PDUG

Krakow 9-10 June 2022

Myth Busters: Db2 latches vs. page latches vs. IRLM locks vs. IRLM latches

Bart Steegmans

IBM

Notices and Disclaimers

- © 2019 International Business Machines Corporation.
- No part of this document may be reproduced or transmitted in any form without written permission from IBM.
- U.S. Government Users Restricted Rights — use, duplication or disclosure restricted by GSA ADP Schedule Contract with IBM.
- Information in these presentations (including information relating to products that have not yet been announced by IBM) has been reviewed for accuracy as of the date of initial publication and could include unintentional technical or typographical errors. IBM shall have no responsibility to update this information. This document is distributed “as is” without any warranty, either express or implied. In no event, shall IBM be liable for any damage arising from the use of this information, including but not limited to, loss of data, business interruption, loss of profit or loss of opportunity.
- IBM products and services are warranted per the terms and conditions of the agreements under which they are provided.
- IBM products are manufactured from new parts or new and used parts. In some cases, a product may not be new and may have been previously installed. Regardless, our warranty terms apply.”
- Any statements regarding IBM's future direction, intent or product plans are subject to change or withdrawal without notice.
- Performance data contained herein was generally obtained in a controlled, isolated environments. Customer examples are presented as illustrations of how those
- Customers have used IBM products and the results they may have achieved. Actual performance, cost, savings or other results in other operating environments may vary.
- References in this document to IBM products, programs, or services does not imply that IBM intends to make such products, programs or services available in all countries in which IBM operates or does business.
- Workshops, sessions and associated materials may have been prepared by independent session speakers, and do not necessarily reflect the views of IBM. All materials and discussions are provided for informational purposes only, and are neither intended to, nor shall constitute legal or other guidance or advice to any individual participant or their specific situation.
- It is the customer’s responsibility to insure its own compliance with legal requirements and to obtain advice of competent legal counsel as to the identification and interpretation of any relevant laws and regulatory requirements that may affect the customer’s business and any actions the customer may need to take to comply with such laws. IBM does not provide legal advice or represent or warrant that its services or products will ensure that the customer follows any law.

Agenda

- Terminology Soup
- Db2 serialization mechanisms
- Locks
- Latches
  - IRLM latches
  - Db2 latches
    - Internal Db2 latches
    - Page latches
- Myth busting – or perhaps better misconception busting as we go along

“Resource Serialization / Synchronization”

PDUG

- Important in any system, but even more so when multiple processes run in parallel. Basically there are two flavors:
- Resource **serialization** services that programs can use to serialize access to resources, eg.
  - Using data that is referenced by more than one program (eg. rows in a Db2 table accessed by programs executing concurrently on the same or other system (sysplex)
  - Using data in subpools that are shared between programs
- **Synchronize**/serialize execution of programs that depend on the completion of events in other programs
  - Synchronize the execution of programs with minimal overhead (wait/post)

MYTH – MISCONCECPTION

Locks and latches are only used by Db2 (and IRLM)

Or Not ?

Terminology Soup

- Same names as locks and latches are used
  - By different components
    - z/OS locks and IRLM/Db2 locks
    - z/OS latches and IRLM/Db2 latches
  - With a single component
    - Db2 (internal) latches and Db2 page latches
    - Db2 latches and IRLM latches
- They often interact with each other, making it hard to understand which part/component is responsible for which resource
  - Especially important to understand this when some sort of contention issue occurs
- When you talk to other teams be aware that what they call locks and latches could be different from the Db2 / IRLM implementation

z/OS Serialization (1|3)

- Enqueues: ISGENQ, ENQ/DEQ/RESERVE (GRS)
  - ISGENQ: Obtain, change, and release user-defined logical resources (ENQs) and direct access storage device (DASD) Reserves
    - Serialize resources (such as data sets) *within a single address space, single z/OS system or across multiple z/OS systems*
    - IBM recommends using the ISGENQ service over ENQ/DEQ/RESERVE
  - ENQ/DEQ: Obtain, change, and release user-defined logical resources (ENQs)
    - Serialize resources (such as data sets) within a single address space, single z/OS system or across multiple z/OS systems.
  - RESERVE: Allow multiple systems (need not be all z/OS systems) to share direct access storage device (DASD)

z/OS Serialization (2|3)

- Latch Manager Callable Services (GRS)
  - Serialize resources (such as control blocks or data sets) *within a single address space or across several address spaces in a single system* with minimal overhead.
  - Authorized programs only
    (ISV Software also use these services – not used by Db2)
- Locking (SETLOCK macro)
  - Serialize system resources (such as MVS™ system queues or control blocks) within *a single system or address space*. (eg. CPU, CML, local lock)
- Locking (IXLLOCK - XES)
  - Enables authorized applications to obtain S/X serialization on user-defined logical resources.
  - Allows to implement your own locking protocols via the inclusion of user data.
  - Specific to Sysplex environments - *provides high speed cross sysplex sharing*

z/OS Serialization (3|3)

- z/OS locking , latching, ENQ waits often show up in Db2 as “NOT ACCOUNT” time. That is time that we are in Db2, but
  - Not using CPU
  - Nor suspended/waiting for something that Db2 understands (like waiting for an I/O)
  - [This typically does not apply to IXLLOCK requests]
- z/OS documentation:
  - MVS Programming: Authorized Assembler Services Guide – Ch 4. Serialization
    - https://www.ibm.com/support/knowledgecenter/en/SSLTBW_2.4.0/com.ibm.zos.v2r4.ieaa800/serial.htm
  - z/OS Diagnosis: Reference - Ch 6. Serialization summary
    - https://www.ibm.com/support/knowledgecenter/SSLTBW_2.4.0/com.ibm.zos.v2r4.ieav200/sersum.htm
  - z/OS Planning: Global Resource Serialization
    - https://www.ibm.com/support/knowledgecenter/en/SSLTBW_2.4.0/com.ibm.zos.v2r4.ieag400/abstract.htm
  - MVS Programming: Sysplex Services Guide - Ch 10. Using Lock Services (IXLLOCK)
    - https://www.ibm.com/support/knowledgecenter/en/SSLTBW_2.4.0/com.ibm.zos.v2r4.ieai600/lsusing.htm

MYTH – MISCONCECPTION

Locks and latches are only used by Db2 (and IRLM)

Or Not ?

