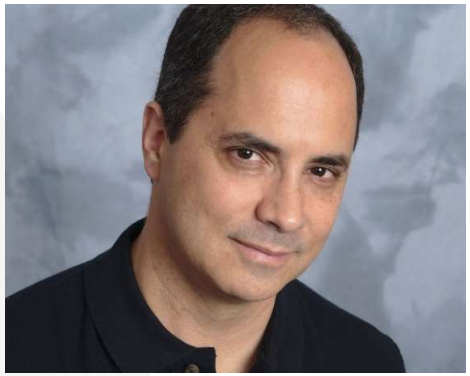
Back to Basics Introduction to Parallel Sysplex and Data Sharing

Peter Enrico

Email: Peter.Enrico@EPStrategies.com

Enterprise Performance Strategies, Inc.
3457-53rd Avenue North, #145
Bradenton, FL 34210
http://www.epstrategies.com
http://www.pivotor.com

Voice: 813-435-2297
Mobile: 941-685-6789



z/OS Performance Education, Software, and Managed Service Providers

Creators of Pivotor®

Contact, Copyright, and Trademarks

Questions?

Send email to performance.questions@EPStrategies.com, or visit our website at https://www.epstrategies.com or http://www.pivotor.com.

Copyright Notice:

Trademarks:

Enterprise Performance Strategies, Inc. presentation materials contain trademarks and registered trademarks of several companies.

The following are trademarks of Enterprise Performance Strategies, Inc.: **Health Check®, Reductions®, Pivotor®**

The following are trademarks of the International Business Machines Corporation in the United States and/or other countries: IBM®, z/OS®, zSeries®, WebSphere®, CICS®, DB2®, S390®, WebSphere Application Server®, and many others.

Other trademarks and registered trademarks may exist in this presentation

Abstract

Back to Basics: Introduction to Parallel Sysplex and Datasharing

- This presentation will provide a comprehensive overview of parallel Sysplex in a z/OS environment. The attendee will learn the basic concepts of parallel Sysplex and data sharing. Covered in the presentation will be an introduction to coupling facility and its resources, coupling facility structures and how they are used, and exploiters of the coupling facility.

  Also covered in the presentation is how data sharing actually works. While billed as a rookie session, this presentation will even teach the seasoned z/OS professional a few new things.

EPS: We do z/OS performance…

- We are z/OS performance!

- Pivotor
  - Performance reporting and analysis of your z/OS measurements
  - Example: SMF, DCOLLECT, other, etc.
  - Not just reporting, but cost-effective analysis-based reporting based on our expertise

- Performance Educational Workshops (while analyzing your own data)
  - Essential z/OS Performance Tuning
  - Parallel Sysplex and z/OS Performance Tuning
  - WLM Performance and Re-evaluating Goals

- Performance War Rooms
  - Concentrated, highly productive group discussions and analysis

- MSU reductions
  - Application and MSU reduction

z/OS Performance workshops available

During these workshops you will be analyzing your own data!

- Essential z/OS Performance Tuning
  - March 20-24, 2023
- Parallel Sysplex and z/OS Performance Tuning
  - May 2-3, 2023
- WLM Performance and Re-evaluating Goals
  - October 2-6, 2023
- Also… please make sure you are signed up for our free monthly z/OS educational webinars! (email contact@epstrategies.com)

Like what you see?

- Free z/OS Performance Educational webinars!
  - The titles for our Fall 2022-2023 webinars are as follows:
    - ✓ *Key Reports to Evaluate z16 Processor Caches*
    - ✓ *Understanding System Recovery Boost's Impact on Performance and Performance Reporting*
    - ✓ *WLM Management of DDF Work: What can you do and what has changed?*
    - ✓ *Intensity! Understanding the Concepts and Usage of Intensity Measurements*
    - ✓ *High, Medium, Low: Understanding how HiperDispatch influences performance in z/OS*
    - ✓ *How and why Pivotor is different than other performance management reporters*
    - *Putting a lid on XCF*
    - *Key Reports to Evaluate Usage of Parallel Access Volumes*
    - *Key Reports to Evaluate Coupling Facility CPU Utilization*
    - *Understanding how memory management has evolved in z/OS*
  - Let me know if you want to be on our mailing list for these webinars

- If you want a free cursory review of your environment, let us know!
  - We're always happy to process a day's worth of data and show you the results
  - See also: http://pivotor.com/cursoryReview.html

EPS presentations this week

What	Who	When	Where
Evolution of z/OS Memory Management: Large Memory, Large Pages, and How to Use Them	Scott Chapman	Mon 1:15	Hanover D
Back to Basics – Introduction to Parallel Sysplex and Data Sharing	Peter Enrico	Wed 9:15	Hanover D
z/OS WLM - Revisiting Goals Over Time	Peter Enrico	Wed 10:30	Hanover D
PSP: z/OS Performance Tuning – Some Top Things You May Not Know	Peter Enrico Scott Chapman	Wed 1:15	Hanover D

Sysplex
and
Parallel Sysplex

Sysplex Checklist

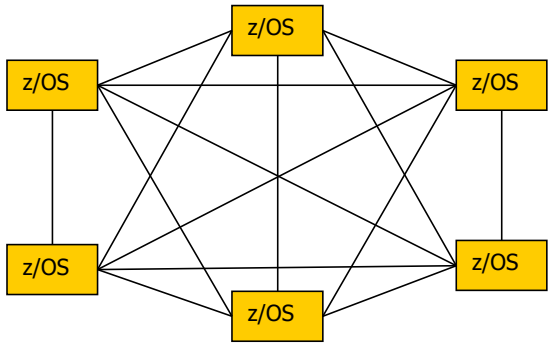
- Important Exercise!
  - Map out your coupling facility hardware and structures

  - What is your CF physical configuration?
  - What CF Link types are in use?
  - What structures are defined in each coupling facility?
    - List structures
    - Lock structures
    - Cache Structures
  - Which of these structures is duplexed, and what is placement of primary & secondary
  - What are the exploiters of each structure?

