Use Profiles to Monitor and Control Db2 Application Context

Maryela Weihrauch

IBM

Acknowledgements and Disclaimers

- **Availability**. References in this presentation to IBM products, programs, or services do not imply that they will be available in all countries in which IBM operates.
- The workshops, sessions and materials have been prepared by IBM or the session speakers and reflect their own views. They are provided for informational purposes only, and are neither intended to, nor shall have the effect of being, legal or other guidance or advice to any participant. While efforts were made to verify the completeness and accuracy of the information contained in this presentation, it is provided AS-IS without warranty of any kind, express or implied. IBM shall not be responsible for any damages arising out of the use of, or otherwise related to, this presentation or any other materials. Nothing contained in this presentation is intended to, nor shall have the effect of, creating any warranties or representations from IBM or its suppliers or licensors, or altering the terms and conditions of the applicable license agreement governing the use of IBM software.
- All customer examples described are presented as illustrations of how those customers have used IBM products and the results they may have achieved. Actual environmental costs and performance characteristics may vary by customer. Nothing contained in these materials is intended to, nor shall have the effect of, stating or implying that any activities undertaken by you will result in any specific sales, revenue growth or other results.

-
  - ***U.S. Government Users Restricted Rights - Use, duplication or disclosure restricted by GSA ADP Schedule Contract with IBM Corp.***
  - IBM, the IBM logo, ibm.com, Db2, and Optim are trademarks or registered trademarks of International Business Machines Corporation in the United States, other countries, or both. If these and other IBM trademarked terms are marked on their first occurrence in this information with a trademark symbol (® or ™), these symbols indicate U.S. registered or common law trademarks owned by IBM at the time this information was published. Such trademarks may also be registered or common law trademarks in other countries. A current list of IBM trademarks is available on the Web at “Copyright and trademark information” at www.ibm.com/legal/copytrade.shtml

