#SHAREorg

SHARE
Technology • Connections • Results

Leveraging the Instrumentation Facility on DB2 for z/OS: Tracing and Monitor Fundamentals

John B. Tobler
IBM, DB2 for z/OS Senior Technical Staff Member
jtobler@us.ibm.com

February 7, 2013 3:00 – 4:00
Session 12783

SHARE in San Francisco 2013

Agenda

- Introduction to instrumentation
- Instrumentation volume control
    - Accounting rollup
    - SMF compression
    - Trace Filtering
- Aggregated performance statistics
- Statement level analysis
- Stored procedure analysis
- Auditing

DB2's Instrumentation Facility

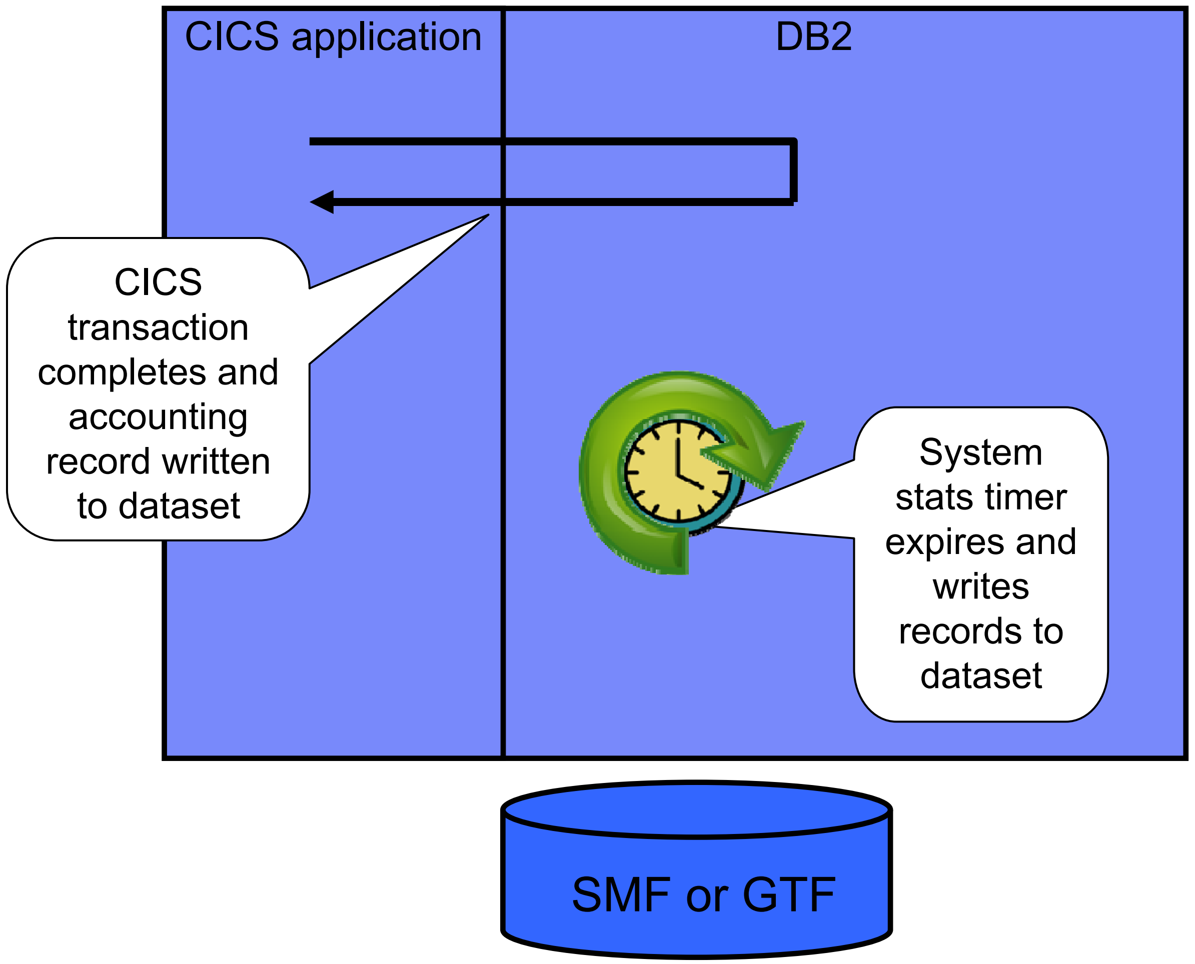
- DB2 defines traces records as IFCIDs
  - Each record has a unique mapping (DSNDQW* external macros or DSNWMSGS)
  - When written to SMF, the DB2 record types are 100 (statistics), 101 (accounting), and 102 (everything else)

- Trace records can be externalized based on
  - A timer (statistics)
  - An event (accounting, performance, audit)
  - A real-time request (monitor)

