Back to Basics Introduction to Parallel Sysplex and Data Sharing

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

Creators of Pivotor®

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

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…

- **Pivotor** – z/OS performance reporting and analysis software and services
  - Not just SMF reporting, but analysis-based reporting based on expertise
  - www.pivotor.com

- **Education and instruction**
  - We teach our z/OS performance workshops all over the world
  - Want a workshop in your area? Just contact me.

- **z/OS Performance War Rooms**
  - Intense, concentrated, and highly productive on-site performance group discussions, analysis and education
  - Amazing feedback from dozens of past clients

- **Information**
  - We present around the world and participate in online forums
  - https://www.pivotor.com/content.html
    https://www.pivotor.com/webinar.html

z/OS Performance workshops available

During these workshops you will be analyzing your own data!

- WLM Performance and Re-evaluating Goals
  - February 19-23, 2024

- Parallel Sysplex and z/OS Performance Tuning
  - August 20-21, 2024

- Essential z/OS Performance Tuning
  - October 7-11, 2024

- 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 Summer / Fall 2024 webinars are as follows:
    - ✓ *What a z/OS Guy Learned About AWS in 10 Years*
    - ✓ *Advantages of Multiple Period Service Classes*
    - ✓ *Understanding z/OS Connect Measurements*
    - *WLM and SMF 99.1 – System Measurements Deeper Dive*
    - *WLM and SMF 99.2 – Service Class Period Measurements Deeper Dive*
    - *Optimizing Performance at the Speed of Light: Why I/O Avoidance is Even More Important Today*
    - *Understanding MVS Busy % versus LPAR Busy % versus Physical Busy %*
    - *Rethinking IBM Software Cost Management Under Tailored Fit Pricing*
    - *Understanding Page Faults and Their Influence on Uncaptured Time*
    - *Response Time Goals: Average or Percentiles?*
    - *Understanding and Using Enclave*

- 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

Like what you see?

- The z/OS Performance Graphs you see here come from Pivotor
- If you don't see them in your performance reporting tool, or you just want a free cursory performance 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
- We also have a free Pivotor offering available as well
  - 1 System, SMF 70-72 only, 7 Day retention
  - That still encompasses over 100 reports!

All Charts (132 reports, 258 charts)
All charts in this reportset.
Charts Warranting Investigation Due to Exception Counts (2 reports, 6 charts, more details)
Charts containing more than the threshold number of exceptions
All Charts with Exceptions (2 reports, 8 charts, more details)
Charts containing any number of exceptions
Evaluating WLM Velocity Goals (4 reports, 35 charts, more details)
This playlist walks through several reports that will be useful in while conducting a WLM velocity goal an

EPS presentations this week

What	Who	When	Where
60 Years of Pushing Performance Boundaries with the Mainframe	Scott Chapman	Sun 17:00	Neptune D
Introduction to Parallel Sysplex and Data Sharing	Peter Enrico	Mon 13:15	Pomona
Macro to Micro: Understanding z/OS Performance Moment by Moment	Scott Chapman	Mon 15:45	Neptune D
WLM Turns 30! : A Retrospective and Lessons Learned	Peter Enrico	Tue 10:30	Neptune D
PSP: z/OS Performance Spotlight: Some Top Things You May Not Know	Peter Enrico Scott Chapman	Tue 13:00	Pomona
More/Slower vs. Fewer/Faster CPUs: Practical Considerations in 2024	Scott Chapman	Tue 14:15	Neptune D
z16 SMF 113s – Understanding Processor Cache Counters	Peter Enrico	Wed 13:15	Pomona

