

DB2 for z/OS Profiles: Customer Experience DB2 Connection Profile to the Rescue

*William Leininger
PNC Financial Services*

*Mark Rader
IBM ATS*

*August 5, 2014
Session 15937*

Test link: www.SHARE.org



#SHAREorg



PNC Bank



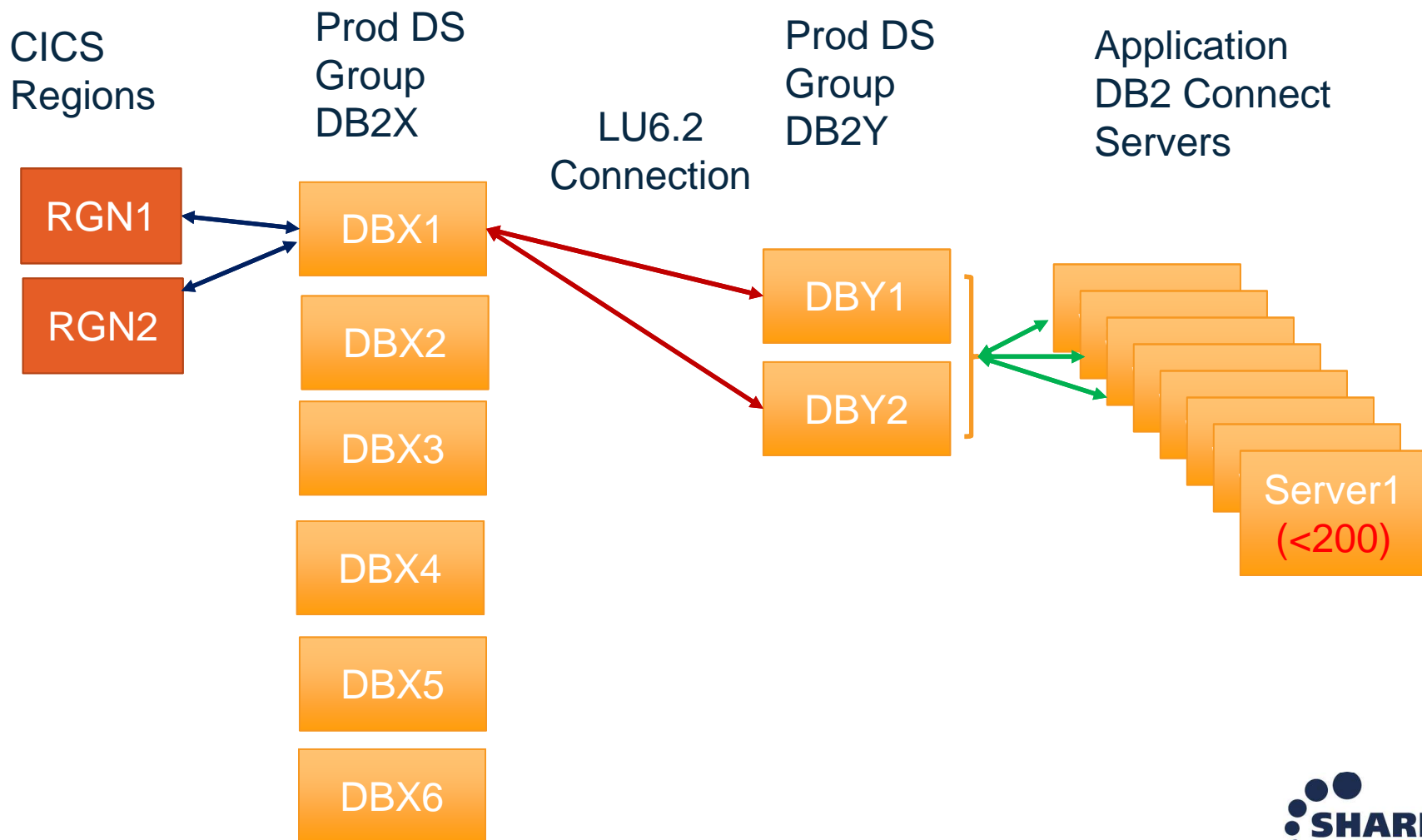
- Acquisitions since 2003
 - 2003 - UNB
 - 2005 - Riggs
 - 2007 - Mercantile
 - 2008 - Yardville, Sterling
 - 2008 - National City
 - 2011 - RBC Bank
 - 2011 – Flagstar (27 Branches)



Description of Environment

- One DSG (DB2X) support home grown applications
 - 6-way datasharing group
 - CICS activity from regions on 3 LPARs
 - TCP/IP DDF inbound activity
 - LU 6.2 outbound activity to DSG(DB2Y) vendor application
- DSG (DB2Y) supports vendor application
 - 2-way datasharing group
 - NO CICS
 - DDF via TCP/IP
 - DDF via LU 6.2 from DB2X