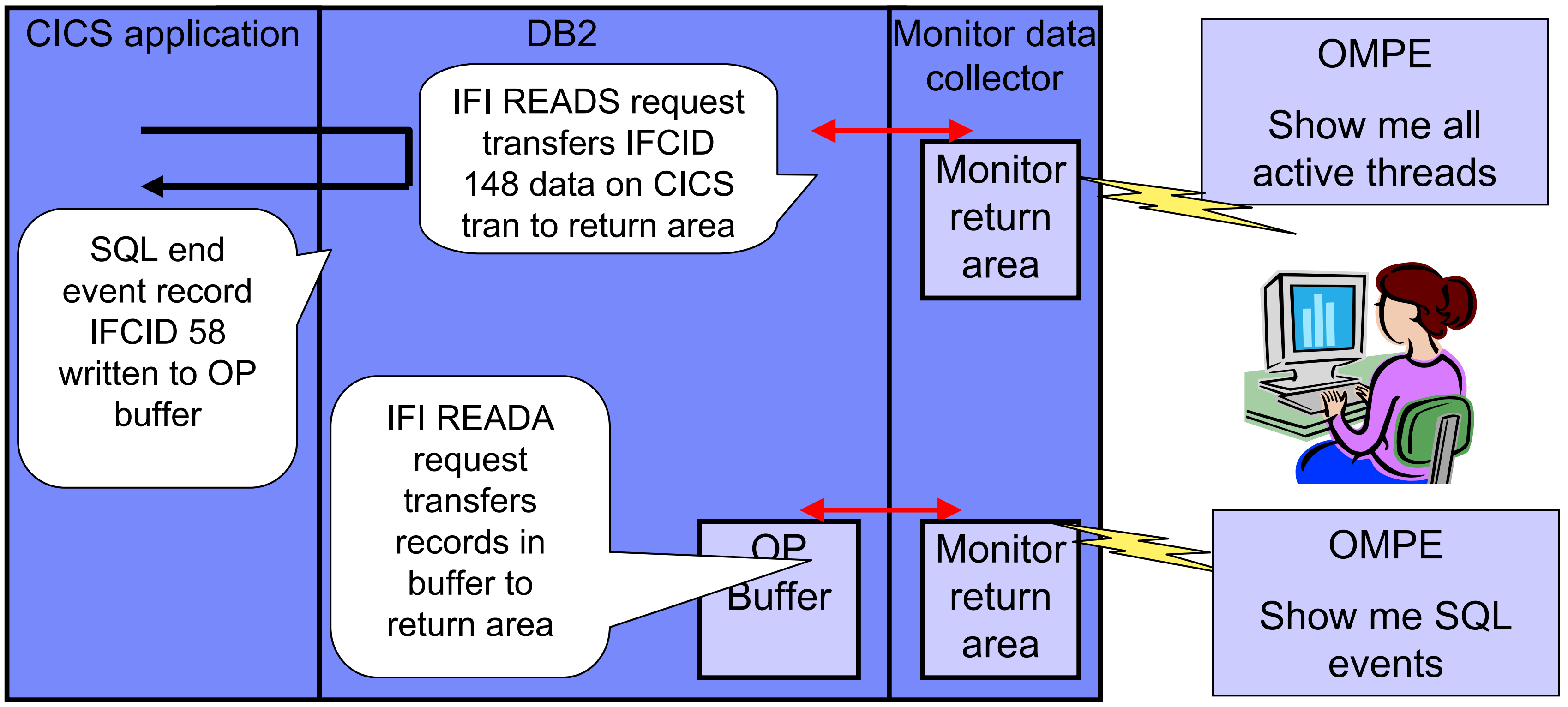
- Managed via –START, –STOP, –DISPLAY, –MODIFY TRACE commands
  - IFCIDs are grouped into 'classes' by trace type for ease of use

    Example: --START TRACE(STAT) CLASS(1) enables IFCID 1, 2, 105, 106, 202, and 225

DB2’s Instrumentation Facility


CICS application
DB2
CICS transaction completes and accounting record written to dataset
System stats timer expires and writes records to dataset
SMF or GTF

DB2's Instrumentation Facility – IFI interface


CICS application
DB2
Monitor data collector
IFI READS request transfers IFCID 148 data on CICS tran to return area
Monitor return area
OMPE
Show me all active threads
SQL end event record IFCID 58 written to OP buffer
IFI READA request transfers records in buffer to return area
OP Buffer
Monitor return area
OMPE
Show me SQL events

Instrumentation Basics Continued…

- The most common and recommended classes are:

Trace Type	Recommended Classes	Typical Overhead
Accounting	1, 2, 3, 7, 8 to track time in and out of DB2 and each package executed 10 for package level drill down	2-3% CPU Classes 7 and 8 can consume ~1K/package/ transaction of ECSA in v9
Stats	1, 3, 9 to track all system level activity (SQL, BP, storage, CPU), deadlock and timeout 5 for data-sharing installations to track group buffer pools	Negligible

- Audit classes are largely determined by security and monitoring needs. The impact of these is typically low.

