Minimize your trips to DB2

Linda Claussen

lclaussen@themisinc.com

www.themisinc.com/webinars

Themis inc.

Questions?

- You can submit questions by typing into the questions area of your webinar control panel.
- Questions will be answered as time permits.
- Any questions not answered due to time constraints can be answered afterward via an email.

DB3052 DB2 for z/OS Database Performance Tuning
Public course offering: 10/11/2016

Agenda

- ✓ Identify Threads with high DB2 Entry/Exit Events
- ✓ Coding Techniques to minimize trips to DB2
- ✓ Optimized Coding techniques unique to DBAT threads
- ✓ SQL and Bind options to optimize multi-row processing

Application Elapsed Time

Application

- Thread Allocation
- Program A
  - Statement 1
  - Statement 2
  - Statement 3
- Program B
  - Statement 1
  - Statement 2
- Thread Deallocation, goes Inactive, Authid Changes or ROLLUP

DB2

- Plan
- Package A
  - Statement 1
  - Statement 2
  - Statement 3
- Package B
  - Statement 1
  - Statement 2
- Trace Record Written

Identify Threads with high DB2 Entry/Exit Events

- Long Running Threads
  - DSNB2601I long-running reader
  - DSNJ031I long running updates
- Code Review
- DB2 Accounting Trace
  - High APPL(Class 1) time
  - High number of DB2 (Class 2) ENT/EXIT
  - High number of DB2 (Class 7) ENT/EXIT

Accounting Trace Report

DB2 ENT/EXIT

TIMES/EVENTS	APPL(CL.1)	DB2 (CL.2)
ELAPSED TIME	~	
CP CPU TIME	~	
SE CPU TIME	~	
SUSPEND TIME	~	
NOT ACCOUNT.	~	
DB2 ENT/EXIT	N/A	#####
EN/EX-STPROC	N/A	0
EN/EX-UDF	N/A	0
DCAPT.DESCR.	N/A	N/A
LOG EXTRACT.	N/A	N/A

SQL DML

SQL DML	TOTAL
SELECT	?
INSERT	?
ROWS	0
UPDATE	0
ROWS	0
MERGE	?
DELETE	0
ROWS	0
DESCRIBE	0
DESC.TBL	0
PREPARE	0
OPEN	?
FETCH	?
ROWS	?
CLOSE	?
DML-ALL	#####

Code Review:
Minimizing Trips to DB2

- Avoid unnecessary SQL Statements
- Perform loops
- Application join vs. SQL Join
- Use output only Updates/Deletes where possible
- Use ROWSET processing

Extended Indicator Variables

What do you do when the input columns received for an update or insert change from execution to execution?

- Dynamic SQL?
- Static code for all possibilities?
- Have 1 update or insert that includes all columns each time?

New for Inserts	New for Updates / Merge Updates
-5 = Use default value	-5 = Use default value
-7 = Use default value	-7 = No Update

Bind / Rebind EXTENDEDINDICATOR (YES)

Prepare WITH EXTENDED INDICATORS

Examples

MOVE -5 (or -7) TO V-ADDRESS2-NI *(no address2, use default)*

```
INSERT INTO EMP (EMPNO, ADDRESS, SALARY)
 VALUE (:V-EMPNO,
            :V-ADDRESS2 :V-ADDRESS2-NI,
            :V-SALARY     :V-SALARY-IND)
```

MOVE 0 TO V-ADDRESS2-NI (*address2 changed*)
MOVE -7 TO V-SALARY-NI *(salary not changed, skip column)*

```
UPDATE EMP
SET ADDRESS2  = :V-ADDRESS2 :V-ADDRESS2-NI,
      SALARY      = :V-SALARY      :V-SALARY-NI
WHERE EMPNO = :EMPNO
```

