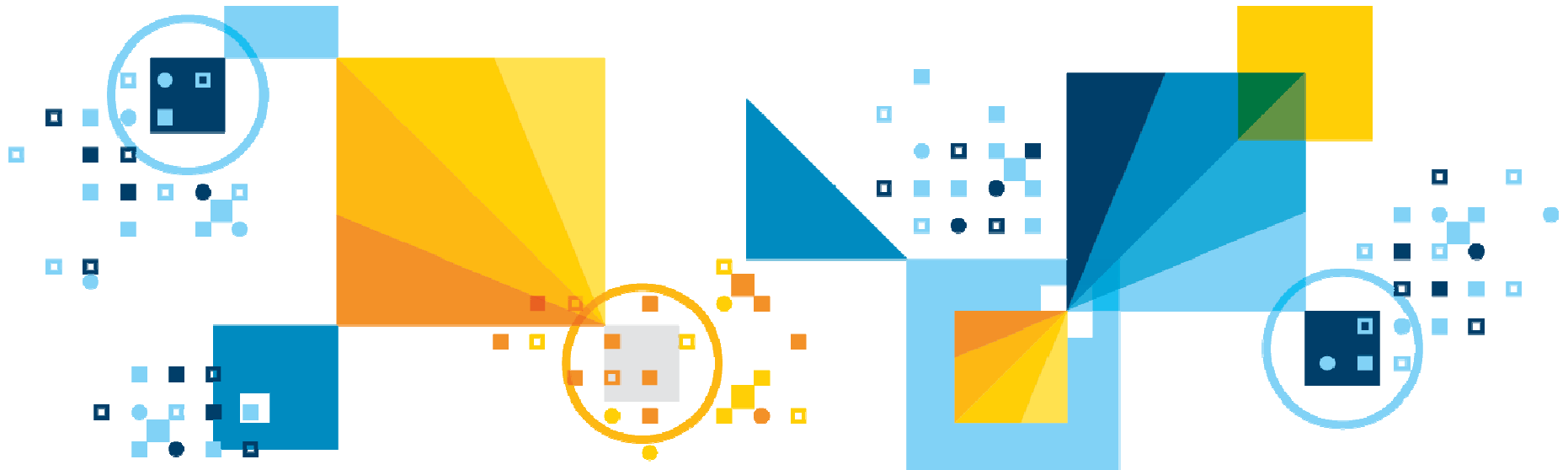


Adrian Burke
DB2 SWAT team
agburke@us.ibm.com

Addressing DDF Issues and Best Practices



Agenda

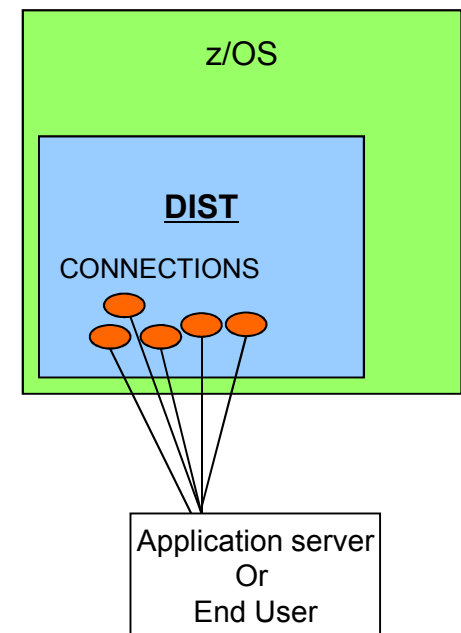
- **Basic concepts and terms**
 - What is a connection?
 - What is a thread?
 - How are resources pooled
- **Stale connections and timeouts**
- **Parameters**
- **Profiles**



Common terms

■ What is a connection?

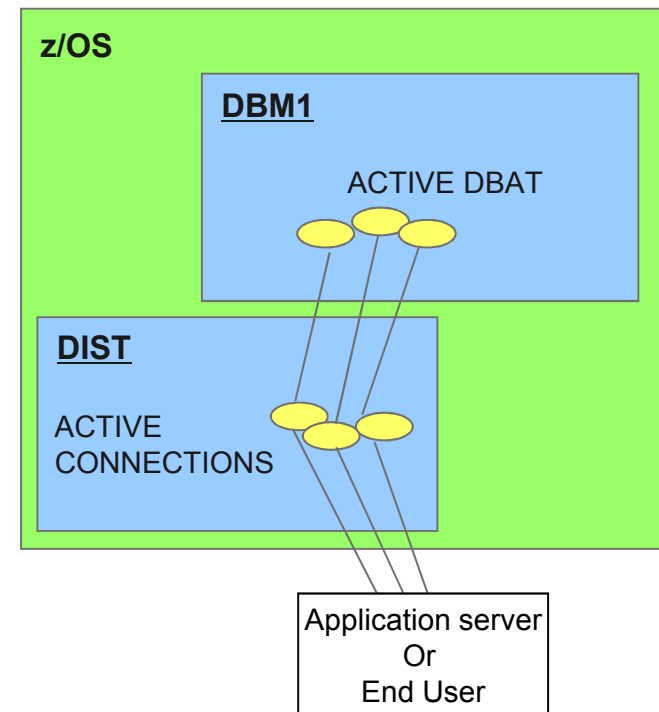
- DRDA over TCP/IP = IP address (domain) and port resulting in an end-point or socket and a control block in DIST address space
 - Total number governed by ZPARM **CONDBAT** (default = 10,000 since V8)
 - ✓ At 80% of CONDBAT, DSLN074I message in MSTR log
 - ✓ If CONDBAT is exceeded, DSNL030I message in MSTR log and connection requests are rejected and DB2 cannot be accessed remotely
 - Socket not released until remote requestor closes connection
 - ✓ No way for DB2 to cancel/close it
 - ZPARM **TCPKPALV** determines interval to ping socket to ensure connection is still there – avoid hung thread in DB2 if connection drops
 - ✓ Default is 120 seconds
 - ✓ ENABLE means takes COMM Server default of 120 *minutes*
- Legacy myth that connections are expensive
 - 2-4K in storage footprint above 2GB bar in DIST address space
 - ✓ Still have 258 bytes in Comm server ECSA
 - Max for CONDBAT has been 150,000 since DB2 V7



Common terms ...

■ What is a database access thread (DBAT)?