Configuration



DB2-to-DB2 connections

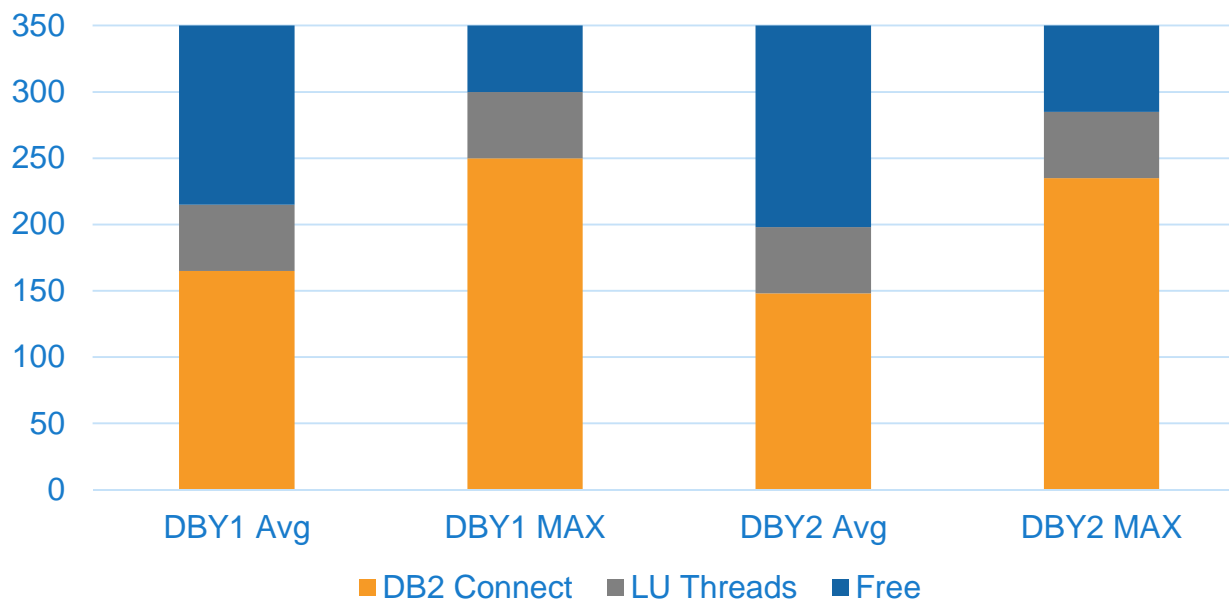
- SYSDDF tables
 - LOCATIONS
 - LUNAME
 - LULIST with both LU names for DB2Y in DB2X table
 - LULIST with all LU names for DB2X in DB2Y table
- DB2 Connect Servers
 - Cluster of servers
 - Spread work
 - Setup so subset can handle all work
 - **Means total threads can be > MAXDBAT!!**

DB2 Configuration Settings

- DB2X
 - MAXDBAT=200
 - IDTHTOIN=0 → not really relevant to problem
- DB2Y
 - MAXDBAT=350 (350+350=700)
 - IDTHTOIN=600
 - MAXDBAT ** Set in V8
 - MAXDBAT ** V9
- DB2 Connect Servers
 - MAX Connections=200 (8*200=1600)

DB2Y Threads

- Trends for DB2Y
 - DBY1 - Avg Threads=165 - Avg Max 250
 - DBY2 – Avg Threads=148 - Avg Max 235



DB2Y – What could go wrong

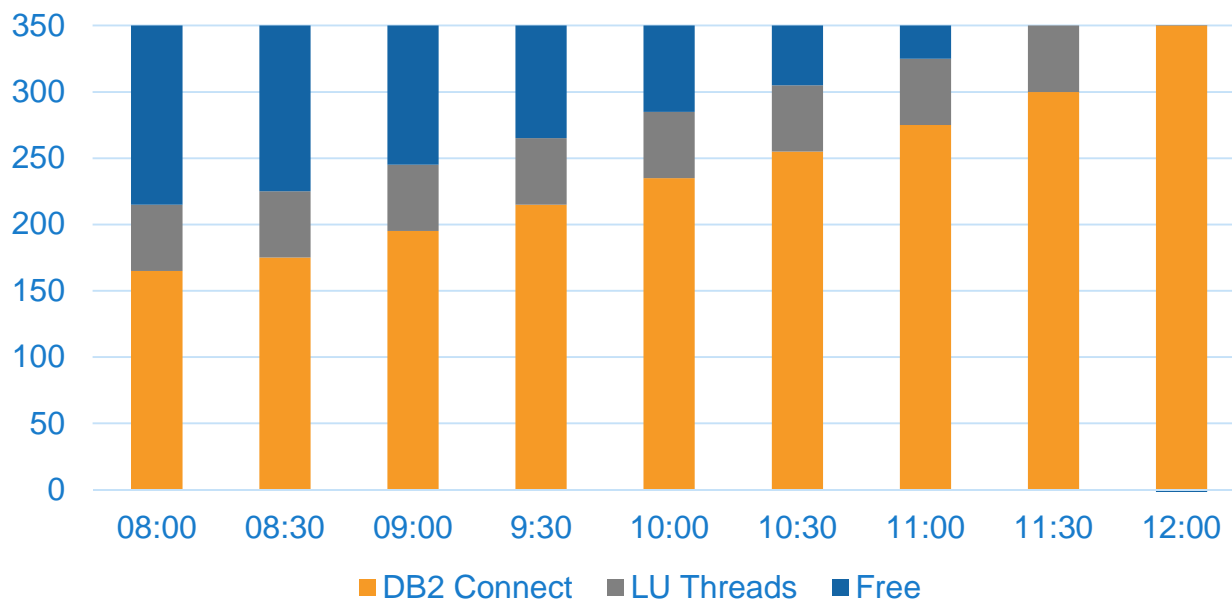
- Diligent monitoring of AVG and MAX threads
- Monitoring was to anticipate need for 3-way DS
- Occasionally DB2Y would spike to 350 for short periods
- Prior to 2014, we looked at options
 1. Evaluate DB2X->DB2Y access
 2. Expand DB2Y to 3-way
 3. Replicate data from DB2Y to DB2X to eliminate LU6.2
- Few occurrences not worth “cost”

DB2Y – What just happened!!!