ROWSET POSITIONING

- Reduce DB2 Entry/Exit using ROWSET POSITIONING
  - Use of Arrays
  - Multi-Row Fetch
  - Multi-Row Insert
- Reduce database calls using the MERGE (upsert) and Multi-Row Fetch
- SELECT from INSERT, UPDATE, DELETE or MERGE
- Use of GET DIAGNOSTICS

Multi-row Fetch

- **1 Trip returns 100 rows**

```
EXEC SQL
  DECLARE CURSOR1  CURSOR WITH ROWSET POSITIONING FOR
              SELECT   COL1,  COL2,  COL3
                 FROM  TABLE1
END-EXEC
MOVE   +100   TO   N
EXEC SQL
    FETCH  NEXT ROWSET FROM CURSOR1 FOR :N ROWS  INTO
      :COL1-ARRAY
   ,  :COL2-ARRAY
   ,  :COL3-ARRAY   :COL3-IND-ARRAY
END-EXEC
```

Multi-Row Fetch

```
    EXEC SQL
      DECLARE TABLE1 TABLE
       ( COL1   CHAR(100)
        ,COL2   DECIMAL(9,2)
        ,COL3   DATE)
    END-EXEC .
01   WORK-AREAS.
    03   COL1-ARRAY        OCCURS 100 TIMES  PIC X(100).
    03   COL2-ARRAY        OCCURS 100 TIMES  PIC S9(07)V9(02)  COMP-3.
    03   COL3-ARRAY        OCCURS 100 TIMES  PIC X(10).
    03   COL3-IND-ARRAY    OCCURS 100 TIMES  PIC S9(04) COMP.
    03   N                                    PIC S9(09) COMP.
```

```
EXEC SQL
DECLARE CURSOR1  CURSOR WITH ROWSET POSITIONING FOR
            SELECT     COL1,  COL2,  COL3
            FROM       TABLE1
END-EXEC
MOVE   +100   TO   N
EXEC SQL
   FETCH  NEXT ROWSET FROM CURSOR1 FOR :N ROWS  INTO
    :COL1-ARRAY ,  :COL2-ARRAY ,  :COL3-ARRAY   :COL3-IND-ARRAY
END-EXEC.
```

Themis inc.

Looping with Multi-Row Fetch

```
 PERFORM UNTIL EOF-FLG = 'Y'
    EXEC SQL FETCH NEXT ROWSET FROM MRFCUR FOR 10 ROWS
      INTO :hv1, :hv2, :hv3
    END-EXEC
    MOVE SQLERRD(3) TO WS-RETURNED
    MOVE SQLCODE TO WS-SQLCODE
    IF WS-SQLCODE = 100
       MOVE 'Y' TO EOF-FLG
    ELSE
       IF WS-SQLCODE NOT = 0
          PERFORM Z100-DB2-ERROR
       END-IF
    END-IF

******* LOOP WITHIN THE LOOP
    PERFORM VARYING SUB1 FROM 1 BY 1 UNTIL SUB1 >
                WS-RETURNED
          DISPLAY hv1(SUB1) hv2(SUB1) hv3(SUB1)
    END-PERFORM

END-PERFORM.
```

Multi-Row Insert

```
WORKING-STORAGE SECTION.
01  NUMBER-TO-INSERT                PIC S9(04) COMP.
01  COL1-ROWSET-ARRAY               OCCURS 10   PIC .....
01  COL2-ROWSET-ARRAY               OCCURS 10   PIC .....
01  COL3-ROWSET-ARRAY               OCCURS 10   PIC .....
01  INSERT-ROWSET-INDICATORS OCCURS 10   PIC .....

PROCEDURE DIVISION.
Populate arrays with values to insert
Set indicator variable elements as required
MOVE +10  TO  NUMBER-TO-INSERT
EXEC SQL
INSERT INTO TABLEA
  (COL1, COL2, COL3)
    VALUES   ( :COL1-ROWSET-ARRAY
               ,:COL2-ROWSET-ARRAY
               ,:COL3-ROWSET-ARRAY   :INSERT-ROWSET-INDICATORS
               )
FOR :NUMBER-TO-INSERT ROWS
NOT ATOMIC CONTINUE ON SQLEXCEPTION
END-EXEC
```

Themis inc.