- Performance traces are often used for detailed analysis. These may be very high volume and can induce overhead of 10-20% CPU (see filtering).

Instrumentation Basics Continued…

- IFCIDs can be destined for SMF and GTF datasets or an in memory OP buffer
  - An IFCID is built 1x and then routed to each destination. Thus a record destined for 2 destination does not incur 2x the overhead. The majority of tracing overhead is in the build phase of the record with the externalization having smaller impact.
  - Some IFCIDs are not external records and are switches to signal the collection of additional data (e.g., accounting classes 7,8 enable the data collection for IFCID239). The destination for switch IFCIDs is ignored and filters do not impact these IFCIDS. The destination for the externalized record is what matters.

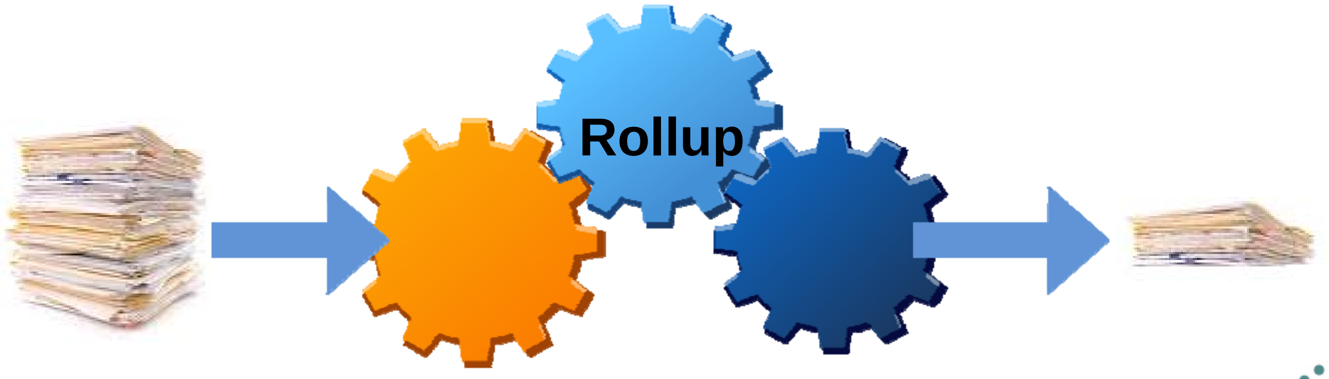
Destination	Data Permanence	Overhead
SMF or GTF	MVS dataset can be saved for post-analysis. Difficult to overrun but can be at high volumes (see filtering)	Low
OP	Limited in memory buffer (64M in v10). May be overrun if monitor program doesn't empty the buffer. Monitor may store permanently for post-analysis but typically these are discarded.	The serialization mechanism is the CML lock of the MSTR address space. This may adversely affect system throughput in v9 (mitigated in v10). This is typically seen as an increase in not accounted for time when a high volume trace is started.

Agenda

- Introduction to instrumentation
- **Instrumentation volume control**
  - **Accounting rollup**
  - **SMF compression**
  - **Trace Filtering**
- Aggregated performance statistics
- Statement level analysis
- Stored procedure analysis
- Auditing

Volume Control – Accounting

- V8 introduced rollup accounting
  - Controlled by ZPARMs ACCUMACC and ACCUMUID
  - Rolls up ACCUMACC number of 'end user instances' into a single accounting record for DDF and RRSAF threads
    Example: userid = 'jtobler' runs 10 distributed transactions and 1 accounting record is written for all 10 transactions
  - Rollup Accounting can reduce SMF volume ACCUMACC times for DDF and RRSAF threads
  - Drawbacks to rollup accounting are:
    - Not all fields are rolled up
    - Package accounting is not useful in v8/v9



