"Run-It-Back"
Db2 for z/OS "2024 SWAT Tales"

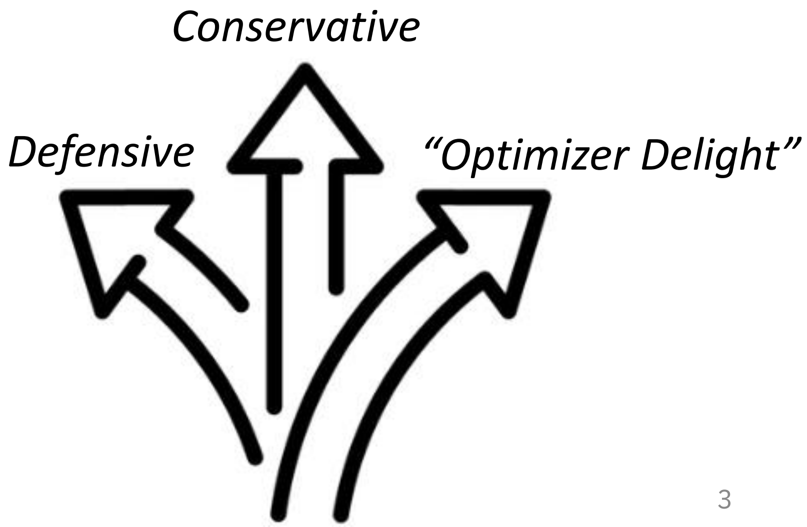
Anthony Ciabattoni
Executive IT Specialist
Db2 for z/OS SWAT Team Leader

Agenda

- Db2 REBIND Access Path Stability
- zHyperLink
- Continuous Availability
- Asynchronous LOCK1 Duplexing
- Questions

Db2 REBIND

- When are Db2 REBINDS are necessary ?
  - After successfully migrating to a new Db2 release and running smoothly, progressively REBIND high used packages
    - Re-enable fast column processing
    - Avoid performance overhead of “puffing” code
    - Pickup latest runtime performance enhancements
    - Pickup latest maintenance to address issues previously seeded
  - Package invalidation following an online REORG to materialize an online schema change
    - NON-UTS to UTS conversions
    - UTS PBG to UTS PBR conversions
  - Exploit RELEASE(DEALLOCATE) optimizations
    - CICS-Db2 Protected threads
    - HP-DBATs
  - Elevate APPLCOMPAT level
  - After applying a Db2 preventative maintenance package
- *What is your appetite for Db2 access path change?*
  - Defensive
    - Adverse to change
  - Conservative
  - *“Optimizer Delight”*
    - Allow optimizer to choose at each REBIND



REBIND Options

- REBIND options
  - APREUSE (ERROR|WARN|NONE)
    - ERROR
      - ✓ Db2 tries to reuse the previous access paths for SQL statements in the package
      - ✓ Will guarantee the same access path or REBIND will fail
      - ✓ Db2 indicates the number of statements that cannot be reused in any package in a message
    - WARN
      - ✓ Db2 tries to reuse the previous access paths for SQL statements in the package
      - ✓ Successful with no warnings if same access path is available
      - ✓ If same access path is not available, optimizer will choose new access path (evaluated in previous step) and will be successful with warnings
    - NONE
      - ✓ Db2 does not try to reuse previous access paths for statements in the package

REBIND Options …

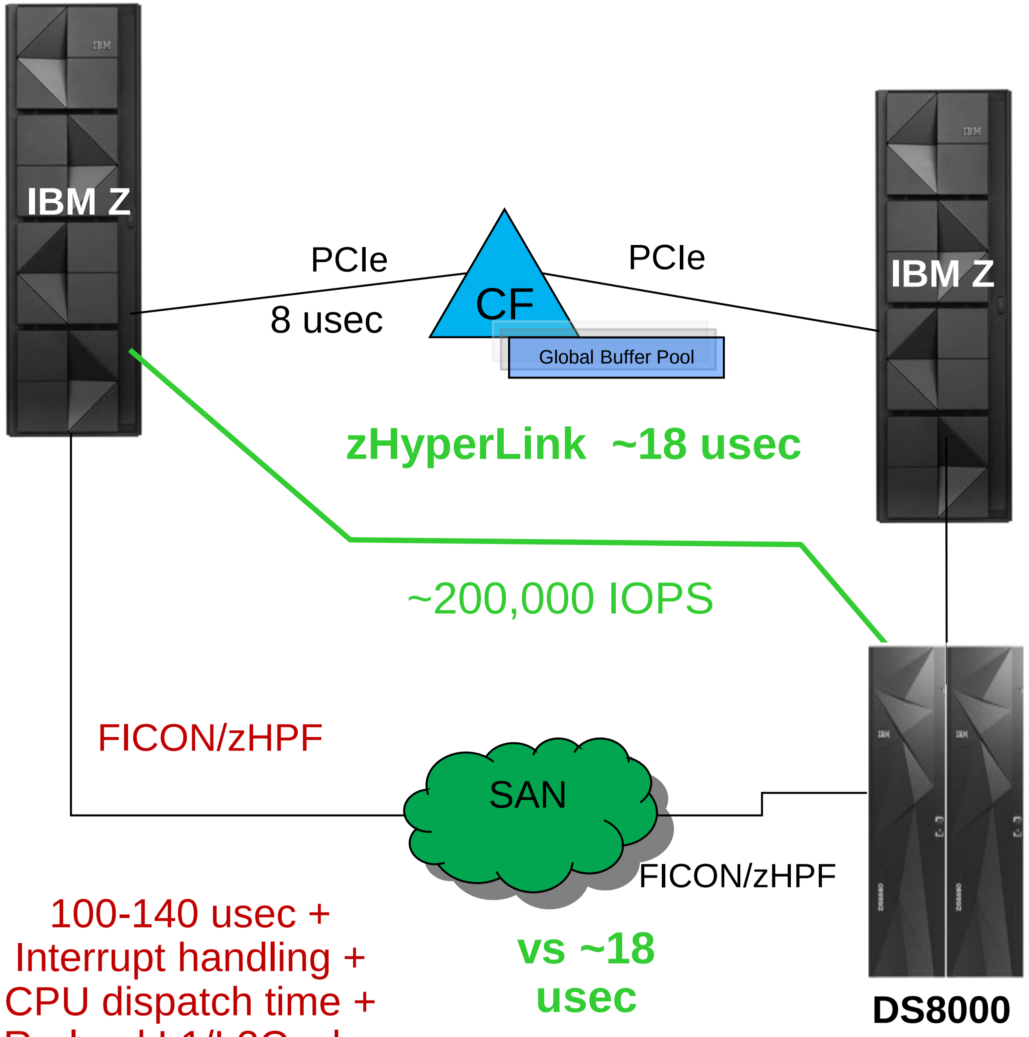
- REBIND options …
  - APCOMPARE (ERROR|WARN|NONE)
    - ERROR
      - ✓ Optimal access path will be selected (no guarantee the same access path will be selected)
      - ✓ If access path is structurally dissimilar when compare previous access path to current
        - REBIND will fail
    - WARN
      - ✓ Optimal access path will be selected (no guarantee the same access path will be selected)
      - ✓ If access path is structurally dissimilar compare previous access path to current
      - ✓ REBIND will be successful with warnings
        - Dissimilar SQL statements will be reported
    - NONE
      - ✓ Db2 does not try to reuse previous access paths for statements in the package
  - APREUSE = APCOMPARE = NONE
    - *“Optimizer Delight”*
      - Allow the optimizer to choose appropriate access path at each REBIND
      - Strongly recommend using Extended Plan Management
        - Revert a package to use previously saved access paths

REBIND Options ...

- To minimize risk when performing a REBIND needed for non-access path operational improvements or when a Db2 package is invalidated
  - APREUSE(ERROR) should be the installation default
    - Defensive option to re-establish previous access path
    - Must apply APAR PH63063 (Db2 13)
      - APREUSE fails when attempting to reuse a Db2 12 access path that involves a view or table expression
    - Addressing failed APREUSE REBINDS
      - ✓ REBIND with APCOMARE(ERROR) EXPLAIN ONLY
        - Db2 13 APAR PH61970 supplies “Phase-In” support for REBIND package EXPLAIN ONLY
        - Will identify access path change
        - Evaluate access path differences
      - ✓ REBIND APREUSE(WARN)
        - Successful with no warnings if same access path is available
        - If same access path is not available, optimizer will choose new access path (evaluated in previous step) and will be successful with warnings
  - If successful REBIND using APREUSE (ERROR or WARN)
    - Develop a process to evaluate potential access path improvements
      - ✓ REBIND APRCOMPARE(WARN) EXPLAIN ONLY
      - ✓ Perform analysis on any access path changes identified by the Db2 Optimizer
      - ✓ REBIND with APCOMPARE(WARN)
        - Optimal access path will be selected (no guarantee the same access path will be selected)

zHyperLink – Can Significantly Reduce Elapsed Times

- zHyperLink is a direct connect physical connection between a Z server and storage system using the same cabling as short distance coupling
  - Reads are exploited by Db2 for synchronous reads
  - Writes are exploited by Db2 for the active log
- zHyperLink is *FAST* enough the CPU can just wait for the data
  - No Un-dispatch of the running task
  - No I/O interrupt delay
  - No CPU Queueing Delays to resume it
  - No host CPU cache disruption
  - Very small I/O service time
- Operating System and Middleware are enhanced for *synchronous I/O*
- *Transparently* gives DB2 and VSAM apps fundamentally better latency than applications on platforms without zHyperLink
  - Excluding 100% in memory databases



100-140 usec + Interrupt handling + CPU dispatch time + Re-load L1/L2Cache

vs ~18 usec

zHyperLink – Extreme Data Access Acceleration

zHyperLink is a *direct connect physical connection* between a Z server and storage system using the same cabling as short distance coupling and supported up to 150m