- DB2 V8/9 – MAXDBAT=350
- Upgrade to DB2 V10 2013 ... Increase MAXDBAT??
- Review potential to increase MAXDBAT
- Dec 2013 - Server Problem ->threads started... not end
- DBY1 & DBY2 go to MAXDBAT
- What happens when MAXDBAT reached??
- Connections from DBX1/CICS ...
- CICS

DB2Y Threads

- When TCP/IP threads do not end
 - IP threads count grows
 - LU6.2 threads get “squeezed” out



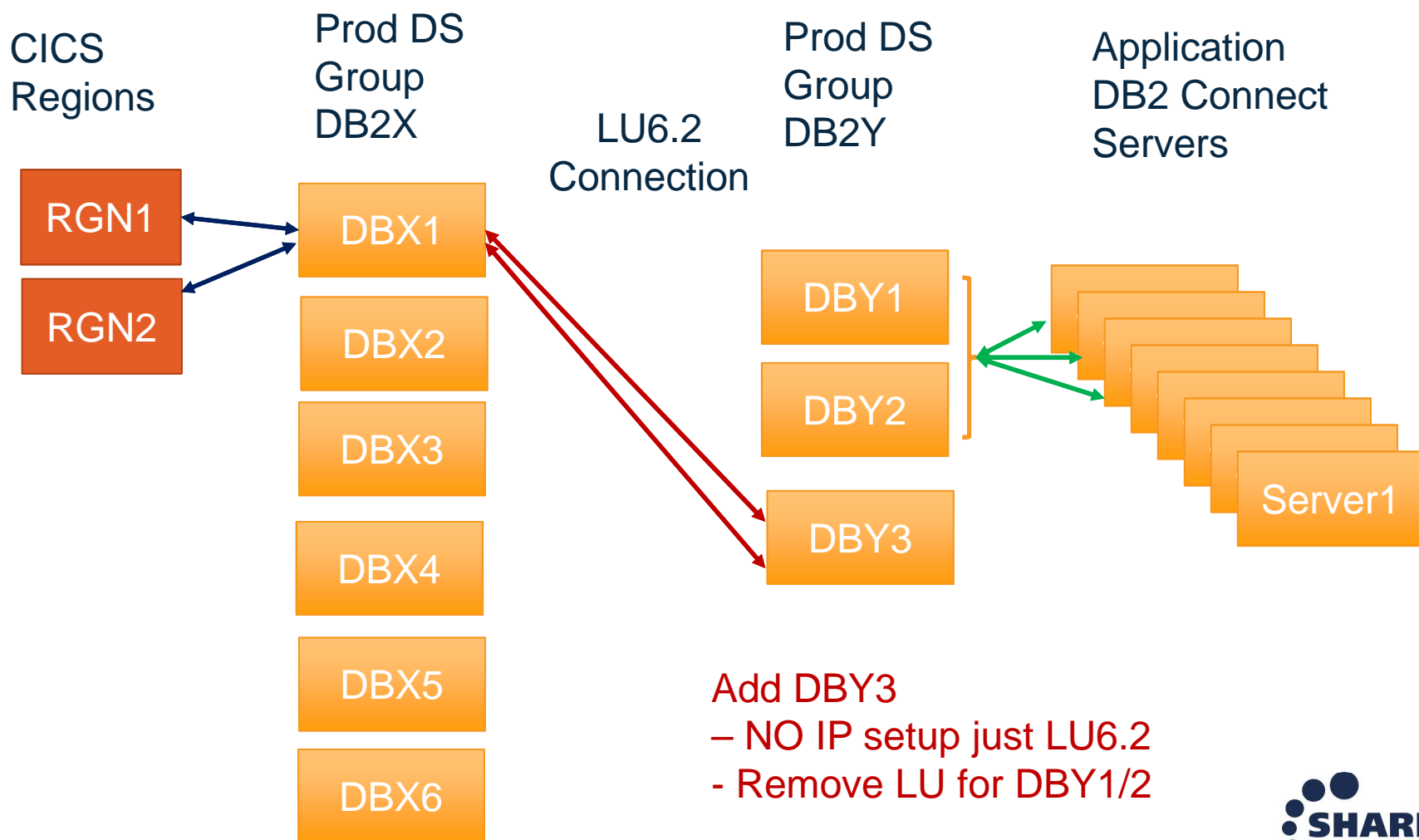
DB2Y – First Reaction

- MAXDBAT analysis ... increase MAXDBAT
 - Option 1 -> increase MAXDBAT dynamically
 - Relieved pressure for short time
 - Option 2 -> Lower Idle Thread Timeout 10 min -> 2 min
-
- MAXDBAT=450
 - MAXDBAT=700
 - MAXDBAT=800
 - MAXDBAT=1000
 - MAXDBAT=1200

DB2Y problems ... again

- Application Issue
- Another Server Issue
- Caused DB2Y to hit MAXDBAT
- MAXDBAT on DB2Y ... caused MAXTASK in CICS
- MAXTASK caused massive slowdowns
- Application issues
- Time for another look at options
 1. Replicate data from DB2Y to DB2X to eliminate LU6.2
 2. Expand DB2Y to 3-way (or 4-way) and put LU 6.2 to 'DBY3' (and 'DBY4'), keep IP traffic to DBY1 and DBY2
 3. CALL IBM

