# Solving an Intermittent Production Problem Using Db2 for z/OS System Profile Monitoring

Mark Rader, Db2 for z/OS

*IBM Washington Systems Center*

Tridex

September 19, 2019

Db2 for z/OS

IDUG

Leading the Db2 User
Community since 1988

# Abstract and objectives

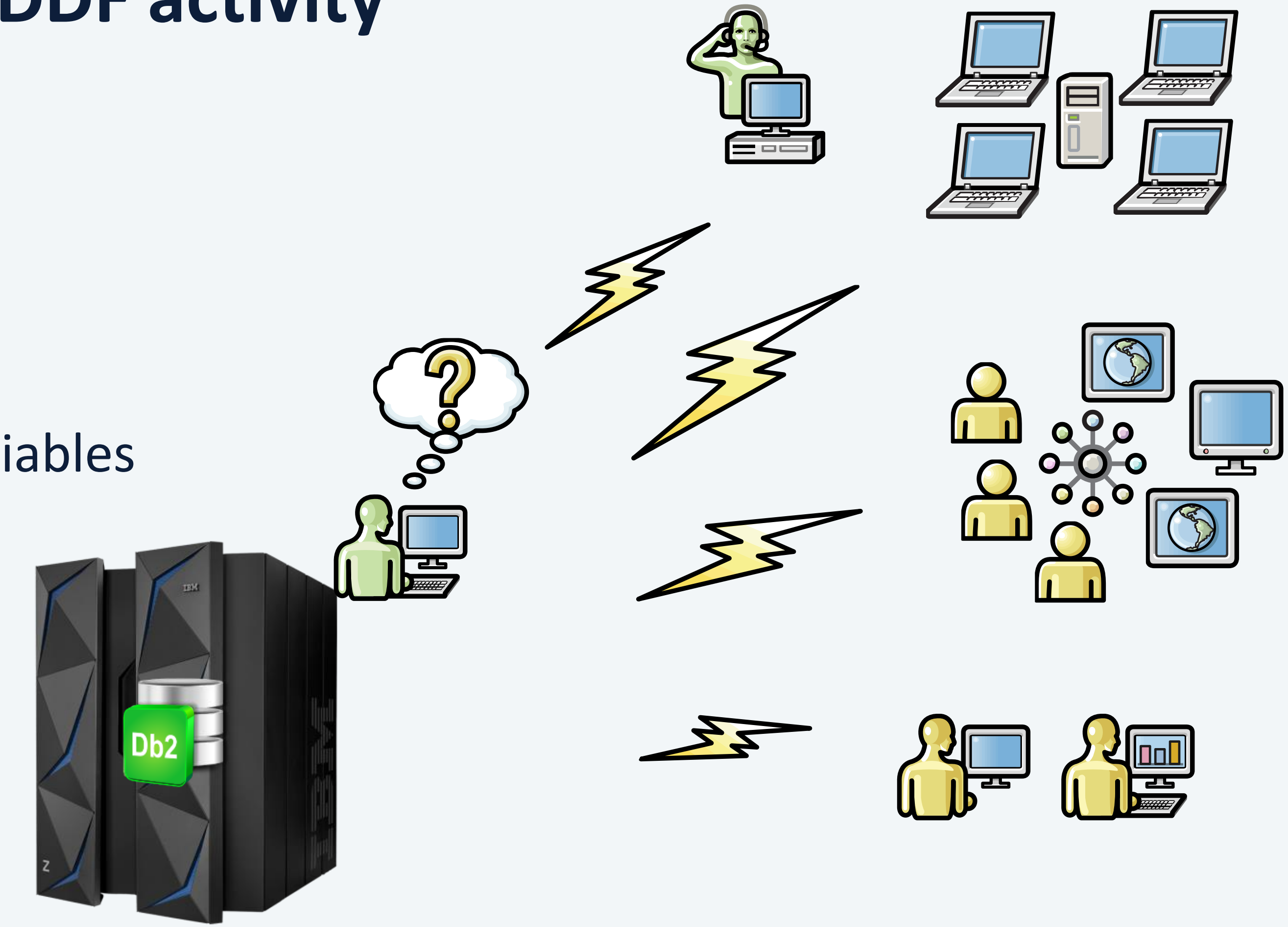
- Based on actual customer experience, this session describes how to implement Db2 System Profile Monitoring to optimize and manage access by applications or clients using DRDA to interact with Db2 data. What is Db2 System Profile Monitoring? Introduced in Db2 9 for z/OS and extended in Db2 10, Db2 11 and Db2 12 for z/OS, System Profile Monitoring provides a variety of exciting capabilities: ZPARMs at a more granular level than the subsystem, modeling production in test, and managing application specific behavior - all available via online changes!
- Objective 1: Describe some challenges that face Db2 administrators in supporting requests from DRDA clients
- Objective 2: Explore use cases for Db2 System Profile Monitoring, including DRDA and modeling production in test.
- Objective 3: Explain how to define profiles and attributes to address the use cases.
- Objective 4: Describe recent developments in Db2 11 and Db2 12 to extend the benefits of System Profile Monitoring

# Agenda

- Challenges relating to DDF activity
- Use case overview
- Adaptation of customer experience presentation
- Details of Db2 System Profile Monitoring for DDF activity
  - Connections, threads, idle thread timeouts
- Other capabilities of Db2 System Profile Monitoring
- Db2 12 enhancements for Db2 System Profile Monitoring

# Challenges relating to DDF activity

- Resources to manage
  - Connections
  - Threads
  - Timeouts
  - Special registers, global variables
- Clients to support
  - Priorities, demands
  - Variety of applications



# Db2 System Profile Monitoring – example use cases

- Prevent denial of service attack
- Manage migration to new driver level
- Reserve threads for critical applications
- Set special registers or global variables
  - Including accelerator, transparent archiving enablement
- Non-distributed: anticipate PROD access paths in TEST
  - Model production values of RIDPOOL, SORTPOOL, BP allocations

# Customer experience with Db2 System Profile Monitoring

# Customer situation

- Significant growth since 2003, including acquisitions

- Net effect was doubling the workload in the environment, then doubling it again, over the course of eight years

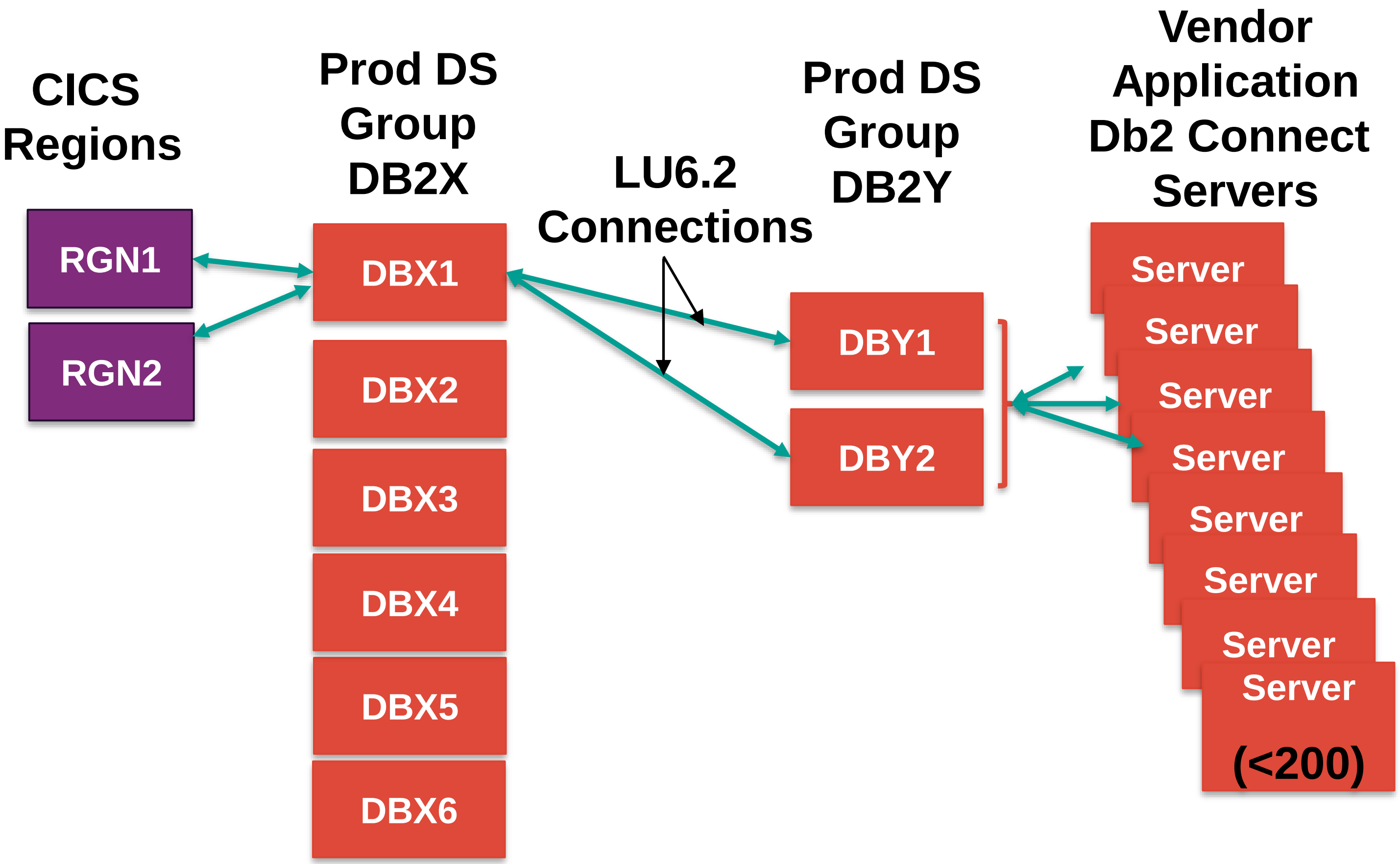
- IBM had been involved in Db2 and systems health checks at various points in that time
  - Including evaluation of DBM1 virtual storage