Agenda

- Db2 serialization terminology
- IRLM locks
  - IRLM lock characteristics
  - What affects locking behavior of an application ?
- Latches in general
- IRLM latches
  - IRLM latch characteristics
  - What can trigger IRLM latch suspensions
  - Monitoring IRLM latch suspensions
- Db2 latches in general
- Db2 (Internal) latches
  - Db2 internal latch characteristics
  - Popular latches classes
  - Monitoring Db2 internal latches
- Db2 page latches
  - Db2 page latch characteristics
  - Monitoring Db2 page latches
  - Interaction between page latches and global contention
- The importance of having enough CPU

Db2 Serialization Terminology

Type	Requested by	Managed by
IRLM locks	Db2	IRLM
IRLM latch	IRLM	IRLM
Db2 (internal) latch	Db2	Db2
Db2 page latch	Db2	Db2
[Claims and drains]	Db2	Db2 and IRLM
[Object restrictive states – DBET]	Db2	Db2
[Utility compatibility matrix]	Db2	Db2
[ ] = Not discussed in the presentation		

Agenda

- Db2 serialization terminology
- IRLM locks
  - IRLM lock characteristics
  - What affects locking behavior of an application ?
- Latches in general
- IRLM latches
  - IRLM latch characteristics
  - What can trigger IRLM latch suspensions
  - Monitoring IRLM latch suspensions
- Db2 latches in general
- Db2 (Internal) latches
  - Db2 internal latch characteristics
  - Popular latches classes
  - Monitoring Db2 internal latches
- Db2 page latches
  - Db2 page latch characteristics
  - Monitoring Db2 page latches
  - Interaction between page latches and global contention
- The importance of having enough CPU

IRLM Locks [1|2]

- Complex but offers a lot of flexibility
- An IRLM lock consists of
  - Resource name (what are we locking)
    - The resource - determines the lock size
      - TS/partition, table, page, row
  - Lock state (how restrictive the lock is)
    - Possible lock states depend on the resource
      - TS/partition: IS,IX,S,U,SIX,X
      - Page/row/LOB/XML: S,U,X
  - Duration (how long the lock is held)
    - Manual, manual+1, commit , commit+1, allocation, interest, plan, utility
  - Scope
    - Global/local

IRLM Locks [2|2]

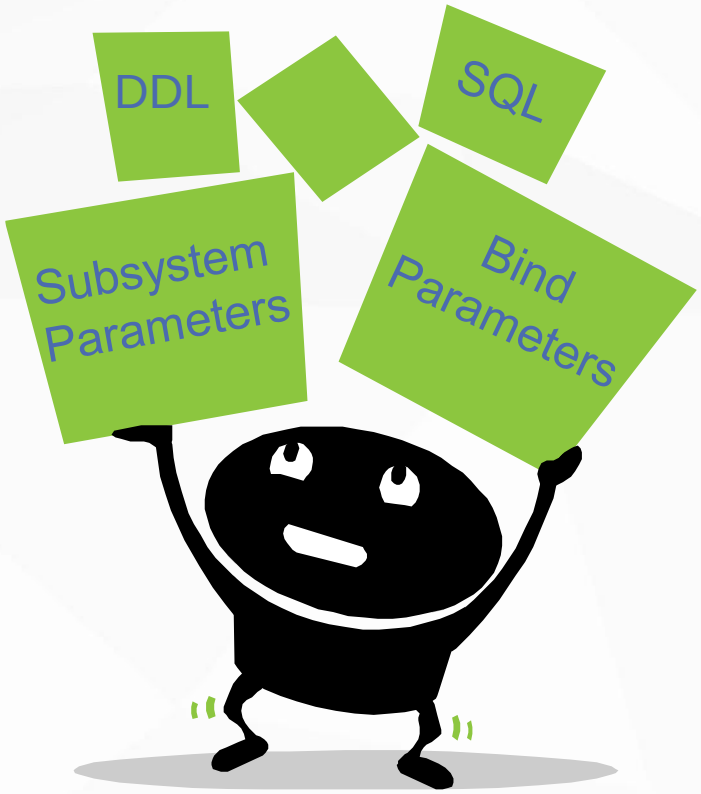
- L-locks and P-locks
  - L-locks – Logical or applications locks
    - Owned by applications to manage concurrency – exist in data sharing and non- data sharing systems
  - P-locks – Physical locks
    - Owned by a member to manage data coherency – only exist in data sharing systems
- Deadlocks/timeouts
  - Every deadlock detection cycle, IRLM will look at lock suspensions to see
    - if a deadlock exists, and will present the info to Db2
    - if tran have been waiting for a lock longer the timeout value, and inform Db2
  - Db2 will decide who the deadlock victim will be, and whether a timeout occurs
- Lock escalation
  - When LOCKMAX is exceeded, Db2 will ‘upgrade’ page/row locks to a higher ‘scope’ lock, eg. All page locks will be released and replaced by one partition lock for UTS

What affects locking behavior of an application ? [1|2]

- Application design / logic has a great impact on which locks are required, when locks are acquired, and how long they are held
  - Type of SQL
  - Sequence of accessing objects
  - Commit frequency
  - …
- IRLM locks have a close interaction with
  - Program isolation level (ISO) – UR, CS, RS, RR
  - CURRENTDATA Bind option
- Many options, parameters to control locking behavior (next foil)
  - [This is much less the case for latches
    - Latches have very few external tuning knobs
    - BUT you can still significantly influence their usage indirectly]

What affects locking behavior of an application ? [2|2]

- Some parameters to control locking behavior
  - DDL options
    - LOCKSIZE
    - LOCKMAX
  - BIND options
    - ISOLATION level
    - CURRENTDATA
    - RELEASE
    - CONCURRENTACCESSRESOLUTION (starting in V10)
  - SQL statement options
    - WITH clause
    - SKIP LOCKED DATA (starting in V9)
    - Lock avoidance
    - …
  - System parameters (ZPARMs)
    - RRULOCK
    - …



- *Lots of presentations about IRLM locking so we will focus on the not so well know aspects of locking and mostly on serialization things besides locking*

Presentations to learn about locking so I don’t have to explain

- Db2 Manuals are a great place to start:
  - Managing Performance and Data Sharing Planning and Administration manuals, or
  - https://www.ibm.com/docs/en/db2-for-zos/12
- Redbook:
  - DB2 9 for z/OS: Resource Serialization and Concurrency Control, SG24-4725-01
- Numerous Previous IDUG Sessions:
  - Analyzing DB2 for z/OS Resource Serialization & Concurrency Problems, Bart Steegmans, IBM, EMEA 2013
  - A Beginners Guide to Locks, Latches, Claims and Drains in DB2 for z/OS, Steve Thomas, CA Technologies (now Broadcom), EMEA 2016
- Blog articles:
  - Series about Db2 for z/OS locking for application programmers, Gareth Jones, Triton, https://www.triton.co.uk/category/tech-blog/?tag=db2-locking