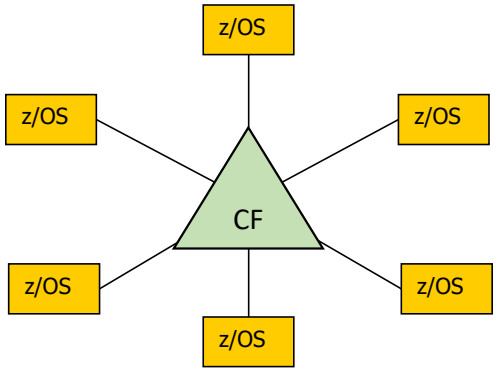
Sysplex concept

- A SYSPLEX (SYStem comPLEX) is a group of z/OS systems that cooperate (via software and sometimes hardware) to:
  - Simplify systems management (single system “image”)
  - Enhance and improve availability
  - Scale to larger usable capacity

Monoplex



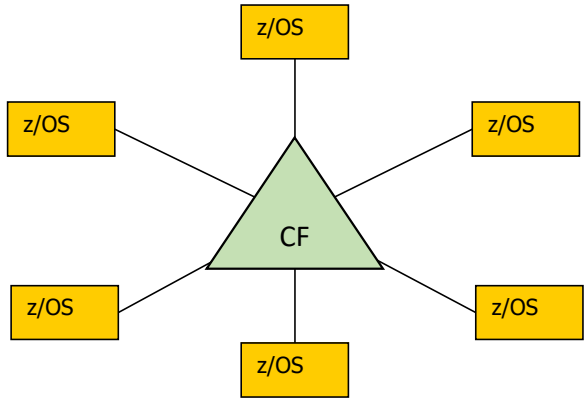
Base Sysplex



Parallel Sysplex

Parallel Sysplex

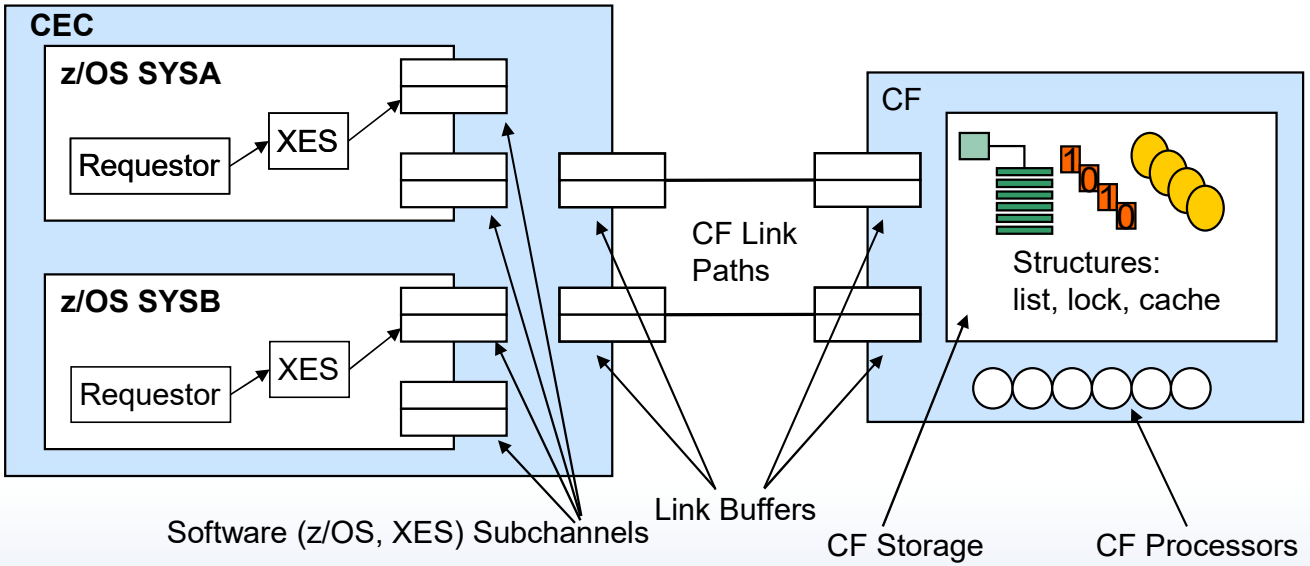
- Base Sysplex + Coupling Facility (CF)
  - CF Links required to connect z/OS to CF
  - STP typically done over same CF links
- CF used for
  - Communication
  - Lock management
  - Shared Cache
- Enables more efficient scalability
  - Linear scalability of signaling paths
  - Enables data sharing coordination by coupling facility
- Used for highest availability and scalability



Parallel Sysplex

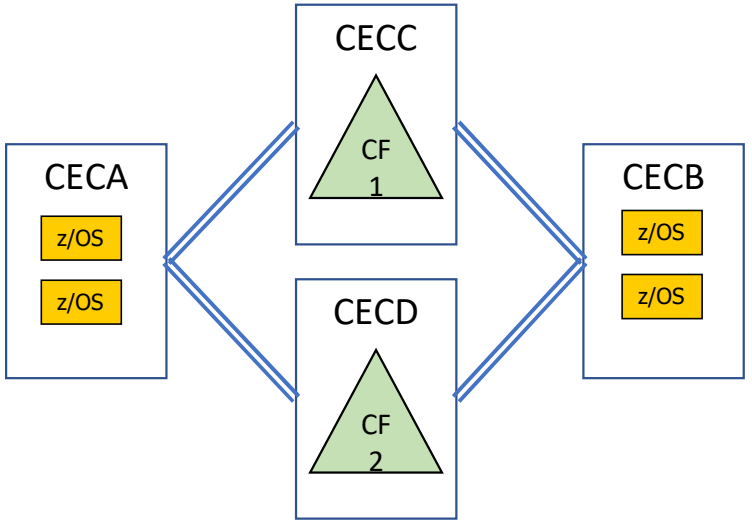
Performance view of CF Requests

- z/OS Processing
  - S/W processing to make CF request
  - Request a sub-channel
  - Request a path
  - Data transfer over link
  - On return, S/W processing to handle CF request
- Coupling Facility Processing
  - Link time (i.e. time on path)
  - CF busy processing request
  - Duplexing
  - List, Lock, Cache structure



