

# Db2 13 for z/OS Application Management Enhancements

*Jammie Dang*

*Senior Software Engineer*

*Db2 for z/OS Development*

**IBM**

# Agenda

- Application-level granularity locking control
  - Profile table enhancement for local threads
  - DDL break-in improvement with Profile table
- 
- Problem statement
  - Db2 12 vs Db2 13 scenario
  - Goal and solution

# Problem statements

- Different applications have different characteristics but have no way to express their own priorities in case of lock contention.
- The subsystem parameter IRLMRWT governs timeout interval for most lock requests and does not allow granular control of setting different timeout values for specific applications. This one-size-fits-all approach does not work in a multi-tenancy cloud environment with disparate application needs.
- Certain DDL activities can be prone to deadlocks. If the thread performing the DDL is chosen as the victim then scheduled DDL activities may fail, requiring repeat attempts and/or impacting subsequent work.


# Db2 12 scenario

- **Application timeout**
  - An application architect determines that the application characteristics require either a longer or shorter lock timeout interval
  - This requirement is not compatible with the requirements of other applications running in the same Db2
  - Thus, the application must be split off into its own Db2 system thereby creating application affinities
- **Deadlock**
  - Batch DDL frequently fails due to deadlocks, thereby requiring restart/retry logic and careful scheduling which is becoming increasingly difficult in 24x7 environments

## Pain points:

- increased management cost and overhead
- decreased availability and resiliency
- unable to perform schema change for new applications
- delay failed DDL may cause disruptive transactions

# Background information

- Subsystem parameter IRLMRWT:
    - 0 – 3600 seconds
    - not online changeable
  - Subsystem parameter DDLTOX: integer (0 – 254), multiple of IRLMRWT, applicable to DDL only
  - Subsystem parameter DLI BATCH TIMEOUT & IMS BMP TIMEOUT, parameter on IMS/CICS CREATE THREAD: 1 - 254, multiple of IRLMRWT, also influence deadlock victim resolution
  - Deadlock victim resolution: based on a number of factors such as
    - NO LOG table
    - Number of log records
    - Weighting factor...
-  Not easy to influence how Db2 chooses a requestor to cancel

# Background information (cont)

- IRLM command to set deadlock interval:

- `F irlmproc,SET,DEADLOCK=nnnn`

*where nnnn is in milliseconds*

- `F irlmproc,STATUS`

- IRLM's deadlock interval affects total lock wait time, especially in data sharing

# Goals

- *The application architect can easily set a lock timeout interval to a value that suits a specific application's or even an individual SQL statement's needs, thereby minimizing application lock contention and simplifying portability of applications to Db2 z without the need to assign the application to a separate Db2 system*
- *The DBA can choose a deadlock weighting value for a scheduled DDL process that helps to ensure that the DDL is not chosen as the victim in an eventual deadlock scenario, thereby optimizing the likelihood of success*

# Db2 13 scenario

- Lock waits
  - An application architect determines that the application characteristics require either a longer or shorter lock timeout interval
  - The application developer codes the SQL statement `SET CURRENT LOCK TIMEOUT=n`
- Deadlocks
  - A DBA or application developer with suitable authority can set the global variable `DEADLOCK_RESOLUTION_PRIORITY` to influence whether the process or application should be chosen as the deadlock victim



# The solution

- New CURRENT LOCK TIMEOUT special register governed by
  - new SQL statement SET CURRENT LOCK TIMEOUT
  - Compatible to Db2 for LUW syntax
- New built-in global variable DEADLOCK\_RESOLUTION\_PRIORITY to influence deadlock victim choosing decision
  - New SQL statement SET SYSIBMADM.DEADLOCK\_RESOLUTION\_PRIORITY

# The solution (cont)

- SET CURRENT LOCK TIMEOUT =
  - Null or Default: use IRLMRWT
    - ❑ Query of CURRENT LOCK TIMEOUT returns IRLMRWT value
  - WAIT or -1: timeouts are not to take place
  - NOT WAIT or 0: if a lock cannot be obtained, an error is to be returned immediately
    - ❑ no DSNT376I message, no IFCID 196
  - Integer/variable: between -1 and 32767
- CURRENT LOCK TIMEOUT is **also** applicable to claims, drains and other waiting periods similar to locks
- CURRENT LOCK TIMEOUT is **not** applicable to certain processes (P-locks, plan/package allocation – package locks)