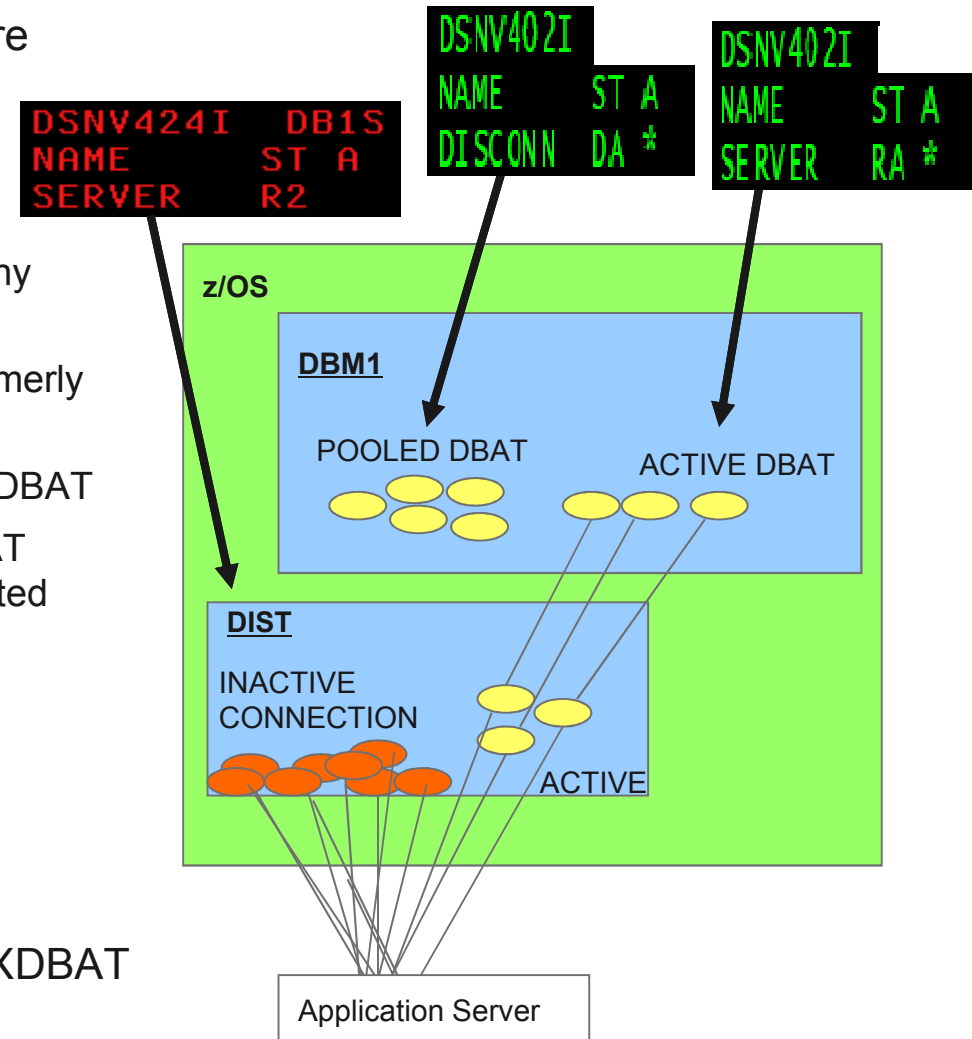
- Executed under WLM managed Enclave SRB mode
- A DBAT is associated with a connection until the connection commits/ terminates
 - ZPARM **MAXDBAT** still 200 by default
 - ✓ If MAXDBAT is hit, DSNL092I in MSTR log and inbound requests queue up to CONDBAT, then requests rejected
 - ZPARM **MAXCONQN** limits the depth of the queue for a DBAT after MAXDBAT has been hit before being canceled/rerouted (default OFF)
 - ZPARM **MAXCONQW** limits the time a DBAT request remains queued until it is canceled/rerouted (default OFF)
 - ZPARM **IDTHTOIN** determines how long a thread remains active between issuances of SQL and COMMIT
 - ✓ Default 120 seconds
- Storage in DBM1 (common range 12KB-2MB, or larger)
 - About 12KB in 31-bit private, the rest in 64-bit shared
 - -DIS THREAD(*) SERVICE(STORAGE) to see 31-bit usage



Thread pooling

■ Thread pooling and inactive connection support

- ZPARM **CMTSTAT** determines whether threads are disassociated with a connection at commit or at connection termination
 - CMTSTAT=**INACTIVE** (default as of V8)
 - ✓ At commit, threads are pooled to be reused by any new/resumed request
 - ✓ Connection becomes an inactive connection (formerly called a type 2 inactive DBAT)
 - » Still holds a socket and counts against CONDBAT
 - ✓ ZPARM **POOLINAC** determines how long a DBAT remains unused in the pool before being terminated
 - » Default 120 seconds
 - CMTSTAT=**ACTIVE**
 - ✓ Thread stays associated with a connection until connection terminates
- Command -DIS THREAD(*) DETAIL for status
- Both active and pooled DBATs count towards MAXDBAT



Thread pooling ...

■ **Recommendation: Utilize thread pooling and inactive connection support**

- Benefits for DB2 z/OS thread pooling:
 - CPU savings in DB2, by avoiding repeated creation and destruction of DBAT
 - Real memory savings in z/OS, by reducing the number of concurrent DBATs
 - Virtual storage savings in DBM1, by reducing the number of DBATs
 - Greater capacity to support DRDA connections since they are disassociated from the thread
 - WLM granularity at a unit of work boundary for enclaves
- What prevents a thread from going inactive?
 - Packages bound with KEEP DYNAMIC(YES)
 - ✓ Statement is held in local statement cache across commit boundary (SAP)
 - Held cursors and/or open LOB streams
 - Declared Global Temporary Tables result sets held beyond commit (no Drop on Commit)
 - High Performance DBATs (JCC packages bound DEALLOCATE and –MODIFY DDF command)
 - ✓ DBAT reused 200 times then recycled
 - ✓ ZPARM POOLINAC controls duration of inactivity
 - ✓ There should be a ‘separate’ allocation of DBATs for applications using High Performance DBATs

Key DB2 DDF Parameters

Parameter	Possible values	Default - V8*	Description
DDF	NO / AUTO / COMMAND	No	DDF Startup
CMTSTAT	ACTIVE / INACTIVE	INACTIVE*	Thread Pooling - Pool inactive threads
CTHREAD	1-2,000	200*	Max users - allied (local) threads RRSAF or CAF (CICS, IMS, TSO, Batch attach, SPUFI, Classic QMF, etc)
MAXDBAT	0-1,999 (V10 – 10,000)	200*	Max remote active DDF Threads - DBM1 Address Space
CONDBAT	0-15,000 (V10 – 10,000)	10,000*	Max remote connections - DDF Address Space
MAXTYPE1	0-CONDBAT (V10 – 150,000)	0	Max inactive DBATs, these are used for private protocol. DRDA uses inactive connections.
POOLINAC	0-9,999	120	Approximate time, in seconds that an inactive/unused DBAT can remain idle in the pool before it is terminated. DBAT deleted after being used 200 times also.
IDTHTOIN	0-9,999	120*	The IDLE THREAD TIMEOUT (IDTHTOIN) parameter specifies the time (in seconds) that an active DBAT can remain idle before it is canceled. – Should set a “little” higher than TCPKPALV
TCPKPALV	ENABLE / DISABLE / 1-65534	120*	TCP/IP keep alive (Goes hand-in-hand with IDTHTOIN)
TCPALVER	NO (server), YES, CLIENT, SERVER, SERVER_ENCRYPT	NO	whether DB2 is to accept TCP/IP connection requests that contain only a user ID (no password, RACF PassTicket, or Kerberos ticket).