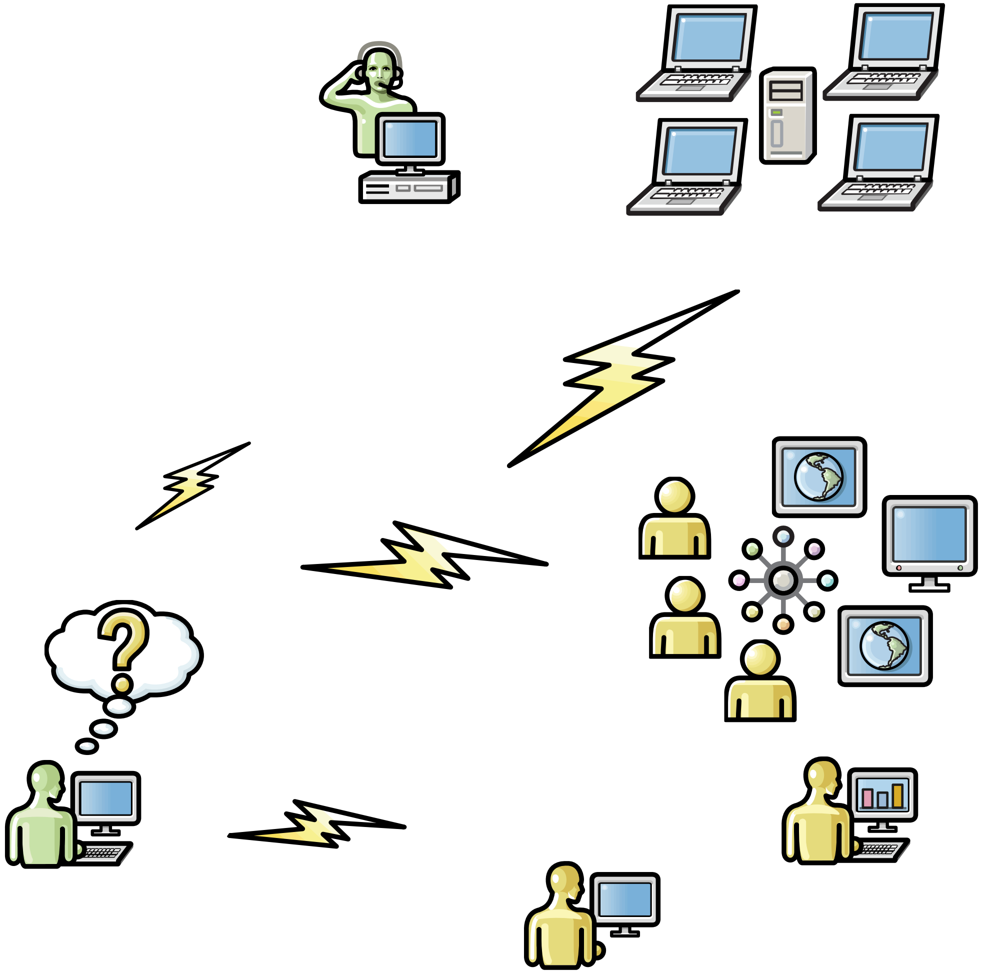
Agenda

- Overview profile monitoring
- Sample definition in profile tables
- Profile table characteristics
- Use cases
- Db2 13 extension to profile table functionality

Db2 for z/OS System Profile Monitoring

- **Resources to manage**
  - Connections
  - Threads
  - Timeouts
  - Special registers, Db2 defined global variables
  - Db2 13 Profile table support for local threads

- **Clients to support**
  - Priorities, demands
  - Variety of applications



Db2 for z/OS: System profile monitoring

- **Use cases:**
  - Prevent workload with denial-of-service attack characteristics
  - 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 production access paths in test
    - Model production values of RIDPOOL, SORTPOOL, BP allocations

- **How to exploit for distributed clients:**
  - Define a distributed application to a profile by qualifier
    - E.g. IP address, or AUTHID and/or ROLE, or driver level, or …
  - Assign values to profiles
    - E.g. threads more granular than single DSNZPARM value for MAXDBAT
    - And/or set special register(s) or global variable(s) for profile(s)

Overview: System Profile Monitoring

- Db2 provided tables
  - SYSIBM.DSN_PROFILE_TABLE
    - Define profile to represent filtering scope or context, a process such as an SQL statement, thread, or connection
  - SYSIBM.DSN_PROFILE_ATTRIBUTES:
    - Define actions for Db2 to take when criteria of the profile is met
    - Specify the
      - Monitoring function type in the KEYWORD column
      - Monitoring action in ATTTRIBUTE1 (warning or exception)
      - Thresholds IN ATTRIBUTE2

- Db2 command START PROFILE after DDF is started
  - START PROFILE issued for each member in a data sharing group
    - START PROFILE status remembered across DDF STOP/START
  - DSN6SYSP PROFILE_AUTOSTART=YES|NO

System profile monitoring: sample tables

SYSIBM.DSN_PROFILE_TABLE

ROLE	AUTHID	LOCATION	PRDID	COLLID	PKGNAME	PROFILEID	PROFILE_ENABLED
	SRVR1					20	Y
			JCC03570			21	Y
		TEST.SVL.IBM.COM				22	Y

SYSIBM.DSN_PROFILE_ATTRIBUTES

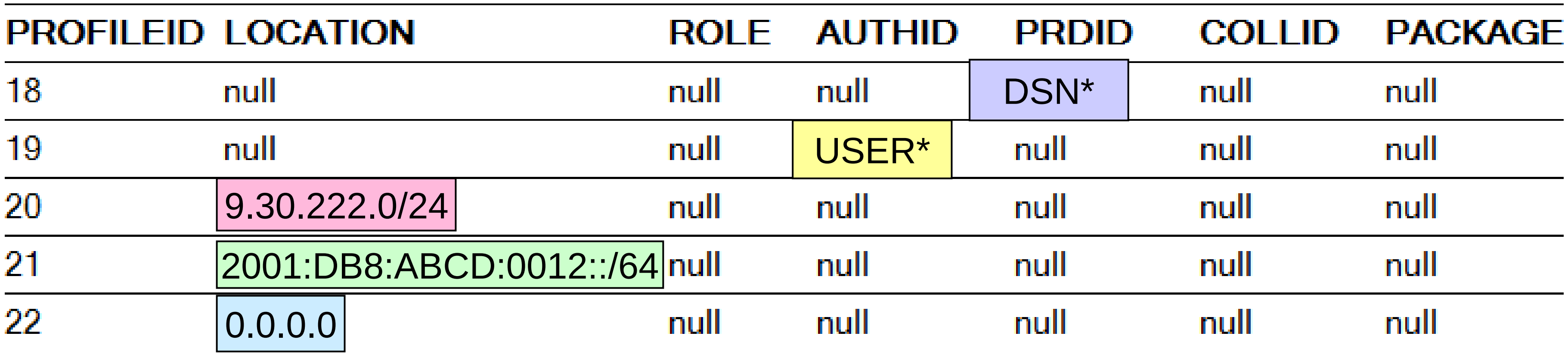
ProfileID	Keywords	Attribute1	Attribute2	Attribute3	Attribute Timestamp
20	MONITOR THREADS	EXCEPTION	10		2011-12-19...
21	MONITOR IDLE THREADS	WARNING	180		2011-12-19…
22	MONITOR CONNECTIONS	WARNING	45		2011-12-19…
22	MONITOR CONNECTIONS	EXCEPTION	90		2011-12-21…

