	2	40	0.024043	0.002116
CALL SPMON_CONF_IOD.GRANDCHILD.V1( )	0	12	0.020522	0.000740
CALL SPMON_CONF_IOD.DAUGHTER.V1( ) \ CALL SPMON_CONF_IOD.GRANDCHILD.V1( )	1	20	0.001797	0.000783
CALL SPMON_CONF_IOD.FAMILY.V1( ) \ CALL SPMON_CONF_IOD.DAUGHTER.V1( ) \ CALL SPMON_CONF_IOD....	2	40	0.001689	0.000641
CALL SPMON_CONF_IOD.SON.V1( ) \ CALL SPMON_CONF_IOD.GRANDCHILD.V1( )	1	12	0.001203	0.000524

Show SQL executed by a SP (1/2)

- Action: Show SQL for **This** Calling Path

Select of Family (0)
shows

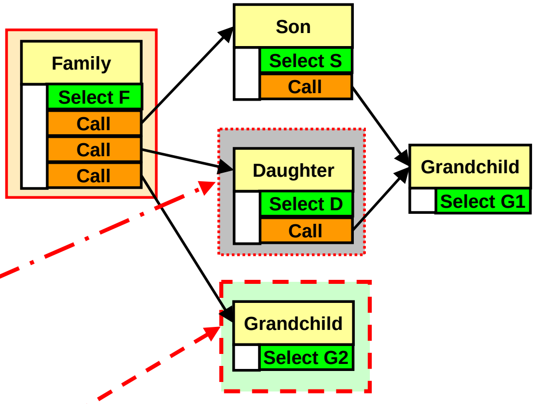
SELECT F *
Σ of Call Son(1) called by Family(0)
Σ of Call Daughter(1) called by Family(0)
Σ of Call Grandchild(1) called by Family(0)

Select of Daughter(1)
shows

SELECT D
Σ of Call Grandchild(2) called by Daughter(1)

Select of Grandchild(1)
shows

SELECT G

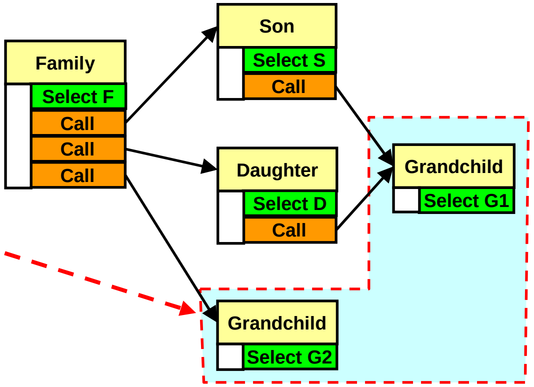


* see next slide

- Action: Show SQL for **All** Calling Paths

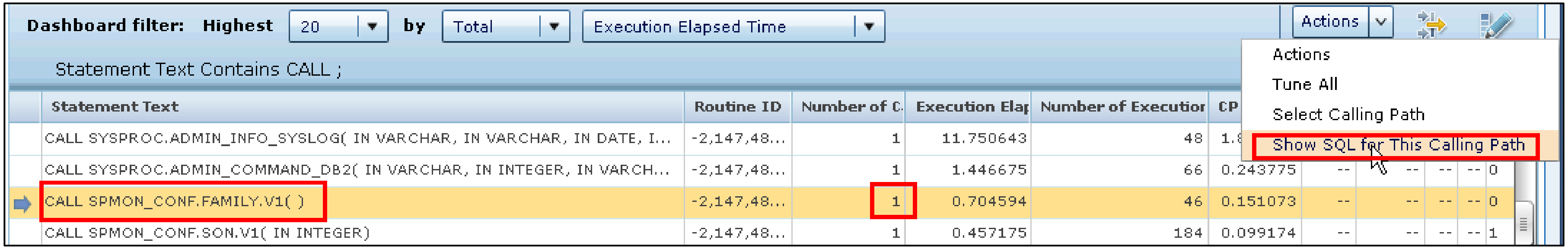
Select Grandchild()
shows Σ

SELECT G1
SELECT G2



Show SQL executed by a SP (2/2)

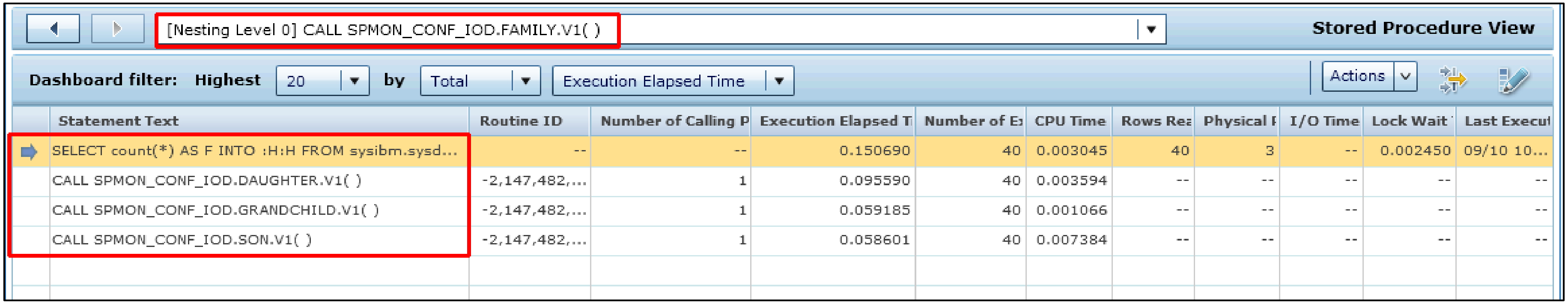
- Show SQL for **<u>This</u>** Calling Path for Family(0)