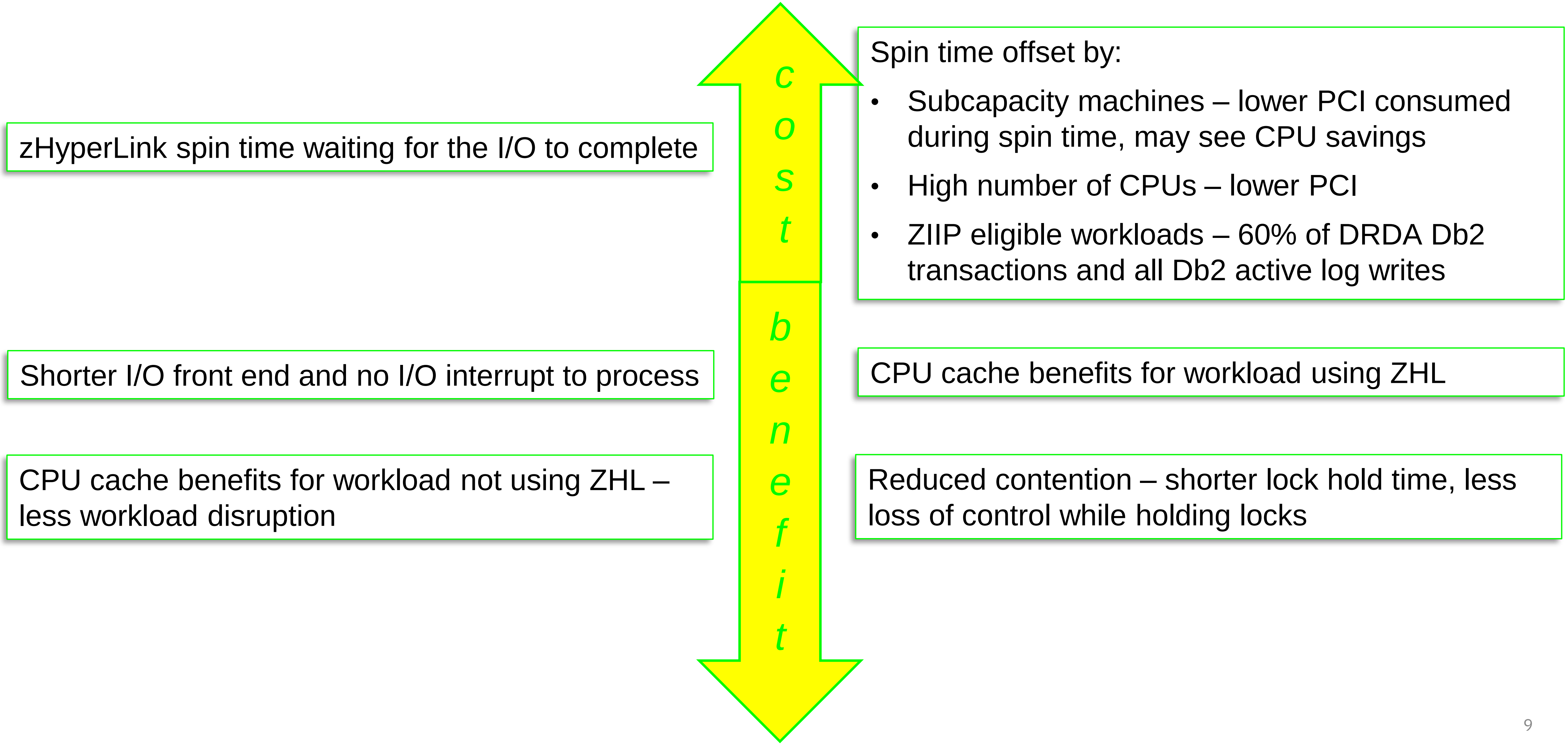
zHyperLink reads are exploited by Db2 for synchronous reads and VSAM for reads except NSR sequential

zHyperLink writes are exploited by Db2 for the active log

zHyperLink writes can be used with local *Metro Mirror* in conjunction with zHyperWrite and with Global Mirror (plus 3-site and 4-site configurations which have local Metro Mirror)

Consistent read from Metro Mirror secondary allows zHyperLink reads to be performed from a secondary device enabling zHyperLink reads for all members of a x-site parallel Sysplex

zHyperLink – CPU Usage …



cost

benefit

zHyperLink spin time waiting for the I/O to complete

Spin time offset by:

- Subcapacity machines – lower PCI consumed during spin time, may see CPU savings
- High number of CPUs – lower PCI
- ZIIP eligible workloads – 60% of DRDA Db2 transactions and all Db2 active log writes

Shorter I/O front end and no I/O interrupt to process

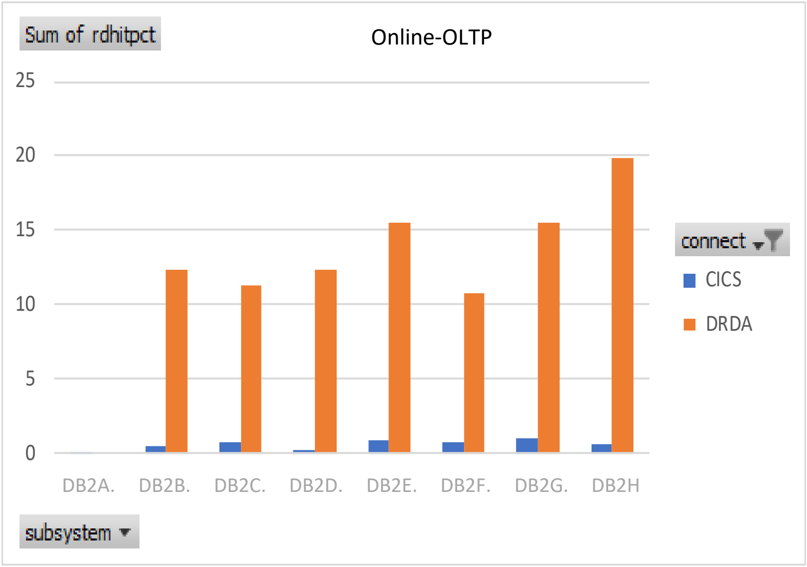
CPU cache benefits for workload using ZHL

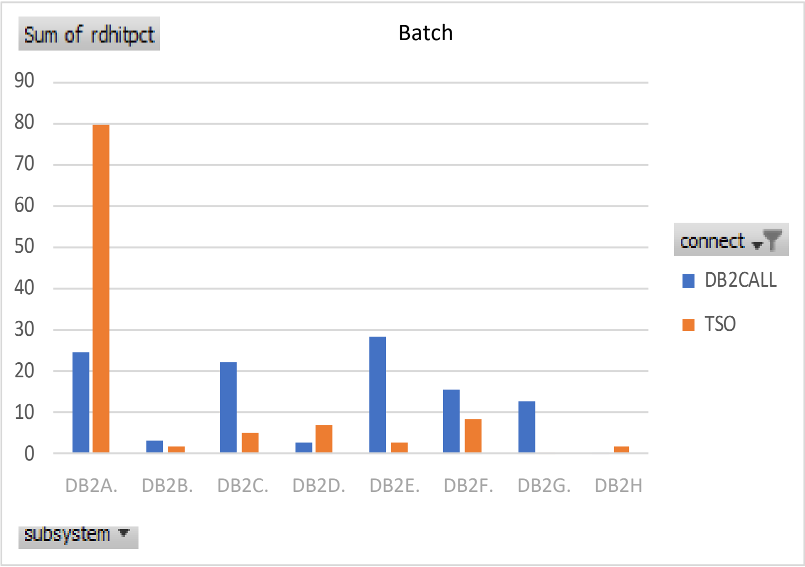
CPU cache benefits for workload not using ZHL – less workload disruption

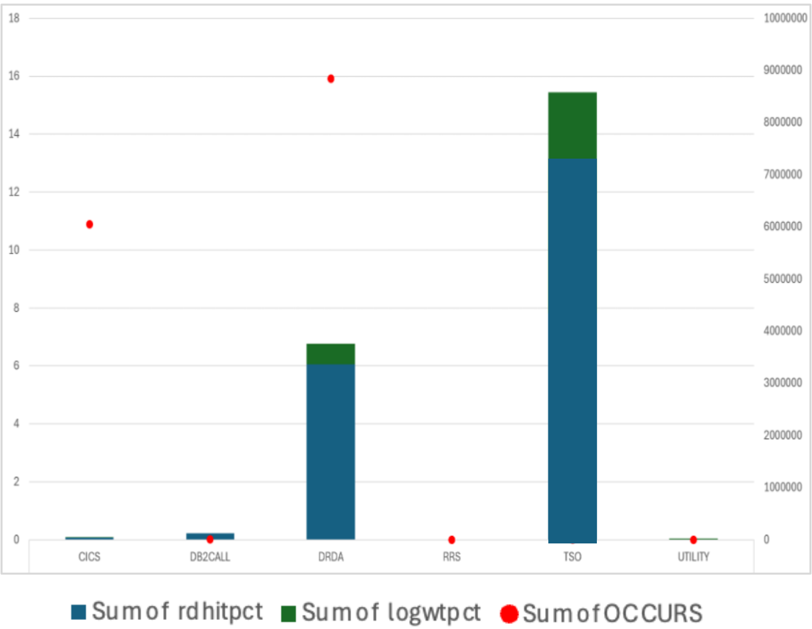
Reduced contention – shorter lock hold time, less loss of control while holding locks

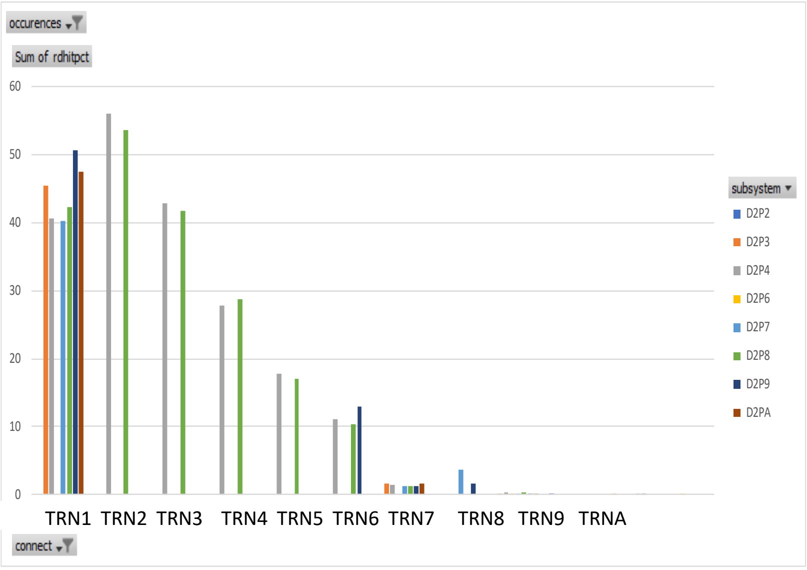
zHyperLink – Eligible Reads

- Typically, small log write benefits
- Online - OLTP
  - DRDA connections
    - ✓ 10-20% of elapsed time is zHyperLink eligible
- Batch
  - TSO connections – DB2A
    - ✓ 80% of elapsed time is zHyperLink eligible
  - DB2CALL
    - ✓ 10-30% of elapsed time zHyperLink eligible
- CICS
  - While overall average CICS improvement with zHyperLink is relatively low
    - ✓ Certain transactions which have much higher potential for improvement
    - ✓ TRN1, TRN2, TRN3, TRN4, TRN5 can see Up to ~10-40% improvement









zHyperLink – CPU Usage

No easy formula to determine the impact, if any, of zHyperLink on CPU usage - interaction is complex and environment dependent