- Values in DSN_PROFILE_TABLE cannot be specified in columns of more than one color (excluding black)
- Multiple rows in DSN_PROFILE_TABLE may apply to one user or application instance:
  for example, one profile based on the authid, another profile based on the product id, etc.
- Multiple rows in DSN_PROFILE_ATTRIBUTES can apply to one profile ID

System profile monitoring: wildcard support

- 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
    - TEMP* in AUTHID would match any authid beginning ‘TEMP’, including TEMP 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

Wildcard support, IP address details

- IPv4 subnet address: initial IPv4 address of a subnet followed by subnet prefix:
  - IPv4address/mm where mm is 8, 16 or 24, representing the number of initial bits of an IPv4 address belonging to the subnet
    - /24 means the first 24 bits of the address. In IPv4, the address is in 4 8-bit nodes. So /24 covers the first three nodes as the 'constant' address value. It is also possible to use /8 or /16.
    - Example: 9.30.222.0/24
    - A profile row with this value in LOCATION would match for any IP address in the range from 9.30.222.1 to 9.30.222.254

- IPv6 subnet address: initial IPv6 address of a subnet followed by subnet prefix:
  - IPv6address/mmm where mmm is 16, 32, 48, 64, 80, 96, or 112, representing the number of initial bits of an IPv6 address which belongs in the subnet
    - In IPv6, the value after the slash is also bits, but in IPv6, each node of the address is 16 bits. So in this example first 4 nodes times 16 bits = 64 bits of the 'constant' address.
    - Example: 2001:DB8:ABCD:0012::/64
    - A profile row with this value in LOCATION would match for any IP address in the range from
      - 2001:DB8:ABCD:0012:0000:0000:0000:0000 to
      - 2001:DB8:ABCD:0012:FFFF:FFFF:FFFF:FFFF

- 0.0.0.0, or ::0, which represents any IP address (IPv4 or IPv6, respectively)

System profile monitoring: ROLLBACK attribute

- MONITOR IDLE THREADS behavior with Attribute1 column value EXCEPTION
  - On exception, thread aborted and connection ***terminated***
  - Client may perceive that Db2 is 'unavailable', especially for those clients that have to re-establish the connection resource
  - Exceeding ZPARM IDTHTOIN has same effect

- ROLLBACK option
  - Thread is still aborted
  - Connection remains ***inactive***
  - Client will not have to re-establish connection
  - Options for ATTRIBUTE1 column
    - EXCEPTION_ROLLBACK
    - EXCEPTION_ROLLBACK_DIAGLEVEL1
    - EXCEPTION_ROLLBACK_DIAGLEVEL2

- DSNZPARM IDTHTOIN does NOT apply to distributed thread that meets the criteria for MONITOR IDLE THREADS profile.

System profile monitoring: ROLLBACK attribute

- 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

Set global variables

- Applies to
  - Transparent Archiving for distr app without application change
  - IDAA query routing from distr app without application change

- 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’

Set global variables

MOVE_TO_ARCHIVE_DEFAULT added to DSNZPARM to set default for SYSIBMADM.MOVE_TO_ARCHIVE

- Rules for defining profiles and criteria for connection attributes do not change
- If unsupported global variables specified in ATTRIBUTE1
  - SYSIBM.DSN_PROFILE_ATTRIBUTES_HISTORY set to ‘REJECTED – UNSUPPORTED SET STATEMENT’
- If supported SET statements contain invalid values
  - SYSIBM.DSN_PROFILE_ATTRIBUTES_HISTORY set to ‘REJECTED – INVALID GLOBAL VARIABLE VALUE’
- Precedence for Global Variables
  - 1. Global variable(s) set explicitly by application SET statement(s)
  - 2. Global variable(s) set in system profile tables
  - 3. Global variable defaults
- In remote applications that match the profile, variable values are set before SQL statements are run
- Authorization
  - Authorization model for setting global variables within application does not apply to global variables set through profile tables
  - User authorized to insert rows into DSN_PROFILE_TABLE and DSN_PROFILE_ATTRIBUTES is considered authorized to set those global variables