Dashboard filter: Highest 20 by Total Execution Elapsed Time

Statement Text Contains CALL ;

Statement Text	Routine ID	Number of C.	Execution Elap	Number of Executior	CP					
CALL SYSPROC.ADMIN_INFO_SYSLOG(IN VARCHAR, IN VARCHAR, IN DATE, I...	-2,147,48...	1	11.750643	48	1.8					
CALL SYSPROC.ADMIN_COMMAND_DB2(IN VARCHAR, IN INTEGER, IN VARCH...	-2,147,48...	1	1.446675	66	0.243775	--		--	--	0
CALL SPMON_CONF.FAMILY.V1( )	-2,147,48...	1	0.704594	46	0.151073	--	--	--	--	0
CALL SPMON_CONF.SON.V1(IN INTEGER)	-2,147,48...	1	0.457175	184	0.099174	--	--	--	--	1



[Nesting Level 0] CALL SPMON_CONF_IOD.FAMILY.V1( )

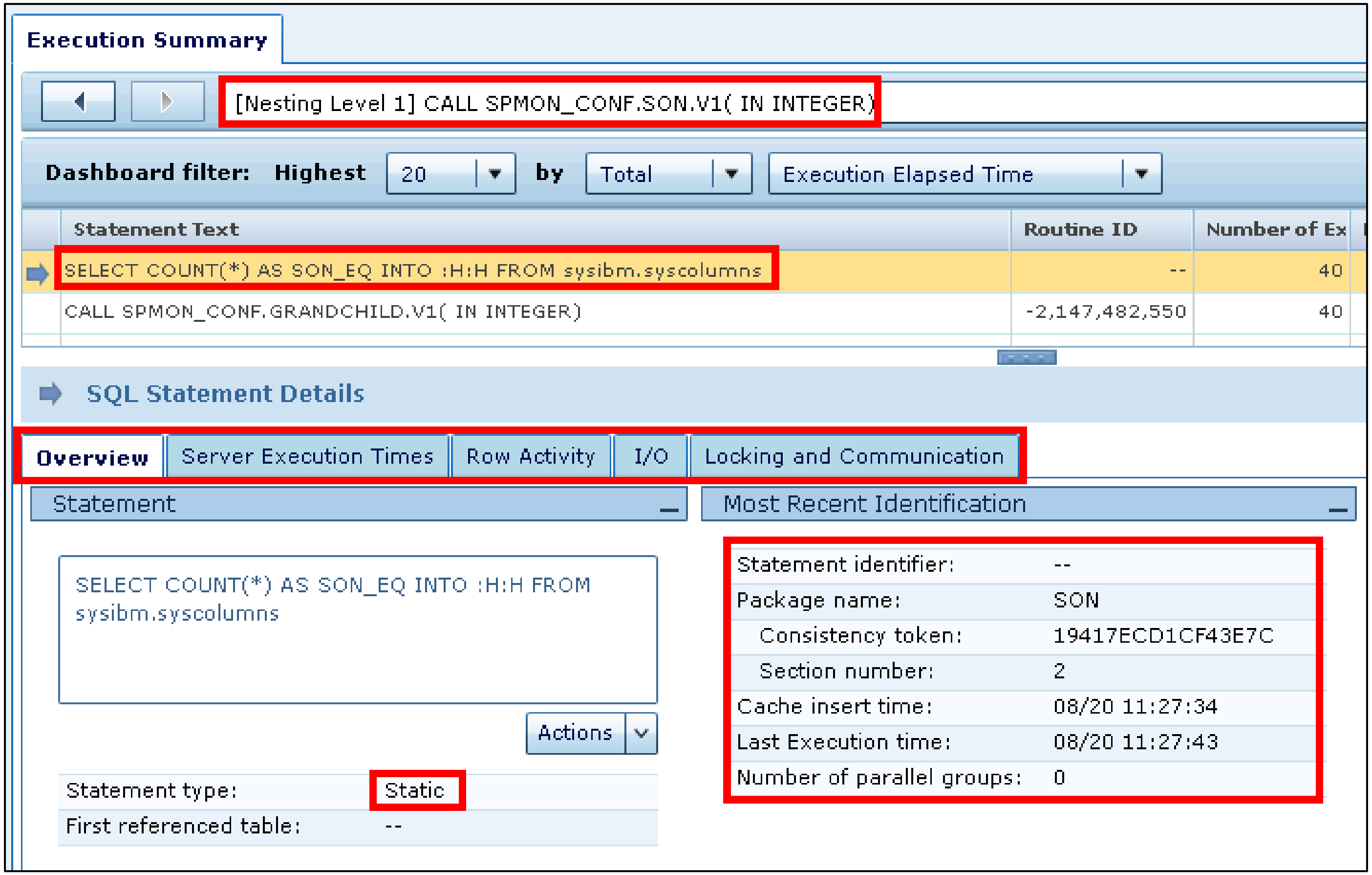
Stored Procedure View

Dashboard filter: Highest 20 by Total Execution Elapsed Time

Statement Text	Routine ID	Number of Calling P	Execution Elapsed T	Number of E:	CPU Time	Rows Rea	Physical I	I/O Time	Lock Wait	Last Execut