Parallel Sysplex with External CFs

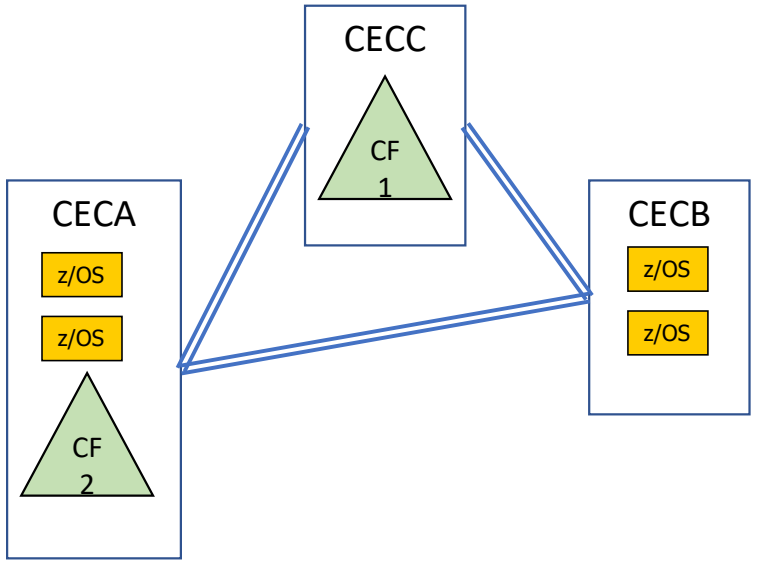
- "External" or "Standalone" Coupling Facilities dedicated to running CF LPARs
- This configuration was very common in the 1990s when CECs were more capacity-constrained and (slightly) less reliable
- No single point of failure from a processing perspective
  - Planned maintenance can be done non-disruptively as well



- Expense of external CFs typically limits their use to larger environments
  - I.E. likely larger than shown here
- More than 2 CFs can be used in a single Sysplex, but that's rare

Parallel Sysplex with 1 External CF

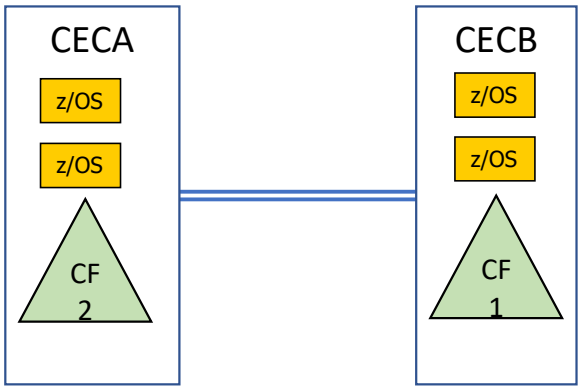
- Single external CF + Single internal CF LPAR
  - CF1 normally used, CF2 as backup
- No single point of failure from a processing perspective
  - Planned maintenance can be done non-disruptively as well
- Saves a bit of money compared to having 2 external CFs
- Connections from z/OS to CF on same CEC are internal links
  - Memory to memory transfers facilitated by microcode (no physical connection)



Parallel Sysplex with Internal CFs

- Dual Internal CFs: one on each CEC

- Can have a single point of failure, e.g. a single CEC failure could impact the sysplex
  - Dual failure of both the CF and the z/OS LPARs that would be needed to rebuild those CF structures
  - CF Structure duplexing is used to address this concern
  - Planned maintenance can be done non-disruptively

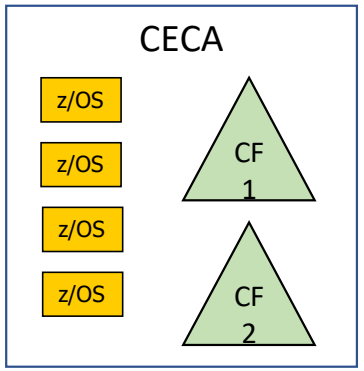


- Least expensive way to get to Parallel Sysplex High Availability without a single point of failure
  - Structure Duplexing does add overhead though

- Most common configuration in mid-size environments

Parallel Sysplex In A Box

- All LPARs (z/OS and CF) in a single CEC



- The CEC becomes a single point of failure
  - CF Structure duplexing would only help a CFCC code failure situation, not a CEC-wide failure
  - Planned maintenance can be done non-disruptively if there are two CF LPARs (highly recommended)

- Least expensive way to get to most of Parallel Sysplex availability benefits for planned maintenance
  - Although the CEC is a single point of failure, smaller sites that are looking to implement Parallel Sysplex for planned maintenance availability sometimes start here

Internal vs. External CFs & Duplexing

- The main goal of your parallel Sysplex should be to avoid the issues around dual failure scenarios
  - I.E. CEC failure that impacts both a CF and one or more z/OS systems using that CF
  - Unplanned CEC outages are generally rare, but can and do happen
- Internal CFs are cheaper but usually require some level of structure duplexing to avoid the dual failure scenarios
  - Because even if you lose the CF, there is a copy of the structures in the other CF
  - Duplexing involves a performance penalty
  - Duplexing lock structures is very expensive, may or may not be required
    - E.G. DB2 can recover from dual failure with a group restart, but that takes time
    - In personal experience, disabling lock duplexing for a busy application saved 5% of our total installed capacity
    - New Async Duplexing option on z14 may make lock duplexing palatable

Coupling links

- CF Link technology regularly changes
- Generally speaking, there are faster short-distance technologies vs. slower long-distance technologies

Link Speed MB/sec	IC	CS5 (ICA SR)	12x IFB3	12x IFB	CL5 (CE LR)	1x IFB3
zBC12	7100		4000	1000		400
zEC12	9400		5000	1000		400
z13s	7300	6000 (0-70 m) 3700 (70-150m)	4000	1000		400
z13	8500	6000 (0-70 m) 3700 (70-150m)	5000	1000	700	400
z14 Model ZR1	7600	6000 (0-70 m) 3700 (70-150m)	4000	1000	700	NA
z14	8900	6000 (0-70 m) 3700 (70-150m)	5000	1000	700	400
z15 T02	7600	6000 (0-70 m) 3700 (70-150m)	NA	NA	700	NA
z15	8900	6000 (0-70 m) 3700 (70-150m)	NA	NA	700	NA

Coupling Facility Request Types

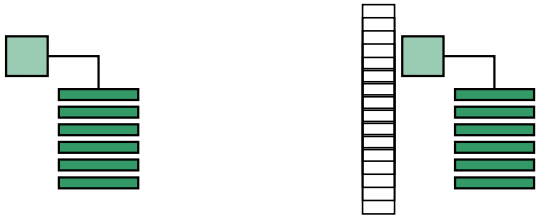
Coupling Requests

- Synchronous requests
  - Fastest: response time as low as 2-3 microseconds (µs)
  - CPU waits for response to come back from CF
    - Sometimes called “spinning” or “dwelling”
    - Consumes CPU capacity while CF request is processing
- Asynchronous requests
  - Slower: response time can be low 100s of microseconds
  - Task goes to wait and CPU used for some other task
    - CPU capacity not consumed while CF request is processing
    - Upon request completion interrupt raised and task has to be re-dispatched
  - Similar to a cached I/O (although generally faster)
- XES heuristic algorithm will convert slow sync requests to async