Sysplex and Parallel Sysplex

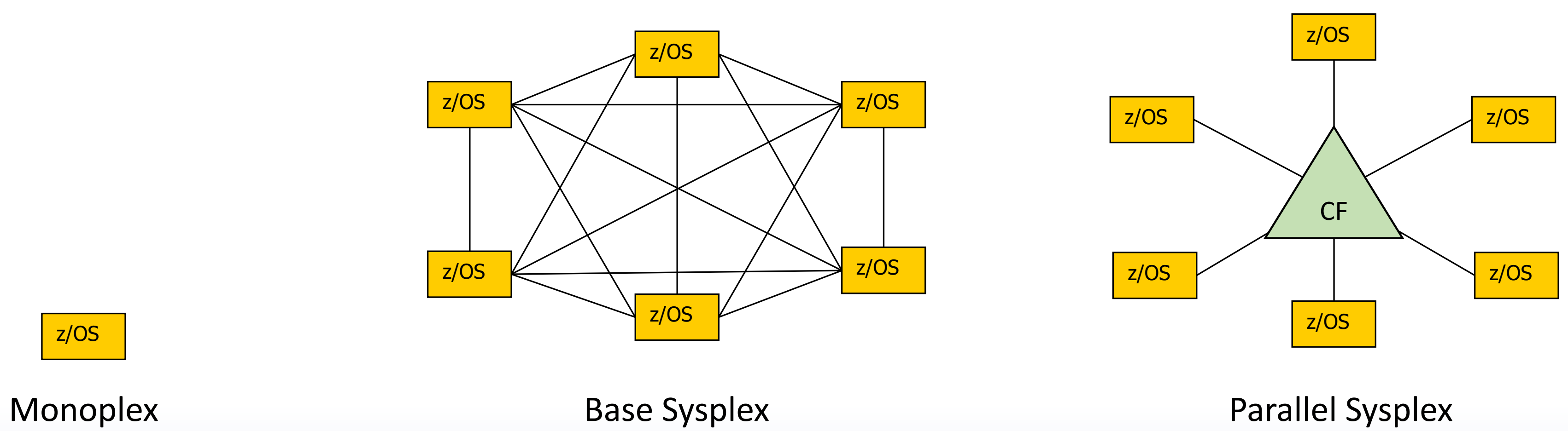
Sysplex Checklist

- 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?