An *overall system level CPU comparison* is the only effective metric. Challenging for real workload – day to day variation may be higher than zHyperLink benefit or cost

Observations from customer environments show anywhere from a *slight CPU improvement to slight CPU increase*

In environments with *large I/O workloads and small amount of business logic* and *competing workloads, a CPU increase* may be observed

zHyperLink – How not to Measure CPU Usage

1

Read the Db2 reports and add up the zHyperLink time

- Db2 is reporting the response time for Db2, not the CPU time for zHyperLink

2

Look at the zHyperLink response times in z/OS or DS8000 and add up the zHyperLink time

- z/OS response times are closer but do not reflect the benefits (e.g., cache benefits)

3

Look at batch job / address space statistics and evaluate the increases CPU time

- This will miss the benefits to other workloads

4

Run a simple single thread test environment and measure the CPU time

- This will miss the cache, contention, and other workload benefits

Remember *- an overall system level CPU comparison is the only effective metric.* It can be challenging to measure for a real workload where the variation in workload day to day is likely higher than any benefit or cost of zHyperLink

Value of zHyperLink Reads

Performance

20-50% of overall SIO can be *converted to zHyperLink* reads in typical exploiting environments – this is a much smaller percentage of I/O bandwidth

10-20% improvements in read intensive batch and occasionally higher

5-10% improvements in *transactional response time* for transactions performing significant synchronous reads and occasionally higher

CPU Usage

zHyperLink activity is on the same processor as the initiating tasks so can be on a *ZIIP* if the workload is ZIIP enabled (e.g., for certain Db2 workloads)

CPU usage is a concern for many clients even though *the absolute numbers are small* and typical experience is *neutral or better overall impact*

The *inability to give assured numbers* can introduce significant friction in deployment

Restrictions

Inability to exploit zHyperLink for *larger read sizes* is an issue with some clients – more so for VSAM but Db2 installations may also move more to larger page sizes over time.

MQ and IMS do not have any *specific read support* today

How have Clients enabled zHyperLink

1

Storage class controls whether an eligible data set can use zHyperLink

2

During early testing, client may *selectively enable data sets*

3

For production, *micro-managing* thousands of data sets may be *time consuming and not practical*

4

Therefore, one might simply *enable existing storage classes* for zHyperLink and do *selective enablement*, for example, by Db2 member

5

Most clients perform a staged implementation and evaluate the behavior after each change

6

Alternatively, you may *choose subsets* of Db2 tables or VSAM files to *enable via storage class* changes

7

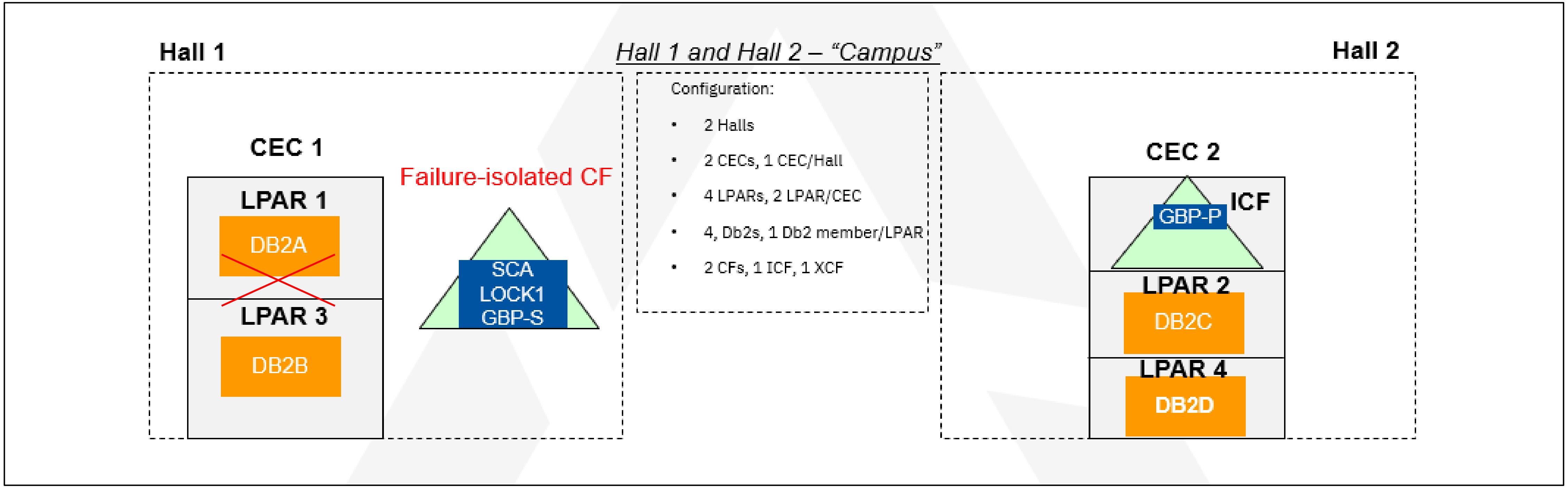
Toggle, dynamically enable during the batch window and disable during online processing day

Continuous Availability

- ***Problem: Big investment in z/OS Parallel Sysplex and Db2 for z/OS data sharing, but still cannot achieve 'true' continuous availability***
  - *Unable to maintain continuous availability and mask planned and unplanned outages from applications*
- z/OS Parallel Sysplex and Db2 Data Sharing are the ***'Gold Standard'*** in terms of continuous availability
  - Provides the base infrastructure to build upon
- <u>Additional ingredients are required to achieve the highest levels of continuous availability</u>
  - Active inter-system read-write data sharing
  - Replicated cloned applications – "Towers"
    - Multiple redundant instances of an application workload running across multiple systems
  - Fine-grained, dynamic transaction routing
    - Use the aggregate capacity of multiple images to satisfy peak demands
    - Improve application availability, throughput and scalability
    - Provide automatic re-route around failure
  - Dynamically add or increase available capacity when needed
  - CF structure duplexing
  - Automated Db2 and system restart

Continuous Availability …

- Parallel Sysplex configuration
  - Common Problems – Db2 Member Failure



Configuration:

- 2 Halls
- 2 CECs, 1 CEC/Hall
- 4 LPARs, 2 LPAR/CEC
- 4, Db2s, 1 Db2 member/LPAR
- 2 CFs, 1 ICF, 1 XCF

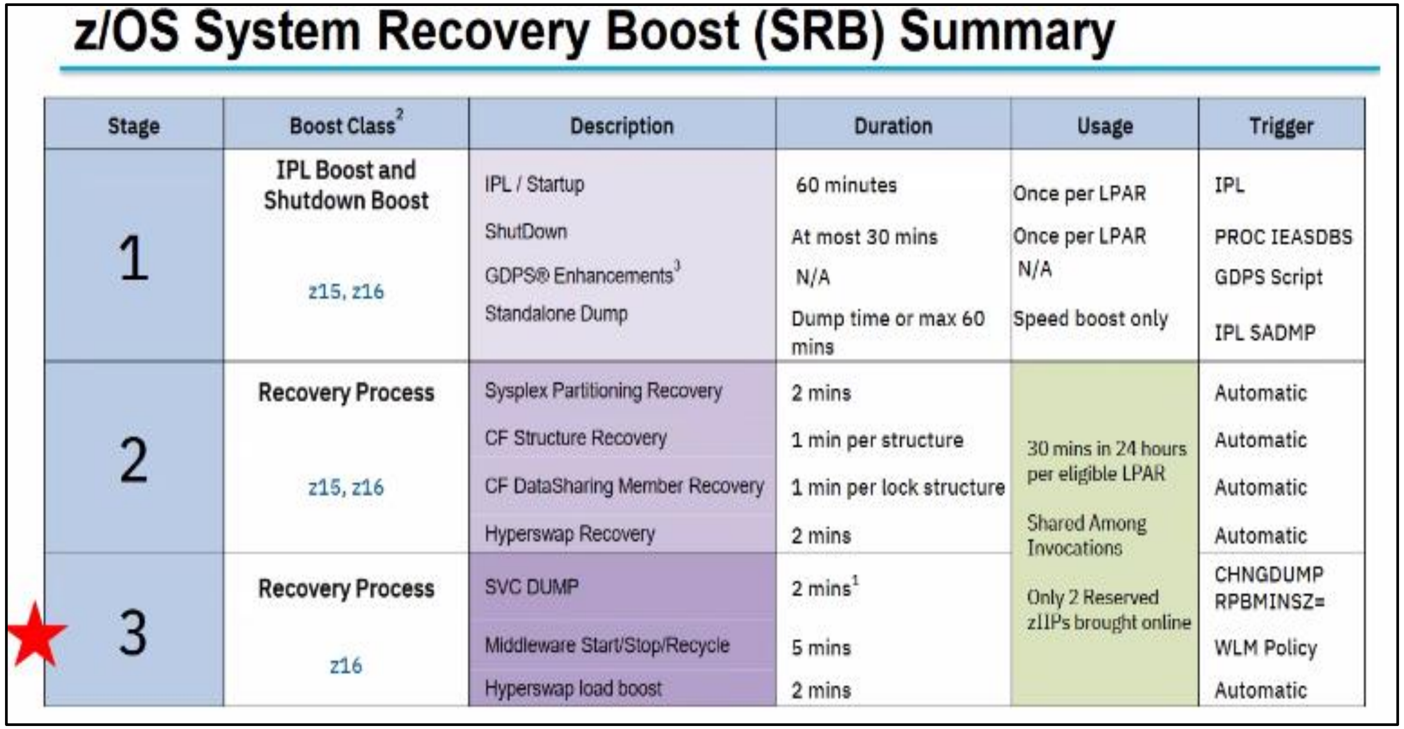
- True "Campus" continuous availability - 4-way active data sharing across 4 LPARs on two CECs (boxes)
  - Infrastructure is designed to support true continuous availability
- Db2 member failure
  - All other components remain active