Alternate Configuration



Option #1

- Data replication from DB2Y->DB2X –NOT an OPTION!!
- Starting DBY3 was contemplated
- Meant another subsystem member
- Might require -
 - Creating Subsystem Alias
 - REBINDing PLANS
 - Modify COBOL that issued CONNECT
 - Modifying ALIAS with 3-part name to DB2Y

Option #2

- Contact IBM and got meeting setup
- After listening to the problem, solution was
- Use DB2 PROFILES
- Skeptical to say the least -> Too Simple
- If this worked, it would mean
 - No new subsystem
 - No VIEW or ALIAS changes
 - No subsystem ALIAS
 - No COBOL changes
 - TOO GOOD TO BE TRUE

DB2 PROFILE

- 4 Tables
 - DSN_PROFILE_TABLE
 - DSN_PROFILE_ATTRIBUTES
 - DSN_PROFILE_HISTORY
 - DSN_PROFILE_ATTRIBUTES_HISTORY
- Define tables
- Load data to define profile
- Issue START PROFILE command

DB2 PROFILE - TESTING

- Could not use development environment
- Used SANDBOX environment
- Defined tables then Setup profile for COLLECTION=NULLID & “maxdbat”=2
- Issued START PROFILE
- Used COMMAND LINE PROMPT windows
 - Connect and run query (with autocommit off)
 - 1st and 2nd window worked
 - 3rd window would “hang” ... until one committed
- **Successful test!!**

DB2 PROFILE – TESTING(2)

- Move to development environment
- ZPARM MAXDBAT=700
- Dynamically set ZPARM to 40
- Ran Stress test (BASELINE)
 - When Active threads=40 ..
 - DDF threads began to queue ...
 - including LU6.2 threads

DB2 PROFILE – TESTING(3)

- Put ZPARM back to 700
- Define tables then Setup profile for COLLECTION=NULLID with “maxdbat”=40
- Issue START PROFILE
- Ran Stress Test
 - When Active threads hit 40 -> no impact to CICS
 - When ACTIVE IP threads hit 40 -> IP threads queued
 - No impact to LU 6.2 threads -> no impact to CICS regions
 - **SUCCESSFUL TEST!!**

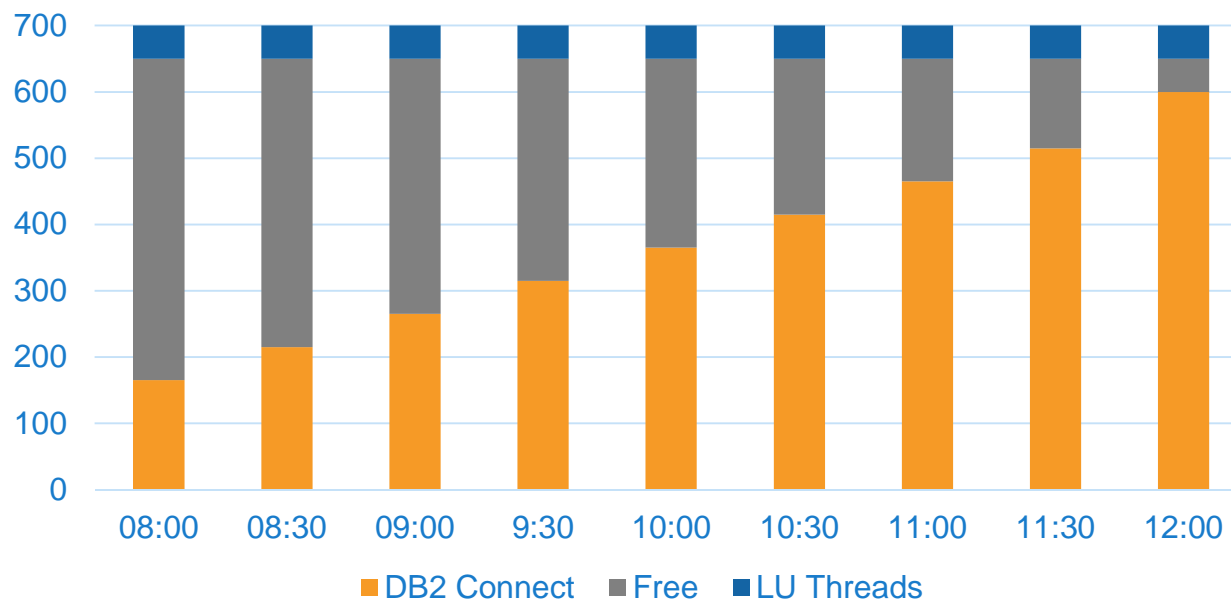
DB2 PROFILE - PRODUCTION

- MAXDBAT=700
- Define tables then Setup profile for COLLECTION=NULLID with “maxdbat”=600
- GOAL – “reserve” 100 for LU6.2 (from CICS)
- Set IDLE thread timeout from 600 -> 115
 - Application saw no impact when lower during incident

- Within 2 weeks, another server issue
- No impact to CICS
- **SUCCESS!!!**

DB2Y Threads –

- When TCP/IP threads do not end
 - IP threads count grows
 - Make sure LU6.2 threads don't get “squeezed” out



DB2 PROFILE – PRODUCTION

- PROFILE configuration working perfectly
- Minor problem with IDLE Thread timeout reduction
- Timeout “catching” weekly tasks
- Time to rethink reduction??
- **Solution Update profile!!!**
- Add PROCESS IDs to profile and assign higher IDLE THREAD TIMEOUT value