# The solution (cont)

- Subsystem parameter IRLMRWT is online changeable
- New subsystem parameter SPREG\_LOCK\_TIMEOUT\_MAX: upper bound on the value that a user can specify for CURRENT LOCK TIMEOUT on the SET statement
  - -1 (default)
    - Rebuild subsystem parameter loadmod after migration
  - 0 – 32767
- High CURRENT LOCK TIMEOUT value or -1 could impact other concurrent applications/processes

# Instrumentation enhancements

- Accounting trace: number of SET CURRENT LOCK TIMEOUT statement executions - QXSTTIMEOUTFromAppl
- New IFCID 437: written when executing the SET CURRENT LOCK TIMEOUT statement

QW0437	DSECT	IFCID	QWHS0437
QW0437FromSource	DS CL1		Source of the SET statement
QW0437OldTimeOutNI	DS CL1		Null indicator for old value
QW0437OldTimeOut	DS F		Old value if not null
QW0437NewTimeoutNI	DS CL1		Null indicator for new value
QW0437NewTimeout	DS F		New value if not null
QW0437Status	DS CL1		Status

- IFCID 196 (timeout event): lock holders and waiters using IRLMRWT or CURRENT LOCK TIMEOUT

QW0196TI: waiter's timeout interval

QW0196TR: source of waiter's timeout interval

QW0196HTI: holder's timeout interval

QW0196HTR: source of holder's timeout interval

# Timeout message enhancement

DSNT376I PLAN=*plan-name1* WITH CORRELATION-ID=*correlation-id1* CONNECTION-ID=*connection-id1* LUW-ID=*luw-id1* THREAD-INFO=*thread-information1* IS TIMED OUT. ONE HOLDER OF THE RESOURCE IS PLAN=*plan-name2* WITH CORRELATION-ID=*correlation-id2* CONNECTION-ID=*connection-id2* LUW-ID=*luw-id2* THREAD-INFO=*thread-information2* ON MEMBER *member-name*. REQUESTER USING TIMEOUT VALUE *<tout-interval1>* FROM *<timeout-source1>*. HOLDER USING TIMEOUT VALUE *<tout-interval2>* FROM *<timeout-source2>*.

timeout-source:

- IRLMRWT- timeout occurred due to the expiration of the zparm IRLMRWT value
- Special Register - timeout occurred due to the expiration of the current lock timeout value set with the CURRENT LOCKTIMEOUT special register
- IRLM – timeout occurred due to other internal reasons in Db2 or IRLM

# The solution for deadlock control

- SET SYSIBMADM.DEADLOCK\_RESOLUTION\_PRIORITY =
  - Null or Default: Db2 to choose deadlock victim
  - Small integer/variable: between 0 and 255
- Higher value – less likely locks are denied
- Write privilege is required to set value
- Read privilege is required to query value
- DEADLOCK\_RESOLUTION\_PRIORITY is not applicable to certain processes (P-locks, plan/package allocation – package locks)
- New field in IFCID 172 to indicate whether a waiter's worth value is from DEADLOCK\_RESOLUTION\_PRIORITY – QW0172WAS

# The Solution (cont)

- CURRENT LOCK TIMEOUT requires APPLCOMPAT V13R1M500
- DEADLOCK\_RESOLUTION\_PRIORITY requires APPLCOMPAT V13R1M501
- Both are applicable to static and dynamic SQL
- IRLM APAR PH43770: data sharing
- IRLM PH45843 and DB2 APAR PH45840: high DEADLOCK\_RESOLUTION\_PRIORITY value