Db2 Member Failure

- Db2 restart after failure
  - Objective is to restore application availability to normal behavior as soon as possible
    - Recommendation is to use automation to detect failures quickly and restart Db2 immediately
      - Must have a laser focused goal of resolving/removing unavailable resources held by the failing Db2 member as soon as possible to minimize any impact on continuous application availability
        - ✓ Retained locks
        - ✓ Indoubt Unit of Recovery (URs)
        - ✓ Excessive postponed abort processing
        - ✓ Resume normal capacity
      - Option 1 - Automated Restart Manager (ARM)
        - ✓ Extremely fast restart with limited additional capabilities
          - Example, RESTART_ATTEMPTS(3,300), 3 restarts as fast as possible within 300 seconds
      - Option 2 - Automated operators (e.g., Tivoli System Automation family)
        - ✓ Ability to add additional restart logic
          - Inject a delay in between successive Db2 member restart attempts after the initial restart failure and before attempting the next restart
          - Force down a “hung” address space that prevents a successful restart e.g., IRLM, DIST

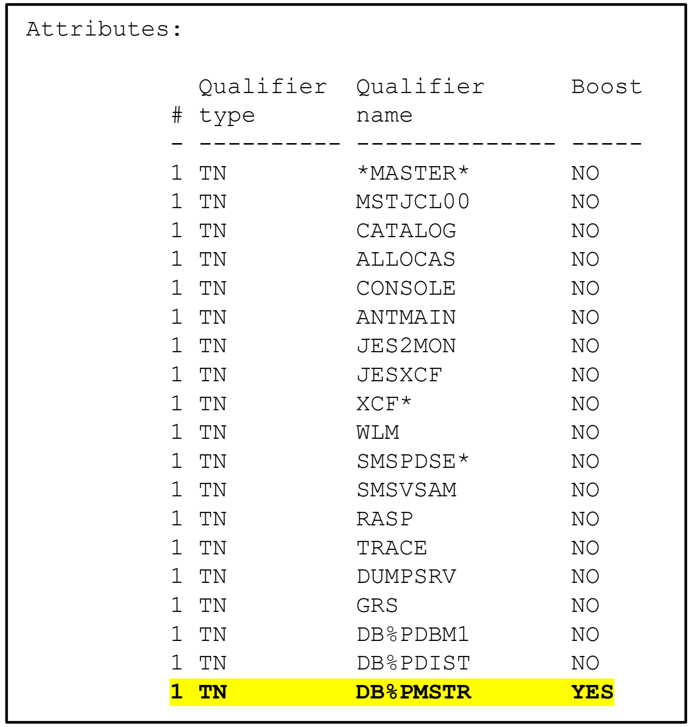
Db2 Member Failure …

- Db2 restart after failure …
  - Exploit Middleware Recovery Boost on z/16 for faster Db2
    - 6 available boosts a day
    - 5 minute in duration



z/OS System Recovery Boost (SRB) Summary

Stage	Boost Class[2]	Description	Duration	Usage	Trigger
1	IPL Boost and Shutdown Boost z15, z16	IPL / Startup	60 minutes	Once per LPAR	IPL
		ShutDown	At most 30 mins	Once per LPAR	PROC IEASDBS
		GDPS® Enhancements[3]	N/A	N/A	GDPS Script
		Standalone Dump	Dump time or max 60 mins	Speed boost only	IPL SADMP
2	Recovery Process z15, z16	Sysplex Partitioning Recovery	2 mins	30 mins in 24 hours per eligible LPAR Shared Among Invocations Only 2 Reserved zIIPs brought online	Automatic
		CF Structure Recovery	1 min per structure		Automatic
		CF DataSharing Member Recovery	1 min per lock structure		Automatic
		Hyperswap Recovery	2 mins		Automatic
★ 3	Recovery Process z16	SVC DUMP	2 mins[1]		CHNGDUMP RPBMINSZ=
		Middleware Start/Stop/Recycle	5 mins		WLM Policy
		Hyperswap load boost	2 mins		Automatic

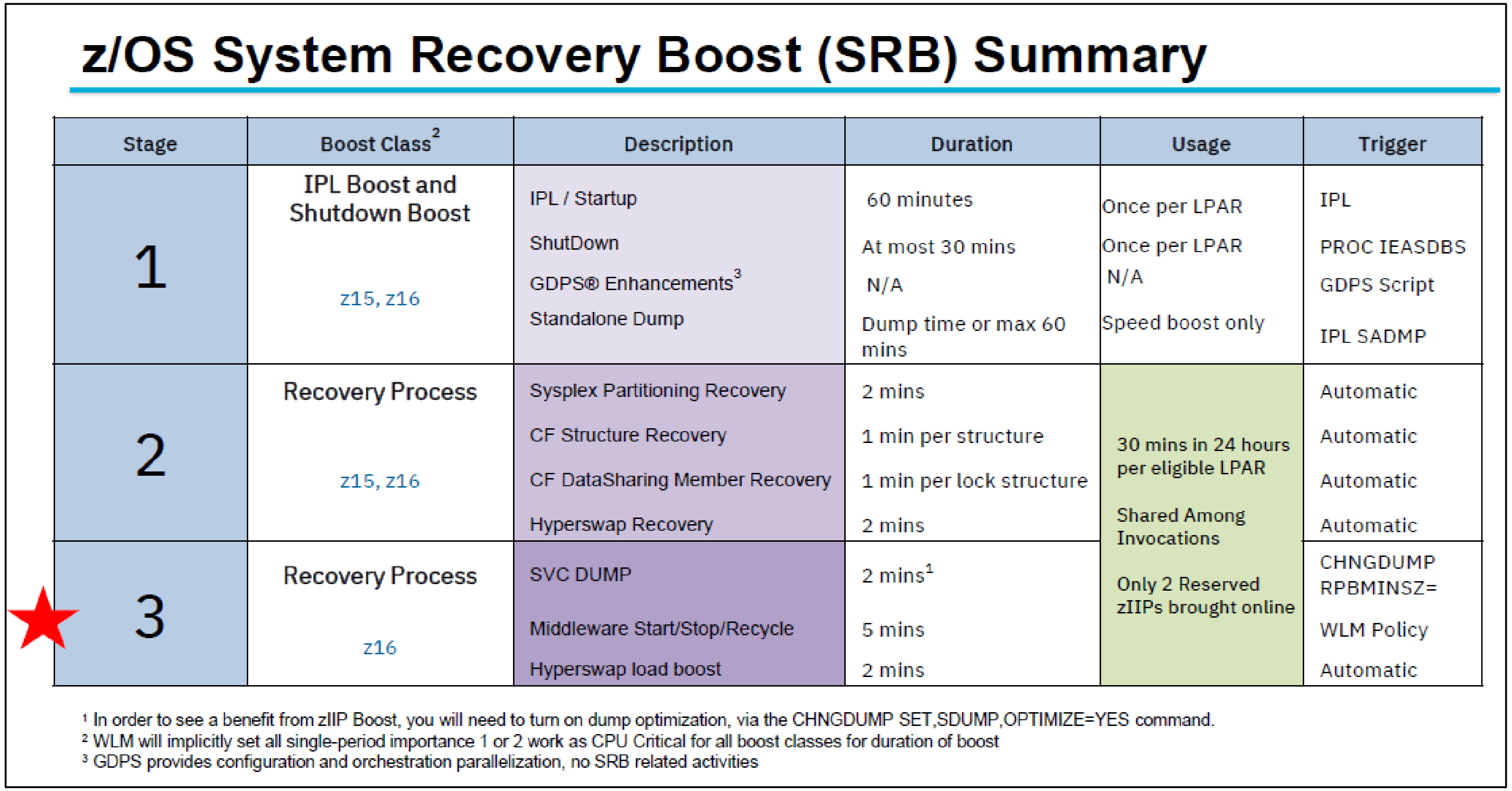


```
Attributes:

    Qualifier  Qualifier       Boost
  # type       name
  - ---------- --------------- -----
  1 TN         *MASTER*        NO
  1 TN         MSTJCL00        NO
  1 TN         CATALOG         NO
  1 TN         ALLOCAS         NO
  1 TN         CONSOLE         NO
  1 TN         ANTMAIN         NO
  1 TN         JES2MON         NO
  1 TN         JESXCF          NO
  1 TN         XCF*            NO
  1 TN         WLM             NO
  1 TN         SMSPDSE*        NO
  1 TN         SMSVSAM         NO
  1 TN         RASP            NO
  1 TN         TRACE           NO
  1 TN         DUMPSRV         NO
  1 TN         GRS             NO
  1 TN         DB%PDBM1        NO
  1 TN         DB%PDIST        NO
  1 TN         DB%PMSTR        YES
```

Db2 Member Failure …

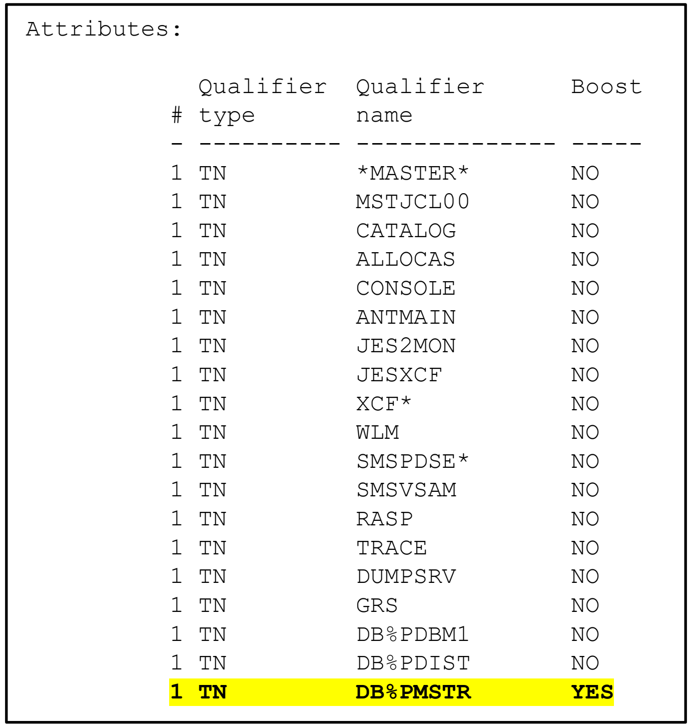
- Db2 restart after failure …
  - Exploit Middleware Recovery Boost on z/16 for faster Db2
    - 6 available boosts a day
    - 5 minute in duration



z/OS System Recovery Boost (SRB) Summary

Stage	Boost Class[2]	Description	Duration	Usage	Trigger
1	IPL Boost and Shutdown Boost	IPL / Startup	60 minutes	Once per LPAR	IPL
		ShutDown	At most 30 mins	Once per LPAR	PROC IEASDBS
	z15, z16	GDPS® Enhancements[3]	N/A	N/A	GDPS Script
		Standalone Dump	Dump time or max 60 mins	Speed boost only	IPL SADMP
2	Recovery Process	Sysplex Partitioning Recovery	2 mins	30 mins in 24 hours per eligible LPAR Shared Among Invocations Only 2 Reserved zIIPs brought online	Automatic
		CF Structure Recovery	1 min per structure		Automatic
	z15, z16	CF DataSharing Member Recovery	1 min per lock structure		Automatic
		Hyperswap Recovery	2 mins		Automatic
★ 3	Recovery Process	SVC DUMP	2 mins[1]		CHNGDUMP RPBMINSZ=
		Middleware Start/Stop/Recycle	5 mins		WLM Policy
	z16	Hyperswap load boost	2 mins		Automatic