System profile monitoring: message granularity

- Requirement: identify applications that exceed system profile thresholds
  - Knowing that a profile was exceeded (WARNING) or taking action when a profile was exceeded (EXCEPTION) is not always sufficient
  - Application information would be helpful to resolve problems or enforce behavior
  - Solution: provide LUWID for threads that exceed profile thresholds

- 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

Prevent denial of service attack characteristics

- High TCP connection count, due to old idle Db2 connections that were abandoned by clients, requests for new connections/work is exhausting private storage
  - *Db2 connections are inactive and do NOT hold Db2 resources (DBAT), so Db2 zparm idle thread timeout does not apply*
  - *Idle connections remain for approx. 30 min before being closed by the client*
  - Old idle connections are essentially blocking new work and there is no way to manage this on the server side -> problem needs to be limited to appServer
- Add a Db2 monitor connection profile threshold per appServer
  - Set to ½ of connections for warning
  - It can limit the scope of a connection storm and ensure connections are not consumed by a few misbehaving appServers

ProfileID	Keywords	Attribute1	Attribute2	Attribute3	Attribute Timestamp
20	MONITOR THREADS	EXCEPTION	10		2011-12-19...
22	MONITOR CONNECTIONS	WARNING	90		2011-12-19…
22	MONITOR CONNECTIONS	EXCEPTION	180		2011-12-19…

Sample: Manage Migration to New Driver Level

- Identify driver level by activating distributed location statistics stats class(7) which is ifcid(365)
- Still difficult to convince clients to upgrade driver level
  - Define profile table entry for PRDID of driver level that needs to be migrated and ATTRIBUTE1 column to WARNING for period of time.
  - Update ATTRIBUTE1 column to EXCEPTION to prevent old driver level to connect to DB2 when ready to enforce

ROLE	AUTHID	LOCATION	PRDID	COLLID	PKGNAME	PROFILEID	PROFILE_ENABLED
			JCC03570			21	Y
		TEST.SVL.IBM.COM				22	Y

ProfileID	Keywords	Attribute1	Attribute2	Attribute3	Attribute Timestamp
21	MONITOR THREADS	WARNING_DIAGLEVEL2	1		2011-12-19…
22	MONITOR CONNECTIONS	WARNING	45		2011-12-19…

Db2 13 extension to profile table functionality
Profile table support for local threads

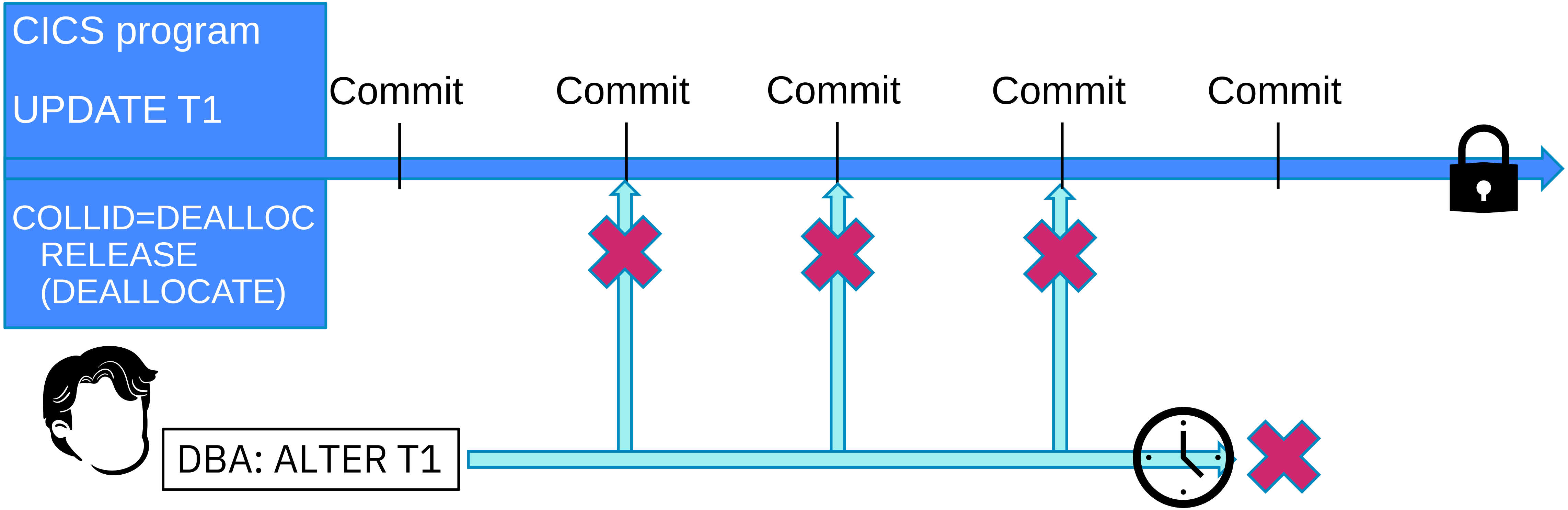
- Current behavior
  - Profile tables can be used by DBA to set special registers for distributed threads only
  - Local applications cannot easily change:
    - Special registers
    - Global variables
  - Application developer required to make changes to local applications