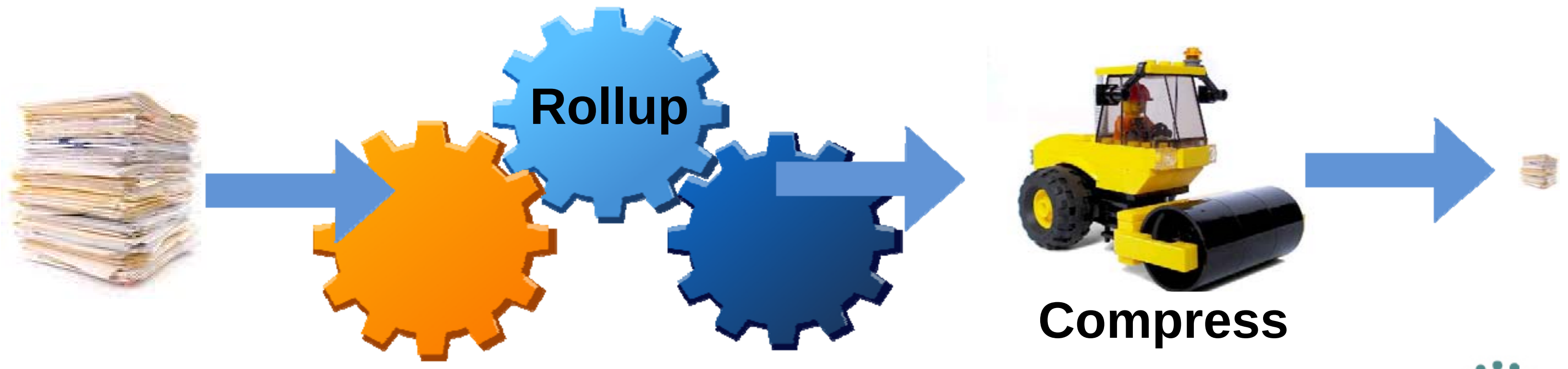
Volume Control – Accounting

- V10 dramatically enhanced rollup accounting
  - Up to 24 packages are now rolled up (DSNDQPAC in IFCID 239)
  - Up to 4 remote locations are now rolled up (DSNDQLAC in IFCID 3)
  - IFI accounting data is now rolled up (DSNDQIFA in IFCID3)
  - Distributed header, DDF accounting information reflect the last thread to roll data
  - Fixed time rather than variable time staleness threshold

Volume Control – Accounting

- V10 also introduced SMF compression
  - Controlled by ZPARM SMFCOMP with the default being OFF
  - Will compress all records written to SMF
  - Simple decompression algorithm
  - Accounting records compress 80-90% in many cases
  - Overhead is ~1%

- A combination of rollup accounting with ACCUMACC at 10 and SMF compression could reduce SMF accounting volume by 99%!



Volume Control – Filtering

- DB2 9 introduced extensive enhancement to instrumentation filtering. Filters now include PLAN, LOCATION, AUTHID, USERID, APPNAME, WRKSTN, PKGPROG, PKGLOC, PKGCOL, CONNID, CORRID, ROLE and eXclude keywords for each (i.e., XPLAN)

- Terminating and positional wildcards are allowed (e.g., PLAN(DSNTEP*) PLAN(PLAN_01))

- For each affirmative filter, multiple values are allowed. The logic between these is OR logic. For example:

  ```
  --START TRACE(PERFM) CLASS(3) PLAN(A, B)
  ```

  will write performance trace records if the PLAN = A OR PLAN = B.
  Note: This will start effectively 2 traces so the total trace limit of 32 may be reached more rapidly. Consider the use of wildcards if possible.

- For each exclude filter, multiple values are allowed. The logic between these is AND logic. For example:

  ```
  --START TRACE(PERFM) CLASS(3) XPLAN(A,B)
  ```

  will write performance trace records if the PLAN is not A AND is NOT B.

Volume Control – Filtering

- Filtering is binary decision to externalize the entire record or to discard it. The state of the thread attempting to write the record is all that is considered. The content of the record does not apply.

  Note: Package filters do NOT filter content from IFCID239. Filtering content may be considered in a future release of DB2.

- Filtering only applies to records that are externalized. IFCIDs that are global switches (e.g., accounting class 2, 3, 7, 8, IFCID318, IFCID400 etc.) cannot be filtered.