DB2 PROFILE – Summary

- Allowed us to control number of threads (by COLLID)
- Allowed us to set lower IDLE THREAD timeout for subsystem
- Allowed us to set higher IDLE THREAD timeout for specific IDs
- BENEFIT
 - No new subsystems
 - No application changes
 - Reduce risk to CICS regions
 - No software to purchase and/or install
- Lower IDLE THREAD Timeout -> Lower # threads active

DB2 PROFILE - Options

- Any of the following combinations for monitoring connections, threads or idle threads, with particular rules for certain KEYWORDS values:
 - LOCATION only
 - PRDID only
 - AUTHID, ROLE, or both.
 - COLLID, PKGNAME, or both
 - One of CLIENT_APPLNAME, CLIENT_USERID, CLIENT_WORKSTNNAME
- Any column can be wild carded
 - Only “*” can be specified, enabling an “All” context for filtering
 - No “like” wild carding

DB2 PROFILE – Options(2)

- PROFILE can
 - MONITOR CONNECTIONS
 - MONITOR THREADS
 - MONITOR IDLE THREADS

Questions



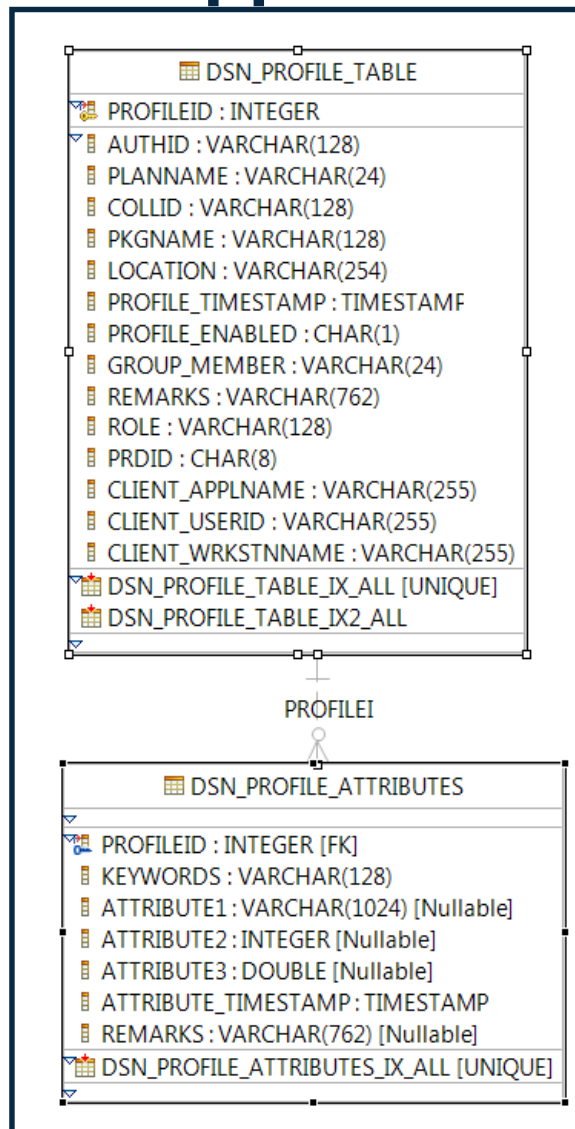
Overview of Profile Support in DB2 for z/OS

- Profiles allow you to:
 - ★ – Monitor remote threads and connections (TCP/IP) – DB2 10
 - MONITOR CONNECTIONS
 - MONITOR THREADS
 - MONITOR IDLE THREADS
 - Set or disable optimization parameters for SQL statements
 - Model your test environment after production
 - Set thresholds for Query Acceleration
 - Set special registers for distributed clients (DB2 11)

Overview of Profile Support in DB2 for z/OS

- **SYSIBM.DSN_PROFILE_TABLE**
 - Defines profile and filtering (e.g. LOCATION, PRDID, AUTHID, etc.)
 - Only certain combinations are allowed
 - Scope varies by the monitoring function specified in Attributes table
 - The PROFILE_ENABLED column indicates whether DB2 activates the profile when you start monitoring
- **SYSIBM.DSN_PROFILE_ATTRIBUTES** table
 - Defines attributes (e.g. MONITOR CONNECTIONS, MONITOR THREADS, MONITOR IDLE THREADS, etc.)
 - One or more attribute rows are required
 - Attribute rows control the actions that DB2 applies

Overview of Profile Support in DB2 for z/OS



Defining Profiles and Attributes

- Defining a profile
 - A row in SYSIBM.DSN_PROFILE_TABLE defines one filtering scope
 - Possible to have multiple rows defining more than one scope
 - In this example, the filtering category is ‘authorization identifier’ expressed in the AUTHID column (example value: DDS2364)
 - The PROFILE_ENABLED column indicates whether DB2 activates the profile when you issue the START PROFILE command

```

000003 INSERT INTO SYSIBM.DSN_PROFILE_TABLE ( AUTHID
000004                                     , PLANNAME
000005                                     , COLLID
000006                                     , PKGNAME
000007                                     , LOCATION
000008                                     , PROFILEID
000009                                     , PROFILE_TIMESTAMP
000010                                     , PROFILE_ENABLED
000011                                     , GROUP_MEMBER
000012                                     , REMARKS
000013                                     , ROLE
000014                                     , PRDID
000015                                     , CLIENT_APPLNAME
000016                                     , CLIENT_USERID
000017                                     , CLIENT_WRKSTNNAME)
000018 VALUES ( DDS2364 , ' ' , ' ' , ' ' , ' ' ,
000019             , 101 , CURRENT_TIMESTAMP , 'Y' ,
000020             ' ' , ' ' , ' ' , ' ' , ' ' )