# Description of environment

- One data sharing group, DB2X, supports home grown applications
  - 6-way data sharing group
  - CICS activity from regions on 3 LPARs
  - TCP/IP DDF inbound activity
  - LU 6.2 outbound activity to DSG DB2Y and vendor application data

- Other data sharing group, DB2Y, supports vendor application
  - 2-way data sharing group
  - No CICS
  - DDF via TCP/IP
  - DDF via LU 6.2 from DB2X

# Configuration – simplified view



CICS Regions

Prod DS Group DB2X

LU6.2 Connections

Prod DS Group DB2Y

Vendor Application Db2 Connect Servers

# Db2 configuration settings

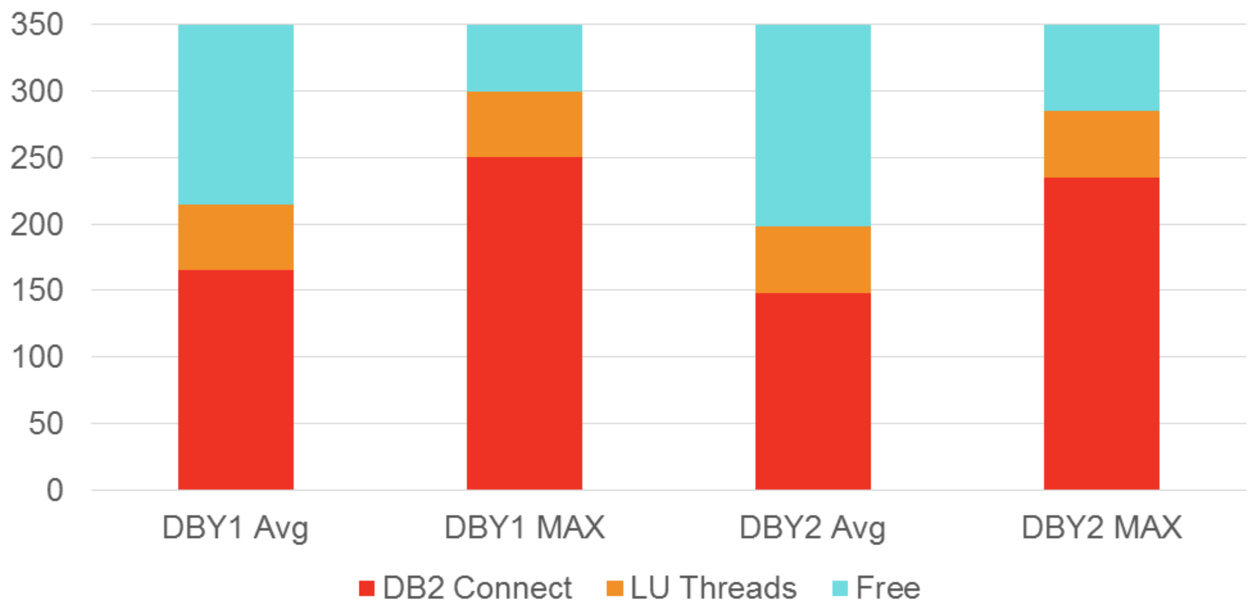
- DB2X (traditional)
  - MAXDBAT=200
  - IDTHTOIN=0 → not really relevant to initial problem

- DB2Y (app server)
  - MAXDBAT=350 (350+350=700 in Data Sharing Group)
  - IDTHTOIN=600
  - MAXDBAT ** Set in V8
  - MAXDBAT ** V9

- Db2 Connect Servers
  - MAX Connections=200 each ( 8*200=1600 total possible)

# 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
  - Anticipated need for 3-way data sharing

- Occasional spike in threads
  - DB2Y members would spike to 350 for short periods

- Prior to 2014, customer looked at options
  - Evaluate DB2X -> DB2Y access
  - Expand DB2Y to 3-way
  - Replicate data from DB2Y to DB2X to eliminate LU6.2 traffic

- Few problem occurrences considered not worth the “cost” of any changes

# DB2Y – What just happened ?!?!

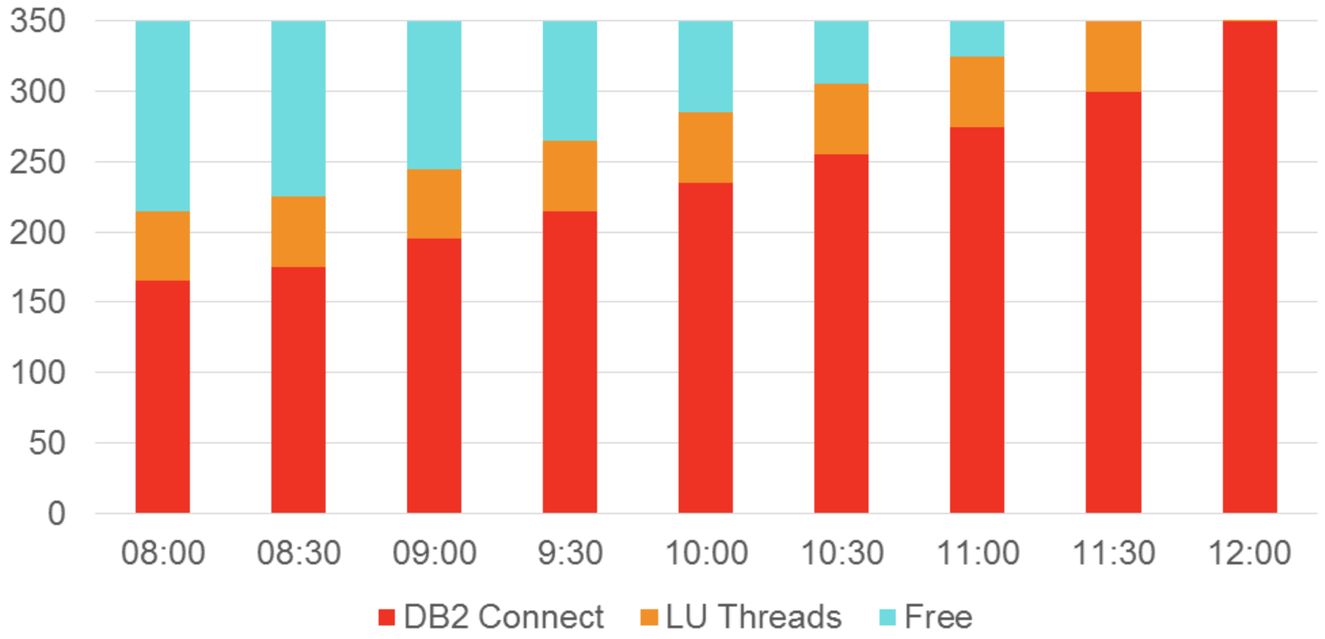
- Db2 V8/V9 – MAXDBAT = 350
- Upgrade to Db2 10 in 2013 … increase MAXDBAT??
  - Reviewed potential to increase MAXDBAT

- Dec 2013 – Server Problem: threads started, but did not end
  - DBY1 and DBY2 both went to MAXDBAT

- What happens when MAXDBAT reached??
  - Connections from DBX1/CICS get squeezed out
  - CICS begins to hit MAXTASK and DB2X sees slowdown

# DB2Y threads

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



- Time not to scale; could happen in a few minutes.