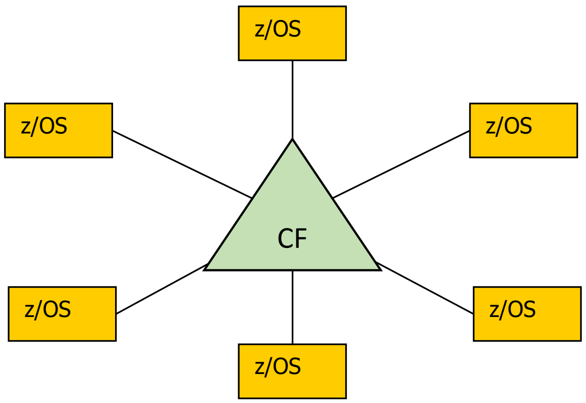
Sysplex concept

- 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



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

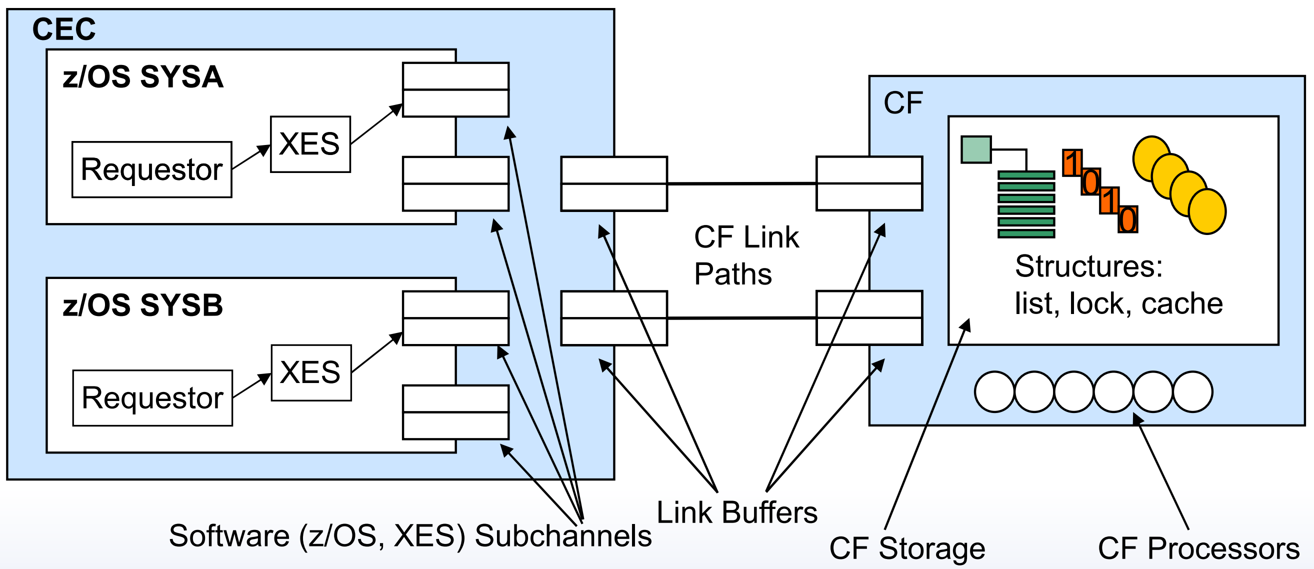
Exploiting Parallel Sysplex

- Parallel Sysplex often implemented for a particular reason
  - E.G. DB2 Data Sharing
- With a parallel Sysplex though, you can exploit the CF in other areas as well
  - Can increase performance (often by reducing DASD I/O)
  - Can simplify operations (e.g. reducing/eliminating XCF CTC links)
  - Can improve availability (e.g. by spreading work across multiple systems)

- If you have a Coupling Facility, use it!