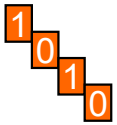
Introduction to Coupling Facility Structures

Introduction to Coupling Facility Structures

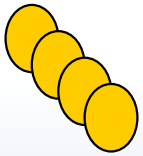
- Data is organized in the coupling facility in one of four different structure types
  - List structures – Simple (un-serialized) and Serialized
    - When data needs to be organized into lists, queues, stacks



  - Lock structures
    - When serialization is required



  - Cache structures
    - When data needs to cached
    - When buffer validation is required

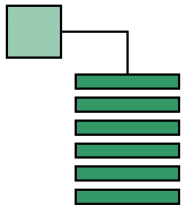


Introduction to List Structures

List Structures

- CF can be used to store data organized into list structures

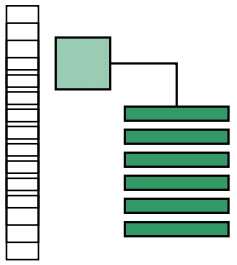
- List structure made up of
  - List entries
  - List elements
  - Optional lock table



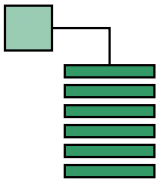
Simple List Structure

- List structures can be organized into
  - FIFO queues
  - Push / Pop structures
  - Static lists

- Uses for list structures include
  - high speed message routing
  - distributing work requests among Sysplex members (as in shared work queues)
  - Maintain shared information such as status



Serialized List Structure



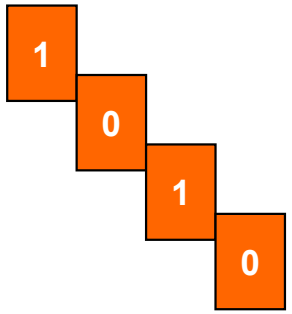
List Structure Exploiters

- XCF signaling
- JES2 Checkpoint data set
- Operlog – Shared operations log stream
- Logrec – Shared Logrec log stream
- VTAM
  - Generic Resources
  - MNPS - Multi-Node Persistent Sessions
- RRS – shared log stream
- SmartPipes – a.k.a BatchPipes
- MQSeries
- Intelligent Resource Director (IRD)
- WLM multi-system enclaves
- DFSMShsm Common Recall Queue
- TCP/IP
  - system-wide security associations
  - TCP/IP Sysplexports
- CICS
  - Shared primary and second system logs
  - Shared journals
  - Forward recovery logs
  - Temporary Storage Queue Pool
  - Named Counter Server
- DB2
  - Shared Communication Area (SCA)
- IMS
  - Shared IMS log
    - Forward Recovery logs
  - Shared message queues

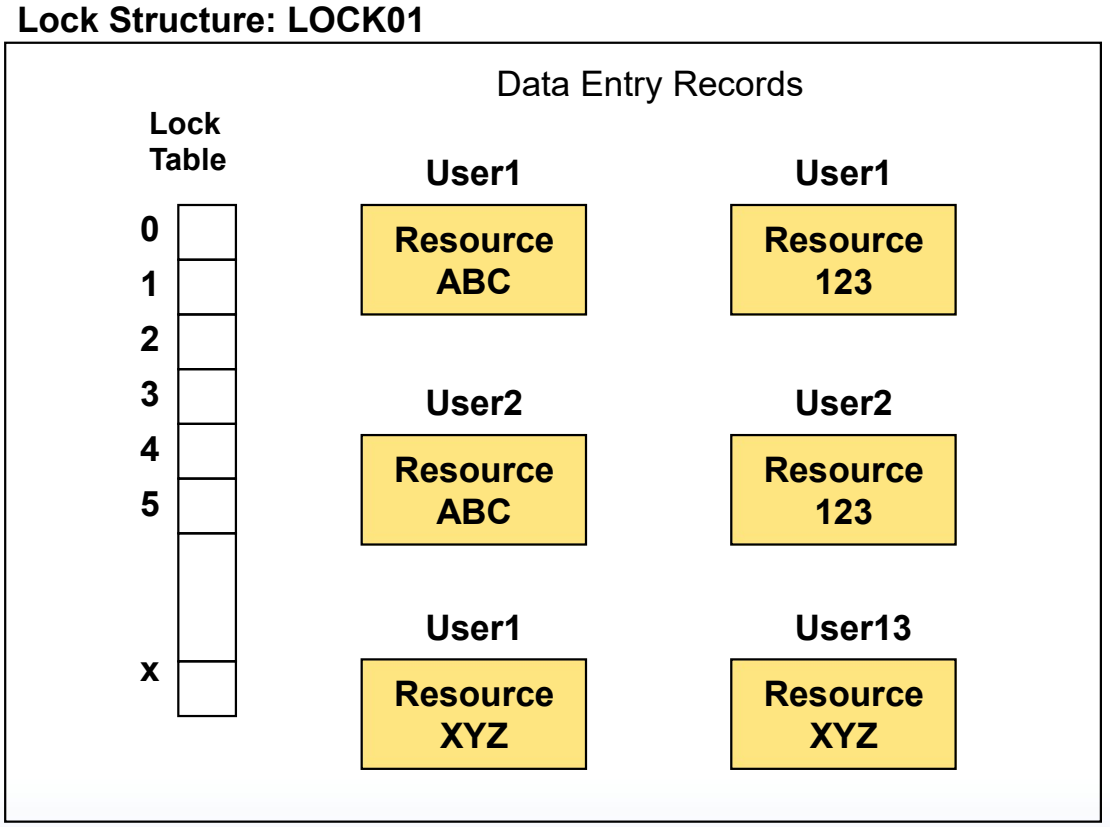
Introduction to Lock Structures

Lock Structures

- CF can be used as a high-speed locking facility by using lock structures
  - Lock structures are centralized lock tables maintained in the CF
- Lock structure made up of
  - Lock table containing information about the serialized resource
  - Lock record containing information about connected users
- Lock structures support
  - Shared lock state
  - Exclusive lock state
  - Application defined lock state
- Uses for lock structures include
  - Synchronous resource serialization
  - Resource contention detection