SELECT count(*) AS F INTO :H:H FROM sysibm.sysd...	--	--	0.150690	40	0.003045	40	3	--	0.002450	09/10 10...
CALL SPMON_CONF_IOD.DAUGHTER.V1( )	-2,147,482,...	1	0.095590	40	0.003594	--	--	--	--	--
CALL SPMON_CONF_IOD.GRANDCHILD.V1( )	-2,147,482,...	1	0.059185	40	0.001066	--	--	--	--	--
CALL SPMON_CONF_IOD.SON.V1( )	-2,147,482,...	1	0.058601	40	0.007384	--	--	--	--	--

SQL Cache Correlation

- For a nested statement correlation to the cache is shown in “SQL Statements Details” area:



[Nesting Level 1] CALL SPMON_CONF.SON.V1(IN INTEGER)

Dashboard filter: Highest 20 by Total Execution Elapsed Time

Statement Text	Routine ID	Number of Ex
SELECT COUNT(*) AS SON_EQ INTO :H:H FROM sysibm.syscolumns	--	40
CALL SPMON_CONF.GRANDCHILD.V1(IN INTEGER)	-2,147,482,550	40

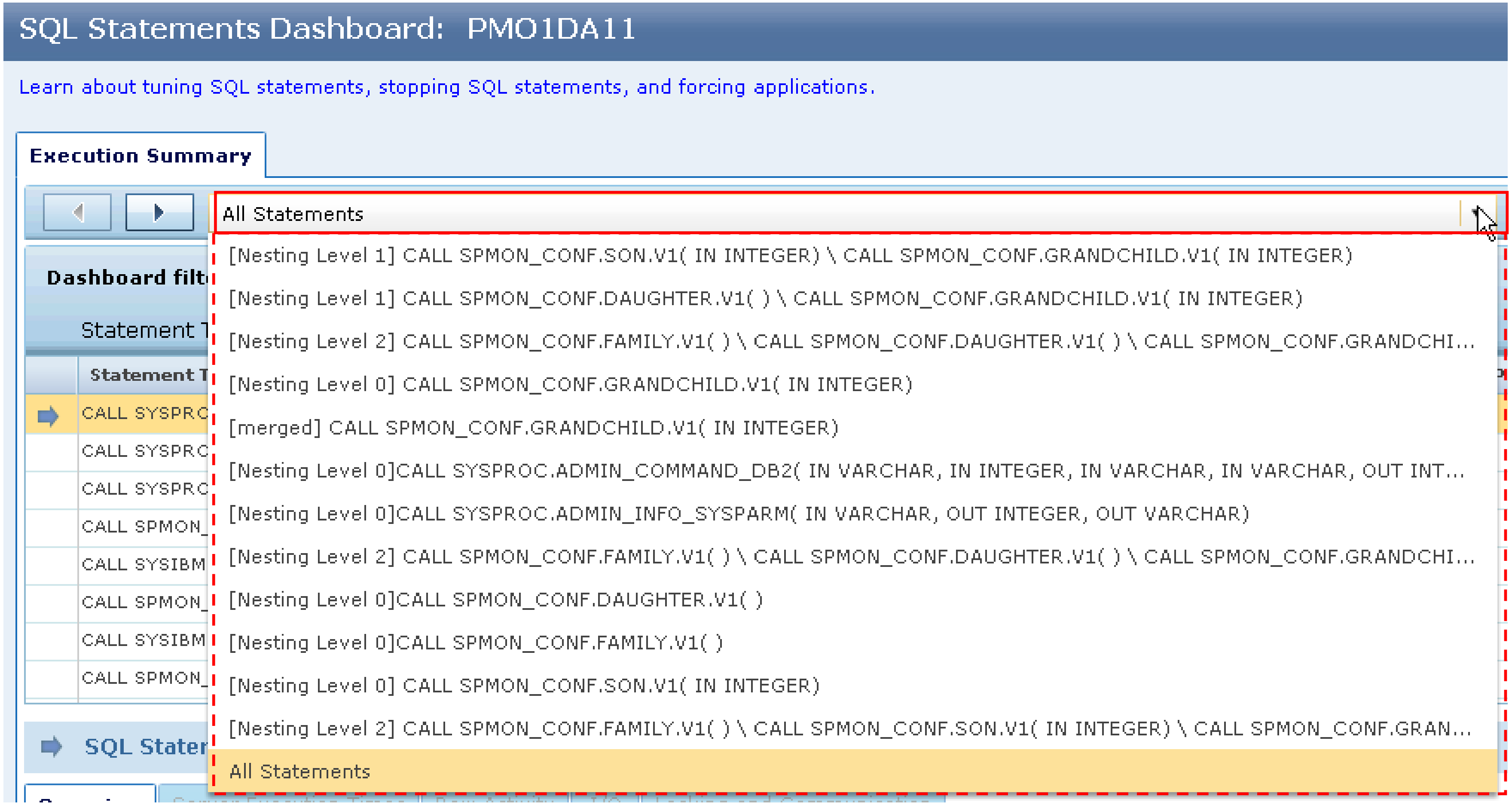
SELECT COUNT(*) AS SON_EQ INTO :H:H FROM sysibm.syscolumns