# PROFILE TABLE ENHANCEMENT



# System profile monitoring

Identify an application process context and actions to be taken

- Filtering criteria in DSN\_PROFILE\_TABLE:
  - LOCATION only
  - PRDID only
  - AUTHID, ROLE, or both
  - COLLID, PKGNAME, or both
  - One of CLIENT\_APPLNAME, CLIENT\_USERID, or CLIENT\_WORKSTNNAME
- Actions in DSN\_PROFILE\_ATTRIBUTES
  - Modeling production environment and IDAA offload
  - Monitor and control **remote server threads/connections**
  - Set special registers/global variables for **remote server threads** 

# System profile monitoring

- The Profile tables are loaded in memory when Profile is started via the `-STA PROFILE` command
- DDF address space needs to be started

```
-DB2ASTA PROFILE  
DSNT761I  -DB2A DSNT1RSP START PROFILE DETECTED  
          DDF IS NOT LOADED. PROFILE ROWS FOR SYSTEM  
          MONITORING WILL NOT BE ACTIVATED.  
DSNT741I  -DB2A DSNT1SDV START PROFILE IS COMPLETED.  
DSN9022I  -DB2A DSNT1STR 'START PROFILE' NORMAL COMPLETION
```

- Accepted or rejected rows are in `DSN_PROFILE_HISTORY` and `DSN_PROFILE_ATTRIBUTES_HISTORY` tables

# Background information

DSN\_PROFILE\_TABLE

PROFILEID	LOCATION	ROLE	COLLID	APPLNAME
1	COLUMBUS			
2		DBA_ROLE		
3			COLLECTION1	
4				NEWAPPL

DSN\_PROFILE\_ATTRIBUTES

PROFILEID	KEYWORD	ATTRIBUTE1	ATTRIBUTE2
1	SPECIAL_REGISTER	SET CURRENT REFRESH AGE = 'ANY'	Null
2	GLOBAL_VARIABLE	SET SYSIBMADM.MOVE_TO_ARCHIVE ='E'	Null

Attribute 2 = null: remote applications only, processed when 1<sup>st</sup> package is loaded

# Background information

DSN\_PROFILE\_TABLE

PROFILEID	LOCATION	ROLE	COLLID	APPLNAME
1	COLUMBUS			
2		DBA_ROLE		
3			COLLECTION1	
4				NEWAPPL

DSN\_PROFILE\_ATTRIBUTES

PROFILEID	KEYWORD	ATTRIBUTE1	ATTRIBUTE2
3	SPECIAL_REGISTER	SET CURRENT SQLID = 'DOE'	Null
4	SPECIAL_REGISTER	SET CURRENT APPLICATION COMPATIBILITY = 'V12R1M505'	Null

Attribute 2 = null: remote applications only, processed when 1<sup>st</sup> package is loaded

# Problem statement

Local applications cannot be changed easily to set special registers or global variables such as those to control timeout or deadlock scenarios.

*Special registers and system built-in global variables can be defined in the Db2 Profile tables but are applicable to remote applications only.*

# As is scenario

- The application developer edits local applications to add the SQL statement SET CURRENT LOCK TIMEOUT to a suitable value
- The application developer edits local applications to add the SQL statement SET DEADLOCK\_RESOLUTION\_PRIORITY to a suitable value

## Pain point:

- wait time for application changed, tested and promoted to production system

# Goal

*The DBA can easily set a lock timeout interval to a value that suits a specific local application process, **without the cost of changing application.***

# Db2 13 scenario

- The DBA inserts a row in the Profile table with the SET CURRENT LOCK TIMEOUT attribute and a suitable value for the local or application process
- The DBA inserts a row in the Profile table with the SET DEADLOCK\_RESOLUTION\_PRIORITY attribute and a suitable value for the local application process
- The DBA issues the –STA PROFILE command



# Profile table example

DSN\_PROFILE\_TABLE

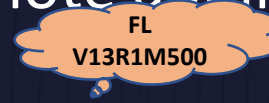
PROFILEID	COLLID	PROFILE_TIMESTAMP
1	COLLECTION1	2021-02-28-14.55.56.780277