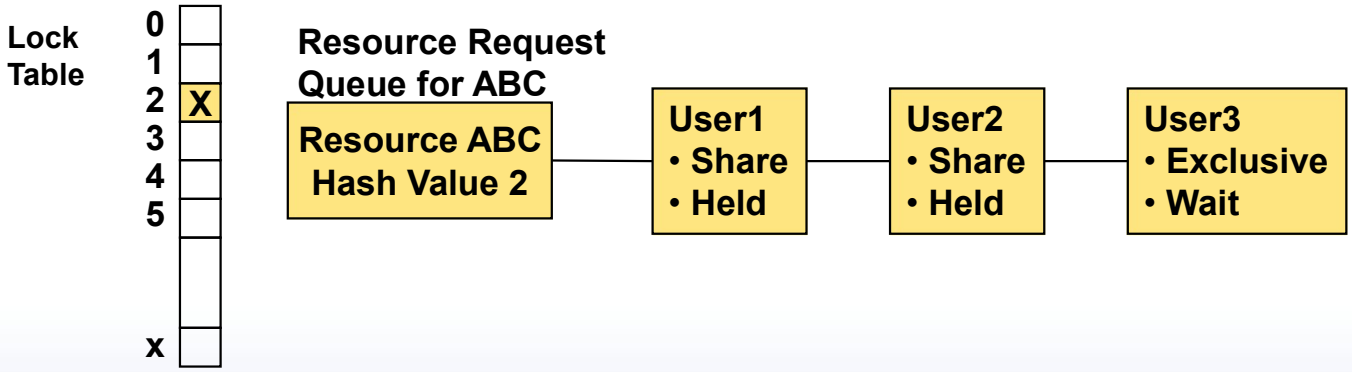


Lock Structure Components



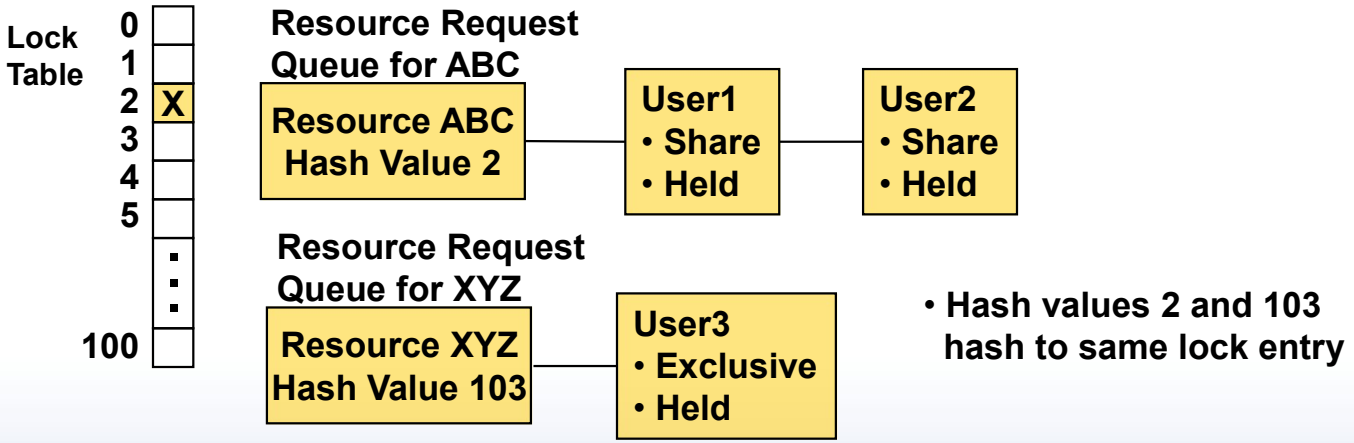
Types of Lock Contention

- Real Lock Contention
  - Contention caused by multiple units of work attempting to serialize on the same resource
  - Factors that influence real lock contention
    - How the locks are being used
    - Amount of time locks are held
    - Degree of data sharing
  - Alleviate real lock contention by tuning the workload (not by tuning the Sysplex or CF structures)



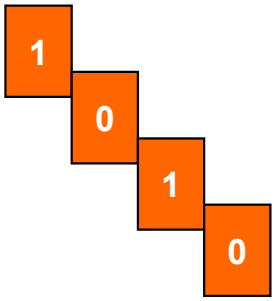
Types of Lock Contention

- False Lock Contention
  - When multiple lock names are hashed to the same lock entry
    - Results in significant excessive processing overhead to resolve
  - Factors that influence false lock contention
    - Size of lock structure
    - Granularity of locking (record, file, block)
    - Concurrent users connected to lock structure
  - Alleviate false lock contention by increasing lock structure size



Lock Structure Exploiters

- GRS Star topology

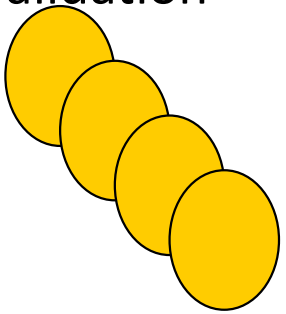


- DB2
  - IRLM lock manager
- VSAM RLS
  - Cross system contention handler (locking)
- IMS
  - IRLM lock manager