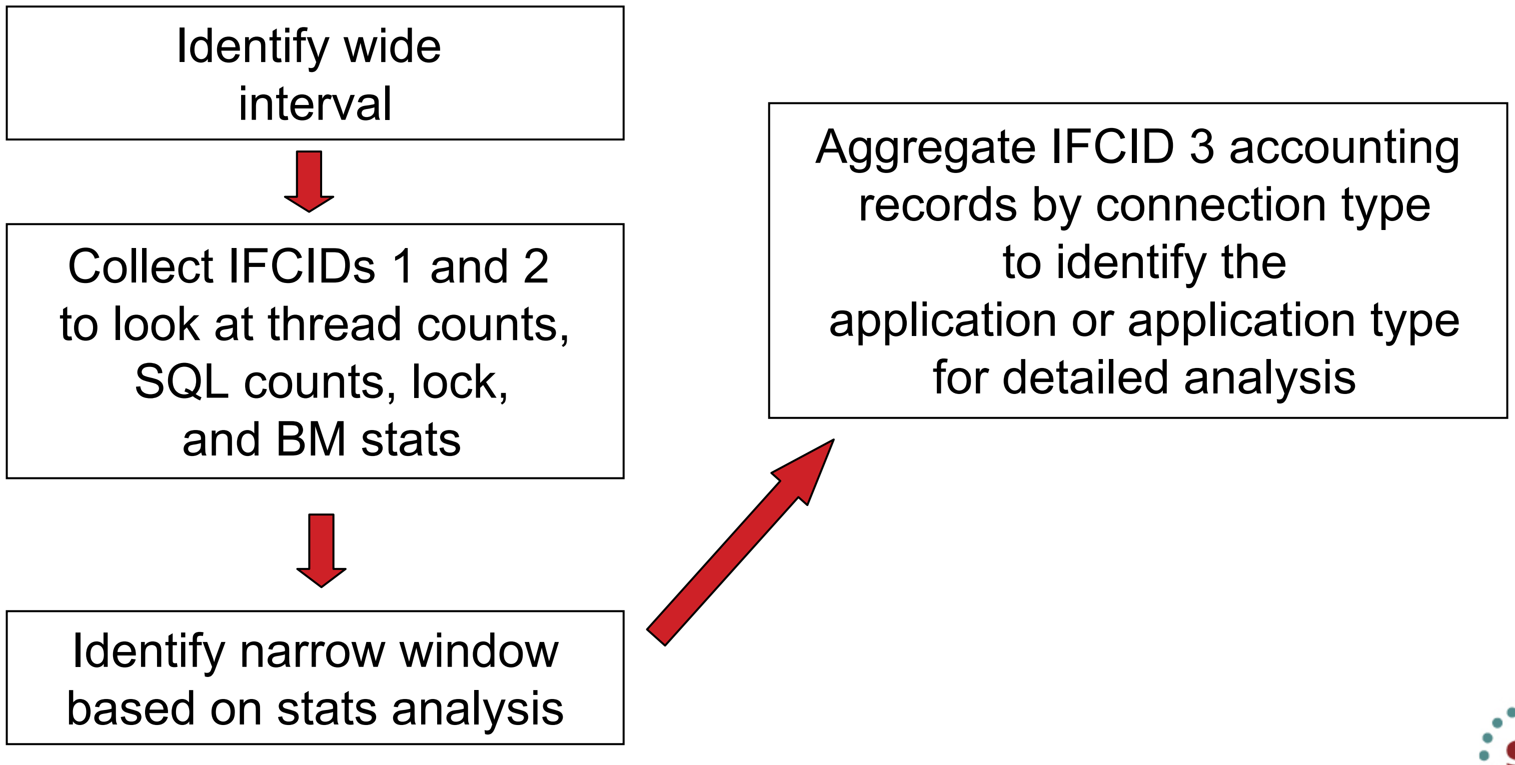
- Filtering is applied during the build phase of each record. Depending on the record, there is still some overhead for records that are discarded. The savings are primarily volume control and a % of the CPU for an unfiltered trace.

Agenda

- Introduction to instrumentation
- Instrumentation volume control
  - Accounting rollup
  - SMF compression
  - Trace Filtering
- **Aggregated performance statistics**
- Statement level analysis
- Stored procedure analysis
- Auditing

Aggregated Performance Statistics – Combining Accounting and Statistics

- Performance analysis and monitoring is typically done by a combination of statistics and accounting
- Commonly use system wide counters as heuristics to ID interval where wait times or CPU usage would have increased (e.g., I see a 2x inflow of DDF threads so this may be a time where CPU spiked).



Identify wide interval

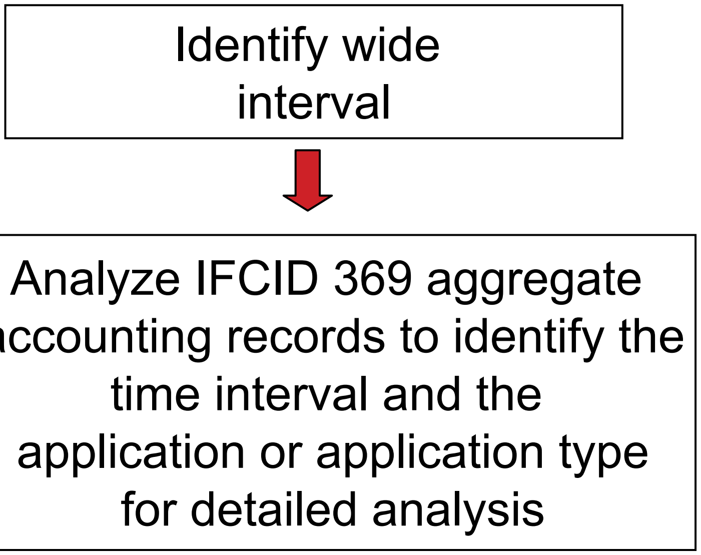
Collect IFCIDs 1 and 2 to look at thread counts, SQL counts, lock, and BM stats

Identify narrow window based on stats analysis

Aggregate IFCID 3 accounting records by connection type to identify the application or application type for detailed analysis

Aggregated Performance Statistics – Combining Accounting and Statistics

- PM62797 simplifies the process where we can look at aggregated accounting data in a single snapshot view. With IFCID369 (stats class 9) accounting wait and CPU data is aggregated per connection type.
- Allows for us to positively ID the interval with high wait times and CPU and then attempt to ID the cause rather than the reverse.