Connection pooling

▪ Connection pooling should be enabled in the application server

- Logical connection from the application to the driver
- Seen as inactive connections in DB2 when no SQL requests are flowing
- No physical connection while in pool and saves creation/destruction of connection object

▪ In WebSphere (defaults) ... others later

- Max Connections (10)
 - max connections from JVM instance
- Min Connections (0)
 - minimum number of connections in pool
- Reap Time (180)
 - How often cleanup of pool is scheduled in seconds
- Unused Timeout (1800)
 - How long to let a connection sit in the pool unused (won't affect Min Conns)
- Aged Timeout (0)
 - How long to let a connection live before recycling helps control inactive connections to DB2
- Purge Policy (entirepool)
 - After StaleConnection, does the entire pool get purged or only individual connection

PercentUsed	connectionPoolModule.percentUsed	9	The average percent of the pool that is in use
PercentMaxed	connectionPoolModule.percentMaxed	10	The average percent of the time that all connections are in use



Connection Pool statistics in WebSphere should be used to determine the applications need for concurrent connections

Recommendations WITHOUT Sysplex WLB

- **WAS settings relative to DB2 timeouts, use <, > suggestions to adjust defaults**
 - TCPKPALV (Enable=120min) < IDTHTOIN (120sec) - we want to ping the socket prior to having DB2 cancel a thread for being idle incase we have lost the TCP/IP connection
 - IDTHTOIN (120) > Aged Timeout (0) : 1) reduce the possible number of inactive connections seen in the DB2 member 2) Allow the connection to be re-driven to another member when Sysplex WLB is being used
 - Small overhead in re-establishing a new connection when the connection is closed
 - Unused Timeout (1800) > Reap Time (180) - means we kill unused connections at the server to free up unused resources, and that no setting should be less than the Reap time, since it is the daemon that goes out to determine if other timeouts have popped
 - But if you have a minimum connection count then those will not be affected by Unused timeout
 - commandTimeout/queryTimeout (driver property) – how long the app waits for response from its request (SQL) before canceling its request
 - Connection Timeout (Driver property) – how long can the application can wait on a connection
 - ****IDTHTOIN > TCPKPALV > Aged Timeout > Unused Timeout > Reap > commandTimeout > ConnectionTimeout - this the end-to-end picture of the major timeout values**

****These settings are to eliminate stale, inactive, and hung connections in DB2**

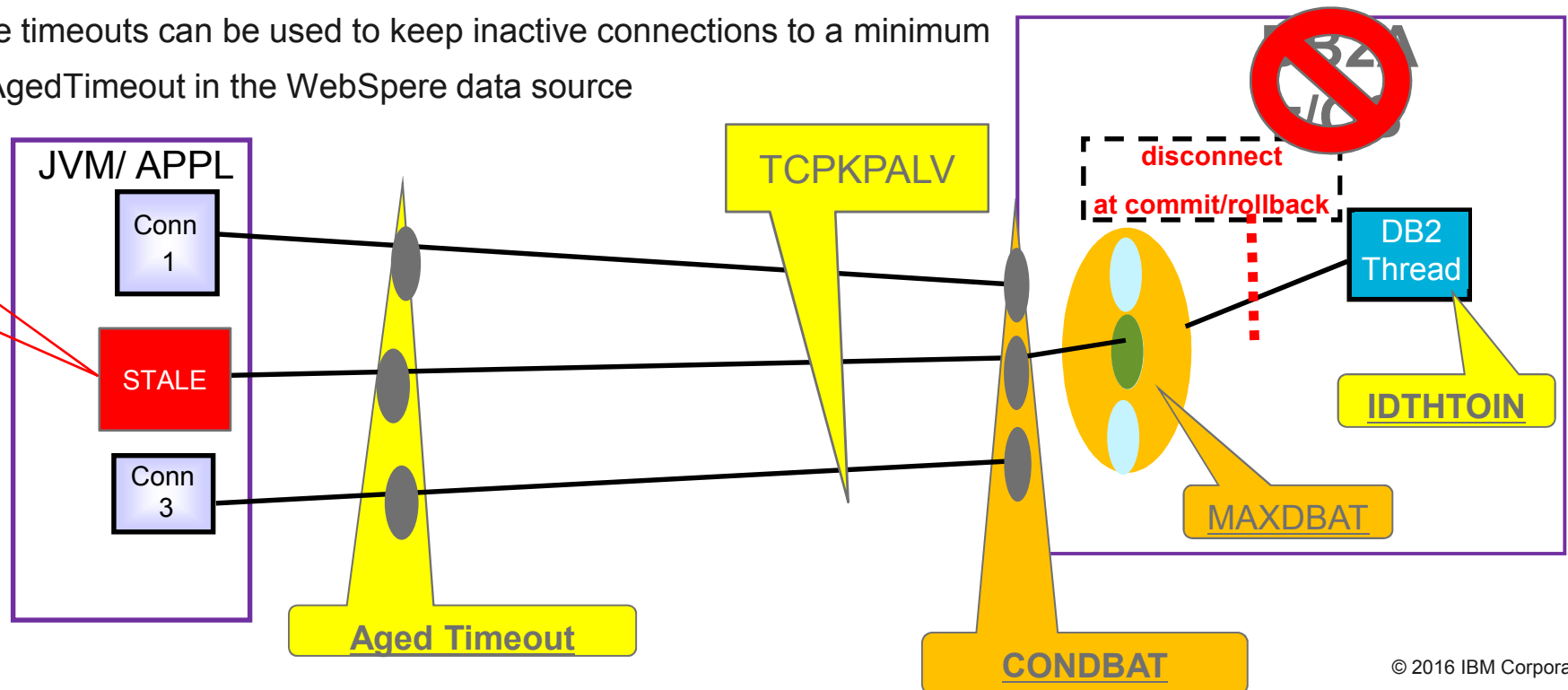
Timeout Values

- **Timeout values should be used to avoid stranded resources**

- If DB2 z is bounced or there is a connection failure and the REAP/AGED timeouts do not run then the next request could get a stale connection exception
- If IDTHTOIN pops, or thread is canceled it leads to stale connection exceptions with DB2 code of SQL1224 (TCPKPALV is to let DB2 know a requestor is no longer there) – PurgePolicy then takes effect