# 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

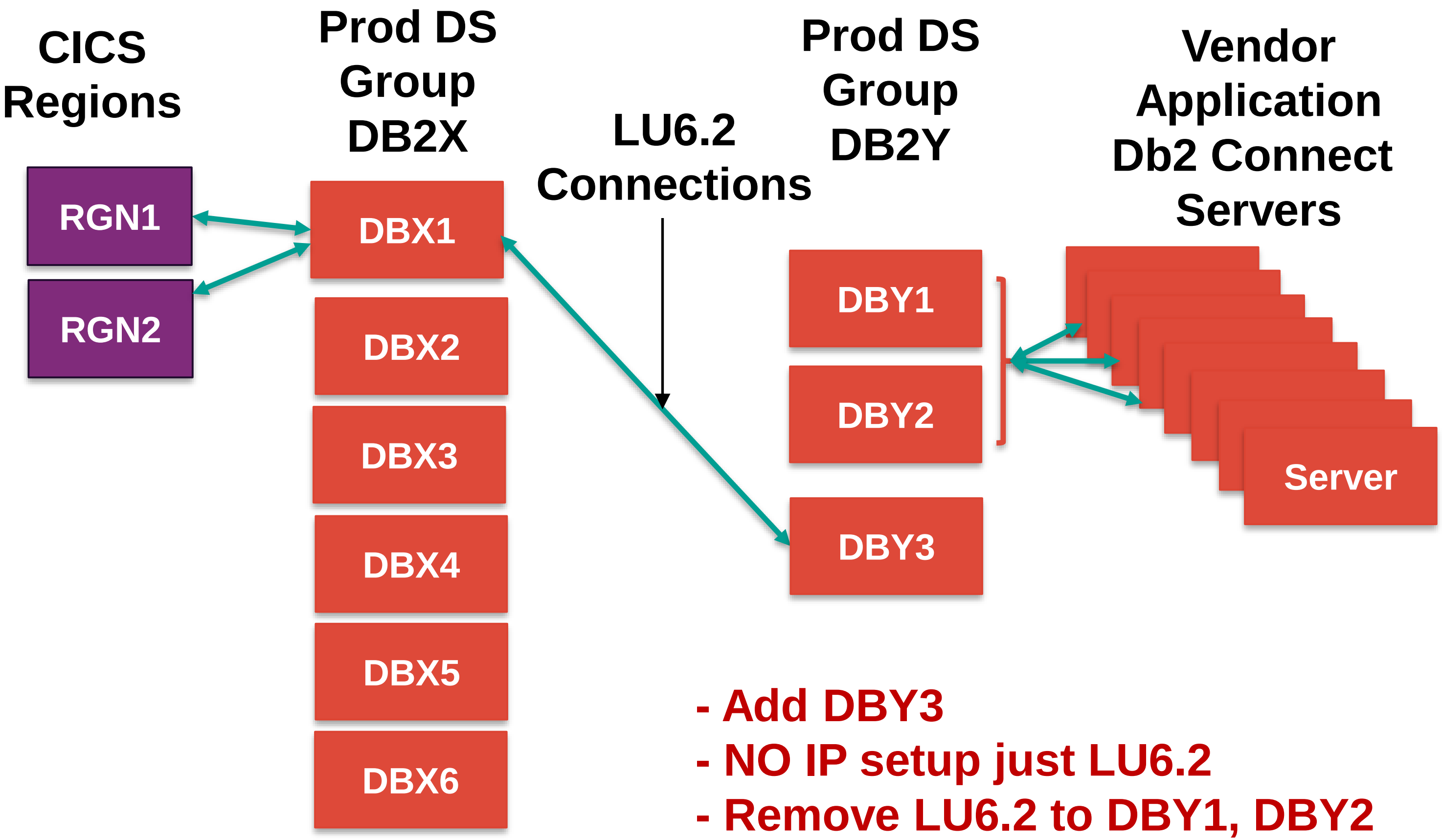
# DB2Y – problems… again

- Application issue
  - Caused DB2Y to hit MAXDBAT
  - MAXDBAT on DB2Y … caused MAXTASK in CICS
  - MAXTASK caused massive slowdowns

- Application issues, therefore DBA staff could not address

- 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 considered



- Add DBY3
- NO IP setup just LU6.2
- Remove LU6.2 to DBY1, DBY2

# Options reconsidered

- 1. Data replication from DB2Y->DB2X
  - Rejected: NOT an OPTION!!

- 2. Starting DBY3 was contemplated
  - Meant another subsystem member
  - Might require -
    - Creating Subsystem Alias
    - REBINDing PLANS
    - Modifying COBOL that issued CONNECT
    - Modifying ALIAS with 3-part name to DB2Y

# Option #3

- Contacted IBM and got meeting set up
- After listening to the problem, solution was to 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 profiles – testing (1|3)

- Could not use development environment
- Used SANDBOX environment
  - Defined tables, then defined profile for COLLECTION=NULLID and “maxdbat”=2
    - MONITOR THREADS, EXCEPTION, 2
  - Issued START PROFILE
  - Used COMMAND LINE PROMPT windows
    - Connect and run query (with autocommit off)
  - 1st and 2nd windows worked
  - 3rd window would ‘hang’ … until one of the others committed

- Successful test of the solution in sandbox!!

# Db2 profiles – testing (2|3)

- Moved to development environment
  - ZPARM MAXDBAT = 700

- Dynamically set ZPARM MAXDBAT to 40
  - To recreate the problem

- Ran stress test (BASELINE)
  - When Active Threads = 40, DDF threads began to queue
  - Including LU 6.2 threads

# Db2 profiles – testing (3|3)

- Reset MAXDBAT to 700
- Defined tables and profile for COLLECTION = NULLID with “maxdbat” = 40
  - MONITOR THREADS, EXCEPTION, 40
- Issued 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 of solution in Development!!

# Db2 profiles - production

- MAXDBAT = 700
- Defined tables then set up profile for COLLECTION=NULLID with “maxdbat”=600
  - MONITOR THREADS, EXCEPTION, 600
  - GOAL – “reserve” 100 threads for LU 6.2 (from CICS)

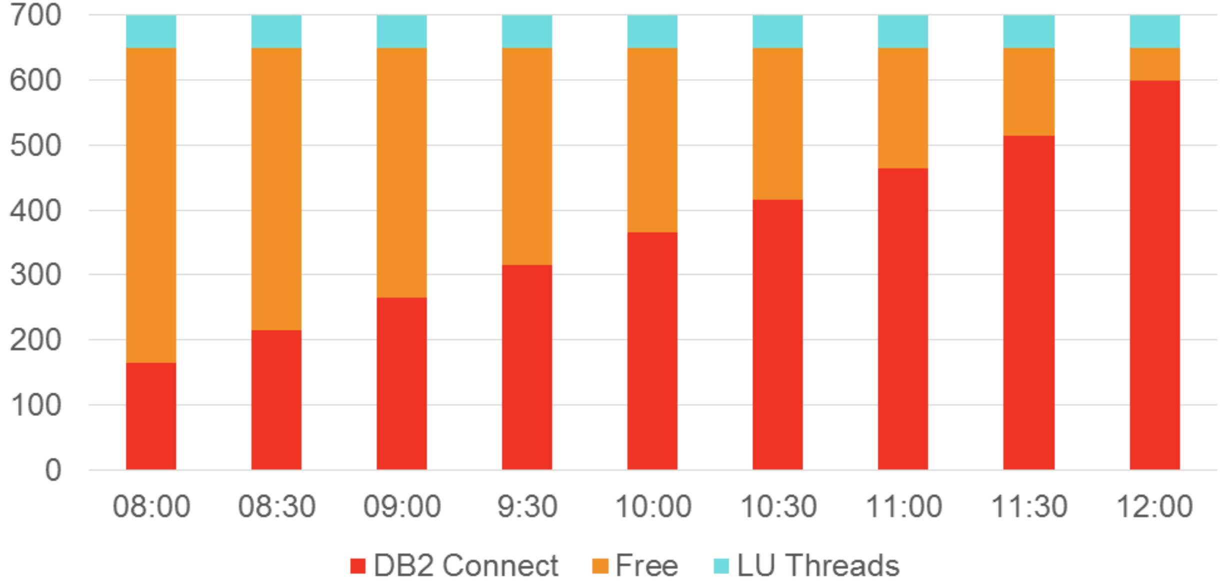
- Set ZPARM IDTHTOIN from 600 -> 115
  - Application saw no impact from lower idle thread timeout during incident

- Within 2 weeks, another server issue
  - This time, no impact to CICS

- Successful Production solution!!