Insert Processing comparison

100 Trips to DB2

```
01 TABLE_DCLGEN
~
PERFORM ~INS-LOOP 100 TIMES
~
INS-LOOP.
Read  1 input record &
  move data to columns
INSERT INTO TABLEA
 (COL1, COL2, COL3)
VALUES
 ( :COL1-ROWSET-ARRAY   ~   )
```

1 Trip to DB2

```
01  NUM-TO-INSERT          PIC S9(04) COMP.
01  COL1-ROWSET-ARRAY   OCCURS 100   PIC ~
~
Read 100  input records & move data to arrays
MOVE +100  TO  NUM-TO-INSERT
~
INSERT INTO TABLEA
 (COL1, COL2, COL3)
VALUES
 ( :COL1-ROWSET-ARRAY   ~   )
FOR :NUM-TO-INSERT ROWS
NOT ATOMIC CONTINUE ON SQLEXCEPTION
~
```

LOAD…… RESUME YES

Merge Processing

2 TRIPS TO DB2

```
SELECT ….
   FROM  T1 ….
IF SQLCODE = 0
  UPDATE  T1 …..
ELSE
IF SQLCODE = +100
  INSERT INTO T1 …
```

1 TRIP TO DB2

```
MERGE INTO T1
  USING (VALUES…)
        AS NEW (…)
 ON I.~ = NEW.~
WHEN MATCHED THEN
  UPDATE …
WHEN NOT MATCHED
THEN
  INSERT …
```

MERGE Statement, 1 trip to DB2


```
EXEC SQL
  MERGE INTO ITEM I
   USING (VALUES(:ITEMNO, :ITEMNAME) FOR 5 ROWS)
      AS NEWITEM ( ITEMNO, ITEMNAME)
   ON  I.ITEMNO = NEWITEM.ITEMNO
  WHEN MATCHED THEN
     UPDATE SET ITEMNAME = NEWITEM.ITEMNAME
  WHEN NOT MATCHED THEN
     INSERT (ITEMNO,ITEMNAME)
      VALUES (NEWITEM.ITEMNO,NEWITEM.ITEMNAME)
  NOT ATOMIC CONTINUE ON SQLEXCEPTION
END-EXEC.
```

GET DIAGNOSTICS Example

Example 1.

```
EXEC SQL
  GET DIAGNOSTICS
        :WS-ROW-COUNT = ROW_COUNT
      , :WS-COND-NUM   = NUMBER
END-EXEC
```

Example 2.

```
EXEC SQL
  GET DIAGNOSTICS      CONDITION 1
        :WS-SQLCODE    =
DB2_RETURNED_SQLCODE
      , :WS-SQLSTATE  = RETURNED_SQLSTATE
END-EXEC
```

Example 3.

```
   EXEC SQL
     GET DIAGNOSTICS
           :WS-TOT-COND = NUMBER
   END-EXEC
PERFORM  VARYING WS-COND-NUM FROM 1 BY 1
              UNTIL WS-COND-NUM IS GREATER THAN WS-TOT-COND
    EXEC SQL
       GET DIAGNOSTICS
         CONDITION :WS-COND-NUM
            :WS-SQLCODE    = DB2_RETURNED_SQLCODE
          , :WS-SQLSTATE  = RETURNED_SQLSTATE
          , :WS-ROW-NUM  = DB2_ROW_NUMBER
    END-EXEC
END-PERFORM
```

Delete Processing

- DELETE … WHERE key=unique_value
- DELETE … WHERE CURRENT OF
- DELETE … WHERE search-condition
- DELETE FROM tbname
- TRUNCATE TABLE tbname
  - IMMEDIATE
  - IGNORE DELETE TRIGGERS
- LOAD …. REPLACE
  //SYSREC DD DUMMY
- REORG …. DISCARD --*will be deprecated in future releases*