Identify wide interval
Analyze IFCID 369 aggregate accounting records to identify the time interval and the application or application type for detailed analysis

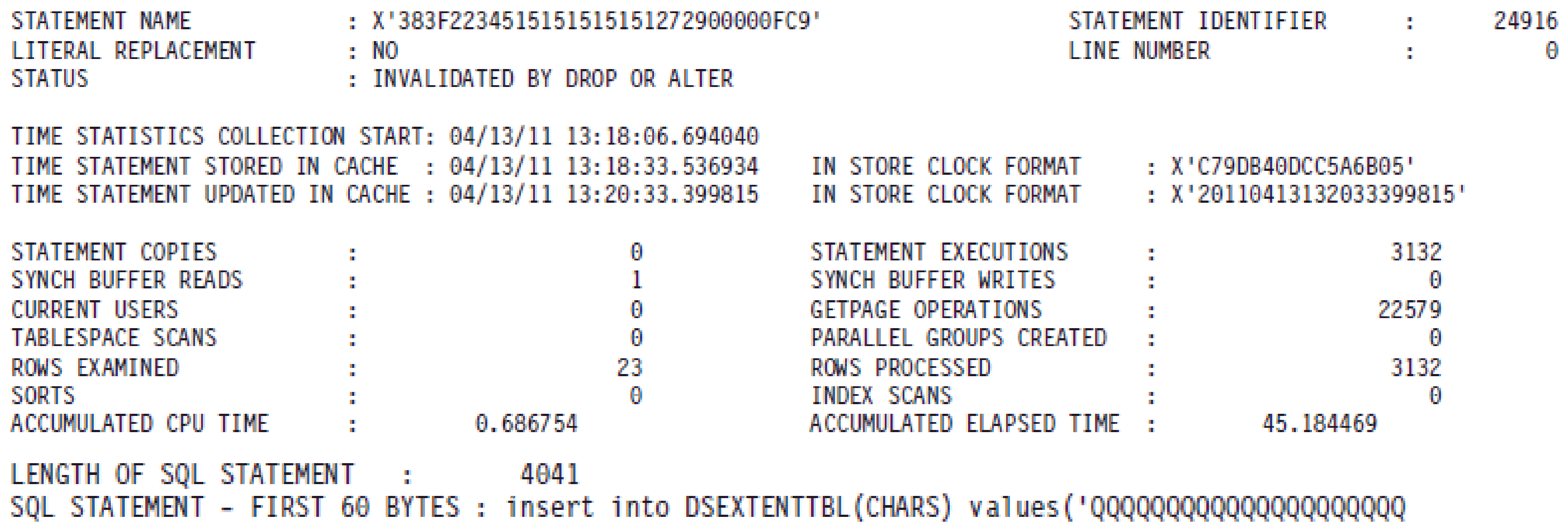
Agenda

- Introduction to instrumentation
- Instrumentation volume control
  - Accounting rollup
  - SMF compression
  - Trace Filtering
- Aggregated performance statistics
- **Statement level analysis**
- Stored procedure analysis
- Auditing

Statement Level Statistics

- Enhance messages and traces to capture statement level information
- Externalize in THREAD INFO messages for deadlocks, timeouts, unavailable resource and lock escalation
- IFCIDs enhanced for statement level information
  - IFCID 53/58 – includes per statement execution statistics
  - IFCID 63/350 – includes STMT_ID with statement text
  - IFCID 124 – statement level accounting
- Monitor Class 29 (overhead is ~1-3%)
  - New for statement level detail
  - Activates IFCID 318 - IFCID 316 information can be collected via READS
  - IFCID 401 for static SQL
- Statement information externalized in real-time
  - STMT_ID - unique statement identifier assigned when statement first inserted into dynamic statement cache
  - Statement Type – static or dynamic
  - Bind timestamp – 10 byte timestamp when stmt was prepared
  - Statement level execution statistics (per execution)