[1] In order to see a benefit from zIIP Boost, you will need to turn on dump optimization, via the CHNGDUMP SET,SDUMP,OPTIMIZE=YES command.
[2] WLM will implicitly set all single-period importance 1 or 2 work as CPU Critical for all boost classes for duration of boost
[3] GDPS provides configuration and orchestration parallelization, no SRB related activities

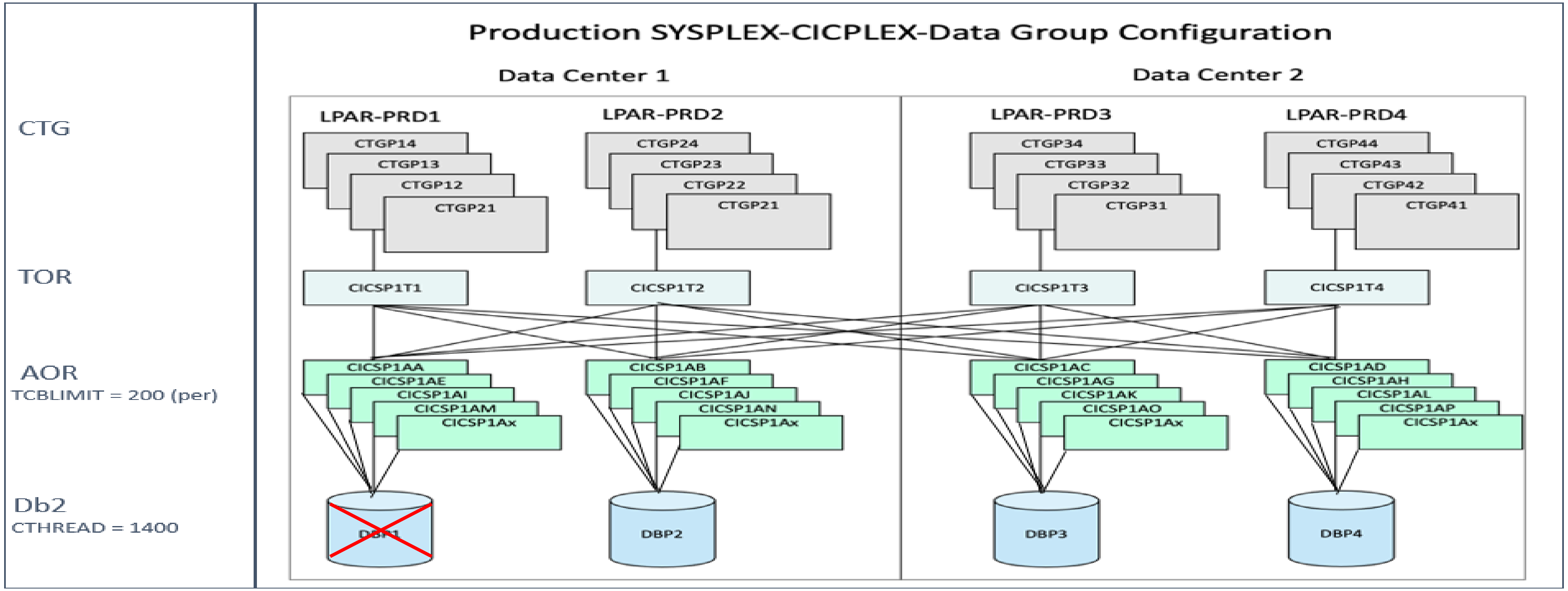


```
Attributes:

                 Qualifier   Qualifier       Boost
               # type        name
               - ----------  --------------  -----
               1 TN          *MASTER*        NO
               1 TN          MSTJCL00        NO
               1 TN          CATALOG         NO
               1 TN          ALLOCAS         NO
               1 TN          CONSOLE         NO
               1 TN          ANTMAIN         NO
               1 TN          JES2MON         NO
               1 TN          JESXCF          NO
               1 TN          XCF*            NO
               1 TN          WLM             NO
               1 TN          SMSPDSE*        NO
               1 TN          SMSVSAM         NO
               1 TN          RASP            NO
               1 TN          TRACE           NO
               1 TN          DUMPSRV         NO
               1 TN          GRS             NO
               1 TN          DB%PDBM1        NO
               1 TN          DB%PDIST        NO
               1 TN          DB%PMSTR        YES
```

Db2 Member Failure ...

- No automation in place to immediately stop the traffic being routed to the CICS AOR regions that were attached to the failed Db2 member which will result in transactions failing fast
  - ”Storm drain” effect of fast failing transactions will continue until manual action is taken to shut off the routing of transaction to the respective CICS AOR regions
  - ”Tsunami" effect will occur and continue until manual action is taken which will likely continue for many minutes until resolution

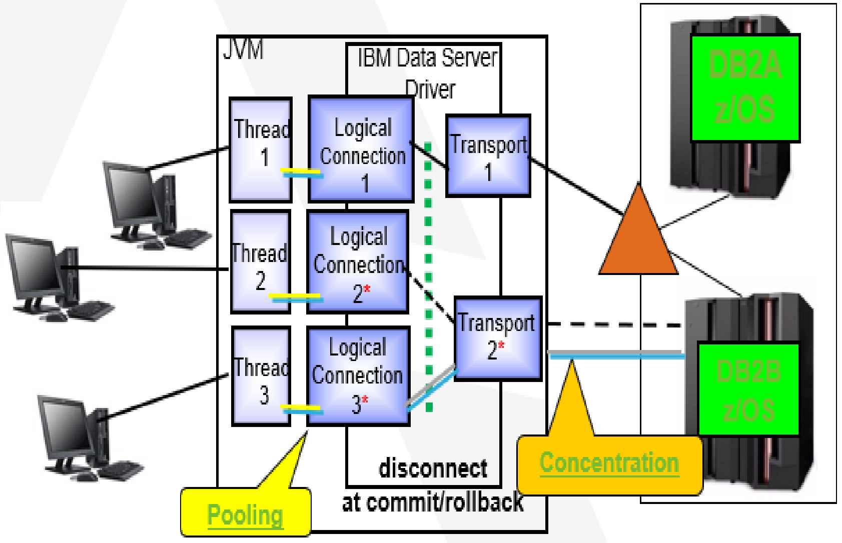


Db2 Member Failure …

- CICS ”storm drain” effect - recommendations
  - Implement CPSM settings in CICS Transaction Server or develop automation to eliminate "storm drain" effect
    - Method 1 – develop automation to
      - ✓ Detect a failed Db2 member and SET CEMT WLMHEALTH OPENSTATUS(IMMCLOSE)
      - ✓ Detect when the Db2 member is restarted and CICS AOR is reconnected SET WLMHEALTH OPENSTATUS(OPEN)
    - Method 2 – CPSM settings
      - ✓ DB2CONN settings
        - CONNECTERROR(SQLCODE)/STANDBYMODE(RECONNECT)
      - ✓ ABENDCRIT
        - Abend probability, CPSM sets to 100% unhealthy when errors occur
        - At intervals AOR(s) become healthier
        - When ABENDCRIT value is reached a “sacrificial” transaction is sent
          - If ”sacrificial” transaction fails, CPMS sets unhealthiness back to 100%
      - ✓ ABENDTHRESH
        - After ABENDCRIT is reached CPMS views AOR(s) as half-healthy and distributes approximately ~50% of workload to the AOR(s)
        - When ABENDTHRESH reached CPMS views AOR(s) as healthy and normal workload distribution is resumed

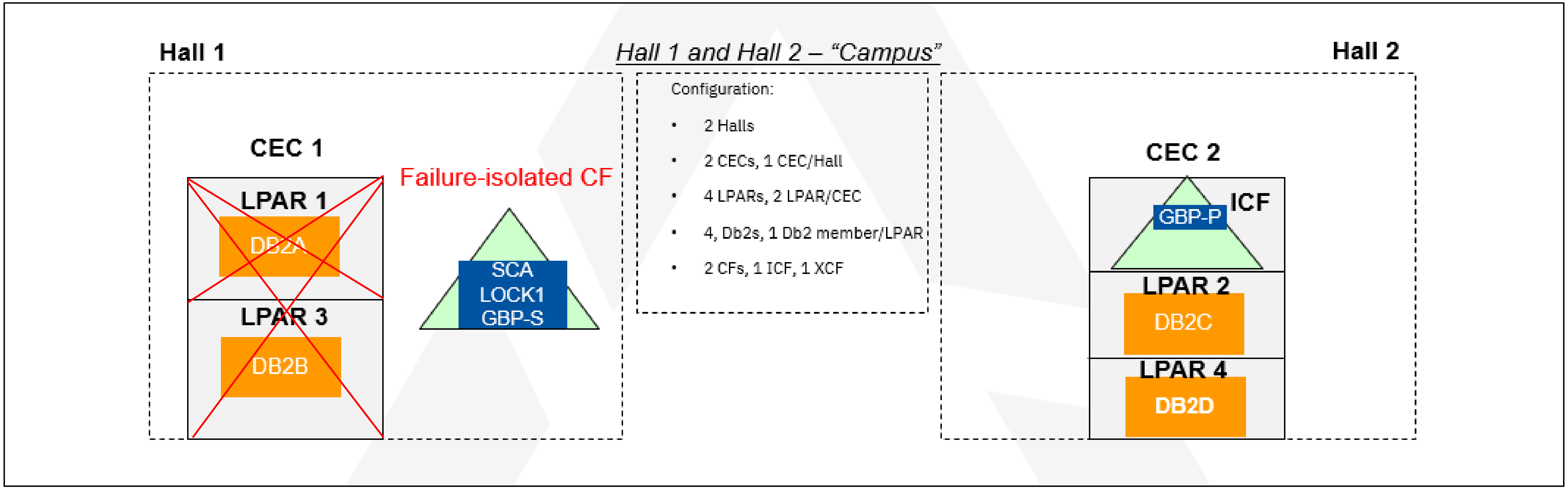
Db2 Member Failure …

- DDF application connections are not cleaned up during a planned or unplanned Db2 restart event which result in transaction failures
  - No automatic client re-route, or transaction retry logic
- Enable Sysplex Workload Balancing
  - Enables transport pooling (connection concentration)
    - Separates the logical connection from the application to the driver and the physical connection from the driver to Db2 for z/OS
    - Global pool of Transport objects per JVM
    - Initial connection goes thru Sysplex Distributor, then driver relies on its WLM server list to route to the member directly
    - Connection reuses/changes transport at transaction boundary (commit)
  - Provides Automatic Client Reroute (ACR)
    - Masks connectivity issue with Db2 including Db2 failure/shutdown
    - If connectivity to DB2A is lost i.e., taken down for maintenance or crashes
      a) Driver receives network failure (-30108)
      b) Driver seamlessly routes transaction to another transport on another member (no negative SQL code)