Returning Modified Data

- SELECT ….. FROM FINAL TABLE
  - (INSERT INTO tbname ……)
  - (UPDATE tbname ……)
  - (MERGE INTO tbname ……)
- SELECT ….. FROM OLD TABLE
  - (DELETE FROM tbname ……)
  - (UPDATE tbname ……)

Minimizing Trips to DB2, DBAT Threads

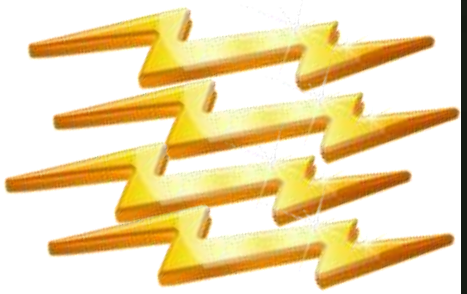
- SQL enabled block fetching for Distributed apps
  FOR FETCH ONLY
  OPTIMIZE FOR n ROWS
  EXTRASRV -- subsystem parameter
- Enable package-based continuous block fetch
  Bind: DBPROTOCOL(DRDACBF).for z/OS client
- FETCH FIRST n ROWS ONLY clause in a distributed query for fast implicit close **Caution!**
- Use V11 Array datatypes
- Move SQL from client to DB2 (Stored Procedures)

Network Traffic

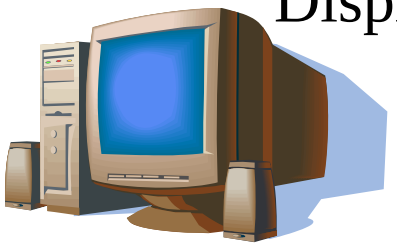
Receive Request

Validation

Get Data
Process Data



Display Result



Themis inc.