Statement Level Statistics Continued…



```
STATEMENT NAME           : X'383F223451515151515151272900000FC9'           STATEMENT IDENTIFIER    :      24916
LITERAL REPLACEMENT      : NO                                              LINE NUMBER             :          0
STATUS                   : INVALIDATED BY DROP OR ALTER

TIME STATISTICS COLLECTION START: 04/13/11 13:18:06.694040
TIME STATEMENT STORED IN CACHE  : 04/13/11 13:18:33.536934   IN STORE CLOCK FORMAT     : X'C79DB40DCC5A6B05'
TIME STATEMENT UPDATED IN CACHE : 04/13/11 13:20:33.399815   IN STORE CLOCK FORMAT     : X'20110413132033399815'

STATEMENT COPIES         :                    0              STATEMENT EXECUTIONS      :                3132
SYNCH BUFFER READS       :                    1              SYNCH BUFFER WRITES       :                   0
CURRENT USERS            :                    0              GETPAGE OPERATIONS        :               22579
TABLESPACE SCANS         :                    0              PARALLEL GROUPS CREATED   :                   0
ROWS EXAMINED            :                   23              ROWS PROCESSED            :                3132
SORTS                    :                    0              INDEX SCANS               :                   0
ACCUMULATED CPU TIME     :          0.686754                 ACCUMULATED ELAPSED TIME  :         45.184469

LENGTH OF SQL STATEMENT   :       4041
SQL STATEMENT - FIRST 60 BYTES : insert into DSEXTENTTBL(CHARS) values('QQQQQQQQQQQQQQQQQQQQQ
```

It is important to note that statement level stats are collected at different intervals than package accounting or plan level accounting. The sum total of statement level stats for all statements in a package will be < the package totals. Deviations are caused by 1) point of collection 2) transaction termination 3) COMMIT.

Agenda

- Introduction to instrumentation
- Instrumentation volume control
  - Accounting rollup
  - SMF compression
  - Trace Filtering
- Aggregated performance statistics
- Statement level analysis
- **Stored procedure analysis**
- Auditing

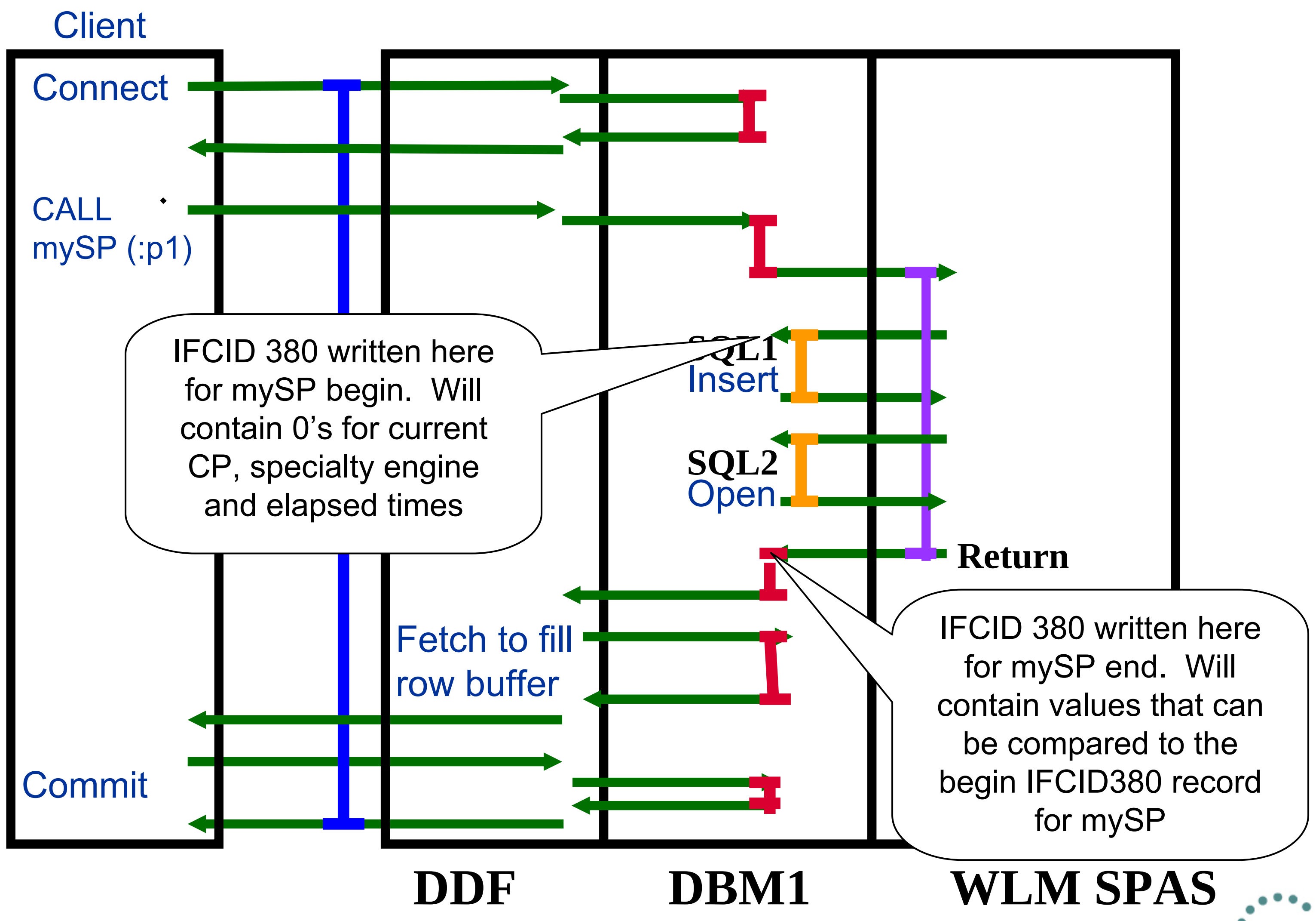
Stored Procedure Performance Analysis – Enhanced Instrumentation