# DB2Y Threads

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



  - Time not to scale; could happen in a few minutes

# Db2 profiles - production

- PROFILE configuration working perfectly
- Minor problem with idle thread timeout reduction
  - Timeout ‘catching’ weekly tasks
- Time to rethink reduction??

- Solution…. Update profile tables!!!
- Add PROCESS IDs to a profile and assign higher IDLE THREAD TIMEOUT value

# Db2 profiles - 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 risks to CICS regions
  - No software to purchase and/or install
- Lower IDLE THREAD Timeout -> Lower # threads active

# Details: Db2 System Profile Monitoring

- Profiles allow customers to:
  - Monitor remote threads and connections (TCP/IP)
    - 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 (since Db2 11)
  - Set global variables (since Db2 12)

# Overview: Db2 System Profile Monitoring (1|2)

- SYSIBM.DSN_PROFILE_TABLE

  - Defines profile and filtering (e.g. LOCATION, PRDID, AUTHID, etc.)
    - Only certain filtering combinations allowed
    - Scope varies by monitoring function specified in Attributes table

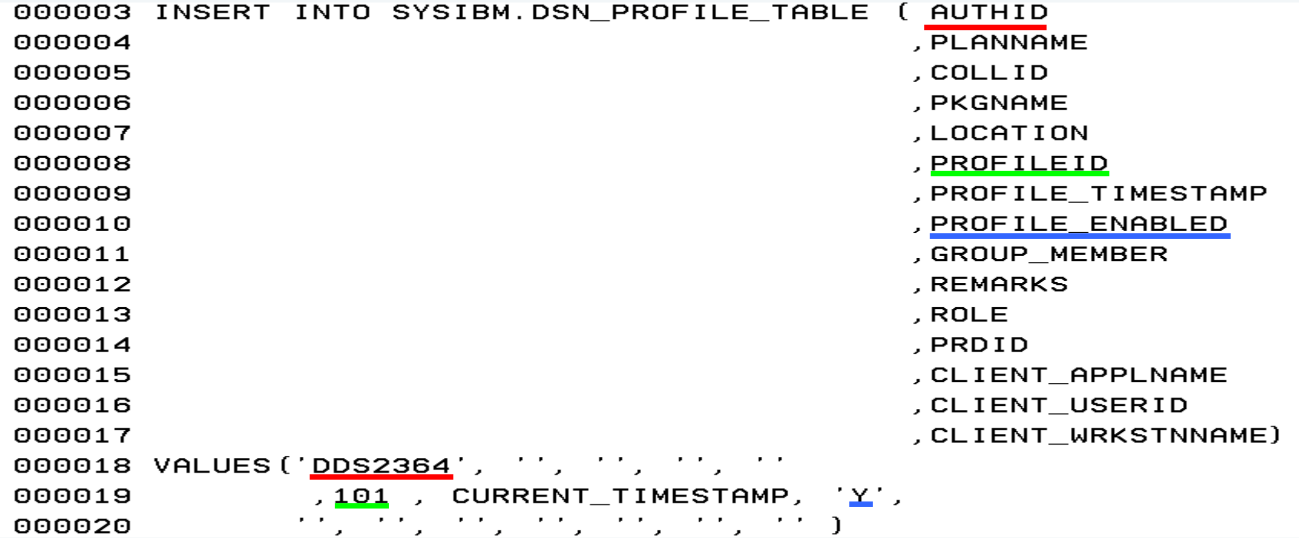
  - The PROFILE_ENABLED column indicates whether Db2 activates the profile when you start monitoring

# Overview: Db2 System Profile Monitoring (2|2)

- SYSIBM.DSN_PROFILE_ATTRIBUTES table
  - Defines actions (e.g. MONITOR CONNECTIONS, MONITOR THREADS, MONITOR IDLE THREADS, etc.)
  - Columns include KEYWORDS, ATTRIBUTE1, ATTRIBUTE2, ATTRIBUTE3
  - One or more attribute rows are required
  - Attribute rows control the actions that Db2 applies

# Defining profiles and attributes (1|4)

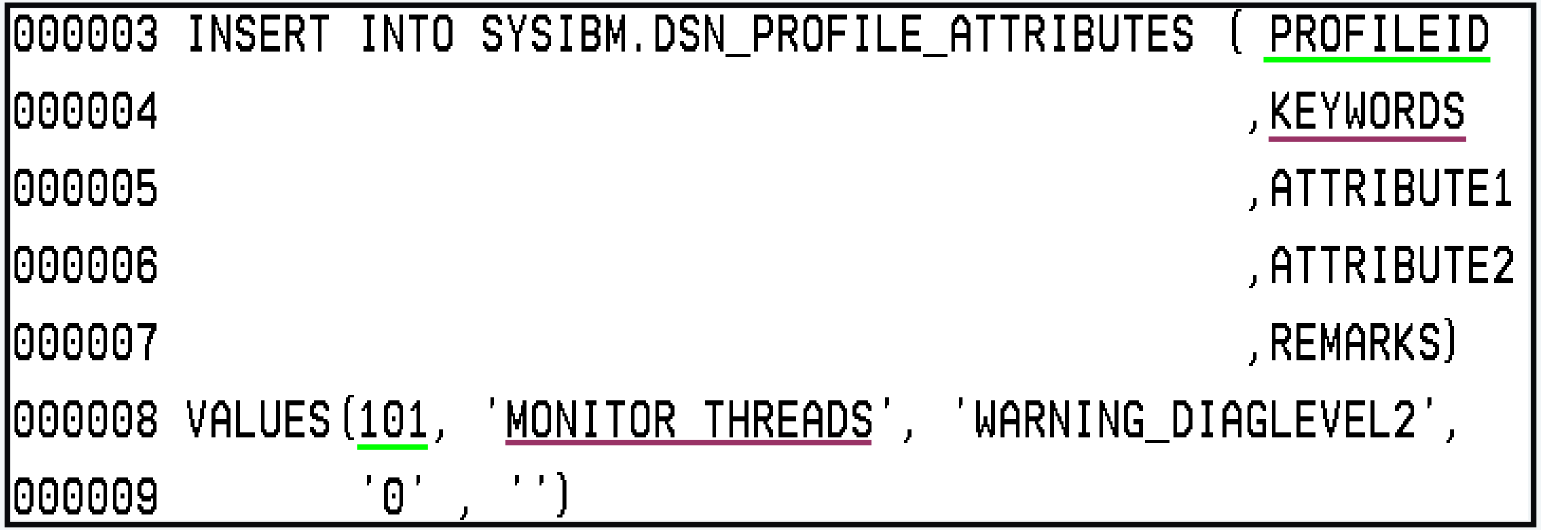
- Defining a profile
  - A row in SYSIBM.DSN_PROFILE_TABLE defines one filtering scope
    - Possible to have multiple rows defining more than one scope
    - In this example, the filtering category is ‘authorization identifier’ expressed in the AUTHID column (example value: DDS2364)
  - The PROFILE_ENABLED column indicates whether Db2 activates the profile when you issue the START PROFILE command



```
INSERT INTO SYSIBM.DSN_PROFILE_TABLE  ( AUTHID
                                       ,PLANNAME
                                       ,COLLID
                                       ,PKGNAME
                                       ,LOCATION
                                       ,PROFILEID
                                       ,PROFILE_TIMESTAMP
                                       ,PROFILE_ENABLED
                                       ,GROUP_MEMBER
                                       ,REMARKS
                                       ,ROLE
                                       ,PRDID
                                       ,CLIENT_APPLNAME
                                       ,CLIENT_USERID
                                       ,CLIENT_WRKSTNNAME)
VALUES('DDS2364', '', '', '', ''
        ,101 , CURRENT_TIMESTAMP, 'Y',
       '', '', '', '', '', '', '' )
```