Introduction to Cache Structures

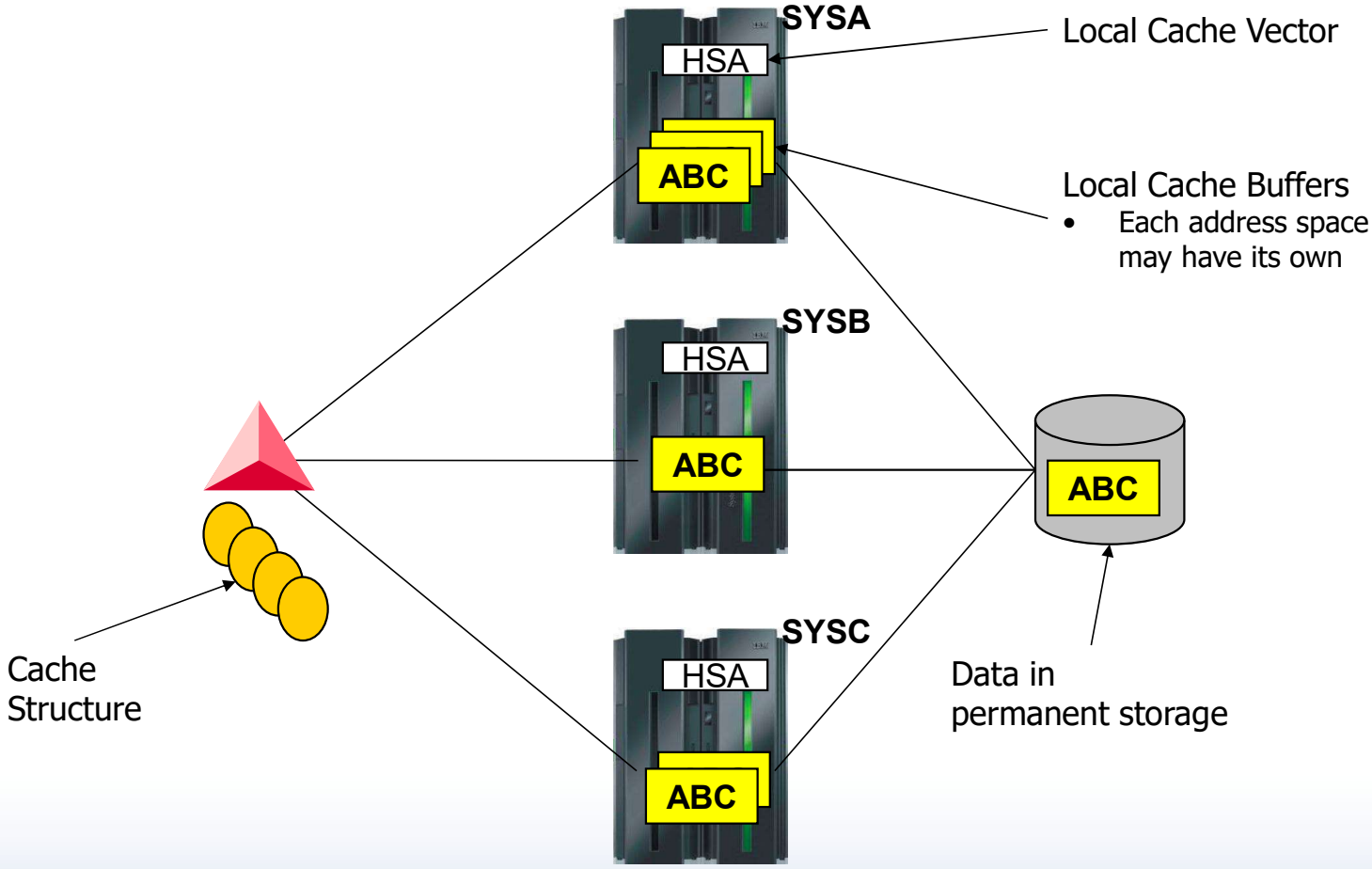
How Data Sharing Works!

Cache Structures



- CF can be used as a high-speed caching facility and buffer validation
- Cache structure made up of
  - directory to keep track of registered data elements
  - optionally, data elements
- Usage of cache structure
  - data consistency / buffer validation
    - ability to maintain a shared copy of data in cache structure in CF
    - ability to keep track of shared data that does not reside in CF
      - permanent storage (i.e. disk)
      - local storage (i.e. z/OS or subsystem buffers)
  - high speed data access
    - Shared data can be stored in cache structure and made available to every system in sysplex
    - Invalid local copy of data can be refreshed with CF cached copy
    - CF access faster than I/O subsystem cache

Cache Structure Components



Cache Structure Terminology

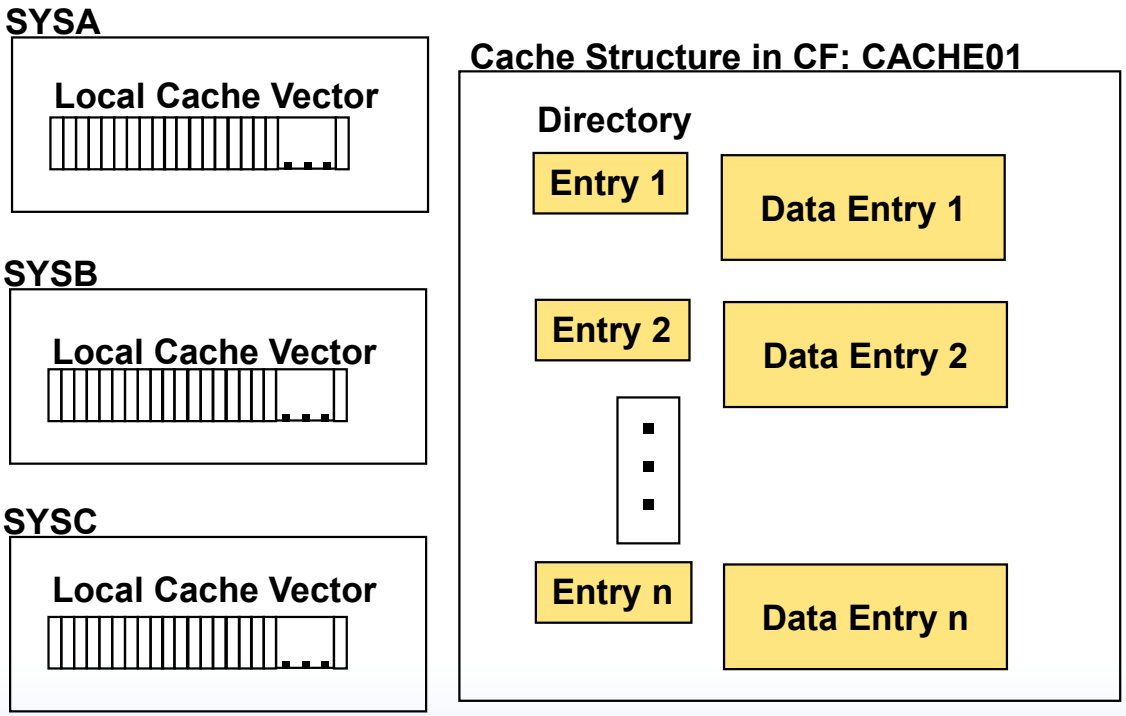
- The cache structure in the coupling facility has two primary components

- Directory Entries
  - Used to keep track of data entries that are shared among multiple systems
  - Every system that has a copy of a particular piece of shared data has a registration entry in this portion of the cache structure.
  - It is this directory whose entries are used to generate cross invalidation signals to indicate that a record in a local cache buffer may be invalid

- Data Entries
  - Used to contain a cached version of the data
  - Optional

Cache Structure Components cont...