- SP and UDF performance and tuning analysis has typically been performed via a combination of IFCID3 and IFCID239. IFCID3 provides plan level information and aggregates all executions of SPs or UDFs into common fields. This can create difficulty when tuning multiple procedures or functions that are executed in a given transaction. IFCID239 is also used for performance and tuning analysis at the package level. This provides better granularity than IFCID3 but still may not be sufficient for all transactions. If a procedure or function is executed multiple times, the variation between executions cannot be identified.

- 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

Stored Procedure Performance Analysis – Enhanced Instrumentation


Client
Connect
CALL mySP (:p1)
SQL1 Insert
SQL2 Open
Return
Fetch to fill row buffer
Commit
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 IFCID380 record for mySP
DDF
DBM1
WLM SPAS

Agenda

- Introduction to instrumentation
- Instrumentation volume control
  - Accounting rollup
  - SMF compression
  - Trace Filtering
- Aggregated performance statistics
- Statement level analysis
- Stored procedure analysis
- **Auditing**

Audit Policies – Satisfy Your Auditor

- New audit policies provide needed flexibility and functionality
- Auditor can define an audit policy to audit any access to specific tables for specific programs during day
  - Audit policy does not require AUDIT clause to be specified using DDL
  - An audit policy generates records for all read and update access for statements with unique statement identifier
  - Audit policy provides wildcarding of based on schema and table names
- Auditor can define an audit policy to identify any unusual use of a privileged authority
  - Records each use of a system authority
  - Audit records written only when authority is used for access
  - External collectors only report users with a system authority

Audit Policies – Exploitation

- Security administrator using the new SECADM authority maintains DB2 audit policies in a new catalog table
  - SYSIBM.SYSAUDITPOLICIES
- Audit policies enabled using –START TRACE command
- Audit policies disabled using –STOP TRACE command
- Up to 8 audit policies can be specified to auto start or auto start as secure during DB2 start up
- Only user with SECADM authority can stop a secure audit policy trace (APAR PM28296)

Audit Policies – Categories

Categories	Mapping IFCIDs
❖ **CHECKING**	❖ **IFCID 83 (**only authentication failures**), IFCID 140**
❖ **VALIDATE**	❖ **IFCIDs 55, 83, 87, 169, 269, 319**
❖ **OBJMAINT**	❖ **IFCID 142**
❖ **EXECUTE**	❖ **IFCIDs 143, 144, 145**
❖ **CONTEXT**	❖ **IFCIDs 23, 24, 25**
❖ **SECMAINT**	❖ **IFCIDs 141, 270, 271**
❖ **SYSADMIN**	❖ **IFCID 361** (Audits installation SYSADM, installation SYSOPR, SYSOPR, SYSCTRL, SYSADM)
❖ **DBADMIN**	❖ **IFCID 361** (Audits DBMAINT, DBCTRL, DBADM, PACKADM, SQLADM, system DBADM, DATAACCESS, ACCESSCTRL, SECADM)

Audit Policies – Example Dynamic Auditing of Tables

- Audit all the tables that start with 'PAY' in EMPLOYEE schema
  - Does not require AUDIT clause to be specified during table definition
  - IFCID 145 trace record contains full SQL statement text and unique statement ID
  - IFCID 143 and 144 trace records contain the unique statement ID that can be used to identify the SQL statement in IFCID 145 record.

```
INSERT INTO SYSIBM.SYSAUDITPOLICIES (AUDITPOLICYNAME,
OBJECTSCHEMA, OBJECTNAME, OBJECTTYPE, EXECUTE)
  VALUES ('TABADT1',‘EMPLOYEE',''’PAY%’'','T','A');

-STA TRACE (AUDIT) DEST (GTF) AUDTPLCY(TABADT1);
```

Thank you !

John Tobler (jtobler@us.ibm.com)

Appendix

- Global switch IFCIDs. Filtering and destinations for these do not apply
  - Accounting and monitor classes 2, 3, 7, 8, 10
  - IFCIDs 318, 400

- IFCIDs available only via IFI READS requests. These cannot be directed to SMF, GTF, or OP buffer destinations.
  - IFCIDs 124, 129, 147, 148, 149, 150, 185, 187, 197, 306

    Note: IFCID 316 and IFCID 401 will only be written to SMF, GTF, or an OP destination if a dynamic statement or static statement is evicted from a cache. To view the entire cache, an IFI READS request must be made.