# Defining profiles and attributes (2|4)

- 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



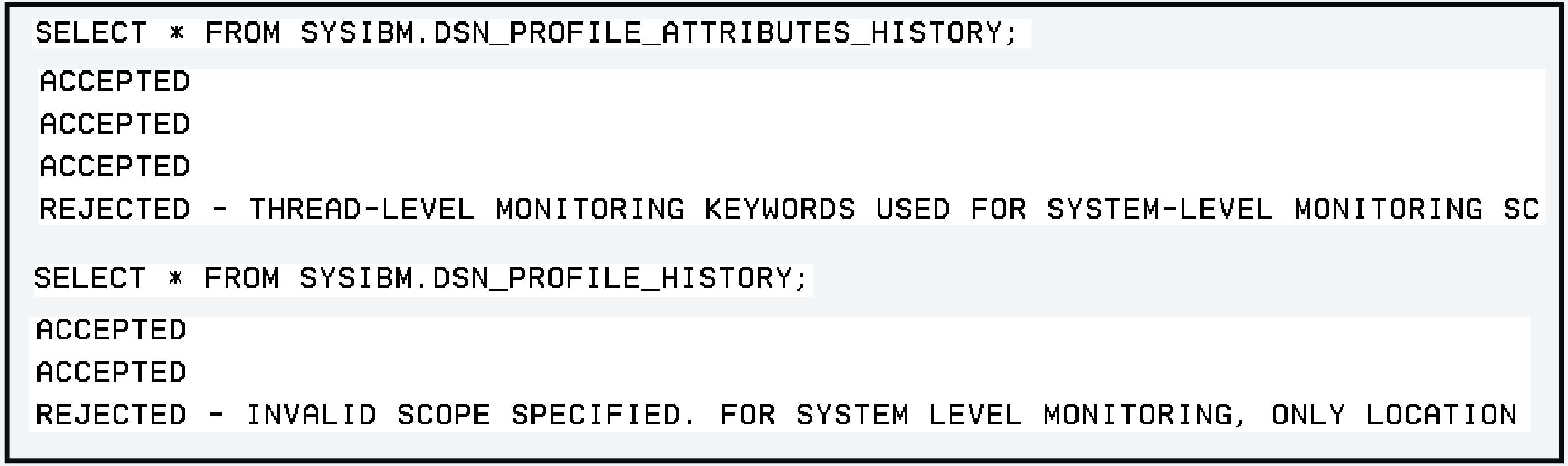
```
000003 INSERT INTO SYSIBM.DSN_PROFILE_ATTRIBUTES ( PROFILEID
000004                                            ,KEYWORDS
000005                                            ,ATTRIBUTE1
000006                                            ,ATTRIBUTE2
000007                                            ,REMARKS)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

# Defining profiles and attributes (3|4)

- 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
    - START PROFILE must be issued separately for each member of data sharing group
  - Db2 activates the functions specified in the profile tables for every valid row of the SYSIBM.DSN_PROFILE_TABLE that contains PROFILE_ENABLE="Y"
    - Profiles in rows that contain PROFILE_ENABLED='N' are not started
  - Issue STOP PROFILE to stop all system profile monitoring
  - Required authority: SQLADM, System DBADM, SYSOPER, SYSCTRL, SYSADM

# Defining profiles and attributes (4|4)

- 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
```

# Monitoring DDF resources with profiles

# Why use profiles to monitor DDF resources?

- A single subsystem value means resources cannot be allocated by business area or priority
  - Vice President’s request for connection or thread is equal to summer intern’s request
  - Numerous low priority threads can reduce access for high priority threads
- Customers may be forced to use other means to prioritize or manage thread and connection requests
  - E.g. Db2 Connect Server (Gateway) instead of direct connections
- Single idle thread timeout may not reflect behavior of all application threads
  - E.g. entire subsystem may disable idle thread timeout because a single application cannot tolerate an idle thread timeout

# Profile table: filtering criteria for DDF (1|2)

## Client IP address or client domain name

**1.** Client IP address or domain name (defined in LOCATION column)

## Client Product ID

**2.** Client Product ID (defined in PRDID column)

## Role name, Authorization ID

**3.** Role name + Authorization ID (defined in ROLE and AUTHID columns)

**4.** Role name (defined in ROLE column)

**5.** Authorization ID (defined in AUTHID column)

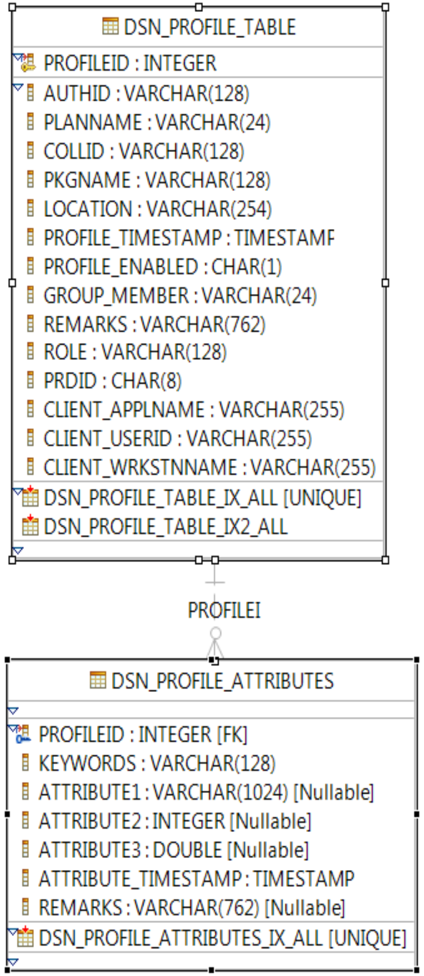
## Collection ID, Package name

**6.** Collection ID + Package name (defined in COLLID and PKGNAME columns)

**7.** Collection ID (defined in COLLID column)

**8.** Package name (defined in PKGNAME column)

IMPORTANT: Criteria in these colored categories are mutually exclusive in a profile row.
Choose one category per profile row.



DSN_PROFILE_TABLE
PROFILEID : INTEGER
AUTHID : VARCHAR(128)
PLANNAME : VARCHAR(24)
COLLID : VARCHAR(128)
PKGNAME : VARCHAR(128)
LOCATION : VARCHAR(254)
PROFILE_TIMESTAMP : TIMESTAMF
PROFILE_ENABLED : CHAR(1)
GROUP_MEMBER : VARCHAR(24)
REMARKS : VARCHAR(762)
ROLE : VARCHAR(128)
PRDID : CHAR(8)
CLIENT_APPLNAME : VARCHAR(255)
CLIENT_USERID : VARCHAR(255)
CLIENT_WRKSTNNAME : VARCHAR(255)
DSN_PROFILE_TABLE_IX_ALL [UNIQUE]
DSN_PROFILE_TABLE_IX2_ALL

DSN_PROFILE_ATTRIBUTES
PROFILEID : INTEGER [FK]
KEYWORDS : VARCHAR(128)
ATTRIBUTE1 : VARCHAR(1024) [Nullable]
ATTRIBUTE2 : INTEGER [Nullable]
ATTRIBUTE3 : DOUBLE [Nullable]
ATTRIBUTE_TIMESTAMP : TIMESTAMP
REMARKS : VARCHAR(762) [Nullable]
DSN_PROFILE_ATTRIBUTES_IX_ALL [UNIQUE]