- New behavior

FL 501

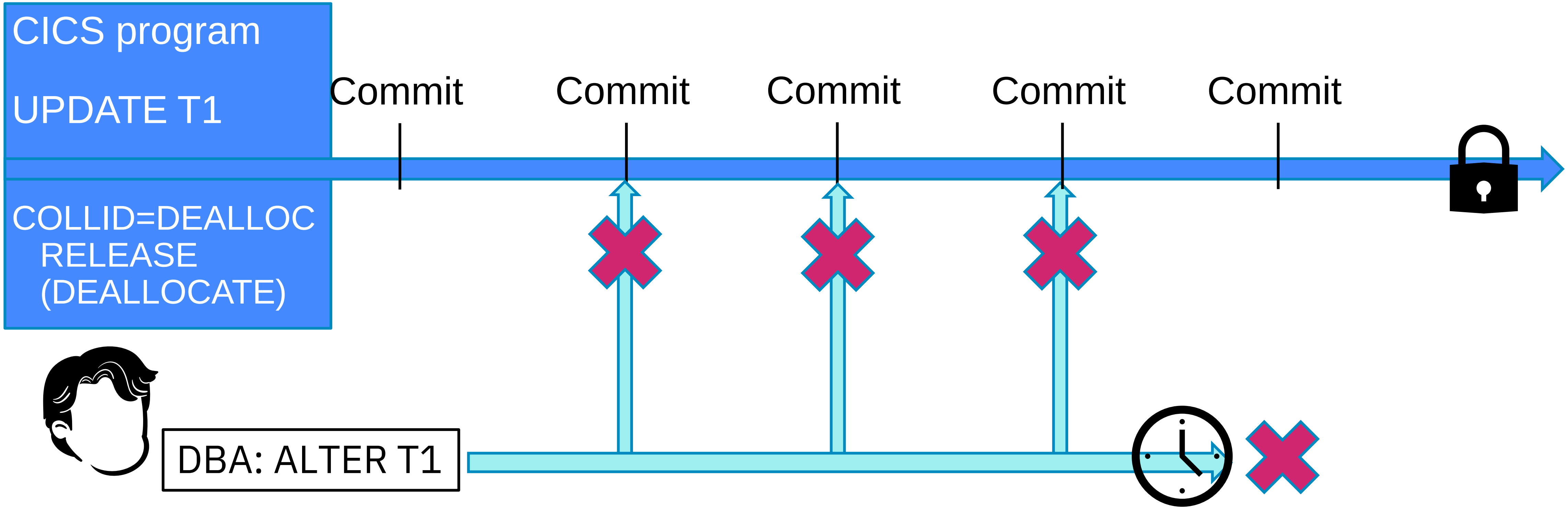
  - Profile tables enhanced
    - Local thread support in some situations
    - New special register:
      - CURRENT LOCK TIMEOUT
    - New built-in global variable:
      - SYSIBMADM.DEADLOCK_RESOLUTION_PRIORITY
    - New keyword: RELEASE_PACKAGE