Agenda

- Db2 serialization terminology
- IRLM locks
  - IRLM lock characteristics
  - What affects locking behavior of an application ?
- Latches in general
- IRLM latches
  - IRLM latch characteristics
  - What can trigger IRLM latch suspensions
  - Monitoring IRLM latch suspensions
- Db2 latches in general
- Db2 (Internal) latches
  - Db2 internal latch characteristics
  - Popular latches classes
  - Monitoring Db2 internal latches
- Db2 page latches
  - Db2 page latch characteristics
  - Monitoring Db2 page latches
  - Interaction between page latches and global contention
- The importance of having enough CPU

Latches in general

- ‘Lightweight’ serialization protocol
- Always within a single (sub)system (Db2 or IRLM) – member
  - If you need to serialize across a data sharing group, a (global) lock will be used
- X and S state only
- Typically (very) short duration
  - Acquire the latch, do whatever needs to be done, and release it
  - Eg. Get latch for a Db2 storage pool, acquire the storage you need, release the Db2 storage pool latch
- If the latch is not available, the request for the latch is QUEUED
  => Latch Contention
- No direct application control knobs, BUT application design can have a significant (indirect) effect on the amount of latch activity
  - E.g. TS scan will acquire many more page latches compared to accessing a page via IX access, hereby increasing the chance of hitting contention
- Can become a serious performance bottleneck
  - Continuous development effort to reduce the chances that this happens

Agenda

- Db2 serialization terminology
- IRLM locks
  - IRLM lock characteristics
  - What affects locking behavior of an application ?
- Latches in general
- IRLM latches
  - IRLM latch characteristics
  - What can trigger IRLM latch suspensions
  - Monitoring IRLM latch suspensions
- Db2 latches in general
- Db2 (Internal) latches
  - Db2 internal latch characteristics
  - Popular latches classes
  - Monitoring Db2 internal latches
- Db2 page latches
  - Db2 page latch characteristics
  - Monitoring Db2 page latches
  - Interaction between page latches and global contention
- The importance of having enough CPU

IRLM Latches

- ‘Lightweight’ serialization protocol
- Different flavors of IRLM latches are used to serialize different things in IRLM, typically access to internal control blocks
- IRLM has:
  - One main latch
  - Many secondary latches
- IRLM latches ‘live’ in the IRLM address space
  - Requested and managed by IRLM

IRLM Latches

- IRLM main latch
  - When held, only the holder can process things
    - No other requests can run in IRLM
    - They are all suspended (IRLM latch suspension)
  - Used by deadlock detection
  - Used by IRLM QUERY requests (not fastquery used by break-in)
- IRLM secondary latches
- IRLM only traces latch suspensions, not latch requests (latch/unlatch)

What Can Trigger IRLM Latch Suspensions (1|2)

- IRLM is running with a low dispatching priority
  - IRLM should use service class SYSSTC
- IRLM TRACE is on
  - TRACE YES (default is NO) can double IRLM pathlength
  - START irlmproc,TRACE=YES (all (internal) traces are started)
  - F irlmproc,STATUS,TRACE to check
- IRLM Deadlock Cycle frequency too low
  - Especially important if you have a lot of locks held at any one time as IRLM main latch is held when checking for deadlocks
  - DEADLOK parameter – keep to 1 second or higher if IRLM holds a lot of locks

What Can Trigger IRLM Latch Suspensions (2|2)

- Frequent IRLM QUERY requests
  - -DIS DB() LOCKS
  - MODIFY irlmproc,STATUS
  - Not for Fast query requests used by ‘break-in’ support (PKGREL_COMMIT=YES ZPARM)
  - IFI READS requests for IFCID 149 , 150 – often used by online monitors
- Make sure to have good lock avoidance, no lock, no trip to IRLM, so no chance of IRLM latch contention

Monitoring IRLM Latch Suspensions (1|4)

- In IFCID 2 (Db2 stats) - DSNDQTXA section
  - QTXASLAT – # IRLM latch suspensions (locking section)
  - PI93817 added quite a few additional counters (stats only) related to the different flavors of IRLM latch contention
- In IFCID 3 (Db2 accounting)
  - DSNDQTXA section (locking section)
    - QTXASLAT – # IRLM latch suspensions
  - DSNDQWAC (class 3 suspension section)
    - QWACAWTL - Wait time due to *local contention for IRLM locks and IRLM latches*
    - QWACARNL - Number of wait trace events related to waits for *local contention for IRLM locks and latches*
    - Note these counters combine waits for (local) IRLM locks and IRLM latches

Monitoring IRLM Latch Suspensions (2|4)

- In IFCID 239 (package accounting)
  - DSNDQTXA section (class 10 locking section)
    - QTXASLAT – # IRLM latch suspensions
  - DSNDQPAC (class 8 suspension section)
    - QPACAWTL - Wait time due to *local contention for IRLM locks and IRLM latches*
    - QPACARNL - Number of wait trace events related to waits for *local contention for IRLM locks and latches*
    - Note these counters combine waits for (local) IRLM locks and IRLM latches

Monitoring IRLM Latch Suspensions (3|4)

- In IFCID 44 / 45 – lock suspend / resume
  - IFCID 21 traces IRLM lock request – it does not trace IRLM latch requests
  - However, if a lock (unlock, change) request hits IRLM latch contention IFCID 44-45 is produced which give an indication about the reason for the suspension, eg.
    - QW0044W8 'LM' = IRLM MAIN LATCH CONTENTION
    - QW0044W9 'LR' = IRLM RESOURCE LATCH CONTENTION
    - QW0044WA 'LW' = IRLM WORK UNIT CONTENTION
    - See DSNWMSGS for details
  - Obviously if the contention was for an incompatible lock that is also reported and is by far the most common reason for cutting an IFCID 44-45 pair

Monitoring IRLM Latch Suspensions (4|4)

IRLM latch contention should be less than 1-5% of Total IRLM Requests (using the Db2 stats record)

Field Name	*Description*
QTXASLAT	SUSPENSIONS (IRLM LATCH)
QTXALOCK	LOCK REQUESTS
QTXAUNLK	UNLOCK REQUESTS
QTXAQRY	QUERY REQUESTS
QTXACHG	CHANGE REQUESTS

IRLM Latch Contention = SUSPENSIONS (IRLM LATCH) (A)

Total IRLM Requests = LOCK + UNLOCK + QUERY + CHANGE REQUESTS (B)