# Profile table: filtering criteria for DDF (2|2)

## Location name or Location alias

**9.** Server location name or server location alias that Client tries to connect to (defined in LOCATION column)

## Client Application Name

**10.** End Client Application name (defined in CLIENT_APPLNAME column)

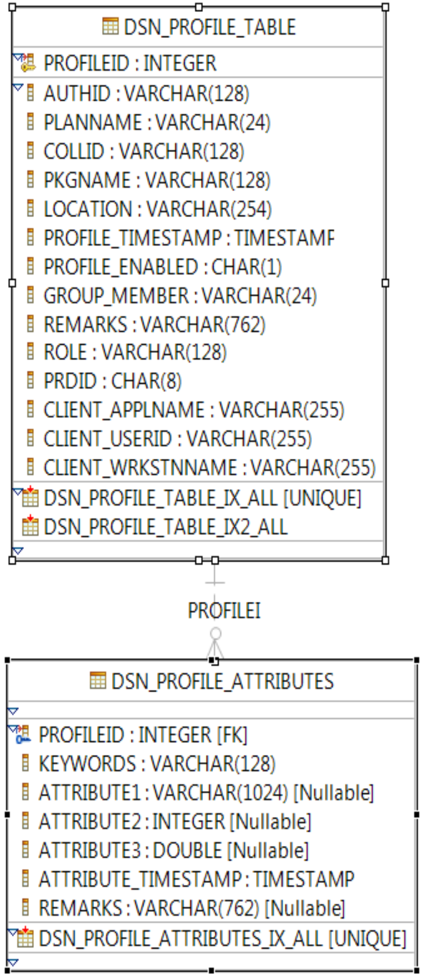
## Client User ID

**11.** End Client User ID (defined in CLIENT_USERID column)

## Client Workstation Name

**12.** End Client workstation name (defined in CLIENT_WRKSTNNAME column)

IMPORTANT: Criteria in these colored categories are mutually exclusive in a profile row.
Choose one category per profile row.



DSN_PROFILE_TABLE
PROFILEID : INTEGER
AUTHID : VARCHAR(128)
PLANNAME : VARCHAR(24)
COLLID : VARCHAR(128)
PKGNAME : VARCHAR(128)
LOCATION : VARCHAR(254)
PROFILE_TIMESTAMP : TIMESTAMF
PROFILE_ENABLED : CHAR(1)
GROUP_MEMBER : VARCHAR(24)
REMARKS : VARCHAR(762)
ROLE : VARCHAR(128)
PRDID : CHAR(8)
CLIENT_APPLNAME : VARCHAR(255)
CLIENT_USERID : VARCHAR(255)
CLIENT_WRKSTNNAME : VARCHAR(255)
DSN_PROFILE_TABLE_IX_ALL [UNIQUE]
DSN_PROFILE_TABLE_IX2_ALL

DSN_PROFILE_ATTRIBUTES
PROFILEID : INTEGER [FK]
KEYWORDS : VARCHAR(128)
ATTRIBUTE1 : VARCHAR(1024) [Nullable]
ATTRIBUTE2 : INTEGER [Nullable]
ATTRIBUTE3 : DOUBLE [Nullable]
ATTRIBUTE_TIMESTAMP : TIMESTAMP
REMARKS : VARCHAR(762) [Nullable]
DSN_PROFILE_ATTRIBUTES_IX_ALL [UNIQUE]

# Profile table: column details (1|2)

- LOCATION column accepts three formats
  - IP address :
    - IPv4 dotted-decimal (e.g. 9.1.2.3)
    - IPv6 colon-hex format (e.g. 2001:DB8::8:800:200C:417A)
  - Domain name
    - E.g. TEST.US.IBM.COM
  - Location name string:
    - LOCATION or LOCATION ALIAS
    - 1-16 byte string, such as STLEC1B
    - Identifies the Db2 for z/OS server, not the client

# Profile table: column details (2|2)

- GROUP_MEMBER applies to members of a data sharing group
  - Blank:
    - Valid profile row applies to any member of the data sharing group
  - Non-blank:
    - Value represents member name and valid row applies only to that member
- Single byte asterisk ‘*’ place in any of the filtering columns defines a default filter scope
  - Not wildcarding, which was introduced in Db2 12, and retro fit to Db2 11

# KEYWORDS column (1|3)

- MONITOR CONNECTIONS
  - Relates to CONDBAT
  - Total number of remote connections from TCP/IP requesters, including current active connections and inactive connections
  - Filtering on Profile table LOCATION column only: IP Address or Domain Name

# KEYWORDS column (2|3)

- MONITOR THREADS
  - Relates to MAXDBAT
  - Total number of concurrent active remote threads that use TCP/IP on the Db2 subsystem or member
  - Filtering on Profile table columns (exclusive OR):
    - 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

- Star indicates filtering used in customer example

# KEYWORDS column (3|3)

- MONITOR IDLE THREADS
  - Relates to IDTHTOIN in DSNZPARM
  - 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

# Attributes columns (1|2)

- ATTRIBUTE1 – specifies how Db2 responds with a  threshold exceeded
  - Action taken:
    - WARNING - A console message is issued at most every five minutes
    - EXCEPTION - Db2 takes action (the connection rejected, or thread queued, or thread canceled), and a message is issued
  - Messaging:
    - DIAGLEVEL1 (default) with minimal information in message  DSNT771I
    - DIAGLEVEL2 includes PROFILEID and reason code in message DSNT772I
    - DIAGLEVEL3 includes PROFILEID, LUWID and reason code
      - Messages DSNT773I or DSNT774I
  - For example: WARNING_DIAGLEVEL1

# Attributes columns (2|2)

- ATTRIBUTE2 - specifies the threshold value
  - Number of connections or threads
  - Number of seconds for idle threads

- Together Keywords, Attribute1, and Attribute2 indicate what Db2 will do
  - Example: MONITOR THREADS, EXCEPTION, 0
  - Result: no DBATs for this profile

# Using profiles: example definitions

SYSIBM.DSN_PROFILE_TABLE

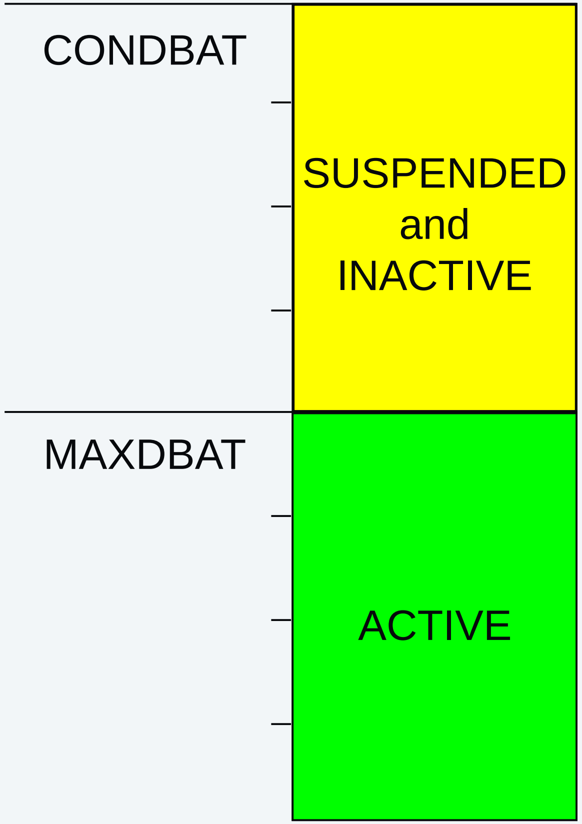
| ROLE | AUTHID | LOCATION | PRDID | COLLID | PKGNAME | PROFILEID | PROFILE_<br>ENABLED |
|---|---|---|---|---|---|---|---|
| | SRVR01 | | | | | 20 | Y |
| | | | JCC03570 | | | 21 | Y |
| | | TEST.SVL.IBM.COM | | | | 22 | Y |

SYSIBM.DSN_PROFILE_ATTRIBUTES

| ProfileID | Keywords | Attribute1 | Attribute2 | Attribute3 | Attribute Timestamp |
|---|---|---|---|---|---|
| 20 | MONITOR THREADS | EXCEPTION | 10 | | 2011-12-19... |
| 21 | MONITOR IDLE THREADS | WARNING | 180 | | 2011-12-19… |
| 22 | MONITOR CONNECTIONS | EXCEPTION | 45 | | 2011-12-19… |
| 22 | MONITOR THREADS | EXCEPTION | 20 | | 2011-12-21… |