Statement type:	Static
First referenced table:	--

Statement identifier:	--
Package name:	SON
Consistency token:	19417ECD1CF43E7C
Section number:	2
Cache insert time:	08/20 11:27:34
Last Execution time:	08/20 11:27:43
Number of parallel groups:	0

History Navigator

- The History Navigator shows the drill down history for Stored Procedures and can be used similar to a Browser History



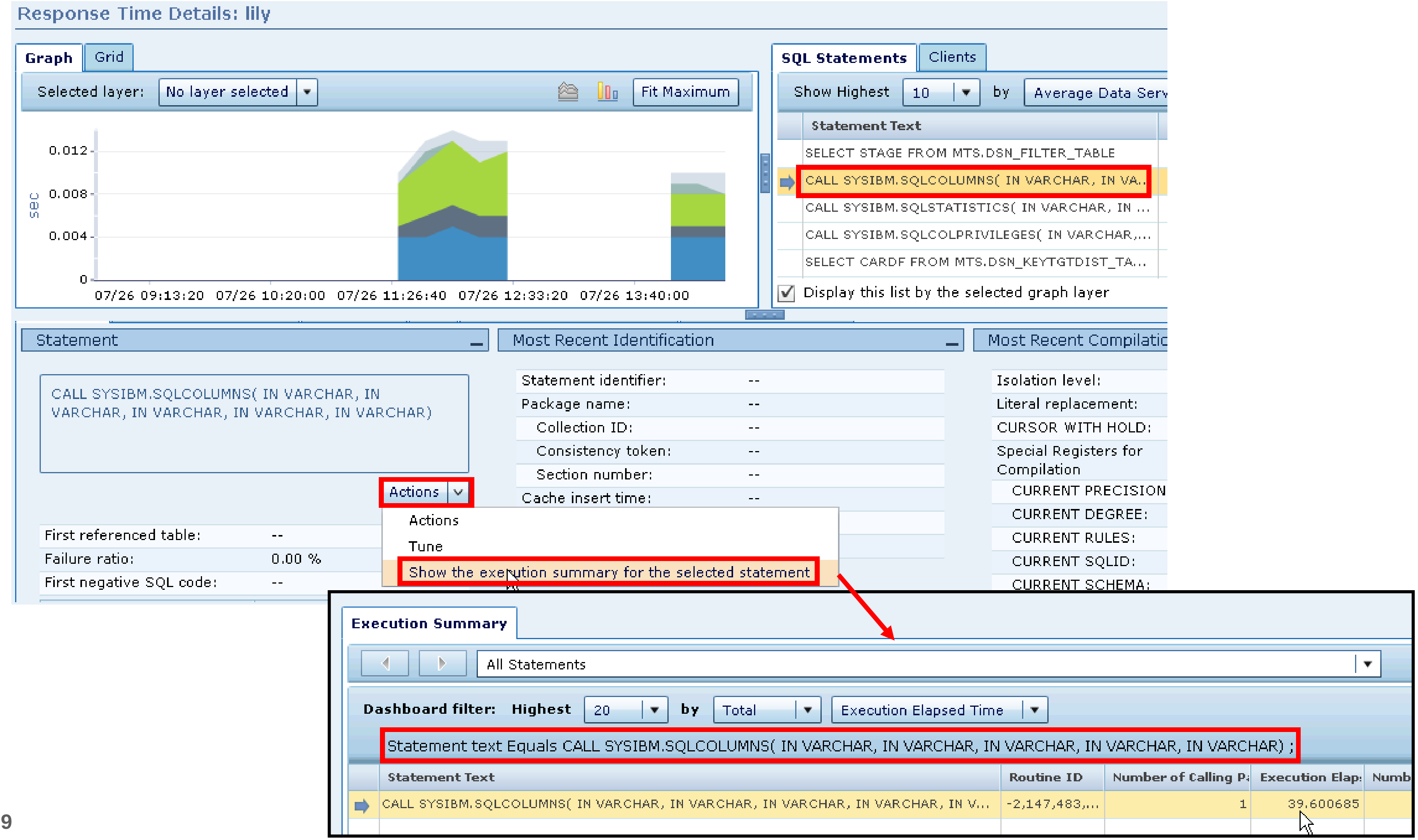
SQL Statements Dashboard: PMO1DA11

Learn about tuning SQL statements, stopping SQL statements, and forcing applications.