IRLM Latch Contention Rate (%) = (A)/(B)*100

CPU Times for IRLM Lock/Latch Requests – L/L Suspensions

- In general, the CPU time is charged to
  - The transaction (class 2 CPU time) for *lock/latch request that are not suspended*
  - The transaction (class 2 CPU time) for *unlock requests by SELECT* statement
    - Eg. ISO(CS) - going to the next (qualifying) row/page we unlock the previous one
  - Db2 MSTR SRB (Update commit) for *unlocking locks acquired by I/U/D*
  - IRLM SRB for Db2 *lock and IRLM latch suspension* (resume)

Agenda

- Db2 serialization terminology
- IRLM locks
  - IRLM lock characteristics
  - What affects locking behavior of an application ?
- Latches in general
- IRLM latches
  - IRLM latch characteristics
  - What can trigger IRLM latch suspensions
  - Monitoring IRLM latch suspensions
- Db2 latches in general
- Db2 (Internal) latches
  - Db2 internal latch characteristics
  - Popular latches classes
  - Monitoring Db2 internal latches
- Db2 page latches
  - Db2 page latch characteristics
  - Monitoring Db2 page latches
  - Interaction between page latches and global contention
- The importance of having enough CPU

Db2 Latches

- ‘Lightweight’ serialization protocol
- Two flavors:
  - Db2 (internal) latches
    - Mainly used to serialize access to memory structures (eg. Ctl block chains)
  - Db2 (Buffer Manager) page latches
    - Used to serialize access to a page in the (local) buffer pool
- A latch is always limited to just the Db2 subsystem that uses it
  If (data sharing) group wide serialization is needed, a (global) lock is used, eg.
  - BM Page latch to serialize access page access within a member
  - Page P-lock to serialize access to pages across members (data coherency)

Agenda

- Db2 serialization terminology
- IRLM locks
  - IRLM lock characteristics
  - What affects locking behavior of an application ?
- Latches in general
- IRLM latches
  - IRLM latch characteristics
  - What can trigger IRLM latch suspensions
  - Monitoring IRLM latch suspensions
- Db2 latches in general
- Db2 (Internal) latches
  - Db2 internal latch characteristics
  - Popular latches classes
  - Monitoring Db2 internal latches
- Db2 page latches
  - Db2 page latch characteristics
  - Monitoring Db2 page latches
  - Interaction between page latches and global contention
- The importance of having enough CPU

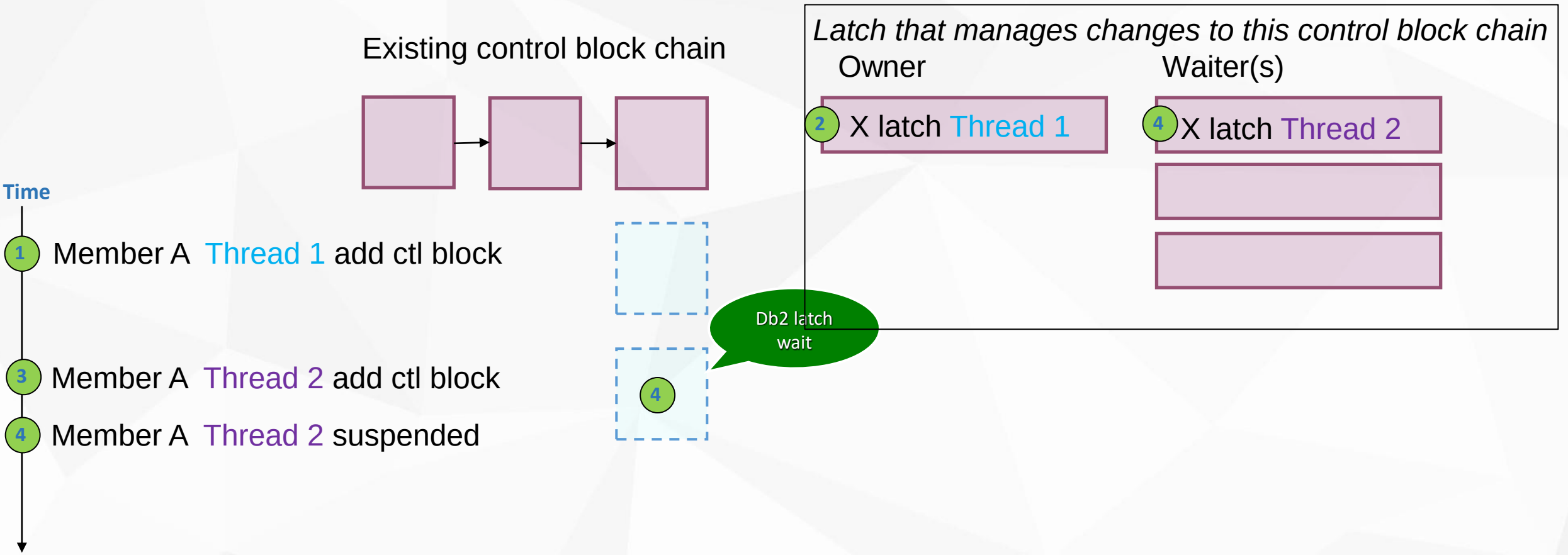
Db2 (Internal) Latches (1|6)

- Used when Db2 needs to serialize DB2 internal resources, not user data
  - Typically used to serialize access to memory structures, eg. control block(chains)
- S/X mode only
- In V12 there are *256 latch types* (not all used) organized in *32 latch classes*
  - Trace record IFCID 51 52 56 57 will show the latch type (and latch address)
  - Trace record IFCID 1 (Db2 stats) only show the latch class
    - [Sometimes you may see that latch types (256) are called latch classes, and latch classes (32) are called latch levels - That is IBM internal terminology - I will try to be consistent and use type (256) and class (32)]
- No deadlock detection mechanism but 'hierarchy' checking is done to try to avoid deadlocks