- Directory - Used to keep track of share entries
- Data Entries - Used to optionally cache data



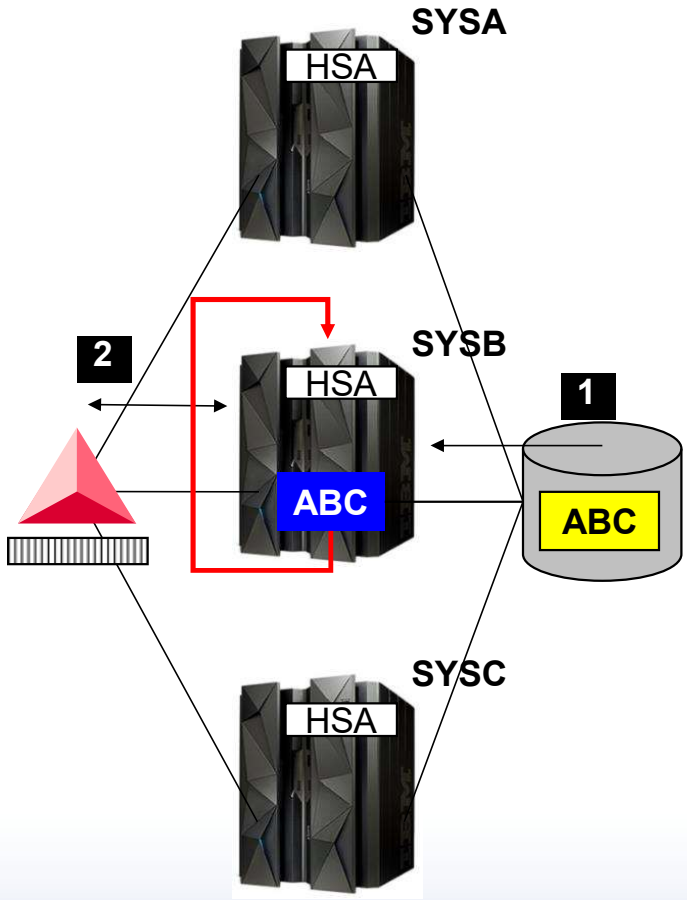
Parallel Sysplex Data Sharing – High Level

1) System SYSB reads record ABC in
   - Assume this the very first time that ABC is accessed
   - SYSB first determines there is no version of ABC in the CF
2) SYSB registers its interest in ABC into the coupling facility
   - ☐ Directory entry in associated cache structure
   - ☐ The HAS of SYSB indicates SYSB has most up-to-date version of the data
   - ☐ Typically the data is not written into the coupling facility upon read

Summary:

Current state: SYSB has read ABC into local memory and has registered interest in ABC in the coupling facility

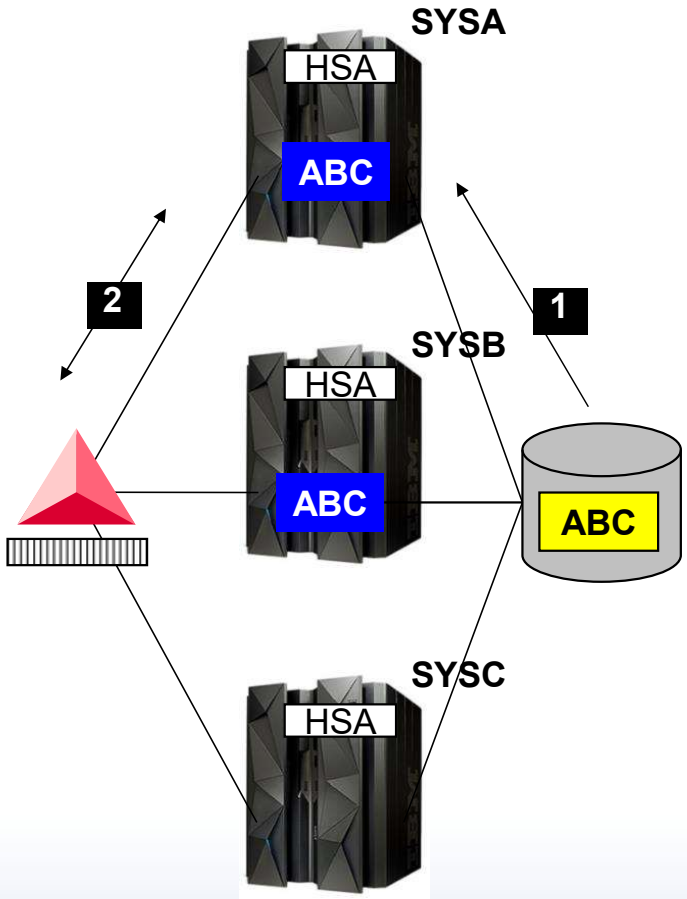
Whenever SYSB references this data, it checks the HSA to determine if the data is the most current version.



Parallel Sysplex Data Sharing – High Level

1) System SYSA reads record ABC in
   - Assume this the very first time that ABC is accessed by SYSA
   - SYSA first determines there is no version of ABC in the CF
   - SYSA then reads ABC in from disk into local memory

2) SYSA registers its interest in ABC into the coupling facility
   - ☐ Directory entry in associated cache structure
   - ☐ Typically the data is not also written into the coupling facility upon read

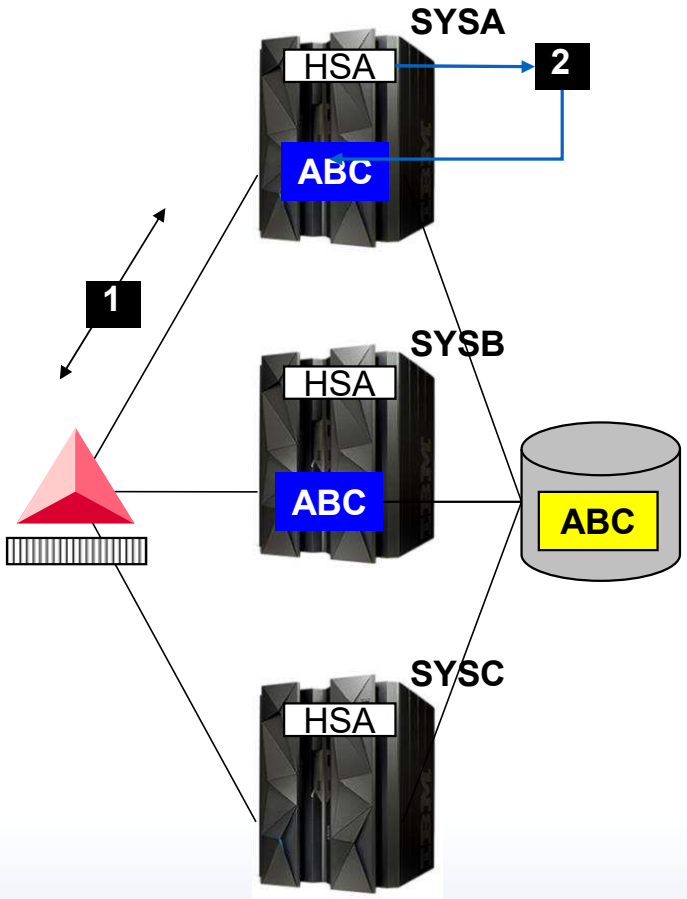


Summary:

Current state: Both systems SYSA and SYSB have read ABC into local memory and has registered their interest in ABC in the coupling facility

Parallel Sysplex Data Sharing – High Level

1) Via a Coupling Facility lock structure, SYSA requests an exclusive lock to allow SYSA to update ABC
   - Assume SYSA gets the exclusive lock
2) System SYSA now wants to make an update to ABC
   - SYSA uses local cache vector table in hardware HSA to determine if ABC in local buffer is valid
   - In this case, SYSA determines that its version of ABC is the most recent version
   - If it were not valid then SYSA would have had to re-read in the data from disk or CF

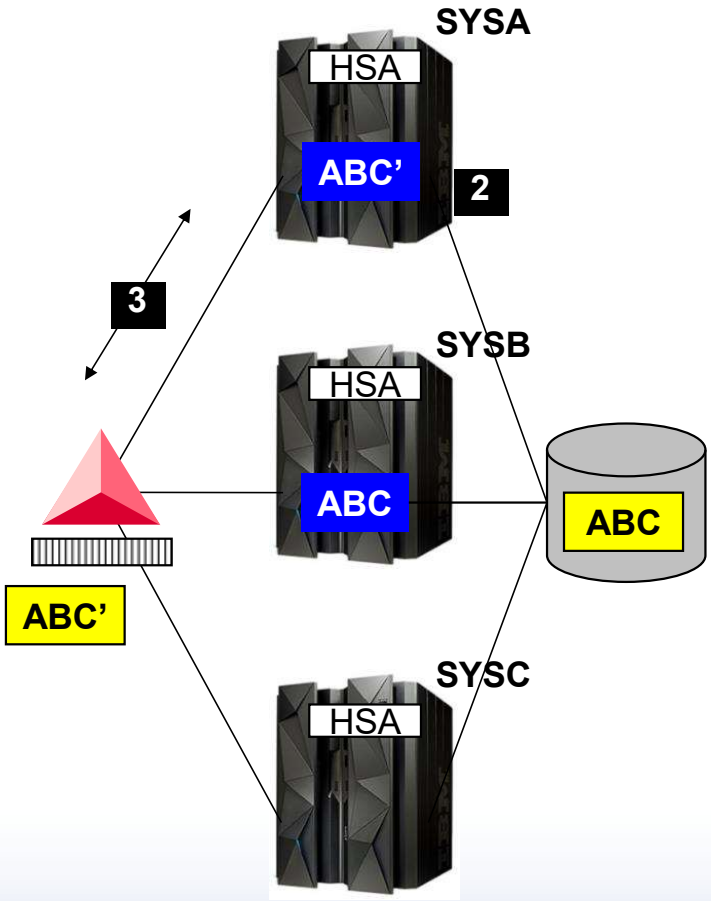


Parallel Sysplex Data Sharing – High Level

1) SYSA can now updates ABC to ABC'
   - SYSA is allowed to do this because it has the exclusive lock for ABC

2) SYSA changes local copy of ABC to ABC'
   - Local cache buffer

3) The change must be duplicated in case SYSA goes down
   - How and where it is duplicated is dependent on the type of cache structure defined and associated with ABC
     - CF cache structure and async later to disk (if store-in algorithm) (*this example)
     - CF cache structure and disk (if store-through algorithm)
     - Disk only (if directory only algorithm)



Parallel Sysplex Data Sharing cont...

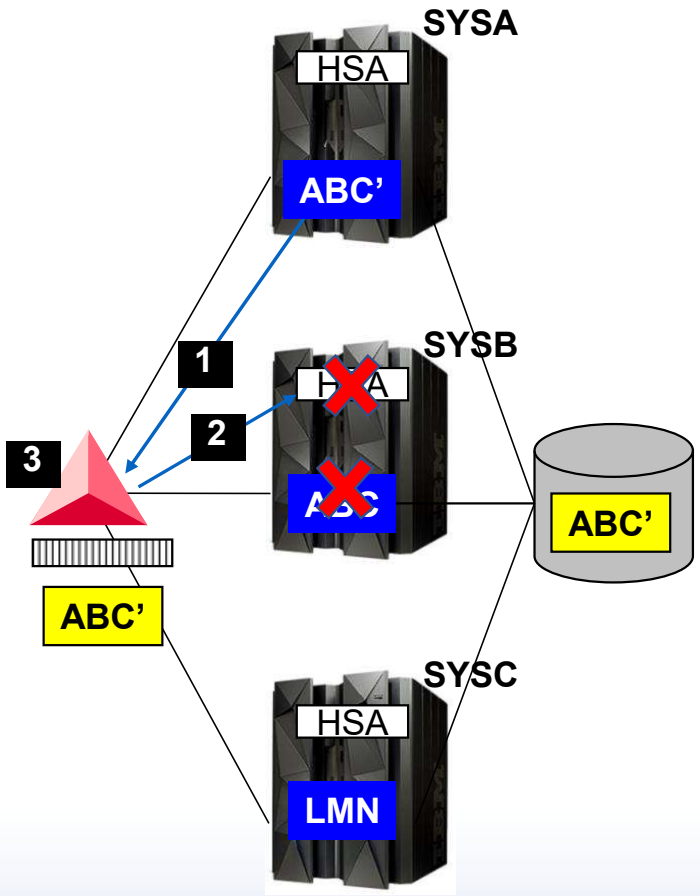
1) Signal sent by database manager to CF to indicate that record ABC has been updated
   - CF cache structure updated so all systems will know ABC has been updated

6) CF invalidates all the local buffers for ABC
   - In this case on SYSB
   - It does this by setting a bit in the local cache vector in the HSA
   - This *cross invalidation* is done with no interrupts to other systems

7) Update is now complete and serialization of record ABC is now released
   - This is known as *lock release*

- Next time SYSB attempts to access record ABC it will know to get the fresh copy, ABC', from CF or disk

- Next time SYSA attempts to access record ABC' it will know it already has the latest copy in its buffers



Sysplex Checklist

- Important Exercise!
  - Map out your coupling facility hardware and structures

  - What is your CF physical configuration?
  - What CF Link types are in use?
  - What structures are defined in each coupling facility?
    - List structures
    - Lock structures
    - Cache Structures
  - Which of these structures is duplexed, and what is placement of primary & secondary
  - What are the exploiters of each structure?