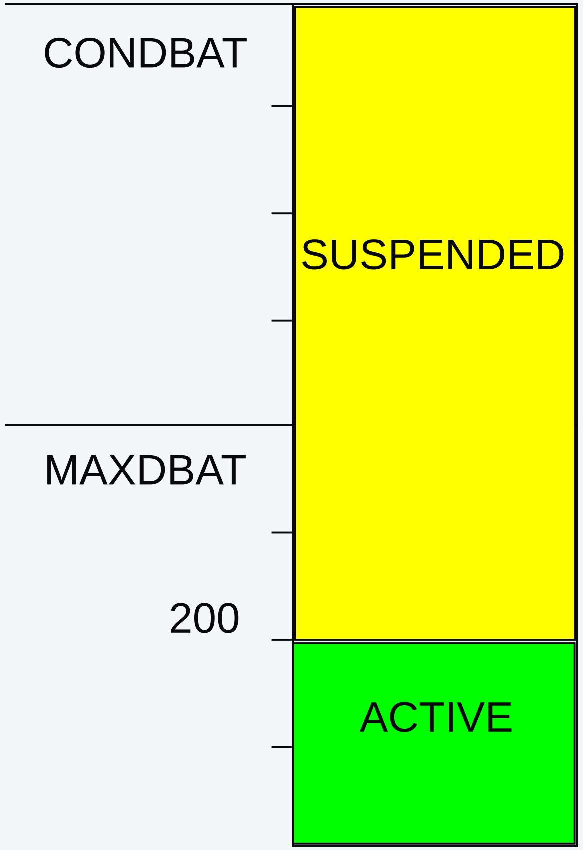
# Monitor threads and connections (1|5)

- Results: Without Profiles enabled
  - Number of active threads controlled by MAXDBAT
  - Number of connections controlled by CONDBAT
  - These values apply at the subsystem level and remain in effect
  - Implementing Db2 System Profile Monitoring does not change this behavior



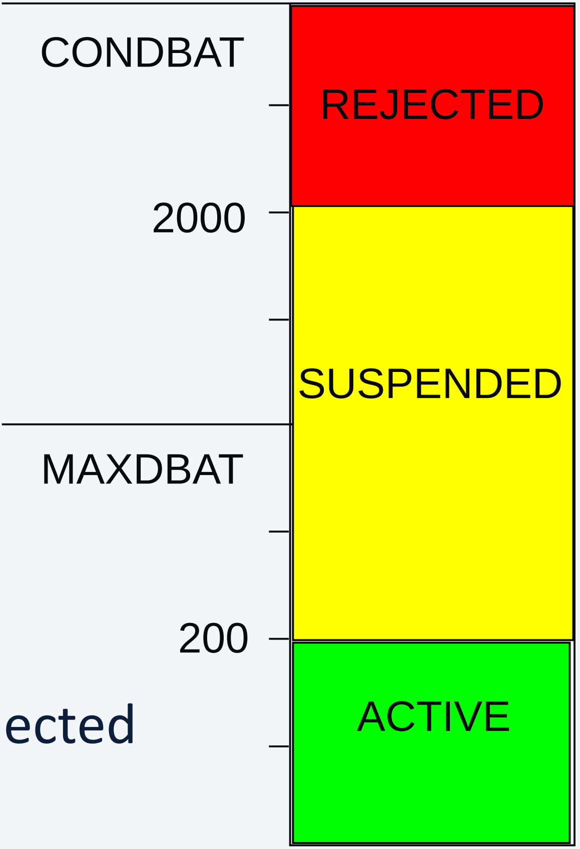
# Monitor threads and connections (2|5)

- Results: Profile enabled on 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



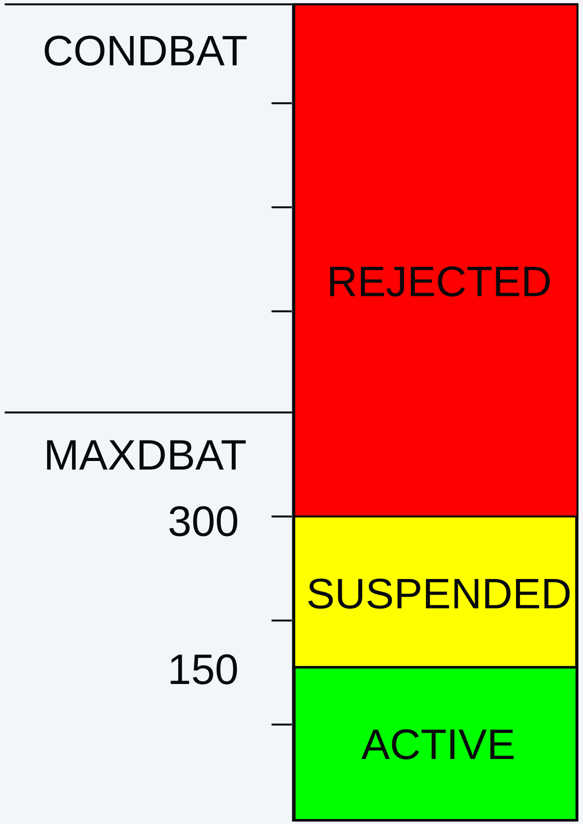
# Monitor threads and connections (3|5)

- Results: Profile enabled on 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



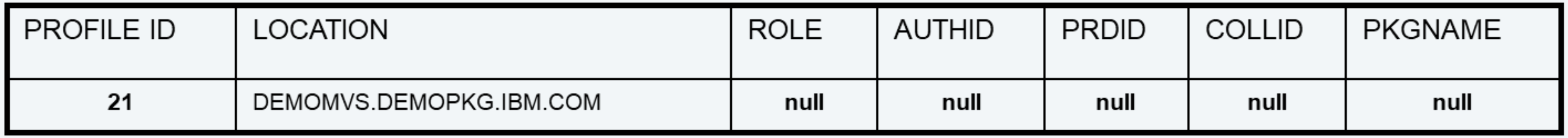
# Monitor threads and connections (4|5)

- Results: Profile enabled on 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

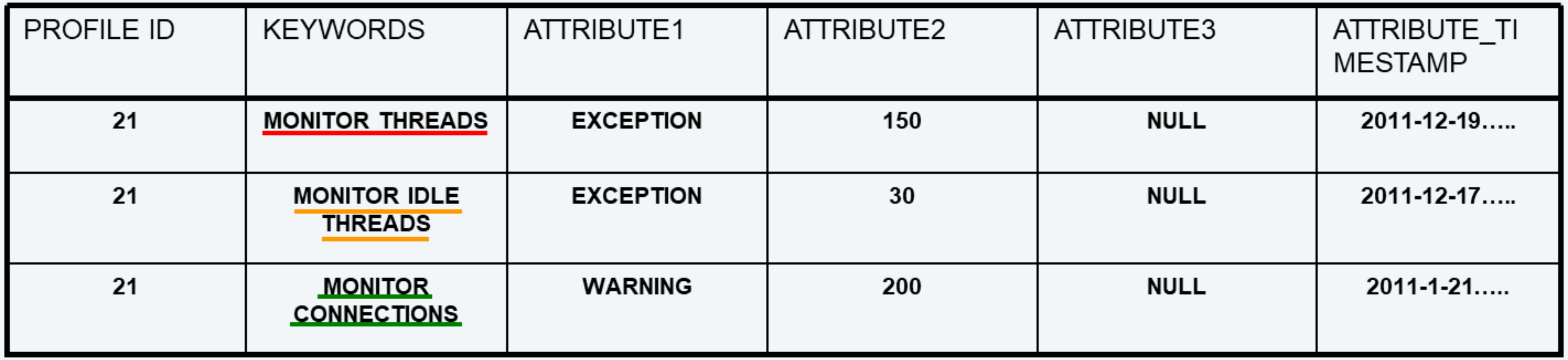


# Monitor threads and connections (5|5)

- 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 | MONITOR THREADS | EXCEPTION | 150 | NULL | 2011-12-19….. |
| 21 | MONITOR IDLE THREADS | EXCEPTION | 30 | NULL | 2011-12-17….. |
| 21 | MONITOR CONNECTIONS | WARNING | 200 | NULL | 2011-1-21….. |

# Other features of Db2 System Profile Monitoring

- SQL optimization parameters
- Modeling PROD in TEST
- Setting Accelerator thresholds
- Setting Special Registers

# SQL optimization parameters

- Set or disable optimization parameters at a granular level
  - Profile enabled on
    - PLANNAME, set to (*), COLLID and PKGNAME
  - Specify the subsystem parameter to modify in the Attributes table
  - KEYWORDS and Attributes:

| ZPARM | KEYWORDS | Attributes |
|---|---|---|
| NPGTHRSH | NPAGES THRESHOLD | ATTRIBUTE2 = number of pages |
| STARJOIN | STAR JOIN | ATTRIBUTE1 = ENABLE \| DISABLE |
| SJTABLES | MIN STAR JOIN TABLES | ATTRIBUTE2 = number of tables |