```

Defining Profiles and Attributes

- Defining actions
 - Rows in the SYSIBM.DSN_PROFILE_ATTRIBUTES table control the actions
 - The values of the PROFILEID columns of each table associate each profile with the corresponding actions for that profile (e.g.101)
 - The value in KEYWORDS column determines the action

```

000003 INSERT INTO SYSIBM.DSN_PROFILE_ATTRIBUTES ( PROFILEID
000004                                     , KEYWORDS
000005                                     , ATTRIBUTE1
000006                                     , ATTRIBUTE2
000007                                     , REMARKS)
000008 VALUES (101, 'MONITOR THREADS', 'WARNING_DIAGLEVEL2',
000009          '0' , '')

```

Defining Profiles and Attributes

- Starting and stopping profiles
 - You must enable and start profiles before DB2 can use the information in the profile tables
 - Issue a **START PROFILE** command
 - DB2 activates the functions specified in the profile tables for every valid row of the **SYSIBM.DSN_PROFILE_TABLE** table that contains **PROFILE_ENABLED='Y'**
 - Profiles in rows that contain **PROFILE_ENABLED='N'** are not started
- To stop monitoring, issue **STOP PROFILE** command
- Required authority: **SQLADM**, **System DBADM**, **SYSOPR**, **SYSCTRL**, or **SYSADM**

Defining Profiles and Attributes

- Validating profiles are ‘accepted’
 - Select from SYSIBM.DSN_PROFILE_HISTORY
 - Select from
SYSIBM.DSN_PROFILE_ATTRIBUTES_HISTORY
 - Example:

```
SELECT * FROM SYSIBM.DSN_PROFILE_ATTRIBUTES_HISTORY;  
ACCEPTED  
ACCEPTED  
ACCEPTED  
REJECTED - THREAD-LEVEL MONITORING KEYWORDS USED FOR SYSTEM-LEVEL MONITORING SC  
  
SELECT * FROM SYSIBM.DSN_PROFILE_HISTORY;  
ACCEPTED  
ACCEPTED  
REJECTED - INVALID SCOPE SPECIFIED. FOR SYSTEM LEVEL MONITORING, ONLY LOCATION
```

Using Profiles to Monitor DDF Resources (1)

- Granular control – KEYWORDS column
 - MONITOR CONNECTIONS (relates to CONDBAT)
 - Total number of remote connections from TCP/IP requesters, including current active connections and inactive connections
 - Filtering on LOCATION column only: IP Address or Domain Name

Using Profiles to Monitor DDF Resources (2)

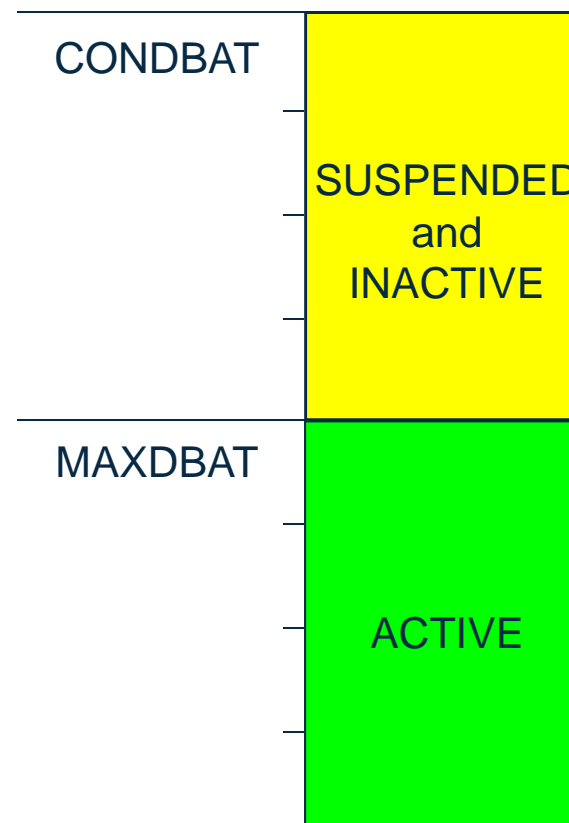
- Granular control – KEYWORDS column
 - MONITOR THREADS (relates to MAXDBAT)
 - Total number of concurrent active remote threads that use TCP/IP on the DB2 subsystem
 - Filtering on
 - *LOCATION* column (IP Address, domain name, location or location alias), or
 - *PRDID*, or
 - *ROLE* and/or *AUTHID*, or
 - ★ – *COLLID* and/or *PKGNAME*, or
 - One of *CLIENT_APPLNAME*, *CLIENT_USERID*, or *CLIENT_WORKSTNNAME*

Using Profiles to Monitor DDF Resources (3)

- Granular control – KEYWORDS column
 - MONITOR IDLE THREADS (relates to IDTHTOIN)
 - Approximate time (in seconds) that an active server thread is allowed to remain idle
 - *A zero value means that matching threads are allowed to remain idle indefinitely*
 - Same filtering as Monitor Threads
 - Can be set independently of IDTHTOIN – higher or lower