Continuous Availability

- Parallel Sysplex configuration …
  - Common Problems – LPAR or CEC Failure



- 1 failed LPAR (25%)
  - 3 surviving LPARs
- 1 CEC failure (50%)
  - Resulting in losing 2 of 4 LPARs

LPAR or CEC Failure

- Objective is to take immediate action to allow the surviving infrastructure to maintain application availability until the failing component can be successfully be restarted
  - Resolve/removed Db2 resources held by the Failing Db2 member(s)
  - Add temporary capacity to support the ***surge*** of workload on the surviving infrastructur
- Resolve/removed Db2 resources held by the Failing Db2 member(s)
  - Db2 cross-system restart
    - "Restart Light" = START DB2 LIGHT(YES|NOINDOUBTS|CASTOUT)
    - Design point of Restart Light
      - ✓ **ONLY** for cross-system restart following LPAR/CEC failure, where the 'home' LPAR will be recovered as soon as possible, including a normal 'full' Db2 restart
      - ✓ **NOT** intended for restart in place on the 'home' LPAR
      - ✓ **NOT** intended for Db2 disaster recovery crash restart
    - Small memory footprint
      - ✓ Used to avoid ECSA/CSA virtual storage shortage on the alternative LPAR
      - ✓ Avoids real memory shortage on alternative LPAR by severely constraining pool sizes
    - Simplified management
      - ✓ Does not allow new workload to come in
      - ✓ ***Can*** shut down Db2 member automatically
  - ***BUT … Restart Light can be much slower than normal Db2 crash restart if not configured properly …***

LPAR or CEC Failure …

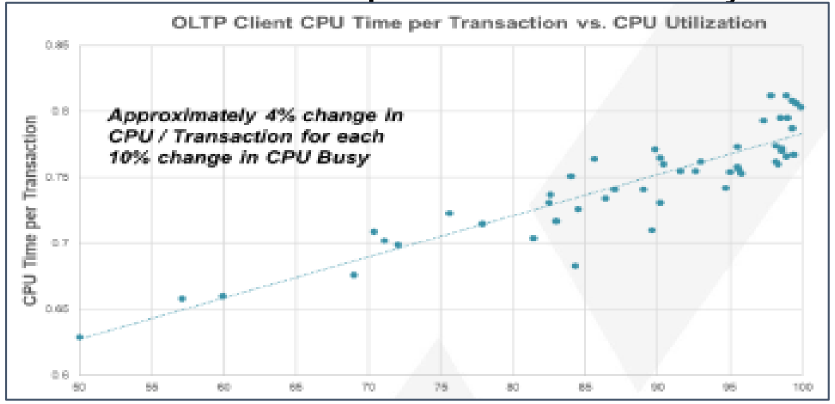
- Db2 cross system restart …
  - Restart Light and Postponed Abort
    - Combination of Restart Light with ZPARM LBACKOUT=AUTO setting (default)
      - ✓ LBACKOUT ignored and is treated as LBACKOUT=NO
        - Abort URs will not be postponed, and restart will wait until complete processing of Abort URs are successful
        - Restart could take a take a long time to complete if there are very long running URs to abort
    - Recommendation: LBACKOUT=LIGHTAUTO to enable postponed Abort for Restart Light
  - Restart Light and indoubt URs
    - With LIGHT(***YES***), Db2 will not terminate automatically until all indoubt URs are resolved
    - **Option #1**: Provided there is sufficient spare REAL memory available on the alternate LPAR, could use automation to restart CICS and/or IMS/TM infrastructure on the same alternate LPAR as the Db2 member being restarted
    - **Option #2** (usually preferred): Use option LIGHT(NOINDOUBTS)
      - ✓ Db2 will terminate automatically and not resolve indoubt URs
      - ✓ But keep in mind that retained locks cannot be freed until the indoubt UR is resolved
    - Restart Light and retained locks
      - ✓ Restart Light(YES|NOINDOUBTS) will not clean up all retained locks
        - Should expect to see message DSNT804I at the end of restart

```
DSNT804I - THERE ARE MODIFY LOCKS OWNED BY THIS DB2 THAT HAVE BEEN RETAINED
```

        - Retained transaction locks and X-mode pageset P-locks are released, SQL DML processing will not be blocked (exception mass delete)
        - But retained pageset p-locks in IX or SIX mode will NOT be released
          - No Db2 utility that is a claimer will be blocked e.g., COPY SHRLEVEL(CHANGE) will work
          - But drainers will be blocked e.g., REORG including REORG SHRLEVEL(CHANGE), SQL DDL
        - Will be cleared once the Db2 is restarted normally in its 'home' LPAR

LPAR or CEC Failure …

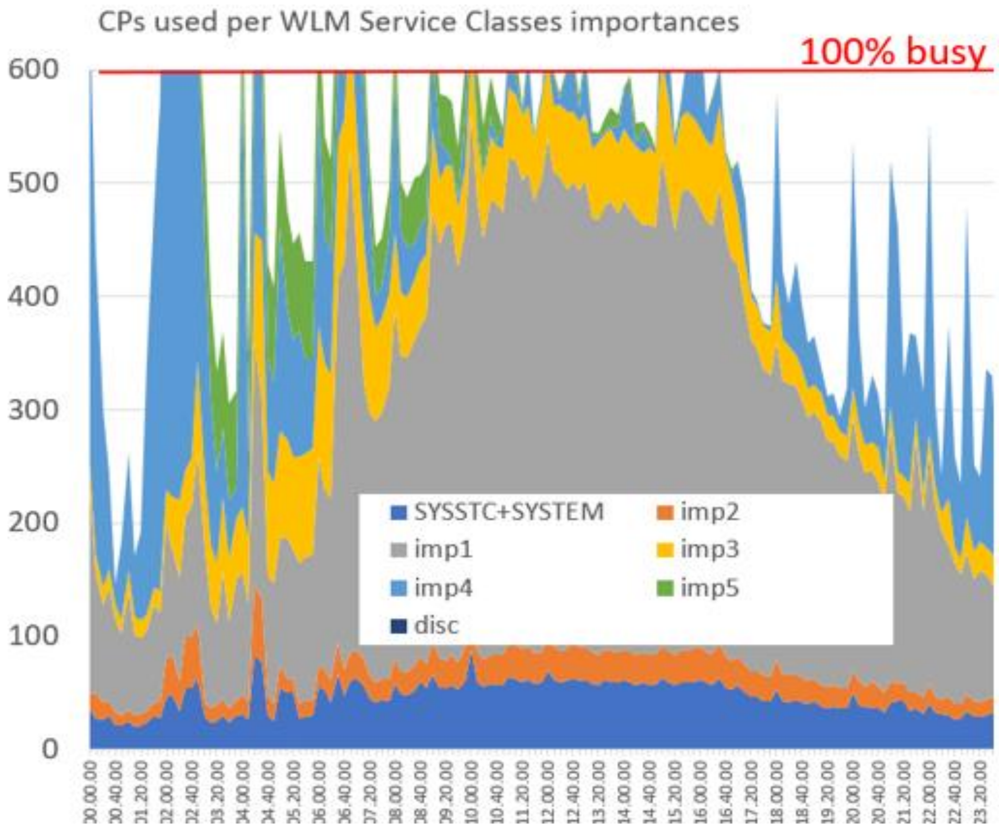
- Understanding available capacity
  - MSU Consumption Sensitivity to Utilization – 4-10 Rule of Thumb

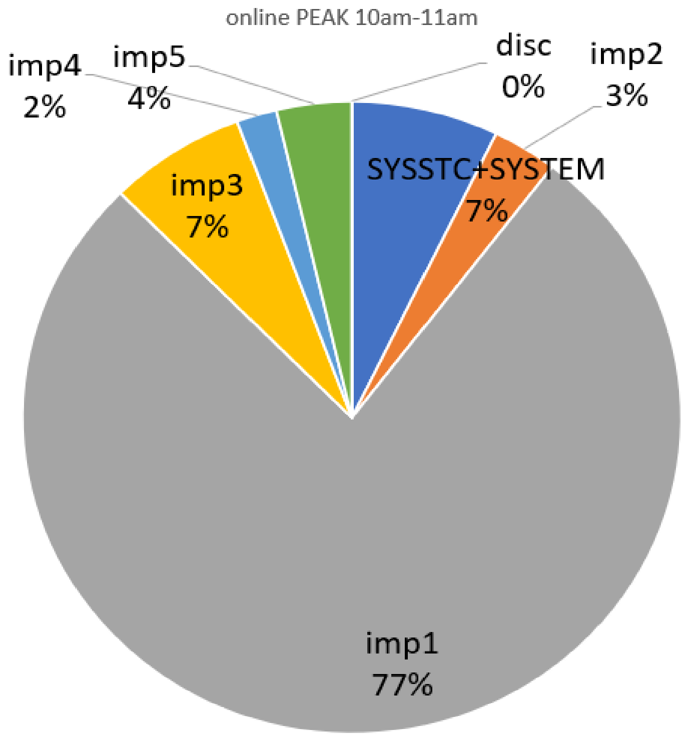


  - Machines run more efficiently at lower utilizations
    - More HW cache per SW work unit
    - Cost per transaction drops
  - Magnitude of effects varies by workload and N-way
    - Lower N-way have smaller effect
    - Lower RNI workload have smaller effect
  - The 4-10 ROT: on average, a 10% change in processor utilization results in a 4% change in cost (CPU time, MIPS, MSUs) per transaction. (3-10 for LOW RNI, and 5-10 for HIGH RNI)
  - Processor utilization can be lowered by:
    - Running less workload on a constant HW configuration
    - Running a constant workload on a larger HW configuration
  - Thus, it is possible for clients to realize a 4% MSU savings (with some variation) by adding enough HW to lower absolute processor utilization by 10%
    - ✓How much HW would need to be added to lower utilization by 10%?
      - If utilization was 100%, would need to add 100/90 = 1.11 or +11% capacity
      - If utilization was 50%, world need to add 50/40 = 1.25 or +25% capacity

LPAR or CEC Failure ...