DSN\_PROFILE\_ATTRIBUTES

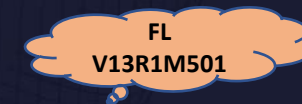
PROFILEID	KEYWORD	ATTRIBUTE1	ATTRIBUTE2
1	SPECIAL_REGISTER	SET CURRENT LOCK TIMEOUT = 10	Null
1	SPECIAL_REGISTER	SET CURRENT LOCK TIMEOUT = 15	1
1	GLOBAL_VARIABLE	SET SYSIBMADM.DEADLOCK_RESOLUTION_PRIORITY = 5	2

# The solution

- Support keywords SPECIAL\_REGISTER and GLOBAL\_VARIABLE in DSN\_PROFILE\_ATTRIBUTES for local and remote applications:
  - CURRENT LOCK TIMEOUT special register
  - DEADLOCK\_RESOLUTION\_PRIORITY built-in global variable
- Column ATTRIBUTE2 in DSN\_PROFILE\_ATTRIBUTES:
  - **Null**: remote applications only. SET statements are processed when the **first** package is loaded (1<sup>st</sup> SQL in first package is loaded)
  - **1**: local threads only. SET statements are processed when **each** package is loaded (1<sup>st</sup> SQL in each package is executed)
  - **2**: both local and remote applications
- Accounting trace: number of SET CURRENT LOCK TIMEOUT statement executions
  - QXSTTIMEOUTFromProf



FL  
V13R1M500



FL  
V13R1M501

# The solution (cont)

Recommendation: Don't mix SQL statement SET in application and Profile table

Note: special register values are *not* saved/restored at package switching

Example: SET CURRENT LOCK TIMEOUT = 10 for PackageA in Profile table

PackageA:

```
SELECT ... FROM T1  
SELECT ... FROM T2  
(Language) CALL PackageB;
```

PackageB:

```
SET CURRENT LOCK TIMEOUT = 20  
INSERT INTO ..  
return;  
UPDATE T3 ...
```

What's the timeout interval for each SQL statement?

# The solution (cont)

Recommendation: Don't mix SQL statement SET in application and Profile table

Note: special register values are *not* saved/restored at package switching

Example: SET CURRENT LOCK TIMEOUT = 10 for PackageA in Profile table

PackageA:

```
SELECT ... FROM T1  
SELECT ... FROM T2  
(Language) CALL PackageB;
```

PackageB:

```
SET CURRENT LOCK TIMEOUT = 20  
INSERT INTO ..  
return;  
UPDATE T3 ...
```

locks timeout after 10 seconds  
locks timeout after 10 seconds

locks timeout after 20 seconds

locks timeout after 20 seconds



# DDL BREAK-IN ENHANCEMENT

# Problem statement

DBA cannot easily route threads using `RELEASE(DEALLOCATE)` packages to `RELEASE(COMMIT)` behavior when issuing DDL statements, resulting in DDL timeout.

- Packages are bound with `RELEASE(DEALLOCATE)` for better performance with the drawback that they may prevent successful DDL.
- DDL and **static SQL** in packages are serialized using the package lock
  - Package A has **`SELECT * FROM ORDERS`** -> share lock on package A
  - `ALTER TABLE ORDERS ADD COLUMN SHIPDATE DATE`** -> exclusive lock on package A
- Subsystem parameter `PKGREL_COMMIT`: requires synchronized `COMMIT` among threads
- `-MODIFY DDF,PKGREL(COMMIT)`: applicable to all remote applications

# Db2 12 scenario

- The DBA binds package DEALLOCATE.A with RELEASE(DEALLOCATE) option
- Threads execute DEALLOCATE.A
- The DBA issues a DDL statement
- All threads running DEALLOCATE.A are quiesced
- DDL statement may complete or timeout/deadlock

- Pain point:
  - failed DDL prevents promotion of new applications

# Goal

*The DBA can easily route applications to RELEASE(COMMIT) behavior to optimize the likelihood of DDL success*