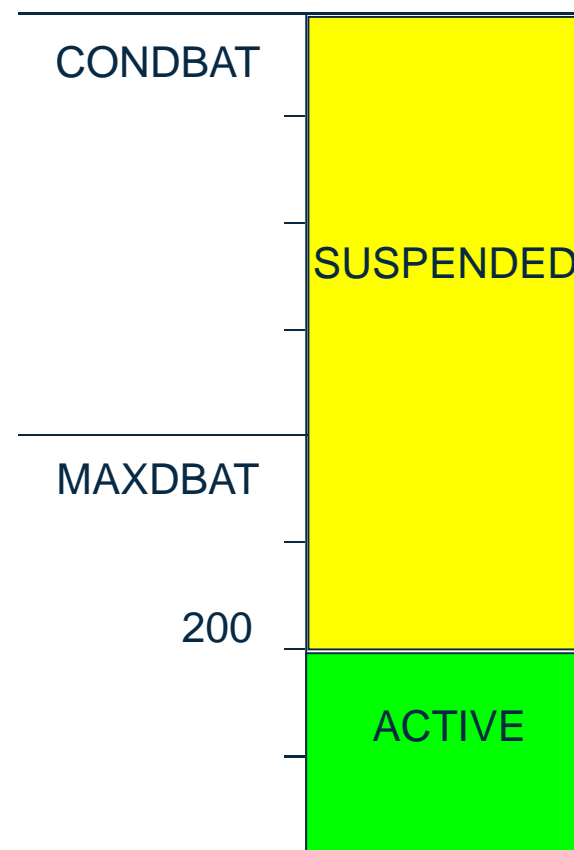
Monitor Threads and Connections (1)

- Results: Without Profiles
 - Number of active threads controlled by MAXDBAT
 - Number of connections controlled by CONDBAT
 - These values apply at the subsystem level and remain in effect



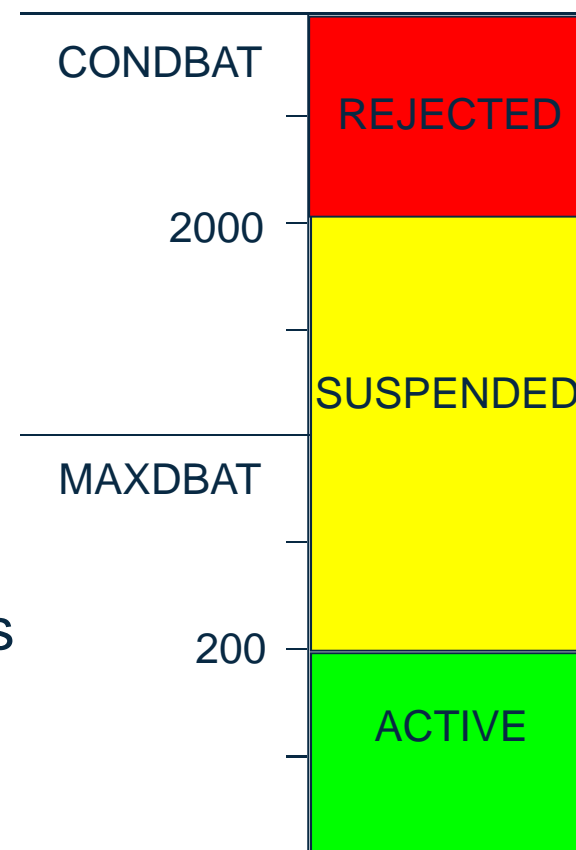
Monitor Threads and Connections (2)

- Results: Profile active: IP Address or Domain Name filter
 - Keyword: MONITOR THREADS
 - Attribute1: EXCEPTION
 - Attribute2: 200
 - Any thread requests *for this profile* in excess of 200 are suspended



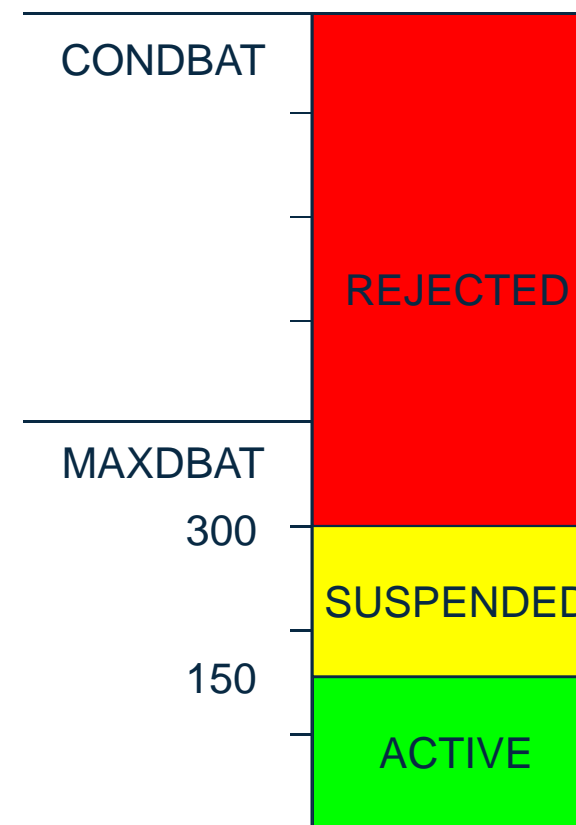
Monitor Threads and Connections (3)

- Results: Profile active: IP Address or Domain Name filter
 - Keyword: MONITOR THREADS
 - Attribute1: EXCEPTION
 - Attribute2: 200
 - Keyword: MONITOR CONNECTIONS
 - Attribute1: EXCEPTION
 - Attribute2: 2000
 - Any thread requests *for this profile* between 200 and 2000 are suspended
 - Thread requests *for this profile* in excess of 2000 are rejected with -30081 Communications Error