- Understanding available capacity ...
  - Additional CPU or designated sacrificial workload is needed for the surviving infrastructure to absorb the workload from that was lost with the failing component(s)
    - Systems are run at very high CPU utilization for elongated periods of time
    - Little or no non-Db2 work to pre-empt when the system becomes 100% busy i.e., no non-Db2 work that can be sacrificed





LPAR or CEC Failure ...

- Adding CPU capacity and/or altering the WLM policy often is a reactionary actions based on exceptions, monitoring and alerting
  - Manual decision to add additional capacity
    - Often takes executive approval
      - ✓ Typically, a binary decision to add capacity in an attempt to resume normal application behavior as fast as possible
    - Must accept the ***"pain and consequences"*** of not adding capacity fast enough
      - ✓ Failed transactions
      - ✓ Slow transactions
      - ✓ Additional threads/agents in the system
- If the decision is to always add capacity, why not automate the solution?
  - Create a "Self Healing" process
    - Establish and document the saturation points/thresholds
    - Acquire executive pre-approval
    - Automatically add capacity

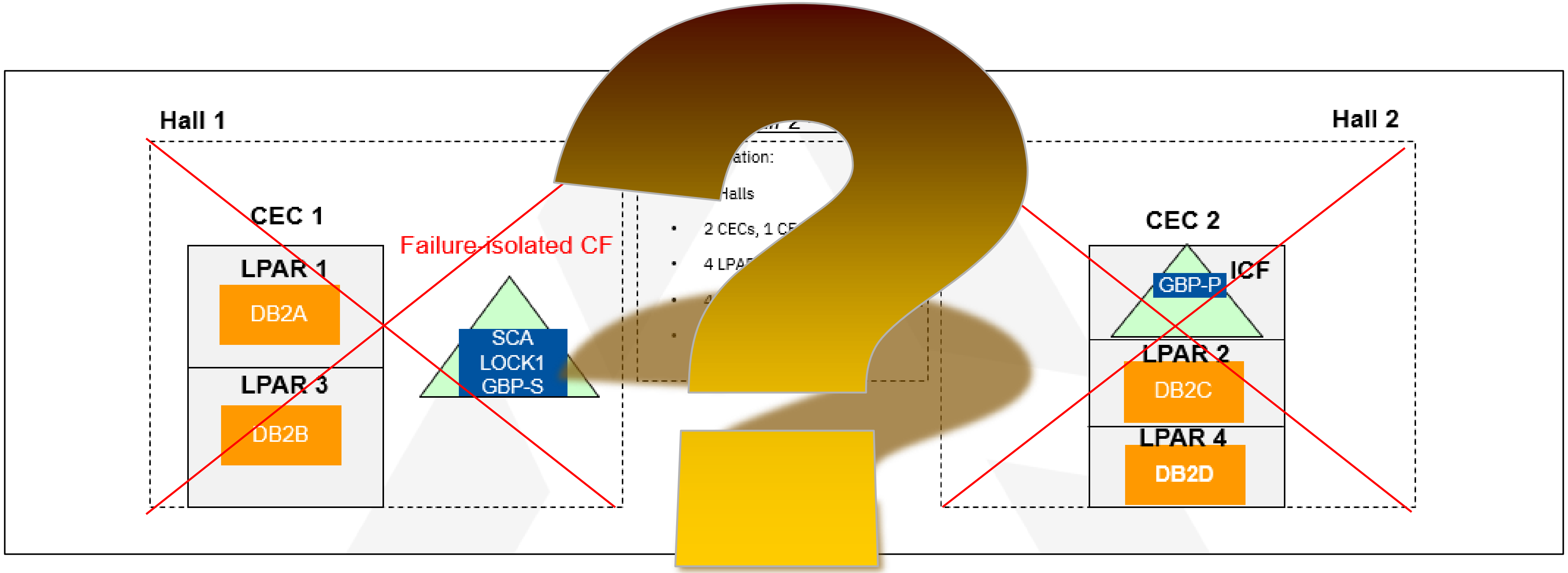
LPAR or CEC Failure …

- Manual decision and process to IPL the failed LPAR
  - Likely take ~1+ hours to resume availability
    - Collect diagnostic information which is needed for root cause analysis
    - Decide (think time) to IPL the failed LPAR
    - Perform the Automated IPL process
  - Resources unavailable until normal restart
    - Db2 pages with indoubt units of work
      - ✓ Needs to successfully reconnect with the commit coordinator
    - Resources held by postponed abort URs
    - Any Single Point of Failure (SPOF) dependent on subject LPAR or Db2 member

- Automatically IPL the failed LPAR in place
  - z/OS Auto IPL with stand-alone dump option
    - AUTOIPL SADMP(xxxx,SM) MVS(LAST)
  - GDPS Auto IPL
    - AUTOIPL=YES, THRESHOLD=(tt,hh:mm)
      - ✓ Default – THRESHOLD=(2,12:00)
    - Create additional automation to perform a stand-alone

Continuous Availability …

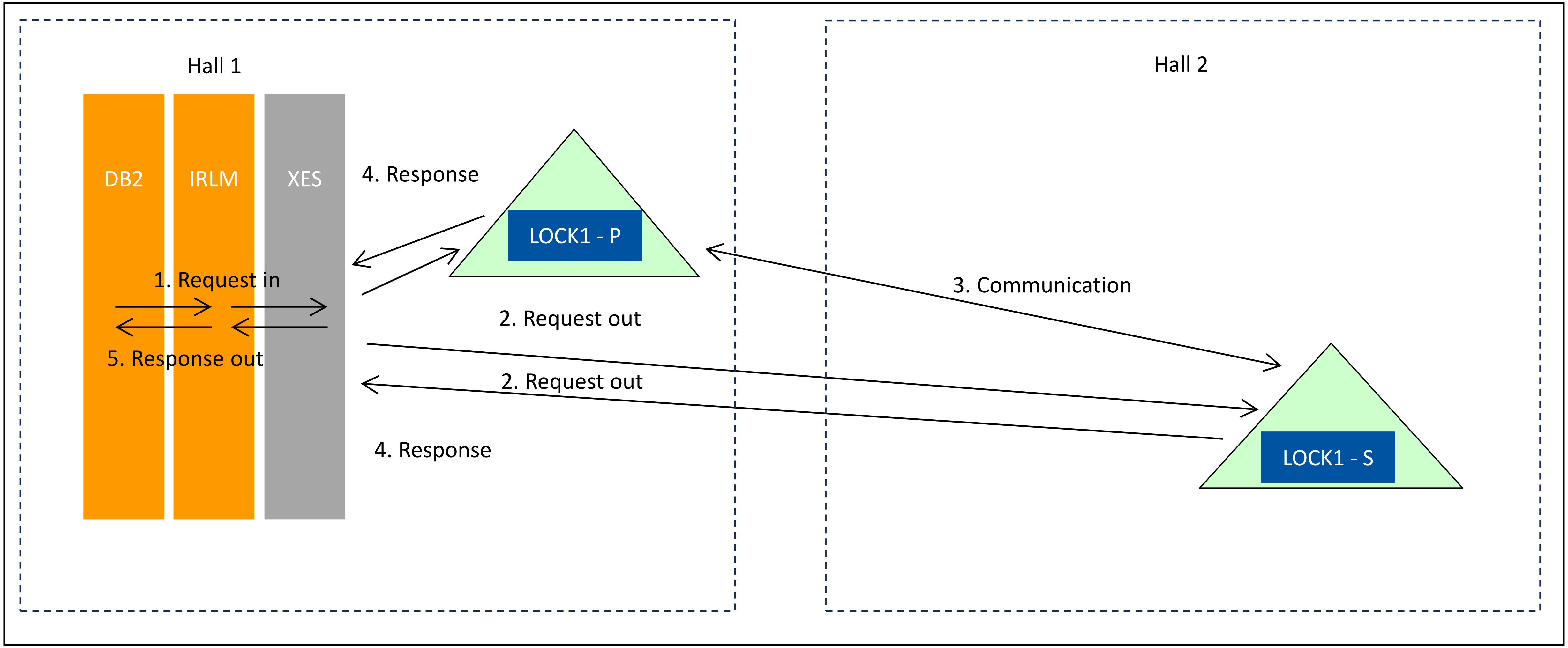
- Parallel Sysplex configuration …
  - Common Problems – Data Center Failure



- Objective is to achieve continuous availability during an unplanned Data Center outage/failure
  - Hall 2 failure would result in the surviving Data Center will need to absorb 50% of the failed workload
  - Hall 1 failure will result in a complete application outage
    - Db2 group restart would occur in Hall 2 on alternative CECs
- In the absence of a 3rd Data Center Db2 CF LOCK1/SCA duplexing would be needed to achieve objective