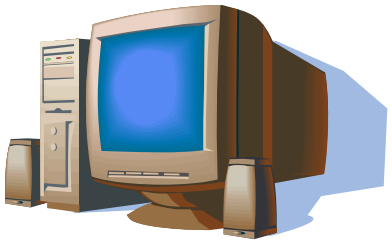
Solution

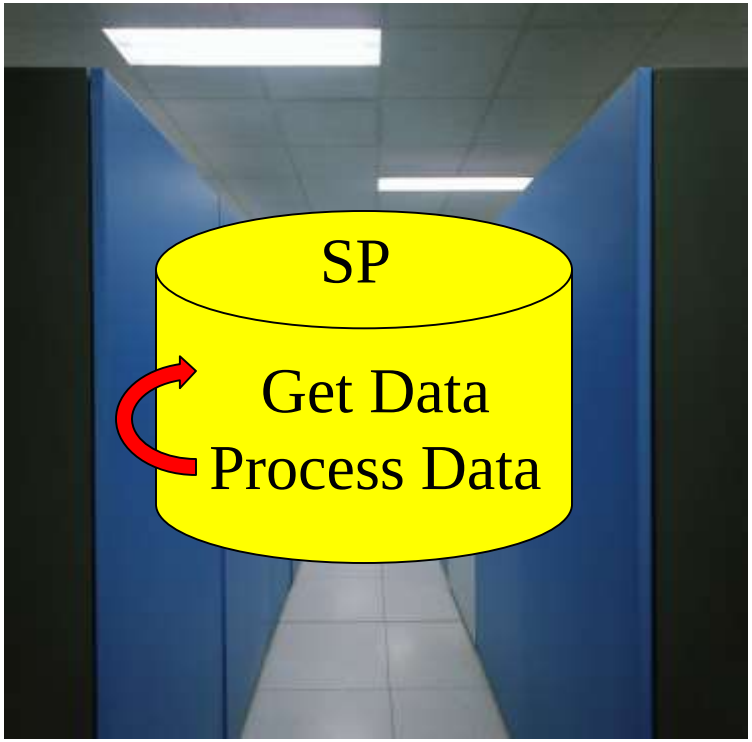
Receive Request

Validation

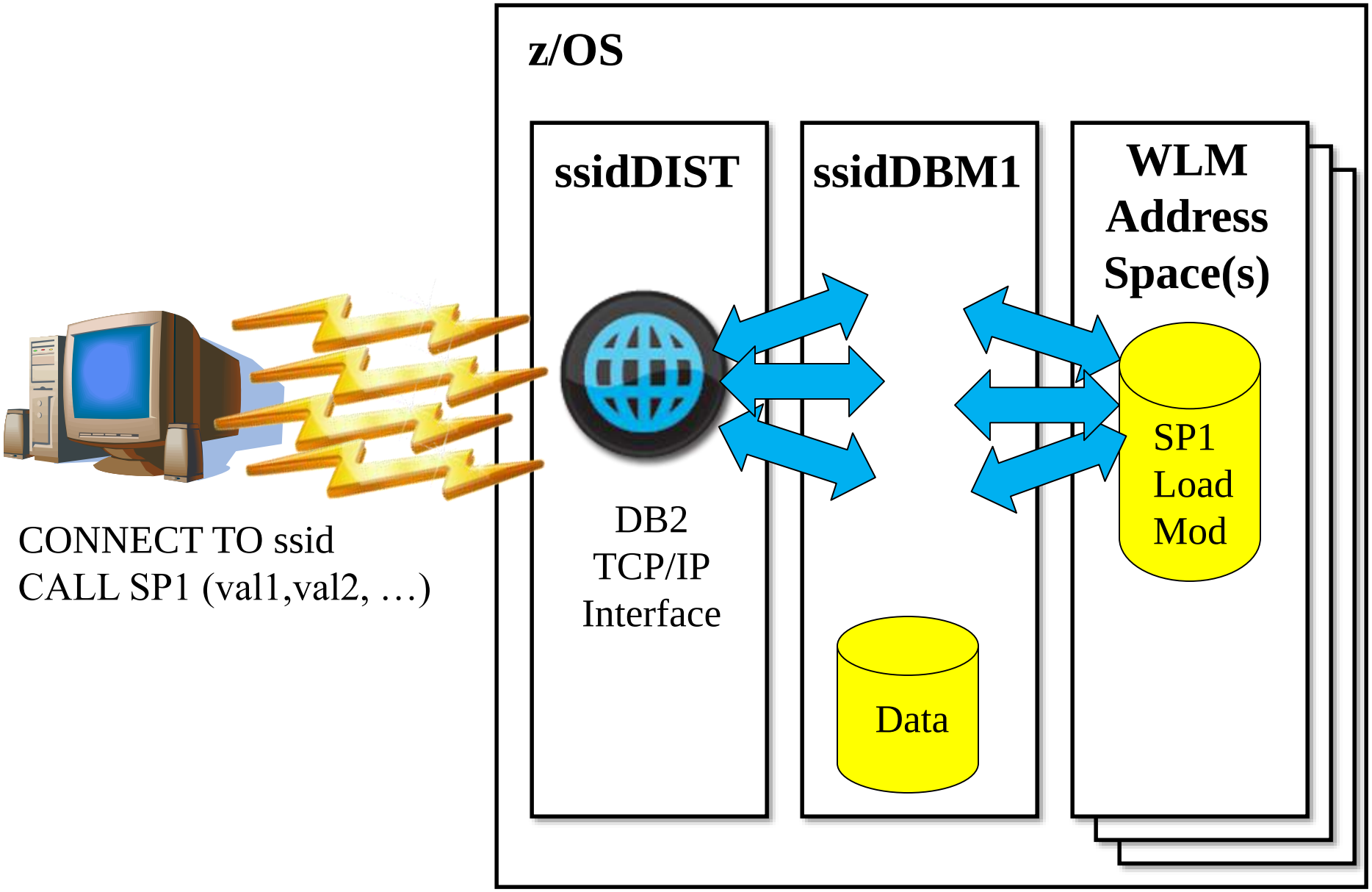
Call Procedure

Display Result

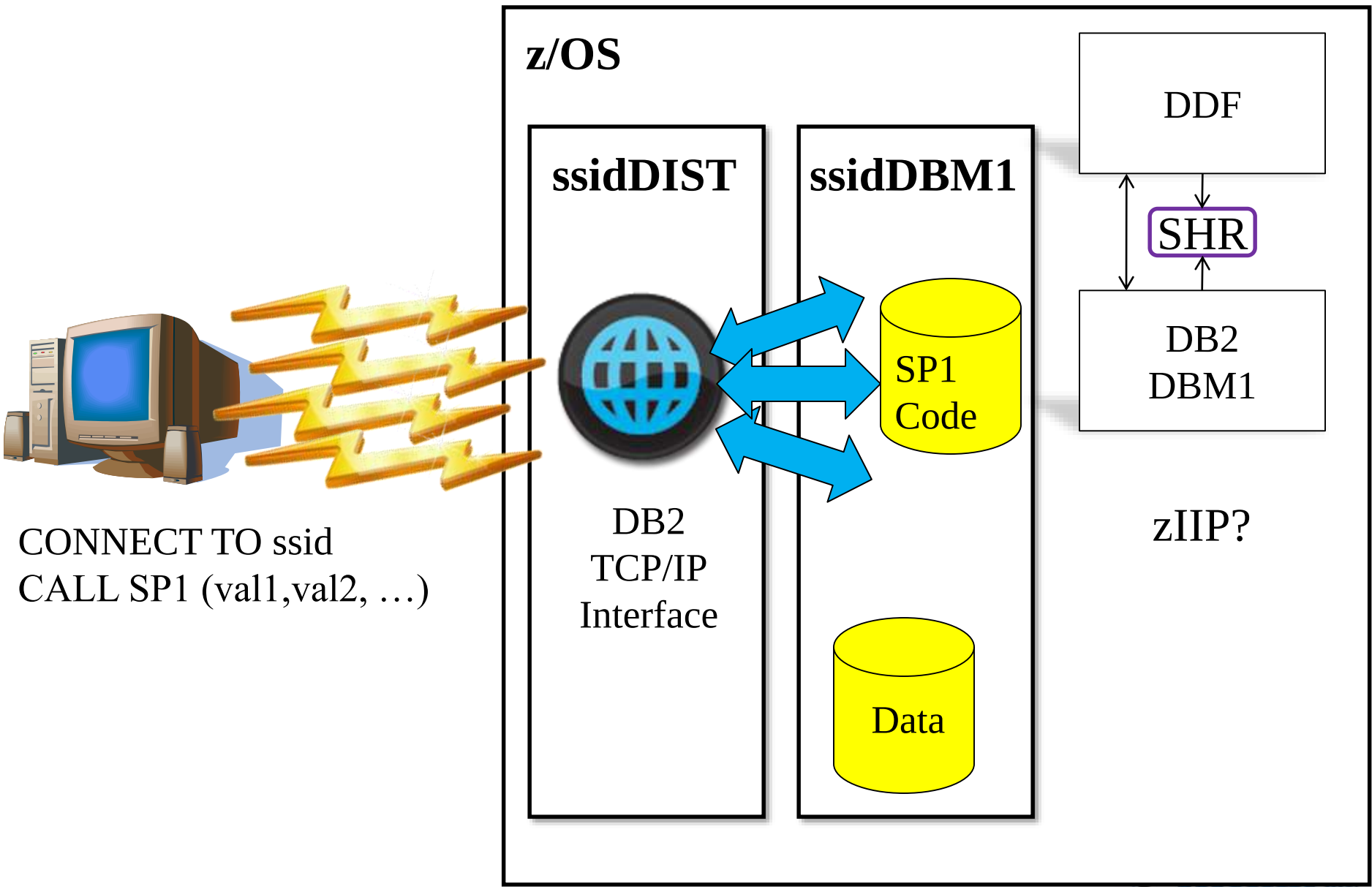




External Stored Procedures