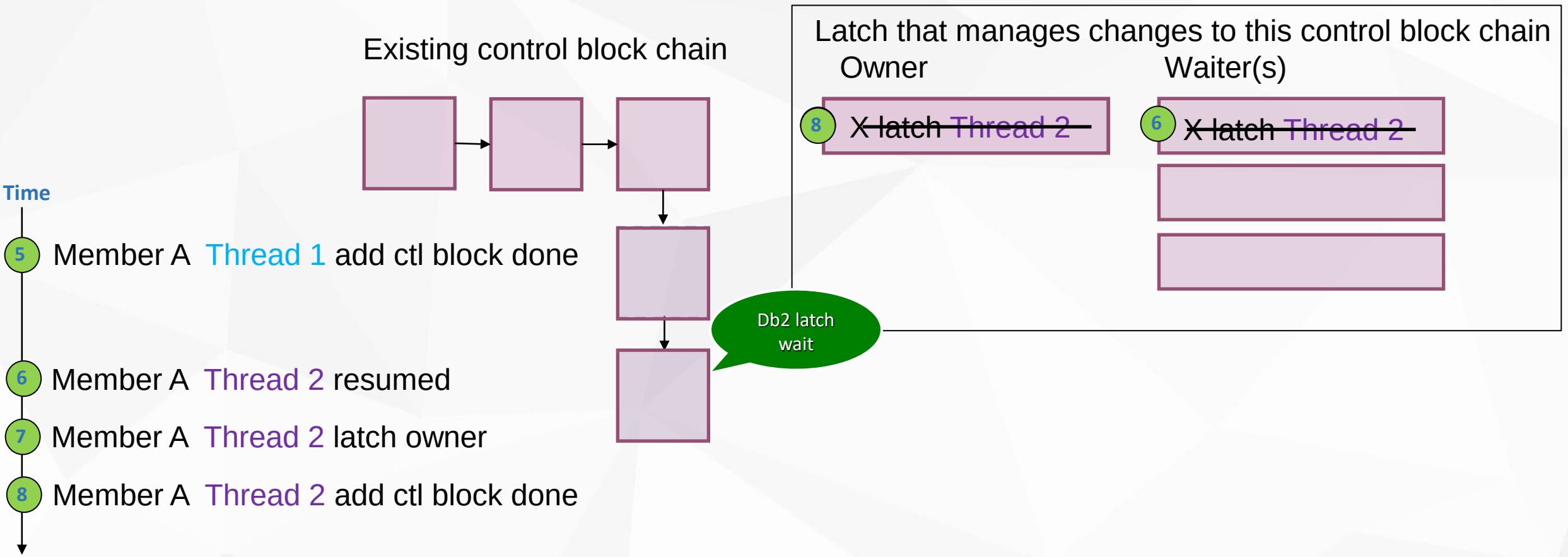
Db2 (Internal) Latches (2|6)

- A latch is nothing more than a storage address that is used to 'protect' whatever it is that needs its access to be serialized
- If the latch is held in an 'incompatible' state (I want the latch in X and there is somebody else with an X already)
  => my process (tran) will be suspended and put on a 'wait queue'
- Once the latch becomes available the waiter(s) will be resumed
  - Prior to V10, all waiters were resumed, only one would get the latch and all others were suspended again. With many waiters and a system with many CPs that can result in many suspend/resume activities for many threads on a busy latch and could use a considerable amount of CPU
  - Since V10 we only resume a 'limited' number of waiters to avoid this

Db2 (Internal) Latches (3|6)



Db2 (Internal) Latches (4|6)



Db2 (Internal) Latches (5|6)

- A Db2 latch is a storage address
- A Db2 latch (address) has a latch type (1-256) associated with it
  - Many different latches (latch addresses) can use the same latch type
    - Latch type 56 is used to see if prefetch is going on against at page set/partition
    - Each page set/partition has its own latch (address),
      but they all use the same latch type(56), and the same latch class(24)
- A latch type is mapped to a latch class
  - Class 1-32 and 254(INDEX)
  - Db2 latch classes are documented in the DSNDQVLS macro in *.SDSNMACS library
  - Several latch types can map to the same latch class, eg.
    - Latch type 24 = EDM POOL LRU Latch is latch class 24 (LC24)
    - Latch type 56 = Prefetch scheduling also maps to latch class 24 (LC24)
    - Context can help to determine which one it is that is driving the contention
      - If not, start traces for IFCID 51 52 56 57

Db2 (Internal) Latches (6|6)

- "Popular" latch classes and their most common usage
  - LC06 = Index tree modification latch (global)
  - LC14 = Buffer pool exclusive latches
  - LC19 = Log buffer latch (pre V12)
  - LC23 = Buffer pool deferred write or page latch timer (pre V12)
  - LC24 = Buffer manager latch or EDM LRU chain latch
  - LC32 = Storage manager latch (changes in V13 to LC63)
  - LC254 = Index tree modification latch (local)

Db2 (Internal) Latches – LC06 (1|2)

- LC06 is most often latch type 70 – Index tree modification latch
  - In combination with index tree P-lock
- Needed during IX structure modification when IX is GBP-dep
- This latch can be held for a longer duration compared to other latches
  - If IX structure modification takes a long time, the latch will be held longer
    - Space search for an empty page to be used during index page split
    - Extending the pageset
    - Forced log write (1 since V11)

Db2 (Internal) Latches – LC06 (2|2)

- Index split time can be significantly reduced by
  - Using faster active log device (zHyperlink write, zHyperwrite for mirrored DASD)
- Options to reduce the number of index splits
  - Index freespace tuning for random insert
  - Use minimum required index key size especially if unique index
  - NOT PADDED index for large varchar columns (V8)
  - Large index page size (V9)
    - May have to increase BP size
    - May result in more page latch/p-lock contention as more IX keys on the page
  - Asymmetric leaf-page split (V9)
- Tracking the number of index splits
  - LEAFNEAR/FAR in SYSINDEXPART and RTS REORGLEAFNEAR/FAR
  - DB2 10 added IFCID 359 to monitor index splits

Db2 (Internal) Latches – LC14

- Buffer manager LRU and hash chain latch
  - When high LC14 contention
    - Spread objects amongst several buffer pools to get an even getpage frequency
    - When objects don't fit in the BP, increase the size of the BP
    - Make sure the BP uses at least 4000 buffers
    - Use PGSTEAL= FIFO or NONE rather the LRU
      - LRU = Least recently used page steal algorithm (default)
      - FIFO = First in first out
        - Cheaper LRU chain management -> CPU reduction and less chance of LC14 contention
      - NONE = in-memory BP – good option when object fits in the BP
        - Data (pre)loaded at first use (physical open) via prefetch
        - V10-V11: no LRU chain mgt
        - V12: contiguous BP – no hash chains , no page replacement (except for the overflow area)

Db2 (Internal) Latches – LC19

- Log latch
  - Lots of work over the last Db2 releases (V9 -> V12) to reduce the chance of (high) LC19 contention
  - If you still see high LC19 contention
    - Increase the size of the log output buffer if ‘unavailable log buffer’ counter > 0
      - Agent that notices it will wait for the log buffers to be written out to disk
      - Subsequent agent will wait for the log latch (LC19)
    - Reduce/minimize the number of log records that need to be created
      - Use LOAD with LOG NO instead of jobs doing massive INSERT, UPDATE, DELETE activity
      - When deleting all rows from a table , make sure mass delete processing can be used (segmented or UTS) instead of deleting and logging each individual row
    - Commit frequently in heavy I/U/D jobs
      - Not only will that improve concurrency, it will also ‘slow down’ the rate at which these jobs produce log records and reduces that chance of hitting latch contention