- **If inactive connections continue to grow and reach CONDBAT they could cause a service outage to other applications**

- These timeouts can be used to keep inactive connections to a minimum
 - AgedTimeout in the WebSphere data source



Stale Connections

- **Stale connections and statement objects seen by the applications often surface with DSRA9110E messages - SQL1224 in JVM logs if:**

- Database is recycled, TCP/IP connection is lost, or Firewall timeout pops and
 - REAP/AGED does not run for apps that do not close() their logical connection at commit
 - REAP/Unused does not run to cleanup existing pooled conns
 - Or Min Connections > 0
- If timeout values on DB2 are lower than your application timeouts
 - IDTHTOIN vs. queryTimeout and commandTimeout are JDBC and .NET driver properties
 - ✓ Defaults are 120 seconds, '0', and '30' seconds respectively
 - Or REAP/UNUSED > IDTHTOIN and apps use CURSOR HOLD or KEEP DYNAMIC

BUT applications SHOULD be able to handle these exceptions, examples here:

- <http://www-01.ibm.com/support/docview.wss?rs=180&uid=swg21063645>
- If apps have this logic then set purge policy to FailingConnectionOnly, if you do not have this logic then use EntirePool to avoid running into the stale connections
- **With enableSysplexWLB=TRUE the driver guarantees a useable connection and cleans up stale transports for you**

Stale Connections...

- **To analyze the victim/culprit of a cancel command, as of V10 we have the extended correlation token to match the thread in DB2 z and the remote unit of work**
- On z/OS
 - DDF DSNV442 (-DIS THREAD), DSNL027I (IDTHTOIN) and DSNL030I (CONDBAT hit) messages and lock-related messages DSNT318I, DSNT375I, DSNT376I, DSNT377I (timeout/deadlock) all contain the token
 - ✓ 00D3003B for idle thread timeout being hit
 - Connection correlation token (CRRTKN) and the logical unit of work ID (LUWID) in the z/OS diagnostic messages
 - Accounting trace information in the LUWID field or token filed (QWHC record header)
- On the distributed platform
 - SQL1224 in JVM trace, or JCC trace
 - Relates to IDTHTOIN, thread cancels, and abends

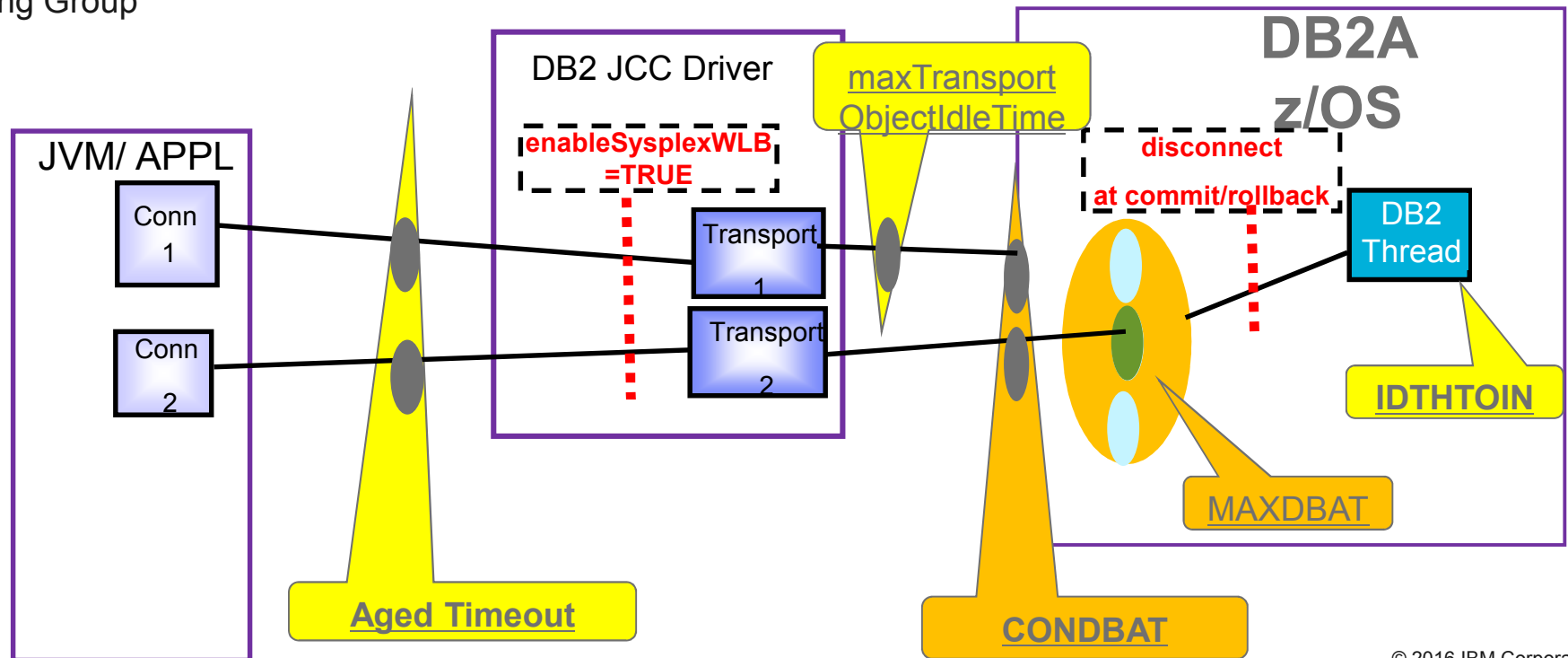
```

DSNL027I @ SERVER DISTRIBUTED AGENT WITH
          LUWID=G91702F8.CD1B.101018185602=16
THREAD-INFO=ADMF001:mask:admf001:db2bp:*:*:*|<9.23.2.248.52507.101018185602>
          RECEIVED ABEND=04E
          FOR REASON=00D3001A
  
```

REQUESTER :	9.23.2.248	CORRNAME:	G1PBTC00	LUW LUN:	DA84
MAINPACK :	YPKSFC00	CORRNMBR:	4	LUW INS:	101018185682
PRIMAUTH :	G1PDBC	CONNTYPE:	DRDA	LUW SEQ:	71
ORIGAUTH :	G1PDBC	CONNECT :	SERVER		