Native Stored Procedures



Bind Options

- BIND/REBIND ……..
- ISOLATION (CS | RR | RS | UR | NC)
- CURRENTDATA (**NO** | YES)
- DBPROTOCOL (DRDA | **DRDACBF**)
- RELEASE (COMMIT | **DEALLOCATE**)
- CONCURRENTACCESSRESOLUTION (USECURRENTLYCOMMITTED | WAITFOROUTCOME)

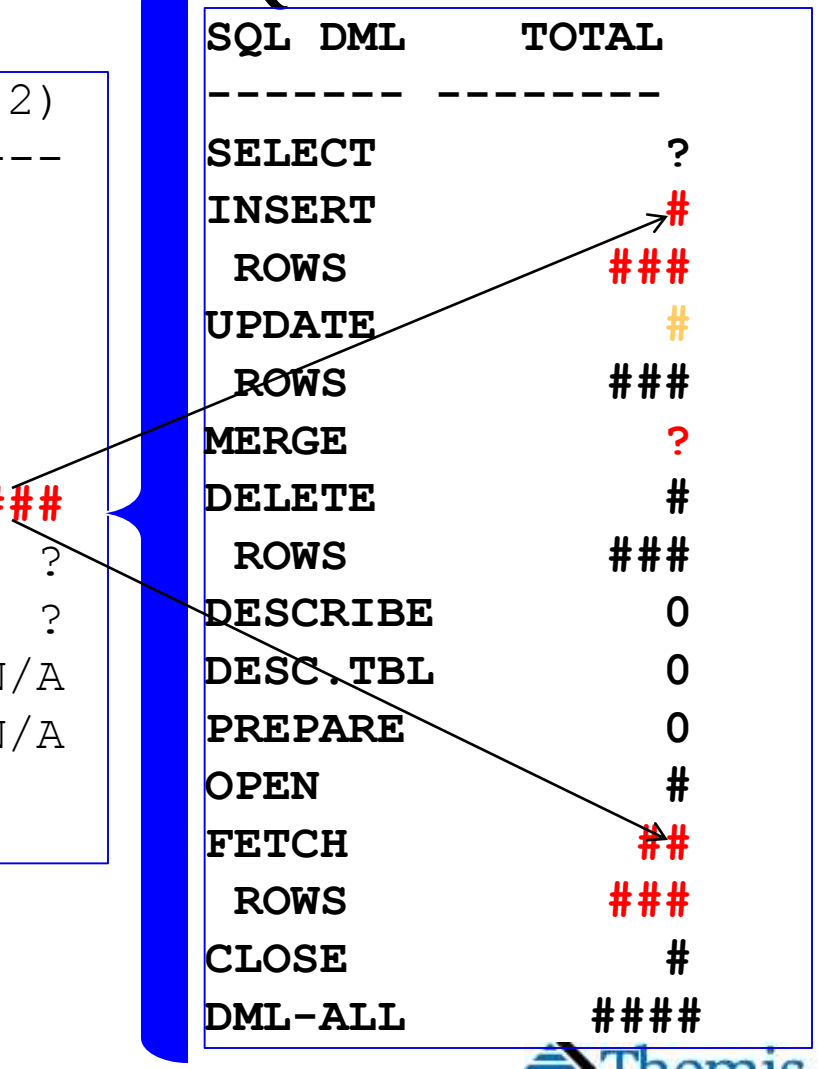
Review Accounting Trace

DB2 ENT/EXIT

TIMES/EVENTS	APPL(CL.1)	DB2 (CL.2)
ELAPSED TIME	~	
CP CPU TIME	~	
SE CPU TIME	~	
SUSPEND TIME	~	
NOT ACCOUNT.	~	
DB2 ENT/EXIT	N/A	###
EN/EX-STPROC	N/A	?
EN/EX-UDF	N/A	?
DCAPT.DESCR.	N/A	N/A
LOG EXTRACT.	N/A	N/A