# Modeling PROD in TEST – specify PROD values in TEST Db2

- DSNZPARMs:
  - SIMULATED_CPU_COUNT and SIMULATED_CPU_SPEED
- Attributes table KEYWORDS
  - SORT_POOL_SIZE
  - MAX_RIDBLOCKS
  - Bp*name*
    - BP0, BP1, …, BP8K1, etc.
- ATTRIBUTE2 specifies the PROD value
- EXPLAIN will reflect “PROFILEID xxxx” concatenated into REASON column of DSN_STATEMENT_TABLE

# Setting Accelerator thresholds

- Attribute table KEYWORDS can influence whether to send a query to an accelerator server
  - For each of these: "If you use an accelerator product, contact IBM Support for that accelerator product for information about how to set this keyword."
    - ACCEL_TABLE_THRESHOLD
      - Total table cardinality for a query to be treated as a short running query
    - ACCEL_RESULTSIZE_TRESHOLD
      - 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
  - ACCEL_NAME_EXPLAIN

# Setting Special Registers for DDF clients (1|2)

- Db2 11 for z/OS NFM
- Profiles use same filtering criteria 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 (2|2)

- CURRENT ACCELERATOR
- 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

# Db2 12 for z/OS enhancements for System Profile Monitoring

# Db2 12 enhancement overview

- PROFILE_AUTOSTART (ZPARM)
  - NO or YES
  - Db2 11: PI89912/UI53164
- MONITOR IDLE THREADS enhancement: ROLLBACK
- Wildcard support for some columns in SYSIBM.DSN_PROFILE_TABLE
- Setting Db2 global variables
- Greater message granularity when system profile thresholds are exceeded

# ROLLBACK for MONITOR IDLE THREADS keyword (1|2)

- Db2 11 MONITOR IDLE THREADS behavior
  - On exception, thread aborted and connection terminated
  - Client may perceive that Db2 is ‘unavailable’
  - Exceeding ZPARM IDTHTOIN has same effect
- Db2 12 adds ROLLBACK option to EXCEPTION
  - Thread is still aborted; Connection remains ***inactive***
  - Client will not have to re-establish connection
  - New options for ATTRIBUTE1 column
    - EXCEPTION_ROLLBACK
    - EXCEPTION_ROLLBACK_DIAGLEVEL1
    - EXCEPTION_ROLLBACK_DIAGLEVEL2

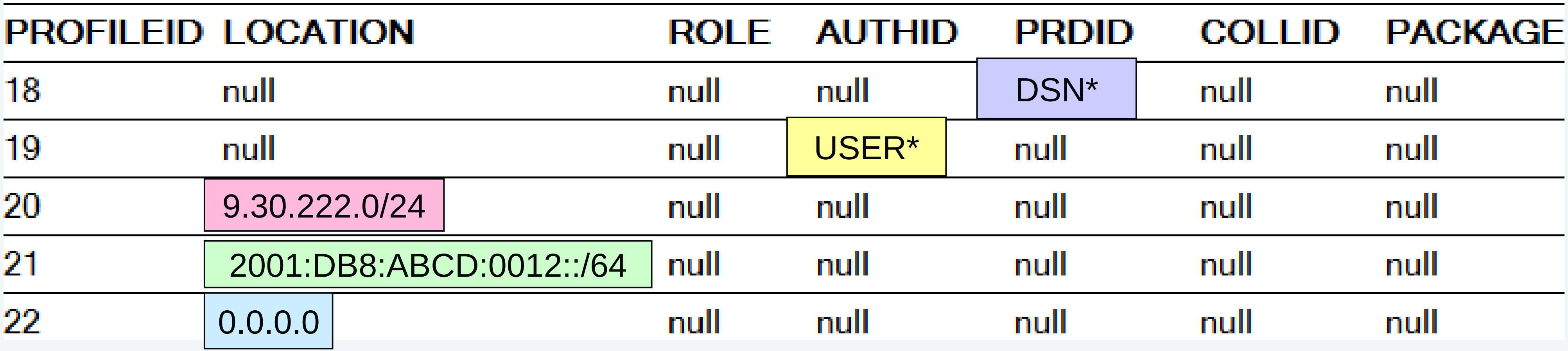
# ROLLBACK for MONITOR IDLE THREADS keyword (2|2)

- If distributed thread qualifies and is idle for longer than the value in ATTRIBUTE2, and
  - Data changes have been made but not committed
    - Thread is aborted, DBAT is pooled
    - Database changes are rolled back
    - Connection is inactive; placed in must-abort
  - Data changes have NOT been made
    - Thread is aborted, DBAT is pooled
    - Connection is inactive; resources may appear to have been lost
      - DGTT, LOB locator, held cursors, kept dynamic statements

# Wildcard support for some Profile table columns

- Use wildcard characters in some profile columns to allow matching against a range of AUTHIDs, or product IDs, or IP addresses
  - AUTHID column: use characters ending in * to match any AUTHID beginning with those characters
    - USER* in AUTHID would match any authid beginning ‘USER’, including USER with no trailing characters
  - PRDID column: multiple product identifiers that began with the same characters could match a single profile definition
    - DSN* would match to any version of a  Db2 for z/OS requestor
  - LOCATION column supports wildcard matching of IP addresses as described on the next page

# Wildcard support: examples



| PROFILEID | LOCATION | ROLE | AUTHID | PRDID | COLLID | PACKAGE |
|---|---|---|---|---|---|---|
| 18 | null | null | null | DSN* | null | null |
| 19 | null | null | USER* | null | null | null |
| 20 | 9.30.222.0/24 | null | null | null | null | null |
| 21 | 2001:DB8:ABCD:0012::/64 | null | null | null | null | null |
| 22 | 0.0.0.0 | null | null | null | null | null |

- LOCATION='9.30.222.251’ matches Profile 20
  - As would any IPv4 address beginning with 9.30.222
- LOCATION=‘2001:DB8:ABCD:0012:0000:0000:0000:1234’ matches profile 21
  - As would any IPv6 address beginning with 2001:DB8:ABCD:0012
- LOCATION='192.168.0.103’ matches profile 22, as does any other IP address that does not match on LOCATION in other DSN_PROFILE_TABLE rows
- The value after ‘/’ in profiles 20 and 21 represents number of bits in the address

# Setting Db2 global variables

- Db2 12 supports setting the following global variables in system profile tables
  - SYSIBMADM.GET_ARCHIVE
  - SYSIBMADM.MOVE_TO_ARCHIVE
  - SYSIBM.TEMPORAL_LOGICAL_TRANSACTIONS
  - SYSIBM.TEMPORAL_LOGICAL_TRANSACTION_TIME
- SYSIBM.DSN_PROFILE_ATTRIBUTES table changes
  - KEYWORD value ‘GLOBAL_VARIABLE’ is available
  - ATTRIBUTE1 accepts SET statement for global variables
  - SET statement requires equal sign (=) between the global variable and the value assigned
  - Example: SET SYSIBMADM.MOVE_TO_ARCHIVE = ‘N’

# Greater message granularity

- New messages, new ATTRIBUTE1 values, improved accounting trace
  - Change to DSNT773I: include LUWID for WARNING
  - New message DSNT774I: include LUWID for EXCEPTION
  - Profile ID added to accounting record if a warning or exception threshold is hit for a profile
  - ATTRIBUTE1 in DSN_PROFILE_ATTRIBUTES includes
    - WARNING_DIAGLEVEL3 to issue DSNT773I
    - EXCEPTION_DIAGLEVEL3 to issue DSNT774I

# Questions

?

# Session summary

- Customers can use Db2 System Profile Monitoring to
  - Manage DDF resources at a more granular level than the system parameter (DSNZPARM) settings, potentially avoiding serious application impacts
    - Connections
    - Threads
    - Idle thread timeout
  - Set optimization parameters, model PROD in TEST, influence query acceleration
  - Set special registers for DDF clients
  - Set global variables for DDF clients (starting in Db2 12)

**Mark Rader**
**IBM**
**mrader@us.ibm.com**

Session code: B17 – Solving an Intermittent Production Problem Using Db2 for z/OS System Profile Monitoring

IDUG

Leading the Db2 User Community since 1988

*Please fill out your session evaluation before leaving!*