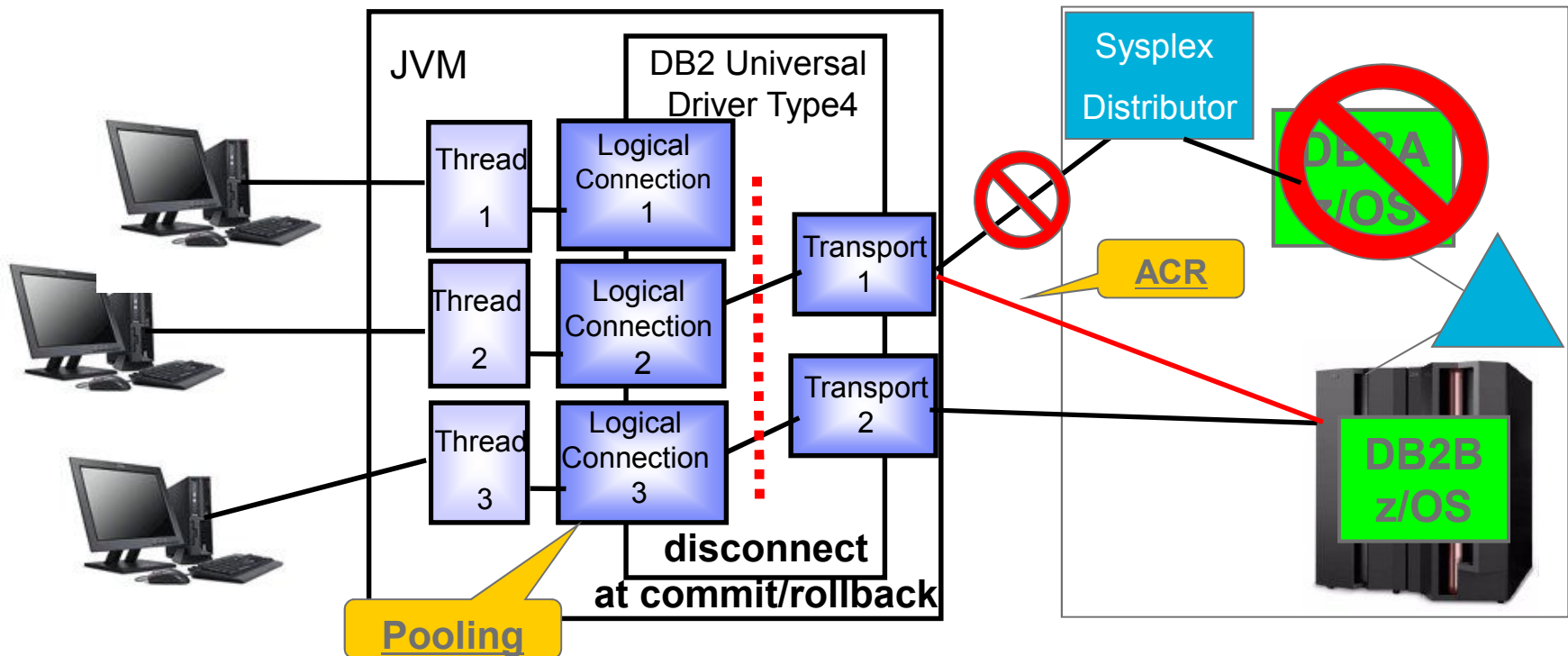
Timeout Values

- With `enableSysplexWLB = TRUE` the transports represent the physical connection to DB2 while the active connections in the pool are simply logical connections = NO stale conns
 - SysplexWLB allows for seamless transaction reroute, and intelligent routing of transactions based on capacity
 - `maxTransportObjectIdleTime` in the JDBC properties is a default of 10 seconds so inactive connections will not remain in DB2 z
 - If IDTHTOIN pops that Transport is destroyed and another one will then service Conn 2's next request
 - If there is a network failure or DB2 failure another transport will route that work to another member of the Data Sharing Group



Sysplex Workload Balancing

- Automatic Client Reroute behavior means if we loose a DB2 or connection, we can re-drive the SQL request across another existing connection to another member of the group without the application's knowledge
- Even without failure WLM will feed back results so transactions, not just connections are managed across the sysplex



Recommendations WITH Sysplex WLB

- **Zero out application server parameters that correspond to Transport object settings**
 - Aged timeout = 0 back to default / Unused = 0
 - Reap Time = 0 (also disables aged and unused timeouts)
 - Purge Policy = FailingConnectionOnly
 - Applications should know how to handle stale connection exceptions :
 - ✓ <http://www-01.ibm.com/support/docview.wss?rs=180&uid=swg21063645>
- **Use the Driver settings to further tune connections/timeouts if there are issues**
 - maxTransportObjects (1,000 default)
 - $\text{TOTAL \# transports} * \text{\#drivers} < \text{CONDBAT} * \text{DB2 Members of DS Group}$
 - maxTransportObjectIdleTime = 10 seconds (default)
 - This is how you get rid of inactive connections (like Aged timeout in WAS)
 - maxTransportObjectWaitTime, default is 1 second
 - SQLCODE -4210, SQLSTATE 57033 for connection waiting for a transport
 - maxRefreshInterval = 10 seconds (for WLM)

Tomcat / WAS/ JCC driver

<u>Tomcat</u>	<u>WebSphere Equivalent</u>	<u>Driver (w/ Sysplex WLB)</u>
Remove abandoned Timeout (seconds)	Aged Timeout	maxTransportObjectIdleTime
maxActive – Tomcat also has settings for maxIdle connections	maxConnections – WAS does not distinguish between active and idle in number	maxTransports
initialSize – Tomcat creates all these at start up	Min Connections- WAS will wait until they are needed then keep them around	minTransportObjects
**validationQuery – string for query to submit (leave it out) – pops on query timeout in driver	Connection validation timeout - SQL validation deprecated, use JDBC validation (JDBC 4.0)	Sysplex WLB should take care of this, re-drive failed connections through sysplex distributor, and automatic client reroute
testOnBorrow- check it when connection used from pool testOnReturn –check in going back to pool	Validate new connections Validate existing connections	** <u>Turn Validation off to avoid overhead..</u>

- Many open source or vendor application servers do not take advantage of availability enhancements in z/OS

As an example Tomcat still has the concept of a validation query

- Enabling this function means the connection object in the pool is tested before the application acquires the connection, as well as when the connection is returned to the pool

The JDBC 4.0 specification made this feature irrelevant as the driver itself (IBM Data Server Driver) guarantees the connection given to the application server from the pool

.NET / WAS/ JCC driver

<u>IBM .NET provider/ Default</u> <u>Connection Pooling = true</u>	<u>WebSphere Equivalent</u>	<u>JDBC Driver</u>
Connection Lifetime (0) – only checked when returned to pool	Aged Timeout (0)	maxTransportObjectIdleTime (10)
Max Pool Size = No Maximum	maxConnections – (10) WAS does not distinguish between active and idle in number	maxTransports (1,000)
N/A	Unused Timeout	N/A
N/A	REAP time	N/A
Min Pool Size = 0	Min Connections- WAS will wait until they are needed then keep them around	minTransportObjects
Connection timeout	Connection timeout	maxTransportObjectWaitTime (1 sec)

**** Settings from PMR →**

	Default

Max Pool Size	No max
Min Pool Size	0
Connection Lifetime	60

Thread pooling ...