SQL DML

SQL DML	TOTAL
SELECT	?
INSERT	#
ROWS	###
UPDATE	#
ROWS	###
MERGE	?
DELETE	#
ROWS	###
DESCRIBE	0
DESC.TBL	0
PREPARE	0
OPEN	#
FETCH	##
ROWS	###
CLOSE	#
DML-ALL	####



Themis inc.

Sample Run Time Results Comparisons

- Run: ODYLC.SP5N00:V1(CHAR(3), INTEGER)
  - Base table cursor
  - Fetch/Insert into DGTT
  - Open Cursor on DGTT
  - Query execution time => 820 ms
- Run: ODYLC.SP5N00:V2(CHAR(3), INTEGER)
  - Mass insert into DGTT from Base Table
  - Open Cursor on DGTT
  - Query execution time => 570 ms
- Run: ODYLC.SP5N00:V3(CHAR(3), INTEGER)
  - Cursor with Return on Base Table
  - Query execution time => 290 ms

Minimizing Trips to DB2 Recap

- Review Long running units of work (DSN messages)
- CODE Reviews
  - Avoid unnecessary SQL Statements
  - Perform loops
  - Application join vs. SQL Join
  - Use output only Updates/Deletes where possible
  - Use ROWSET processing
- Accounting Trace
  - ENT/EXIT Events
  - Compare to SQL DML
  - Plus overall performance
- OPEN Discussion/Questions

Recommendations

1. Identify Threads with high DB2 ENT/EXIT Events
2. Identify Long Running Threads
3. Implement standardized Code Reviews
4. Ban the words ***NEVER & ALWAYS***
5. Choose more than 1 solution
6. Test each solution
7. Implement

Visit www.themisinc.com
for all your training needs

Follow @ThemisTraining

DB3052 DB2 for z/OS Database Performance Tuning
Public course offering: 10/11/2016

Linda F. Claussen
Themis, Inc
lclaussen@themisinc.com
www.themisinc.com/webinar

Themis inc.

Themis On-site & Public, Instructor-led, Hands-on IT Training

800.756.3000Â FREE
(International 01) 908.233.8900

Home | Courses | Schedules | Locations | Training Services | Content | Webinars | Course Request | Contact Us

Themis. Leaders in IT Training!

Two DB2 boot camps starting soon! Click on the links below or call 800-756-3000Â FREE for details.

DB2 for z/OS Database Administration

DB2 for LUW Database Administration

Follow @ThemisTraining

Themis Training Services

Flexible training options designed to meet the needs of your organization.

- On-Site Training
- Public Class Training
- Consortium Training
- Local Training
- Hosting
- Immersion Training
- Short Notice Training
- Savings Plans

Call your Themis Account Executive for details. 800.756.3000Â FREE

Over 400 IT Courses
Search for Yours:

Search

Want to stay ahead and work smarter? Trust Themis to make it happen. Nationwide. On-Site or Public. Courses designed and taught by fulltime, internationally-recognized instructors with real-world answers.

DB2 for z/OS
DB2 for Linux, UNIX, Windows
Structured Query Language (SQL)
Java, Java EE, Python, Frameworks and OO Design
IBM MQ
Microsoft - .NET, SharePoint, SQL Server
Oracle
Big Data (Cognos, Hadoop)
CICS/TS
Mainframe - Assembler, COBOL, z/OS JCL, REXX, SAS
Web Application Development
WebLogic
IMS
Linux
UNIX, C, C++
PowerBuilder / InfoMaker
MySQL
Project Management
Agile

Tweets by @ThemisTraining

Themis Education @ThemisTraining

Themis inc.