Challenge scenario:
RELEASE(DEALLOCATE) and ALTER TABLE



DBA needs to add a new column to the table T1, However packages are bound with the bind option RELEASE(DEALLOCATE) prevents DDL execution

Solution scenario (1|3)



DBA: INSERT INTO SYSIBM.DSN_PROFILE TABLE ... ;
INSERT INTO SYSIBM.DSN_PROFILE_ATTRIBUTES ... ;

Solution scenario (2|3)

- INSERT INTO SYSIBM.DSN_PROFILE_TABLE…

PROFILEID	COLLID	PROFILE_ENABLE
99	DEALLOC	Y

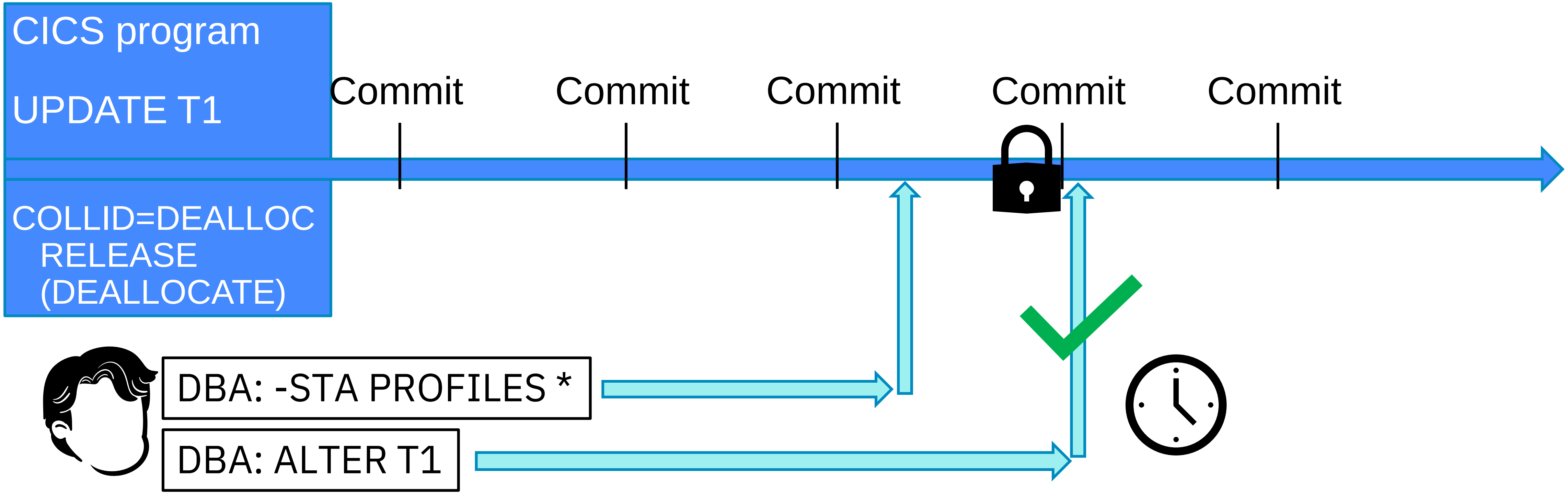
- INSERT INTO SYSIBM.DSN_PROFILE_ATTRIBUTES…

PROFILEID	KEYWORD	ATTRIBUTE1	ATTRIBUTE2
99	RELEASE_PACKAGE	COMMIT	2

ATTRIBUTE1 COMMIT overrides the RELEASE setting for the collection

ATTRIBUTE2 determines scope: NULL – distributed only; 1 – local only; 2 – local and distributed

Solution scenario (3|3)



```
SET CURRENT LOCK TIMEOUT = 60 -- assuming IRLMRWT is 30
SET SYSIBMADM.DEADLOCK_RESOLUTION_PRIORITY = 255
ALTER TABLE T1 ADD C1 INT
```

* as described on prior page

Profile table support for local threads – other details

- The scope of each SET assignment statement (remote, local, or both) is specified through the ATTRIBUTE2 column of the DSN_PROFILE_ATTRIBUTES table.
- **For remote threads:** Profiles are evaluated and SET statements are processed only when the first package is loaded, and the first non-SET statement within that package is executed.
- **For local threads:** Profiles are evaluated and SET statements are processed when each package is loaded.

Summary

- Understand profile monitoring as important capability to influence workload behavior
- Review use cases applicable in your environment
- Db2 13 extended profile table functionality for local threads