- The display DDF detail command gives a snapshot of the remote work going on
 - The presence of DSCDBAT proves the thread pooling is enabled
 - Ideally QUEDBAT and CONQUED will always equal 0
 - PKGREL BNDOPT or BNDPOOL allows the use of high performance DBATs
 - INADBAT is no longer related to Private Protocol, but indicates Monitor Threads profile exceptions

How many threads are currently doing work as well as DSCDBATs

-DIS DDF DETAIL

CMTSTAT=INACTIVE
So thread pooling is enabled

How many connections and threads you think you can handle

```

DSNL090 DT=I CONDBAT= 1000 MDBAT= 200
DSNL091 MCONQN= 0 MCONQW= 0
DSNL092 IADBAT= 31 QUEDBAT= 0 INADBAT= 0 CONQUED= 0
DSNL093 DSCDBAT= 9 INACONN= 168
DSNL094 WLMHEALTH=100 CLSDCONQN= 0 CLSDCONQW= 0
DSNL106 PKGREL = COMMIT
  
```

Monitor thread profile exception hit and DBAT requests queued in DB2

You hit MAXDBAT at some point and requests were queued

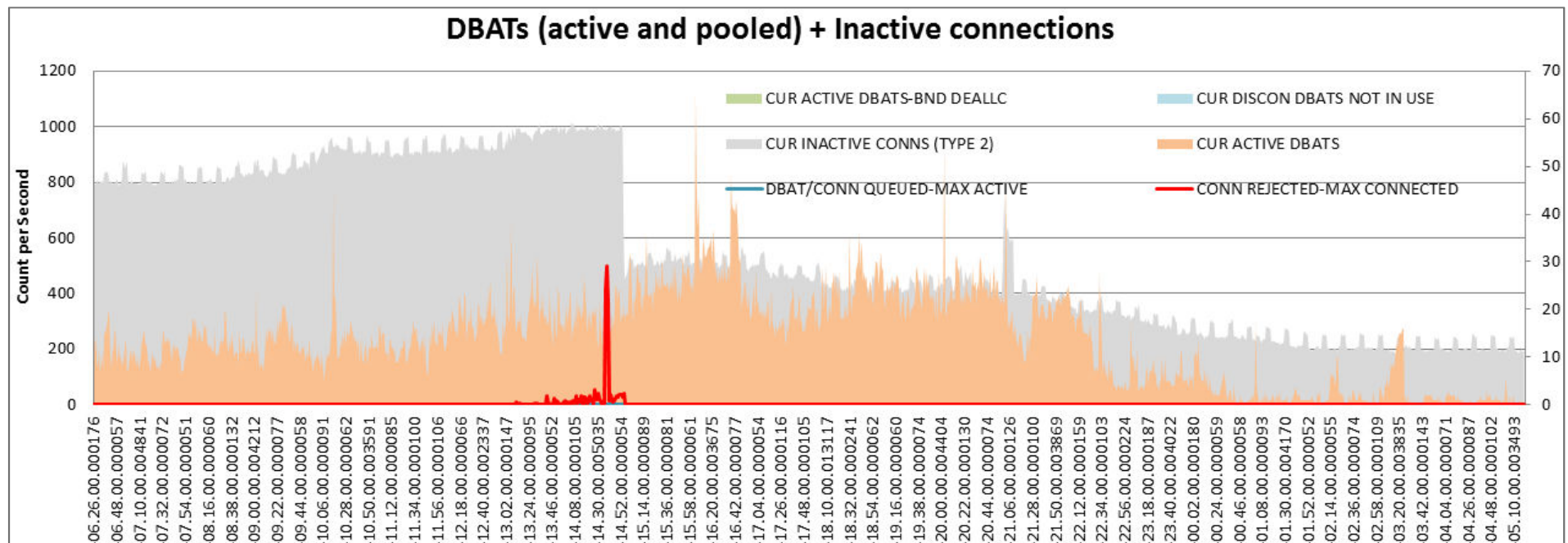
How many threads are lounging in the pool

RELEASE(COMMIT)

Type 2 inactive thread/
inactive connection

Customer Conns Rejected

- A connection storm or incremental growth can occur when database access threads are not even consumed
- Need to monitor the number of inactive connections and ensure CONDBAT is never encroached upon
 - Sum of inactive connections and active DBATS
 - DSNL047I (at 80% of CONDBAT DB2 health lowered, only useful with Sysplex WLB) messages generally come too late to react



What level of the driver is out there?

■ **-DIS LOCATION DETAIL**

– DSNL200I – JCC' is the driver, SQL' is DB2 Connect Server

- *vv* A 2-digit number that identifies the product version number, such as 09 or 10.
- *rr* A 2-digit number that identifies the product release level, such as 01 or 05.
- *m* A 1-digit number that identifies the product modification level, such as 0 or 1

– In the –DIS you can see the IP address it came from, product ID to let you know what level it is at.

- The ATT column shows if the connection is using Sysplex WLB, XA two phase commit, or encryption (AES/ TLS)

JCC03640	<-- 9.7 FP6
SQL09076	<-- 9.7 FP6
JCC03660	<-- 10.5 GA or FP1
JCC04160	<-- 10.5 GA or FP1

Example 2-2 DISPLAY LOCATION report

DSNL200I -D022 DISPLAY LOCATION REPORT FOLLOWS-

LOCATION	PRDID	T	ATT	CONNS
::9.12.4.142	JCC03640	S		50
			WLB	50
			XA	50

What level of the driver is it?

- **What level of the Driver equals what level of Connect?**
 - <http://www-01.ibm.com/support/docview.wss?uid=swg21363866>
- **What driver level came through DB2 for z/OS maintenance stream**
 - https://www-304.ibm.com/support/docview.wss?q1=pk77599&rs=0&uid=swg21428742&cs=utf-8&lang=en&loc=en_US&cc=us

Cancelling Inactive Connections

- If you have 'stale' connections you can get rid of them one-at-a-time

- Display the inactive connections

-DISPLAY THREAD(*) TYPE(INACTIVE) DETAIL

V437-WORKSTATION=9.76.193.254 ← IP address

V448--(1) 446:53637 ← local and foreign ports

- Find the connection identifier (CONN) associated with the IP addresses for that local/foreign port combination:

D TCPIP,,NETSTAT,conn,ipaddr=9.76.193.254

USER ID CONN STATE

DB1SDIST 162724E6 ESTBLSH ← get the CONN identifier

- Drop the inactive connection which corresponds to that identifier:

V TCPIP,,DROP,CONN=162724E6

The Importance of Committing?