Db2 (Internal) Latches – LC24

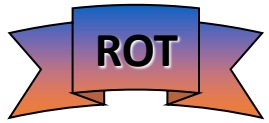
- EDM LRU latch – Latch type 24 (LC24)
  - Also here Db2 code enhancements to reduce hitting latch contention for this latch
    - Removal of EDMBFIT ZPARM (V10)
    - Moving CT, PT from the EDM pool to thread storage (V10)
  - When high
    - Using thread reuse and RELEASE(DEALLOCATE) instead of RELEASE(COMMIT) of frequently executed packages
    - Using high performance DBATs for DDF workload
- Prefetch scheduling – Latch type 56 (LC24)
  - Higher chance of contention with many concurrent prefetch requests (for a P/P)
  - When high
    - Disable prefetch by setting VPSEQT=0 or use PGSTEAL(NONE) buffer pools
    - Use more partitions as the latch is per page set / partition

Monitoring Db2 (Internal) Latches (1|2)

- Db2 only traces latch contentions, not latch obtain, release requests
- Latch (suspension) information can be found in
  - IFCID 1 (Db2 statistics) – LC01-LC32 + LC254 – DSNDQVLS section
    - #latch suspensions for latch types that belong to that latch class
  - IFCID 3 / 239 (Db2 plan / package accounting)
    - QWACAWLH / QPACAWLH - Accumulated wait time due to Db2 latch contention
    - QWACARLH / QPACANLH - Number of times a transaction waited for Db2 latch contention
    - With very high internal latch contention, collecting accounting class 3/8 can be expensive
  - ...

Monitoring Db2 (Internal) Latches (2|2)

- Latch (suspension) information can be found in
  - …
  - Detailed latch trace records IFCID 51-52 (shared latch resume – suspend) and IFCID 56-57 (exclusive latch suspend – resume)
    - Traces the latch type as well as the actual latch address (may require a dump if the latch type is not enough to pinpoint the issue)
    - Typically high volume/overhead trace – activate for a short time to reduce the impact
    - -STA TRA(P) C(30) IFCID(51,52,56,57) TDATA(CPU COR TRA) DEST(xxxx)
    - -STA TRA(P) C(11) TDATA(CPU COR TRA) DEST(xxxx)
      - Perf trace class 11 also contains IFCID 93 94 (suspend/resume) so expect more trace records if you use that
- Try to keep LC rate < 1-10K per second



  Latch contention cannot be avoided – so non-zero numbers are expected

  Most latches are very short duration so 1K/sec can be supported without any perf degradation LC06 – LC254 related to IX split can be held longer so a smaller rate may have a visible effect

Agenda

- Db2 serialization terminology
- IRLM locks
  - IRLM lock characteristics
  - What affects locking behavior of an application ?
- Latches in general
- IRLM latches
  - IRLM latch characteristics
  - What can trigger IRLM latch suspensions
  - Monitoring IRLM latch suspensions
- Db2 latches in general
- Db2 (Internal) latches
  - Db2 internal latch characteristics
  - Popular latches classes
  - Monitoring Db2 internal latches
- Db2 page latches
  - Db2 page latch characteristics
  - Monitoring Db2 page latches
  - Interaction between page latches and global contention
- The importance of having enough CPU

Db2 Page Latches (1|2)

- Used by the Db2 buffer manager component
- Serialize access to a page in the buffer pool
- S/X mode only
  - No latch types or latch classes
- Before you can do anything to a page in the buffer pool, Db2 must acquire a latch on the page
  - S mode latch if you want to read data from a page
  - X mode latch if you intend to change anything on a page
- If the latch you request is already held in an incompatible state, the process is suspended, and must wait for the latch to become available
  - Once it is, the process is automatically resumed

Db2 Page Latches (2|2)

- Before Db2 12, page latch requests could time out (after a long time)
  - To track them they were put on a queue
  - To serialize storing that info a latch (Db2 internal latch) is used
    - That latch maps to LC23
  - As a result, high page latch contention would often result in high LC23 contention
- In Db2 12 the timer queue is only used for requests by a p-lock exit
  - This should result in significant LC23 reduction during period of high page latch contention

Monitoring Db2 Page Latches

- *Db2 only traces page latch contentions, not page latch obtain, page latch release requests*
- No information about page latch contention in Db2 statistics
  - In data sharing the GBP page P-lock section may be able to provide some clues
- IFCID 3 / 239 (Db2 Plan / package accounting)
  - QWACAWTP / QPACAWTP - Wait time due to page latch contention
  - QWACARNH / QPACARNH - Number of times waiting for page latch contention
- IFCID 226 -227 (begin / end page latch suspension)
    - -STA TRA(P) C(30) IFCID(226,227) TDATA(CPU COR TRA) DEST(xxxx)
    - -STA TRA(P) C(4) TDATA(CPU COR TRA) DEST(xxxx)
      - Perf trace class 4 also contains a lot of other IFCIDs so expect more trace records if you use that
    - IFCID 226-227 will allow to determine the object(s) that incur the page latch suspension(s)

MYTH – MISCONCECPTION

Page Latch contention time is the largest contributor to the transaction's elapsed time

Class 2 Elapsed Time ~= CL3 suspension time ~= page latch wait time

This means we have a page latch contention problem

Or Not ?

A Page Latch problem or not ?

AVERAGE	APPL(CL.1)	DB2 (CL.2)
ELAPSED TIME	0.175700	0.175524
CP CPU TIME	0.000455	0.000428
SE CPU TIME	0.000227	0.000217
SUSPEND TIME	0.000081	0.174876
NOT ACCOUNT.	N/A	0.000003
DB2 ENT/EXIT	N/A	2.69
EN/EX-STPROC	N/A	18.59

CLASS 3 SUSPENSIONS	AVERAGE TIME	AV.EVENT	TIME/EVENT
LOCK/LATCH(DB2+IRLM)	0.000018	0.10	0.000177
IRLM LOCK+LATCH	0.000017	0.04	0.000485
DB2 LATCH	0.000000	0.06	0.000005
SYNCHRON. I/O	0.001417	2.21	0.000640
DATABASE I/O	0.001201	2.10	0.000572
READ CACHE HIT	0.001046	1.98	0.000528
LOG WRITE I/O	0.000217	0.11	0.001898
OTHER READ I/O	0.000027	0.01	0.001880
OTHER WRTE I/O	0.000049	0.09	0.000576
SER.TASK SWTCH	0.000000	0.00	N/C
...			
PAGE LATCH	**0.118956**	0.21	0.574268
NOTIFY MSGS	0.000000	0.00	N/C
GLOBAL CONTENTION	0.054408	0.11	0.507807
COMMIT PH1 WRITE I/O	0.000000	0.00	N/C
ASYNCH CF REQUESTS	0.000001	0.06	0.000022
...			
TOTAL CLASS 3	0.174876	2.79	0.062616