Monitor Threads and Connections (4)

- Results: Profile active: Any other filter (not IP Address or Domain Name)
 - ★ – Keyword: MONITOR THREADS
 - Attribute1: EXCEPTION
 - Attribute2: 150
 - Threads *for this profile* in excess of 150 are suspended
 - Until 150 are suspended
 - Threads *for this profile* in excess of 300 are rejected with -30041



Monitor Threads and Connections (5)

- Example with multiple KEYWORDS
 - PROFILEID 21 is associated with three keywords:
 - MONITOR THREADS – issue a message and suspend thread requests beyond 150 active DBATs
 - MONITOR CONNECTIONS – issue a message when there are more than 200 connections, but continue to service the connection requests.
 - MONITOR IDLE THREADS – issue a message and terminate threads idle for more than 30 seconds

PROFILE ID	LOCATION	ROLE	AUTHID	PRDID	COLLID	PKGNAME
21	DEMOMVS.DEMOPKG.IBM.COM	null	null	null	null	null

PROFILE ID	KEYWORDS	ATTRIBUTE1	ATTRIBUTE2	ATTRIBUTE3	ATTRIBUTE_TIMESTAMP
21	<u>MONITOR THREADS</u>	EXCEPTION	150	NULL	2011-12-19.....
21	<u>MONITOR IDLE THREADS</u>	EXCEPTION	30	NULL	2011-12-17.....
21	<u>MONITOR CONNECTIONS</u>	WARNING	200	NULL	2011-1-24.....



Questions



SQL Optimization Parameters

- Set or disable optimization parameters at a granular level
- Specify the subsystem parameter that you want to modify in the DSN_PROFILE_ATTRIBUTES table
 - KEYWORDS and Attributes
 - ZPARAM **NPGTHRSH**
 - *KEYWORDS = NPAGES THRESHOLD*
 - *ATTRIBUTE2 = number of pages*
 - ZPARAM **STARJOIN**
 - *KEYWORDS = STAR JOIN*
 - *ATTRIBUTE1 = ENABLE/DISABLE*
 - ZPARAM **SJTABLES**
 - *KEYWORDS = MIN STAR JOIN TABLES*
 - *ATTRIBUTE2 = number of tables*

Model Production in Test Environment

- APARs and requirements
 - V9 APAR PM26475 & V10 APAR PM26973
 - Each APAR supports optimizer overrides for these system settings
 - New DSNZPARM parameters
 - SIMULATED_CPU_SPEED
 - SIMULATED_COUNT
 - SYSIBM.DSN_PROFILE_ATTRIBUTES KEYWORDS
 - SORT_POOL_SIZE
 - MAX_RIDBLOCKS
 - For bufferpools
 - *Same as the BP names listed in the DSNTIP1 panel*
 - *BP0, BP1,...BP8K, etc.*
- EXPLAIN will reflect “PROFILEID xxxx’ concatenated into REASON column of DSN_STATEMENT_TABLE

Setting Accelerator Thresholds

- In DB2 for z/OS Profiles have the ability to use these functional keywords to influence whether or not to send a query to an accelerator server
 - ACCEL_TABLE_THRESHOLD
 - Total table cardinality for a query to be treated as a short running query
 - ACCEL_RESULTSIZE_THRESHOLD
 - Maximum number of rows that a query that is sent to an accelerator server can return
 - ACCEL_TOTALCOST_THRESHOLD
 - Maximum estimated total cost for a query to be treated as a short running query
- Details are in the Managing Performance manual

Setting Special Registers for DDF Clients

- DB2 11 New Function Mode (PM93658)
- Profiles use same filtering categories as for monitoring threads and idle threads
- New KEYWORDS value: SPECIAL_REGISTER
 - ATTRIBUTE1 contains the ‘SET’ statement
 - E.g. SET CURRENT APPLICATION COMPATIBILITY = ‘V11R1’
- Precedence of the SET *special register*.
 - 1) Special register explicitly set by the application
 - 2) Special register set through Profile Support as above
 - 3) Special register set on the connection property level or data source level

Setting Special Registers for DDF Clients

- CURRENT APPLICATION COMPATIBILITY
- CURRENT DEBUG MODE
- CURRENT DECFLOAT ROUNDING MODE
- CURRENT DEGREE
- CURRENT EXPLAIN MODE
- CURRENT GET_ARCHIVE
- CURRENT LOCALE LC_CTYPE
- CURRENT MAINTAINED TABLE TYPES FOR OPTIMIZATION and CURRENT MAINTAINED TABLE TYPES
- CURRENT OPTIMIZATION HINT
- CURRENT PACKAGE PATH
- CURRENT PATH and PATH and CURRENT FUNCTION PATH
- CURRENT PRECISION
- CURRENT QUERY ACCELERATION
- CURRENT REFRESH AGE
- CURRENT ROUTINE VERSION
- CURRENT RULES
- CURRENT SCHEMA and SCHEMA
- CURRENT SQLID
- CURRENT TEMPORAL BUSINESS_TIME
- CURRENT TEMPORAL SYSTEM_TIME
- ENCRYPTION PASSWORD
- SESSION TIME ZONE and TIME ZONE

Questions



Thank You!

- Bill Leininger
– william.leininger@pnc.com
- Mark Rader
– mrader@us.ibm.com