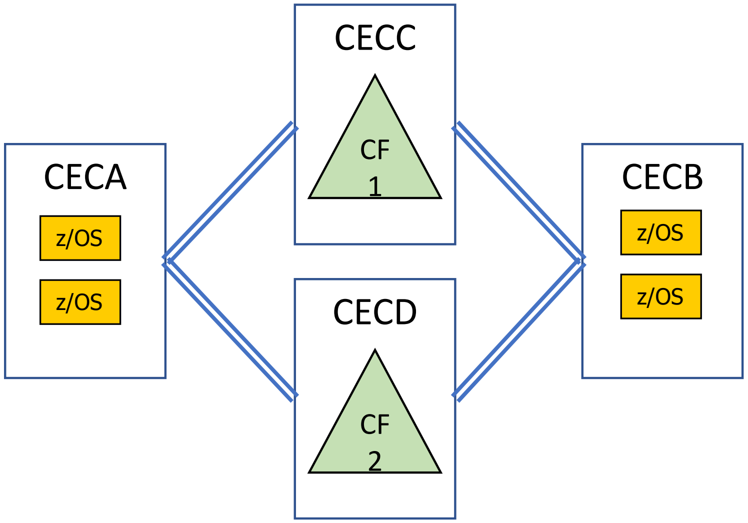
Performance view of CF Requests

- 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



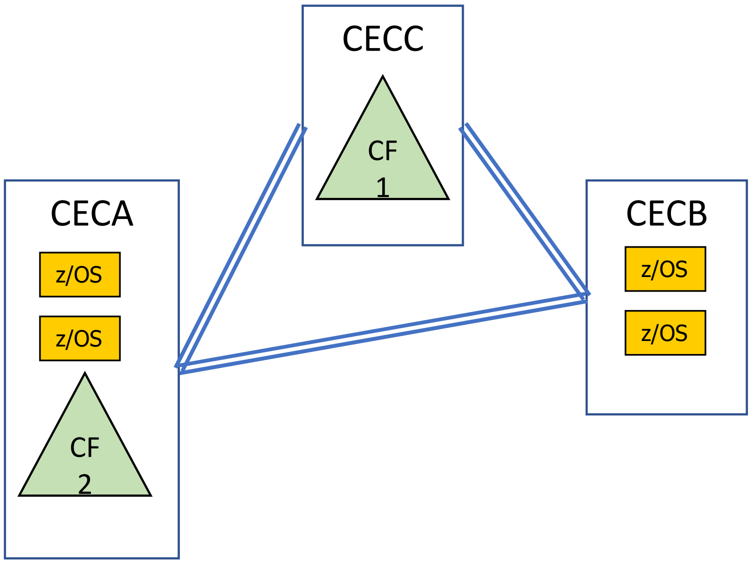
Parallel Sysplex with External CFs

- "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



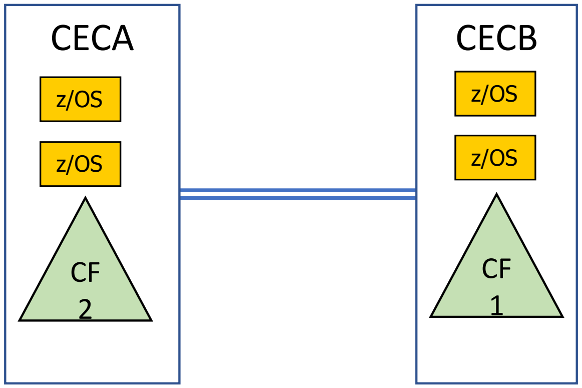
Parallel Sysplex with 1 External CF

- 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

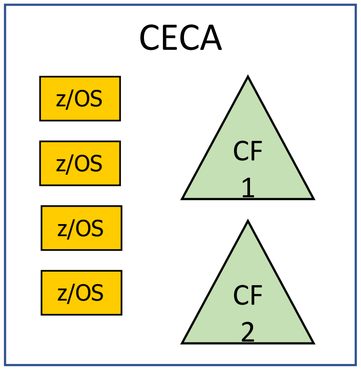
- Dual Internal CFs: one on each CEC
- Can have a single point of failure, e.g. a single CEC failure could impact the sysplex
  - Dual failure of both the CF and the z/OS LPARs that would be needed to rebuild those CF structures
  - CF Structure duplexing is used to address this concern
  - Planned maintenance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