▪ **Uncommitted units of recovery (URCHKTH, URLGWTH):**

- Elongated DB2 restart and recovery times (especially with GBP dependent objects)
- Reduced availability due to retained locks held for a long time (data sharing)
- Lock contention due to extended lock duration >> timeouts for other applications
- Ineffective lock avoidance (data sharing)
- Problems getting DB2 utilities executed
- Lack of space reuse for Universal table spaces

▪ **Readers (LRDRTHLD):**

- REORG and other utilities drain process could fail, or necessitate the FORCE option for the readers still holding a claim
- If the application is not actively reading any rows, but has not committed it could be causing objects to remain group buffer pool dependent, hence increasing the risk of objects going into GRECP/LPL status in the event of a failure
- Space growth in LOB objects due to global CLSN for data sharing environments (including catalog objects)

Driver properties

▪ Cursor Hold

- Driver defaults open result set cursors using the **WITH HOLD** option
- WITH HOLD cursor result sets persist across COMMITs
 - Java driver can override default using **setResultHoldability=2**
 - Non-java driver can override default using **CursorHold=0**

▪ Implicit Close – default is (0) which means close and commit in V10

- Specifies whether result set cursors are closed automatically after all rows are fetched
 - this could affect LOB locators, cursor positioning, etc.
 - Disabling leaves result set cursors open on DB2 after all rows are fetched
- Java driver **queryCloseImplicit** property
- Non-java driver **CursorTypes** set to static forward only cursors
- AUTOCOMMIT will affect behavior as well

Setting Driver Properties

- **DataSource is preferred over DriverManager for portability/consistency/admin**

- **DriverManager (specified in URL)**

- ```
Class.forName("com.ibm.db2.jcc.DB2Driver");
Connection con = DriverManager.getConnection("jdbc:db2://localhost:446/sample[property= value,...]");
```

- **DataSource**

- ```
Context ctx=new InitialContext();  
DataSource ds=(DataSource)ctx.lookup("jdbc/myDB");  
Connection con=ds.getConnection();
```

- **Central Properties file preferred over the URL method**

- **For the JAVA driver**

- DB2JccConfiguration.properties in CLASSPATH can be used
 - db2jcc.override.*property-name* – overrides any setting with same name

- **For non-JAVA driver how do I maintain standards?**

- db2cli.ini (text) and/or db2dsdriver.cfg (XML file) – **preferred**
 - ✓ **Order of precedence** - http://www-01.ibm.com/support/knowledgecenter/SSEPGG_10.5.0/com.ibm.swg.im.dbclient.config.doc/doc/r0061062.html?cp=SSEPGG_10.5.0%2F2-0-2-5-7-0-4&lang=en
 - ODBC, CLI, OLE-DB and.NET or open source (Perl, PHP)

Monitor profiles to limit threads/connections

■ Catalog table holds profiles

- DSN_PROFILE_TABLE
 - Describe threads/connection targets
- DSN_PROFILE_ATTRIBUTES
 - What you limit and how

■ Can limit:

- Connections
- Active threads
- Idle thread timeout (up or down) – CAUTION: IDTHTOIN will no longer apply to these threads

■ Limit based on

- LOCATION only
- PRDID only
- AUTHID, ROLE, or both.
- COLLID, PKGNAME, or both
- One of CLIENT_APPLNAME, CLIENT_USERID, CLIENT_WORKSTNNAME

PROFILEID	AUTHID/ ROLE	LOCATION – IP address, domain name, Alias	PRDID
1	WASPRD*		
2			JCC03570

PROFILEID	KEYWORDS	ATTRIBUTE1	ATTRIBUTE2
1	MONITOR THREADS	EXCEPTION_DIAG LEVEL3	10
2	MONITOR IDLE THREADS	WARNING_DIAGL EVEL2	180

■ Warning or Exception

- 1 attribute could be warning (just once or for each threshold), another could be exception where incoming connections are queued or failed
- DSNT77xI reason code 00E3050x with various degrees of detail

Limiting Total Connections

- **Could use Profiles to protect DB2 from connections spawned by application servers in a loop (garbage cleanup), poorly behaving application, or a denial of service attack**
 - ** If implemented, previous equation for stale connection objects would need to be revisited **
- **PI70250 (PI74729) added wildcarding and more granular messaging to V11**
 - Profile #1: Wildcard location so any remote IP address or domain using more than 1,000 connections to any member writes a warning to the master log (DSNT773I)
 - Profile #2: That IP address can only use up to 500 connections to a member where the profile is started, and the 501st is rejected

SYSIBM.DSN_PROFILE_TABLE

PROFILEID	LOCATION
1	*
2	9.76.193.254

SYSIBM.DSN_PROFILE_ATTRIBUTES

PROFILE ID	KEYWORDS	ATTRIBUTE1	ATTRIBUTE2
1	MONITOR CONNECTIONS	WARNING_DIAGLEVEL3	1000
2	MONITOR CONNECTIONS	EXCEPTION_DIAGLEVEL3	500

http://www.ibm.com/support/knowledgecenter/SSEPEK_11.0.0/com.ibm.db2z11.doc.perf/src/tpc/db2z_monitorconnectionprofiles.data?lang=en

Profile monitoring

■ System profile monitoring – DSN_PROFILE_ATTRIBUTES

– ATTRIBUTE2 – where you set the limits

- Number of connections for MONITOR CONNECTIONS
- Number of threads for MONITOR THREADS
- Timeout value for idle threads in seconds for MONITOR IDLE THREADS

■ -START PROFILE command needed on each member of DS group

- Needs to be issued each time DBM1 is started (until V12 or V11+PTFs coming late 2017)
- Starts all profiles in SYSIBM.DSN_PROFILE_TABLE table where column PROFILE_ENABLED='Y'
- If you change a profile it must be reloaded in memory for changes to take affect

■ SYSIBM.DSN_PROFILE_HISTORY and ATTRIBUTES_HISTORY tables contain STATUS column

- ACCEPTED or REJECTED to determine if your profile was deployed successfully
 - ✓ i.e. 'ACCEPTED - DOMAIN NAME IS RESOLVED INTO IP ADDRESS'

■ Order of precedence in evaluation of profiles

- https://www.ibm.com/support/knowledgecenter/SSEPEK_11.0.0/perf/src/tpc/db2z_systemprofileinteract.html

Profile monitoring ...