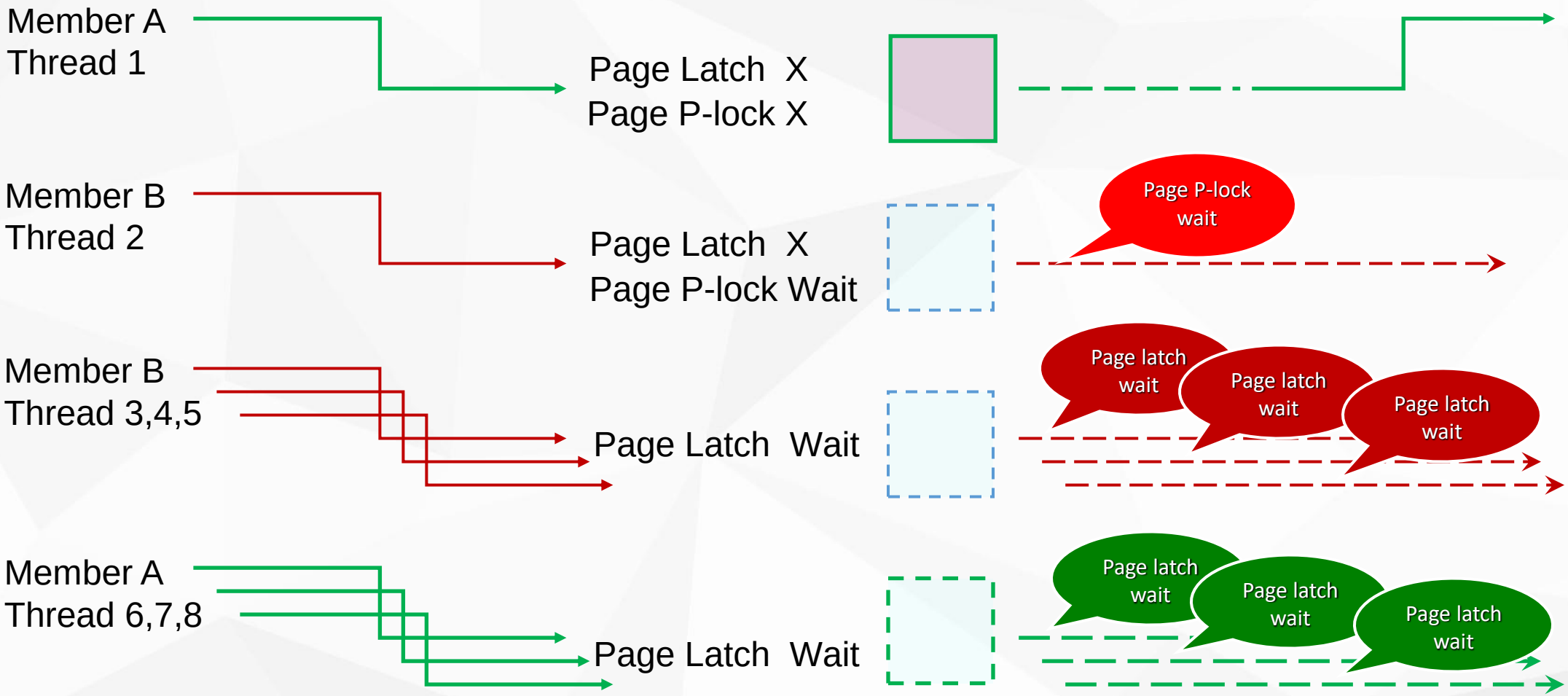
HIGHLIGHTS	
#OCCURRENCES :	140
#ALLIEDS :	0
#ALLIEDS DISTRIB:	0
#DBATS :	140
#DBATS DISTRIB. :	0
#NO PROGRAM DATA:	0
#NORMAL TERMINAT:	140
#ROLLUP TRAN :	140
#DDFRRSAF ROLLUP:	7
#ABNORMAL TERMIN:	0
...	
#COMMITS :	257
#ROLLBACKS :	0
...	
UPDATE/COMMIT :	0.06
SYNCH I/O AVG. :	0.000640
MAX WFILE BLKS :	0
#ZHL READ I/O :	0.00

Largest component of the CL2 ET by far, so must be a page latch problem

GLOBAL CONTENTION	L-LOCKS	AVERAGE TIME	AV.EVENT
PARENT (DB,TS,TAB,PART)		0.000000	0.00
CHILD (PAGE,ROW)		0.000000	0.00
OTHER		0.000000	0.00

GLOBAL CONTENTION	P-LOCKS	AVERAGE TIME	AV.EVENT
PAGESET/PARTITION		0.000000	0.00
PAGE		0.054364	0.10
OTHER		0.000044	0.01