All Statements

- [Nesting Level 1] CALL SPMON_CONF.SON.V1(IN INTEGER) \ CALL SPMON_CONF.GRANDCHILD.V1(IN INTEGER)
- [Nesting Level 1] CALL SPMON_CONF.DAUGHTER.V1( ) \ CALL SPMON_CONF.GRANDCHILD.V1(IN INTEGER)
- [Nesting Level 2] CALL SPMON_CONF.FAMILY.V1( ) \ CALL SPMON_CONF.DAUGHTER.V1( ) \ CALL SPMON_CONF.GRANDCHI...
- [Nesting Level 0] CALL SPMON_CONF.GRANDCHILD.V1(IN INTEGER)
- [merged] CALL SPMON_CONF.GRANDCHILD.V1(IN INTEGER)
- [Nesting Level 0]CALL SYSPROC.ADMIN_COMMAND_DB2(IN VARCHAR, IN INTEGER, IN VARCHAR, IN VARCHAR, OUT INT...
- [Nesting Level 0]CALL SYSPROC.ADMIN_INFO_SYSPARM(IN VARCHAR, OUT INTEGER, OUT VARCHAR)
- [Nesting Level 2] CALL SPMON_CONF.FAMILY.V1( ) \ CALL SPMON_CONF.DAUGHTER.V1( ) \ CALL SPMON_CONF.GRANDCHI...
- [Nesting Level 0]CALL SPMON_CONF.DAUGHTER.V1( )
- [Nesting Level 0]CALL SPMON_CONF.FAMILY.V1( )
- [Nesting Level 0] CALL SPMON_CONF.SON.V1(IN INTEGER)
- [Nesting Level 2] CALL SPMON_CONF.FAMILY.V1( ) \ CALL SPMON_CONF.SON.V1(IN INTEGER) \ CALL SPMON_CONF.GRAN...
- All Statements

Finally: Link to ‘Extended Insight’ functionality



Response Time Details: lily

Graph | Grid

Selected layer: No layer selected ▾ | Fit Maximum

SQL Statements | Clients

Show Highest 10 ▾ by Average Data Serv

Statement Text
SELECT STAGE FROM MTS.DSN_FILTER_TABLE
CALL SYSIBM.SQLCOLUMNS(IN VARCHAR, IN VA...
CALL SYSIBM.SQLSTATISTICS(IN VARCHAR, IN ...
CALL SYSIBM.SQLCOLPRIVILEGES(IN VARCHAR,...
SELECT CARDF FROM MTS.DSN_KEYTGTDIST_TA...

☑ Display this list by the selected graph layer

Statement

CALL SYSIBM.SQLCOLUMNS(IN VARCHAR, IN VARCHAR, IN VARCHAR, IN VARCHAR, IN VARCHAR)

Actions ▾

- Actions
- Tune
- Show the execution summary for the selected statement

First referenced table:	--
Failure ratio:	0.00 %
First negative SQL code:	--

Most Recent Identification

Statement identifier:	--
Package name:	--
Collection ID:	--
Consistency token:	--
Section number:	--
Cache insert time:	--

Most Recent Compilatic

Isolation level:
Literal replacement:
CURSOR WITH HOLD:
Special Registers for Compilation
CURRENT PRECISION
CURRENT DEGREE:
CURRENT RULES:
CURRENT SQLID:
CURRENT SCHEMA:

Execution Summary

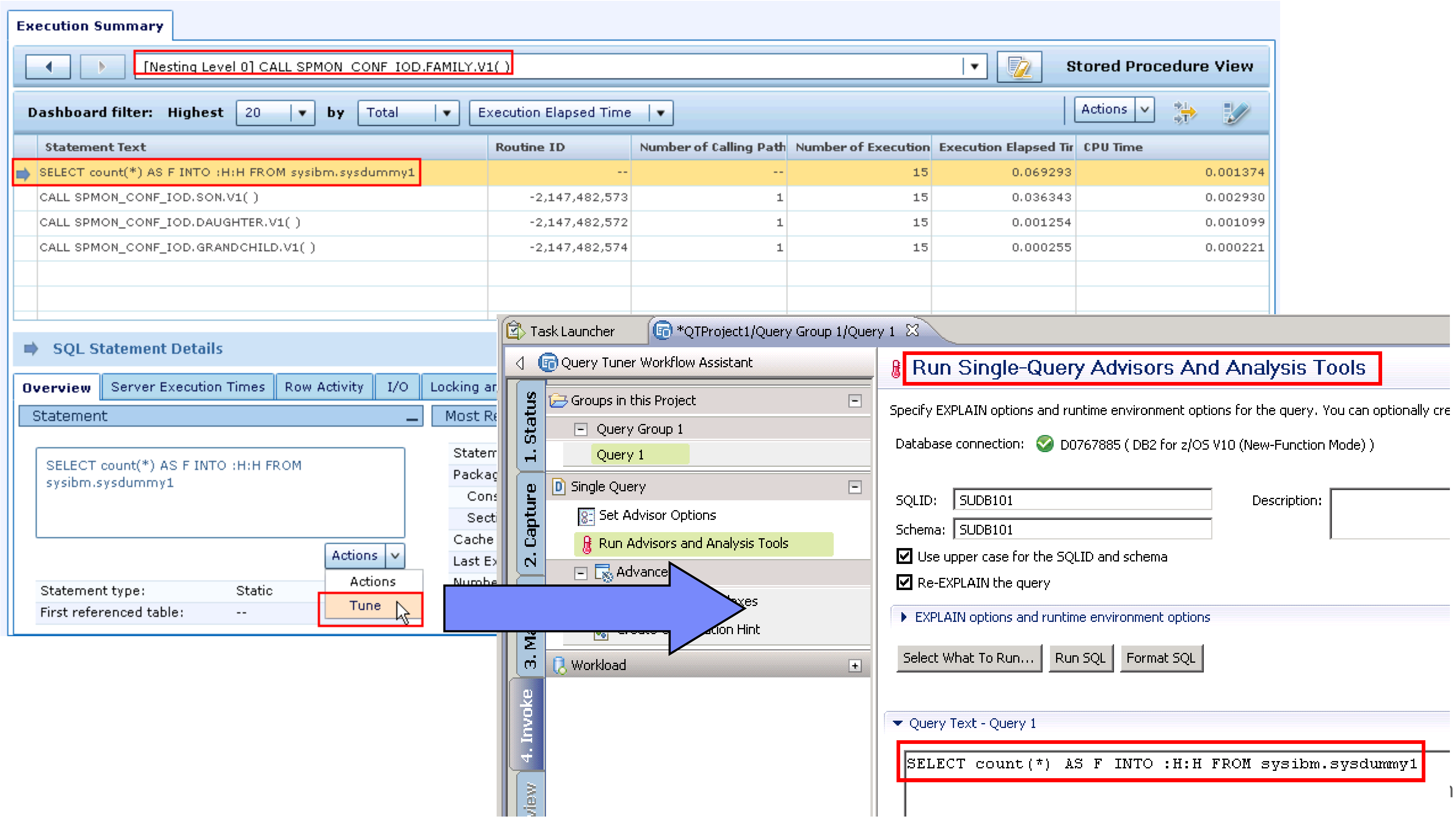
All Statements ▾

Dashboard filter: Highest 20 ▾ by Total ▾ Execution Elapsed Time ▾

Statement text Equals CALL SYSIBM.SQLCOLUMNS(IN VARCHAR, IN VARCHAR, IN VARCHAR, IN VARCHAR, IN VARCHAR) ;

Statement Text	Routine ID	Number of Calling P:	Execution Elap:	Numb
CALL SYSIBM.SQLCOLUMNS(IN VARCHAR, IN VARCHAR, IN VARCHAR, IN VARCHAR, IN V...	-2,147,483,...	1	39.600685	

Integration with Optim Query Workload Tuner for z/OS – Single query tuning



Execution Summary

[Nesting Level 0] CALL SPMON_CONF_IOD.FAMILY.V1( )

Stored Procedure View

Dashboard filter: Highest 20 by Total Execution Elapsed Time

Actions

Statement Text	Routine ID	Number of Calling Path	Number of Execution	Execution Elapsed Tir	CPU Time
SELECT count(*) AS F INTO :H:H FROM sysibm.sysdummy1	--	--	15	0.069293	0.001374
CALL SPMON_CONF_IOD.SON.V1( )	-2,147,482,573	1	15	0.036343	0.002930
CALL SPMON_CONF_IOD.DAUGHTER.V1( )	-2,147,482,572	1	15	0.001254	0.001099
CALL SPMON_CONF_IOD.GRANDCHILD.V1( )	-2,147,482,574	1	15	0.000255	0.000221

SQL Statement Details

Overview | Server Execution Times | Row Activity | I/O | Locking ar

Statement

```
SELECT count(*) AS F INTO :H:H FROM
sysibm.sysdummy1
```

Statement type: Static

First referenced table: --

Task Launcher | *QTProject1/Query Group 1/Query 1

Query Tuner Workflow Assistant

Run Single-Query Advisors And Analysis Tools

Specify EXPLAIN options and runtime environment options for the query. You can optionally cre

Database connection: D0767885 (DB2 for z/OS V10 (New-Function Mode))

SQLID: SUDB101

Schema: SUDB101

Description:

☑ Use upper case for the SQLID and schema

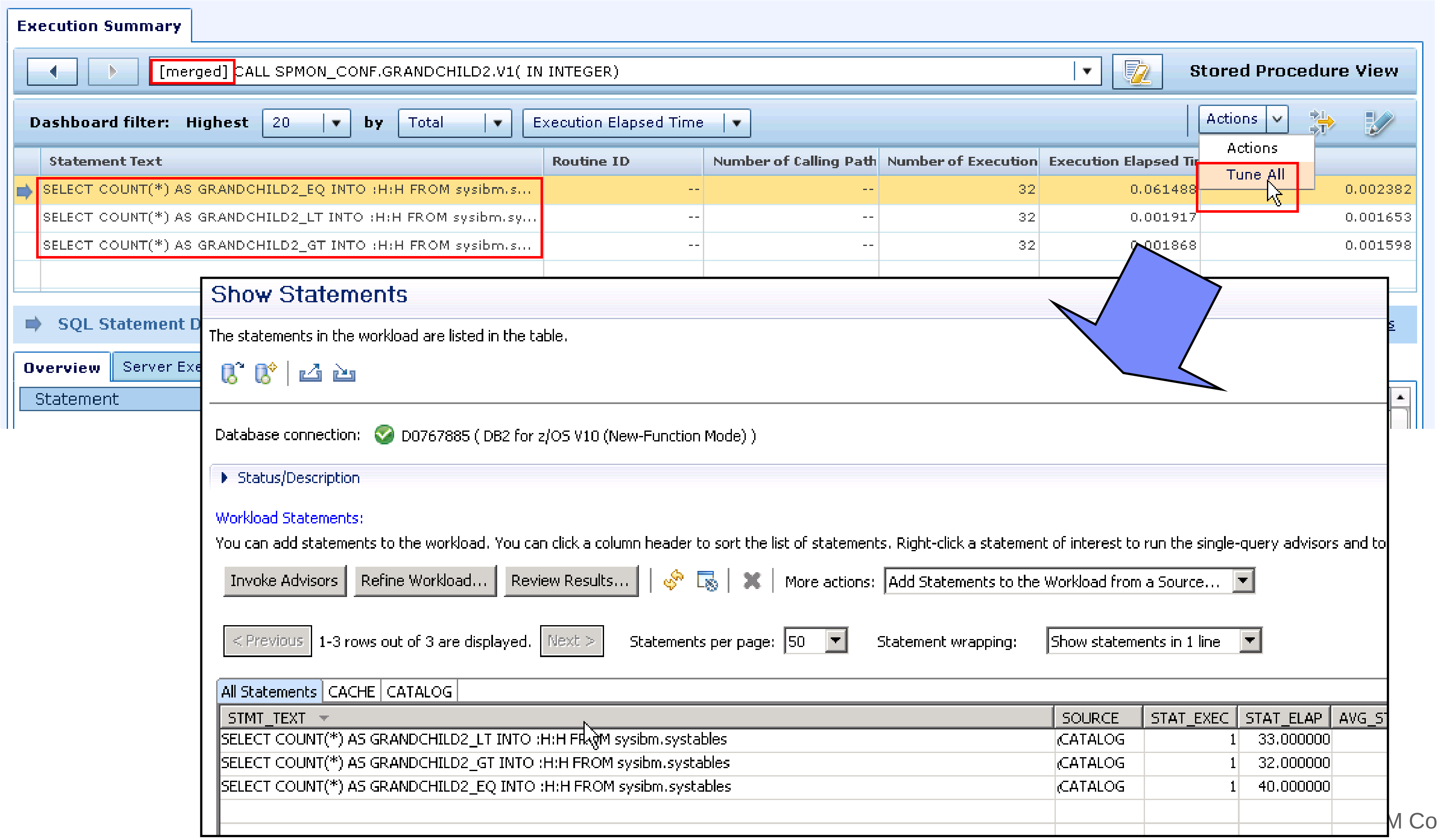
☑ Re-EXPLAIN the query

EXPLAIN options and runtime environment options

Query Text - Query 1

```
SELECT count(*) AS F INTO :H:H FROM sysibm.sysdummy1
```

Integration with Optim Query Workload Tuner for z/OS – Workload level tuning



[merged] CALL SPMON_CONF.GRANDCHILD2.V1(IN INTEGER)

Dashboard filter: Highest 20 by Total Execution Elapsed Time

Statement Text	Routine ID	Number of Calling Path	Number of Execution	Execution Elapsed Ti	
SELECT COUNT(*) AS GRANDCHILD2_EQ INTO :H:H FROM sysibm.s...	--	--	32	0.061488	0.002382
SELECT COUNT(*) AS GRANDCHILD2_LT INTO :H:H FROM sysibm.sy...	--	--	32	0.001917	0.001653
SELECT COUNT(*) AS GRANDCHILD2_GT INTO :H:H FROM sysibm.s...	--	--	32	0.001868	0.001598

Show Statements

The statements in the workload are listed in the table.

Database connection: ✓ D0767885 (DB2 for z/OS V10 (New-Function Mode))

Status/Description

Workload Statements:

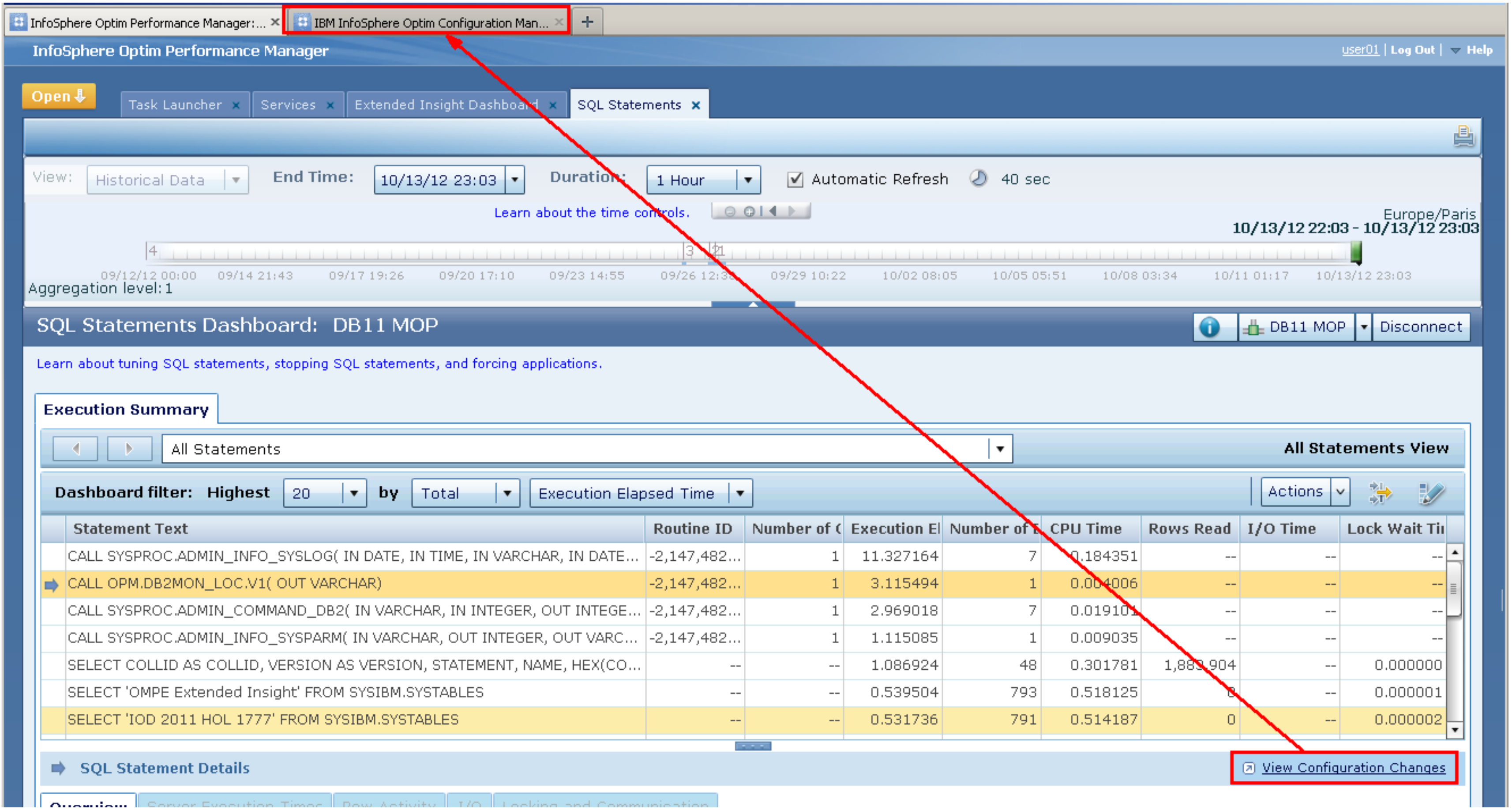
You can add statements to the workload. You can click a column header to sort the list of statements. Right-click a statement of interest to run the single-query advisors and to