# Db2 13 scenario

1. The DBA binds package DEALLOCATE.A with `RELEASE(DEALLOCATE)`
2. Thread 1 executes package DEALLOCATE.A
3. The DBA inserts a row in the Profile table with the `RELEASE_PACKAGE` and `COMMIT` attribute for the application process
4. The DBA issues the `-STA PROFILE` command
5. The DBA issues the `SET CURRENT LOCK TIMEOUT, SET SYSIBMADM.DEADLOCK_RESOLUTION_PRIORITY` before the DDL statement
6. New threads load package DEALLOCATE.A and set `RELEASE(COMMIT)` option
7. Existing thread 1 releases package DEALLOCATE.A on next `COMMIT`

# The solution

- New keyword **RELEASE\_PACKAGE** in DSN\_PROFILE\_ATTRIBUTES to demote RELEASE(DEALLOCATE) to RELEASE(COMMIT) behavior
  - ATTRIBUTE1 = **COMMIT**
  - ATTRIBUTE2
    - 1 - local threads only
    - null – remote threads only
    - 2 – **both** local and remote threads
- Filtering criteria in DSN\_PROFILE\_TABLE for local threads:
  - AUTHID, ROLE, or both
  - COLLID, PKGNAME, or both
  - One of CLIENT\_APPLNAME, CLIENT\_USERID, or CLIENT\_WORKSTNNAME

# The solution (cont)

- Filtering criteria in DSN\_PROFILE\_TABLE for remote server threads:
  - LOCATION only
  - PRDID only
  - AUTHID, ROLE, or both
  - COLLID, PKGNAME, or both
  - One of CLIENT\_APPLNAME, CLIENT\_USERID, or CLIENT\_WORKSTNNAME
- Profile applied at
  - Each package is loaded (1<sup>st</sup> SQL statement execution)
  - COMMIT, ROLLBACK

# Finding package dependency

```
SELECT BNAME, BQUALIFIER, BTYPE, DCOLLID, DNAME FROM SYIBM.SYSPACKDEP  
WHERE BNAME = 'xxxx' AND BQUALIFIER = 'yyyy' AND BTYPE = 'z'
```

SYSPACKDEP

BNAME	BQUALIFIER	BTYPE	DCOLLID	DNAME
EMPLOYEE	REGION1	T	COLLID1	PKGA
EMPLOYEE	REGION1	T	COLLID2	PKGB
EMP_IDX1	REGION1	I	COLLID1	PKGA
EMP_IDX2	REGION2	I	COLLID3	PKGC
EMP_IDX2	REGION2	I	COLLID2	PKGB

# Finding package dependency

```
SELECT BNAME, BQUALIFIER, BTYPE, DCOLLID, DNAME FROM SYIBM.SYSPACKDEP  
WHERE BNAME = 'EMPLOYEE' AND BQUALIFIER = 'REGION1' AND BTYPE = 'T'
```

SYSPACKDEP

BNAME	BQUALIFIER	BTYPE	DCOLLID	DNAME
EMPLOYEE	REGION1	T	COLLID1	PKGA
EMPLOYEE	REGION1	T	COLLID2	PKGB
EMP_IDX1	REGION1	I	COLLID1	PKGA
EMP_IDX2	REGION2	I	COLLID3	PKGC
EMP_IDX2	REGION2	I	COLLID2	PKGB

# Profile tables

DSN\_PROFILE\_TABLE

PROFILEID	COLLID	PKGNAME	PROFILE_TIMESTAMP
99	COLL1	PKGA	2021-02-28-14.55.56.780277
99	COLL2	PKGB	2021-02-28-14.55.57.780277

DSN\_PROFILE\_ATTRIBUTES

PROFILEID	KEYWORD	ATTRIBUTE1	ATTRIBUTE2
99	RELEASE_PACKAGE	COMMIT	2

# Notes

- RELEASE(COMMIT) may keep package lock at end of unit of work
  - WITH HOLD cursors
  - KEEP\_DYNAMIC(YES) bind option
- -STOP PROFILE command or disable/delete rows in DSN\_PROFILE\_ATTRIBUTES table and restart Profile
- Selectively disable High-Performance DBATs:
  - connection will be made inactive
  - thread will be pooled
- IFCID 177 written at package loading



**THANK YOU!**

Tammie Dang

IBM

[tammied@us.ibm.com](mailto:tammied@us.ibm.com)