Db2 Synchronous CF lock duplexing

- Synchronous CF lock structure duplexing – how it works today

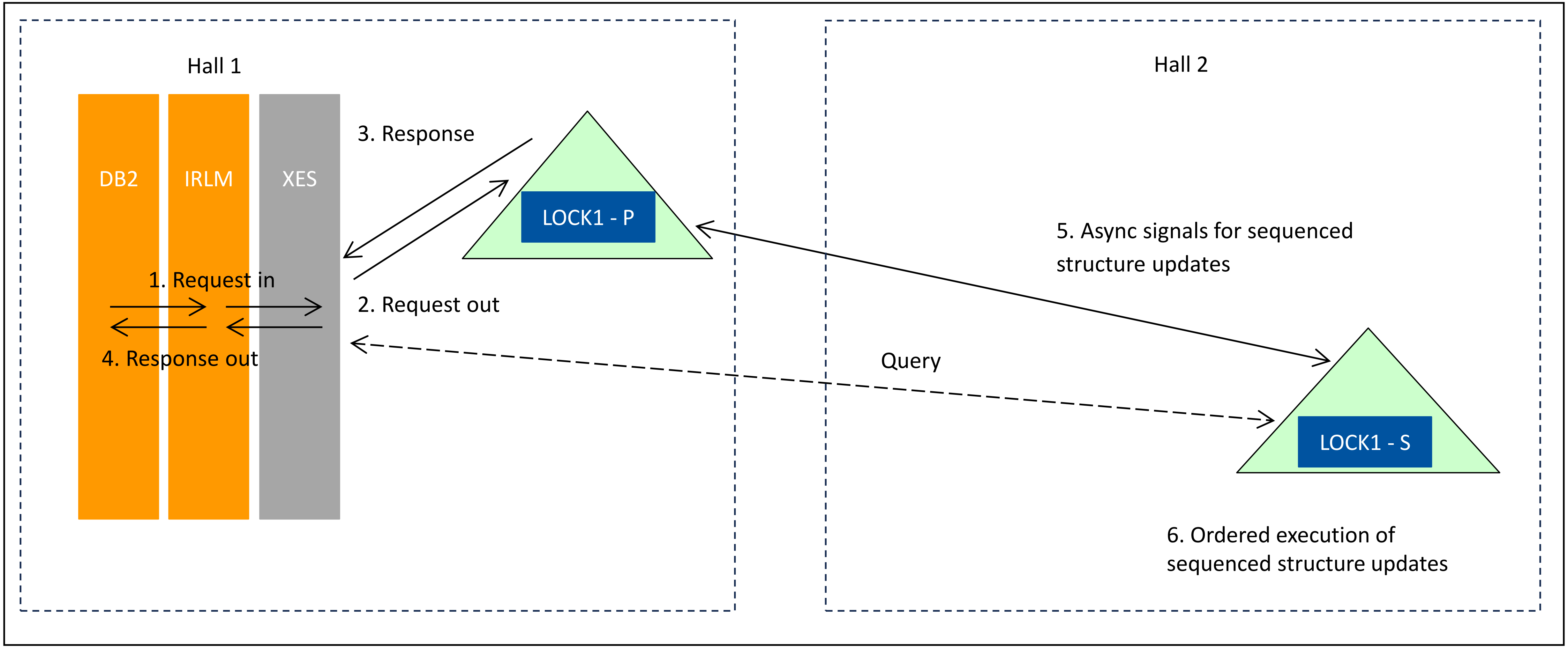


Db2 Asynchronous CF lock duplexing

- Introduced in Db2 12
  - Reduces overhead for System Managed Duplexing (SMD) of CF LOCK1 structure
  - Secondary structure updates are performed asynchronously with respect to primary updates
  - Db2 will sync up with z/OS to ensure data integrity i.e., all modify locks have been "hardened" in the secondary lock structure before the corresponding undo/redo record for the update is written to the Db2 active log on DASD
  - The physical log writer performs the 'sync' call to query the secondary, and it happens whenever log records get physically written to DASD, which can be earlier than commit
- Increases the practical distance for multi-site sysplex operations whilst duplexing of CF LOCK1 structure

Db2 Asynchronous CF lock duplexing …

- Asynchronous CF lock structure duplexing – how it will now work

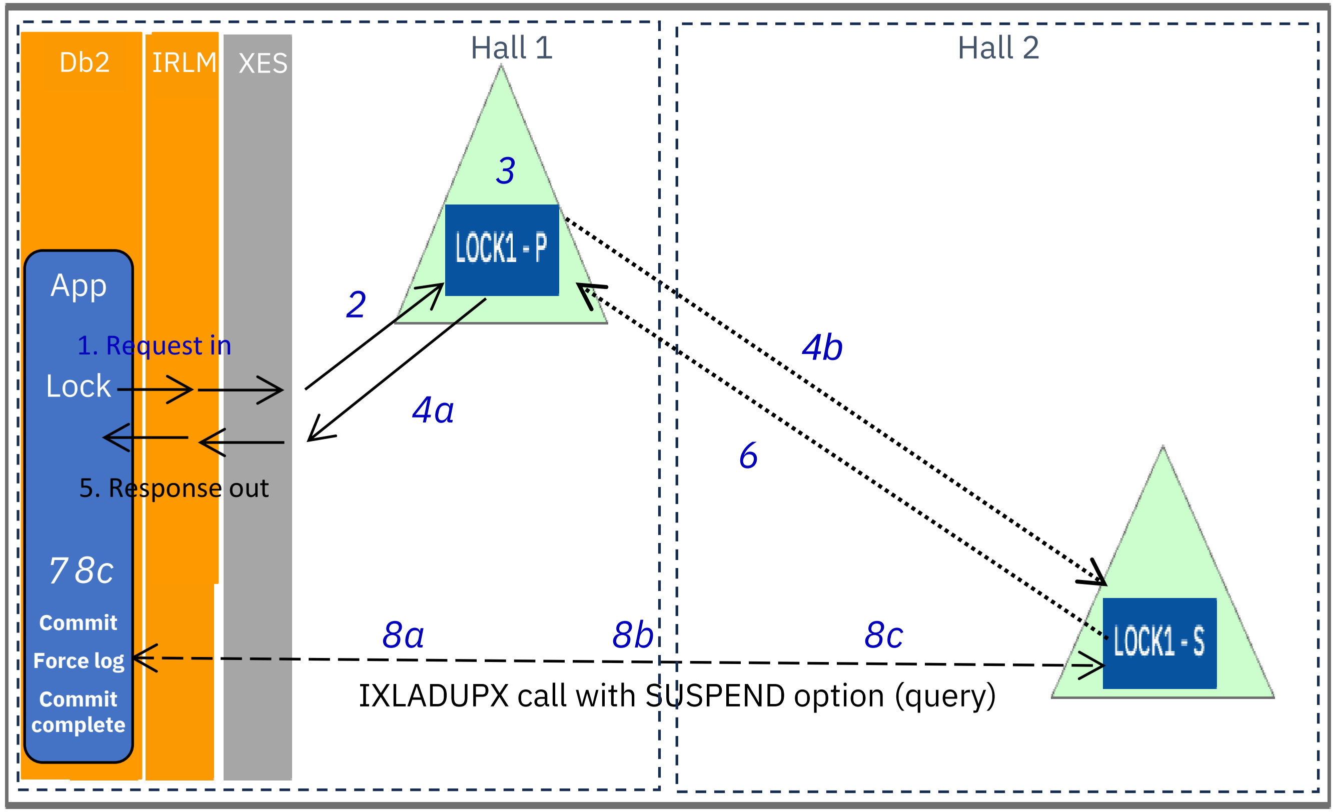

Hall 1
DB2
IRLM
XES
3. Response
LOCK1 - P
1. Request in
2. Request out
4. Response out
Hall 2
5. Async signals for sequenced structure updates
Query
LOCK1 - S
6. Ordered execution of sequenced structure updates

Db2 12 asynchronous CF lock duplexing

- Terminology
  - Unique Sequence Number (USN)
  - Last Committed Operational Sequence Number (LCOSN)
    - This field is maintained in the secondary structure
    - Updated every time CFCC completes a request to update that structure.
  - Last Committed Operational Sequence Number known to Primary (LCOSNP)
    - This field is maintained in the primary structure
    - Updated based on the LCOSN value that the secondary CF included in the response to the request sent over from the Primary CF
    - Typically, a little behind where the secondary CF is
      - ✓ Because it is based on the LCOSN value at that time the secondary CF received the most recent request, not when it completed the request

Db2 12 asynchronous CF lock duplexing ...

- Asynchronous CF lock structure duplexing
  - How it will now work (details)



1. Request in
2. Request Unique Sequence Number (USN)
3. Generate Unique Sequence Number (USN)
4. Lock result
   a. USN & LCOSNP returned
   b. USN placed on queue in secondary
5. Response out
6. Secondary sends response to primary
   a) Confirms request has been received
   b) Sends sequence number of the most recently completed request (LCOSN)
7. If LCOSNP >= Unique Sequence Number (USN)
   a) Force log, complete commit
8. If LCOSNP < Unique Sequence Number (USN)
   a) IRLM makes IXLADUPX call with the SUSPEND option to receive LCOSN
   b) SUSPEND will continue until LCOSN >= USN
   c) When LCOSN >= USN
      i. Commit/Log Write

Db2 Asynchronous CF lock duplexing ...

- Benefits
  - Cost of lock structure duplexing is significantly lower
    - Host CPU for lock requests decreases
    - IRLMs receive responses sooner
  - Existing sites using synchronous SMD should see lower host CPU cost and better elapsed times
  - More environments can now achieve higher availability in all-ICF configurations
    - Reduce risk with asynchronous CF lock duplexing with less cost all round
      - ✓ Hardware maintenance
      - ✓ Capital cost for extra frames
  - Processor technology refresh applies to both host GCP and ICF engines
- But it is not free for simplex users
  - Will have to acquire ICF engines and coupling links for CF-to-CF connectivity
  - CF utilization is higher for asynchronous System-Managed Structure Duplexing relative to simplex case, but it is much less than synchronous CF Lock duplexing
    - Expected to be higher than simplex because there is simply more work for the CF to do
      - ✓ Host CPU: 1.2x times simplex cost
    - Estimated CF utilization for Primary Lock structure:
      - ✓ CF utilization for the Simplex Lock structure * 7 * 2/3
    - Estimated CF utilization for Secondary Lock structure:
      - ✓ CF utilization for Simplex Lock structure * 7 * 1/3
    - CF links: Links between the PRIM and SEC CFs need to support 1.5x times the Lock rate

Summary

- Options exist to minimize risk when performing a Db2 package REBIND
- Availability options can significantly reduce the risk of performing a REBIND
- If your installation objective is to reduced application elapsed time zHyperLink is a technology that should be explored and tested
- z/OS Parallel Sysplex and Db2 Data Sharing are the ***'Gold Standard'*** in terms of continuous availability
- Reality is failure, defects, hardware malfunctions and operational error are all going to happen at some point in time
  - Additional ingredients are required to achieve the highest levels of continuous availability
  - Must build and design to 'mask' unplanned and planned outages from the application
    - Redundancy
    - Resiliency
  - Financial investment
    - Hardware
    - Software
- 'True' continuous availability can be achieved with z/OS Parallel Sysplex and Db2 for z/OS data sharing continuous availability with a proper design and vision

Questions

Thank You