▪ DSN_PROFILE_ATTRIBUTES

- ATTRIBUTE1 has two parts: (warning or exception) and the amount of information (DIAGLEVEL)
 - These messages are issued to the console at most every 5 minutes when profile exceeded
 - WARNING and WARNING_DIAGLEVEL1 – DSNT771I
 - EXCEPTION and EXCEPTION_DIAGLEVEL1 - DSNT771I
- WARNING_DIAGLEVEL2 – **DSNT772I** includes profile ID and **reason code**
 - ✓ MONITOR CONNECTIONS – **00E30503**
 - ✓ MONITOR THREADS – **00E30505**
 - ✓ MONITOR IDLE THREAD – **00E30501**

DSNT771I -DB2B DSNLAGNT A MONITOR
PROFILE WARNING
CONDITION OCCURRED 1 TIME(S)

DSNT772I -DB2B DSNLAGNT A MONITOR PROFILE WARNING
CONDITION OCCURRED 1 TIME(S) IN PROFILE ID=4
WITH PROFILE FILTERING SCOPE=IPADDR
WITH REASON=00E30505

Profile monitoring ...

▪ DSN_PROFILE_ATTRIBUTES

– ATTRIBUTE1

- EXCEPTION_DIAGLEVEL2 – **DSNT772I** and action is take to limit connections, threads, or timeout value
 - ✓ MONITOR CONNECTIONS get **00E30504** reason code and application receives **SQLCODE -30081**
 - ✓ MONITOR THREADS (N threshold) get:
 - » **00E30506** if filtering on LOCATION (*IP address or domain name*) threads simple queue after the threshold is hit; this will continue until a thread opens up or CONDBAT is reached
 - » **00E30507** if filtering on PRDID, ROLE, AUTHID, ROLE_AUTHID, LOCATION (*location name, alias name, database name*)
 - » SQLCODE -30041 sent to driver when threads = 2N+1
 - » No Automatic Client Reroute, application must re-drive connection
 - » **00E30508** if filtering on COLLID, PKGNAME, COLLID_PKGNAME, CLIENT_APPLNAME, CLIENT_USERID, CLIENT_WRKSTNNAME
 - » SQLCODE -30041 sent to driver when threads = 2N+1
 - » No ACR, application must re-drive connection
 - ✓ MONITOR IDLE THREAD gets **00E30502** reason code
 - » **CAUTION: if you MONITOR IDLE you also need EXCEPTION**
 - » If it is an exception then same old DSNL027I and 00D3003B reason code

DSNT772I -DB2B DSNLAGNT A MONITOR
PROFILE EXCEPTION CONDITION OCCURRED 10
TIME(S) IN PROFILE ID=3
WITH PROFILE FILTERING SCOPE=AUTHID
WITH REASON=00E30507

Profile monitoring ...

■ DSN_PROFILE_ATTRIBUTES

- PI60841 brought in more options for idle threads
 - WARNING_MESSAGE_FOR_IDLE_THREAD_TIMEOUT (with V11 PI60841)
 - ✓ DSNT773I message for a thread that remains in an idle state. When a client request message is received, and a COMMIT or ROLLBACK is completed with no resources active past the end of the unit-of-work, DB2 removes the warning against the thread
 - EXCEPTION_ROLLBACK_DIAGLEVEL2 (with V11 PI60841) – only applies to idle threads and causes resources to be freed and work to be rolled back BUT the connection remains intact
 - ✓ **use caution with this setting as if there are result sets or DGTTs left beyond the COMMIT they will be rolled back and the application will not know what happened to them – NO SQL CODE
 - ✓ DSNT772I and **DSNL030I** message with **00D30050** reason code, SQL -919 only if changes rolled back
- Within V11 and PI31957 there are three levels of detail for each type
 - *TYPE_DIAGLEVEL3* (in PI31957) – includes DSNT772I with detailed information about what filtering criteria triggered the WARNING/EXCEPTION plus thread and connection information when available
 - ✓ Issued once per occurrence – so you would need to alter and stop/start profile if it floods the console
 - ✓ This would be very useful for testing as well as cases where ‘well-behaved’ applications grow
 - » DSNT773I – for warning
 - » DSNT774I – for exception

```
DSNT774I -DB2B DSNLAGNT SERVER DISTRIBUTED AGENT 732 WITH
LUWID=G91EDE68.G403.000000000000
THREAD-INFO=-UNKNOWN:*****
FOR LOCATION=::FFFF:9.30.222.104
RECEIVED MONITOR THREADS
EXCEPTION DUE TO PROFILE ID=4
```

And for all your efforts..

WLM and MOBILE Pricing

- **WLM service classification rules can be used to subset transactions coming in from mobile devices using client accounting information**
 - You could also use client IP address to distinguish them; this IP address would be the same one you would see in the accounting string if you did a –DIS THREAD
- **Using a sub-rule for the classification of mobile WebSphere transactions set the Reporting Attribute in WLM to MOBILE**
- **The CPU consumption will then be reported in the WLM activity report, and SCRT report, which allows customers to reduce their MLC bill**

```

Modify Rules for the Subsystem Type                               Row 1 to 4 of 4
Command ==> _____ Scroll ==> PAGE

Subsystem Type . : DDF          Fold qualifier names?  Y  (Y or N)
Description . . . Distributed DDF work

Action codes:  A=After      C=Copy      M=Move      I=Insert rule
               B=Before     D=Delete row R=Repeat    IS=Insert Sub-rule
                                   <=== More

Action  -----Qualifier-----  Storage  Reporting  Manage Region
Type    Name      Start    Critical  Attribute  Using Goals Of

_____ 1 UI        WAS*      _____  N/A      NONE      N/A
_____ 2 CAI       WASMOB    _____  N/A      MOBILE    N/A
  
```

TRANSACTION APPL% :	TOTAL :	CP	N/A	AAP/IIP ON CP	N/A	AAP/IIP	N/A
	MOBILE :	CP	N/A	AAP/IIP ON CP	N/A	AAP/IIP	N/A

References

- **PI63531 – Improve granularity of monitor profile messages**
 - <http://www-01.ibm.com/support/docview.wss?crawler=1&uid=swg1PI63531>
- **Techdoc for V10 and V11 MEMU2 with spreadsheet sample**
 - <https://www-01.ibm.com/support/docview.wss?uid=tss1prs5279>
- **DB2 for z/OS and WebSphere Integration for Enterprise Java Applications**
 - <http://www.redbooks.ibm.com/abstracts/sq248074.html?Open>
- **Subsystem and Transaction Monitoring and Tuning with DB2 11 for z/OS**
 - <http://www.redbooks.ibm.com/redpieces/abstracts/sq248182.html?Open>
- **Upcoming Webcasts - World of DB2**
 - <https://event.on24.com/eventRegistration/EventLobbyServlet?target=reg20.jsp&referrer=&eventid=1360591&sessionid=1&key=22E701220E804AC43B6075FB7DA64318®Tag=84051&sourcepage=register>