More actions: Add Statements to the Workload from a Source...

1-3 rows out of 3 are displayed. Statements per page: 50 Statement wrapping: Show statements in 1 line

STMT_TEXT	SOURCE	STAT_EXEC	STAT_ELAP	AVG_ST
SELECT COUNT(*) AS GRANDCHILD2_LT INTO :H:H FROM sysibm.systables	CATALOG	1	33.000000	
SELECT COUNT(*) AS GRANDCHILD2_GT INTO :H:H FROM sysibm.systables	CATALOG	1	32.000000	
SELECT COUNT(*) AS GRANDCHILD2_EQ INTO :H:H FROM sysibm.systables	CATALOG	1	40.000000	

Integration with Optim Configuration Monitor for z/OS – Configuration optimization



InfoSphere Optim Performance Manager:...
IBM InfoSphere Optim Configuration Man...
InfoSphere Optim Performance Manager
user01 | Log Out | Help
Open
Task Launcher
Services
Extended Insight Dashboard
SQL Statements

View: Historical Data

End Time: 10/13/12 23:03

Duration: 1 Hour

☑ Automatic Refresh 40 sec

Learn about the time controls.

Europe/Paris
10/13/12 22:03 - 10/13/12 23:03

Aggregation level: 1

SQL Statements Dashboard: DB11 MOP

Learn about tuning SQL statements, stopping SQL statements, and forcing applications.

Execution Summary

All Statements

All Statements View

Dashboard filter: Highest 20 by Total Execution Elapsed Time

	Statement Text	Routine ID	Number of (	Execution El	Number of E	CPU Time	Rows Read	I/O Time	Lock Wait Ti
	CALL SYSPROC.ADMIN_INFO_SYSLOG(IN DATE, IN TIME, IN VARCHAR, IN DATE...	-2,147,482...	1	11.327164	7	0.184351	--	--	--
➡	CALL OPM.DB2MON_LOC.V1(OUT VARCHAR)	-2,147,482...	1	3.115494	1	0.004006	--	--	--
	CALL SYSPROC.ADMIN_COMMAND_DB2(IN VARCHAR, IN INTEGER, OUT INTEGE...	-2,147,482...	1	2.969018	7	0.019101	--	--	--
	CALL SYSPROC.ADMIN_INFO_SYSPARM(IN VARCHAR, OUT INTEGER, OUT VARC...	-2,147,482...	1	1.115085	1	0.009035	--	--	--
	SELECT COLLID AS COLLID, VERSION AS VERSION, STATEMENT, NAME, HEX(CO...	--	--	1.086924	48	0.301781	1,883,904	--	0.000000
	SELECT 'OMPE Extended Insight' FROM SYSIBM.SYSTABLES	--	--	0.539504	793	0.518125	0	--	0.000001
	SELECT 'IOD 2011 HOL 1777' FROM SYSIBM.SYSTABLES	--	--	0.531736	791	0.514187	0	--	0.000002

SQL Statement Details

View Configuration Changes

More Information

- **Websites**
  - DB2 for z/OS home page
  - DB2 Tools for z/OS home page
  - OMEGAMON XE for DB2 PE on z/OS home page
  - Optim Query Workload Tuner for z/OS home page
  - Optim Configuration Manager for z/OS home page
  - DB2 for z/OS: Information Roadmap
- **DB2 Stored Procedure Redbooks**
  - IBM DB2 Stored Procedures: Through the Call & Beyond
  - Triggers and User Defined Functions on DB2
- **Online demo**
  - Stored procedure monitoring

Thank you !

presenter@us.ibm.com