Interaction between Page Latches and Global Contention



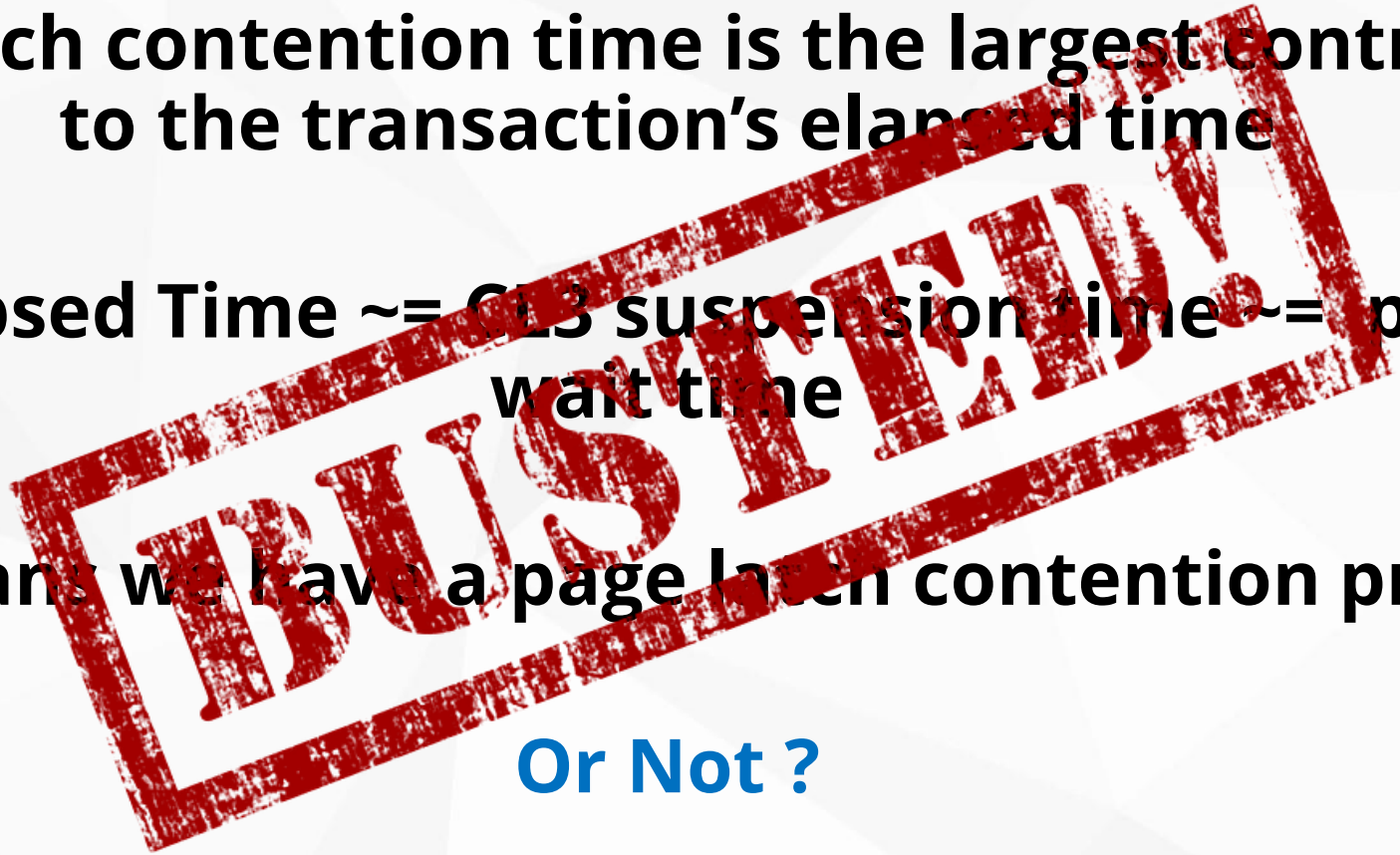
Interaction between Page Latches and Global Contention

- High page latch suspension can be the result of page P-lock contention
- In data sharing when an object is GBP-dependent Db2 will acquire
  - Page latch
  - Page P-lock (if needed)
- 1st thread on a member will wait on the page P-lock
- All subsequent threads will wait on the page latch (as that needs to be obtained first)
  - Db2 accounting reports can give the impression that the problem is page latch related (as many threads are waiting for the page latch) where in fact it could just be one P-lock negotiation that is slow and you need to investigate why that is taking so long more than finding out why an object or page is ‘hot’

MYTH – MISCONCECPTION

Page Latch contention time is the largest contributor to the transaction's elapsed time

Class 2 Elapsed Time ~= Cl.3 suspension time ~= page latch wait time



This means we have a page latch contention problem

Or Not ?

Agenda

- Db2 serialization terminology
- IRLM locks
  - IRLM lock characteristics
  - What affects locking behavior of an application ?
- Latches in general
- IRLM latches
  - IRLM latch characteristics
  - What can trigger IRLM latch suspensions
  - Monitoring IRLM latch suspensions
- Db2 latches in general
- Db2 (Internal) latches
  - Db2 internal latch characteristics
  - Popular latches classes
  - Monitoring Db2 internal latches
- Db2 page latches
  - Db2 page latch characteristics
  - Monitoring Db2 page latches
  - Interaction between page latches and global contention
- The importance of having enough CPU

Before I forget but Oh So Important !

- The lack of CPU resources for a transaction, Db2, LPAR, CF can have a dramatic effect on system performance
  - CPU 100% busy
  - Low dispatching priority
  - Enough CPU = Enough GCP , as well as enough zIIP capacity
- Such problems often show up as
  - Db2 “Not Accounted” time
  - Locking/latching problems
    - Transaction that are waiting for the CPU will hold locks/latches longer, elongating your transaction’s ET, but also others if they are waiting for a lock/latch you are holding
  - Long I/O response times (redispatch)
  - In data sharing, not only affect transactions on the member that is short on CPU, but also transactions on other members
- Having a good WLM policy is especially important when you run the machine at high utilization

Db2 Helping - Db2 Thread Boosting

- When a thread is stuck, Db2 will try to boost the thread
  - -DIS THD(*) SERVICE(WAIT)
    - Identifies Db2 threads that have been suspended for more than x seconds
      - x = MAX(60, 2x the IRLM timeout interval)
      - Note that for this type of boosting to occur you have to be suspended for at least 60 seconds
    - Be careful when reading the output to avoid false positives as DBATs that have not been in use for more than x seconds will also be listed
  - Internal Db2 monitor
    - The command is also driven by the Db2 internal system monitor every minute but no diagnostic output is written
  - Recommendation: Issue the command via automation at 1 min intervals and save the output as diagnostics

z/OS Helping - z/OS WLM Blocked Workload Support

- Allows small amounts of CPU to be provided to stalled dispatchable work units on the z/OS system ready queue
  - Not specific to Db2 workloads
- Controlled by IEAOPTxx parmlib settings
  - BLWLINTHD – Threshold time for which work must have been waiting for being considered for promotion – Default 20 (minimum is 1 (since OA44536))
    - Recommend to set 2-5 sec to get better overall system throughput at very high CPU utilization rates
  - BLWLTRPCT – How much CPU capacity of the LPAR can be used to promote blocked workloads – Default= 5 , meaning 0.5%
- z/OS doesn't support blocked workloads on specialty engines
- RMF CPU and workload manager activity report will show its usage

Summary – Hopefully this Presentation

- Gave you a better understanding of
  - IRLM locks, IRLM latches,
  - Db2 latches and Db2 page latches
  - How they all interact with each other
- Allows you to identify and zoom in on issues with these using
  - Db2 statistics data
  - Db2 accounting data
  - Using Db2 performance traces
    - IFCID 44, 45, 21 for IRLM lock/latch issues
    - IFCID 51, 52, 56, 57 for Db2 latch issues
    - IFCID 226, 227 for Db2 page latch issues
- Provided some insight on
  - Why it is important to have enough available CPU on your system
  - How Db2 and z/OS try to help work from getting stuck for too long

PDUG

Questions

POLISH DB2 USERS GROUP