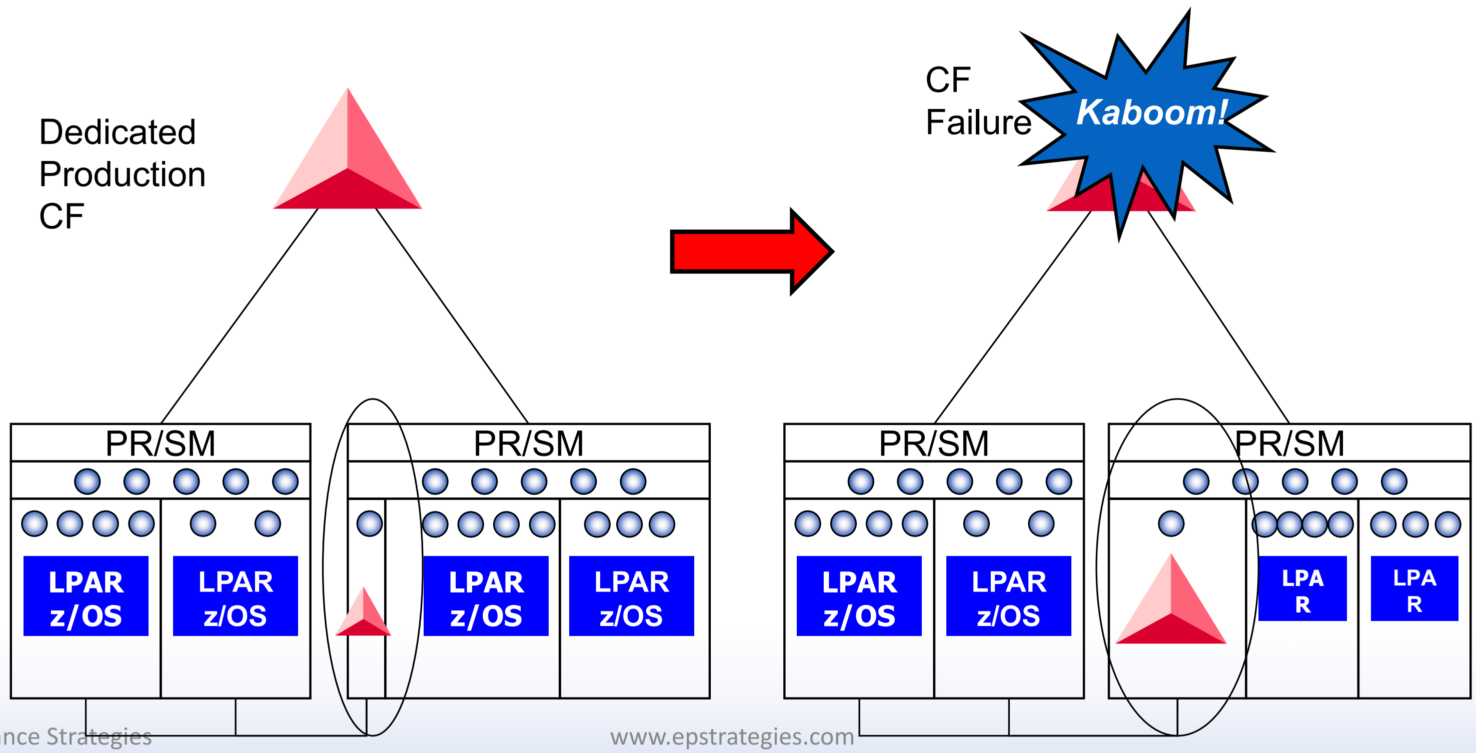
- The CEC becomes a single point of failure
  - CF Structure duplexing would only help a CFCC code failure situation, not a CEC-wide failure
  - Planned maintenance can be done non-disruptively if there are two CF LPARs (highly recommended)



- 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

CF with CP Engines

- Some installations do use CP engines for coupling links
  - MSUs consumed count towards pricing
  - Variety of reasons why customers do this. One example:



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

- IC
  - Internal Coupling Link
  - Memory to memory
- ICA SR (CS5
  - Integrated Coupling Adapter
  - Short reach links
- 12x IFB3
  12x IFB
  1x IFB3
  - Coupling over Infiniband
  - Longer reach links
- CL5 (CE LR)
  - Coupling Express
  - Longer reach links

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

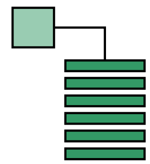
Coupling Requests

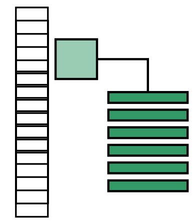
- Synchronous requests
  - Fastest: response time as low as 2-3 microseconds (µs)
  - CPU waits for response to come back from CF
    - Sometimes called “spinning” or “dwelling”
    - Consumes CPU capacity while CF request is processing
- Asynchronous requests
  - Slower: response time can be low 100s 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

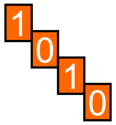
Introduction to Coupling Facility Structures

- 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

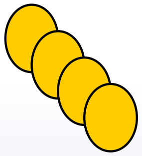




  - Lock structures
    - When serialization is required



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

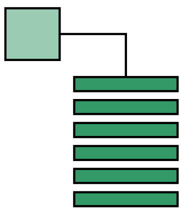


Introduction to List Structures

List Structures

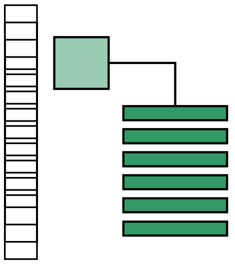
- CF can be used to store data organized into list structures

- 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



Serialized List Structure

- 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

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

Examples of List Structure Exploiter

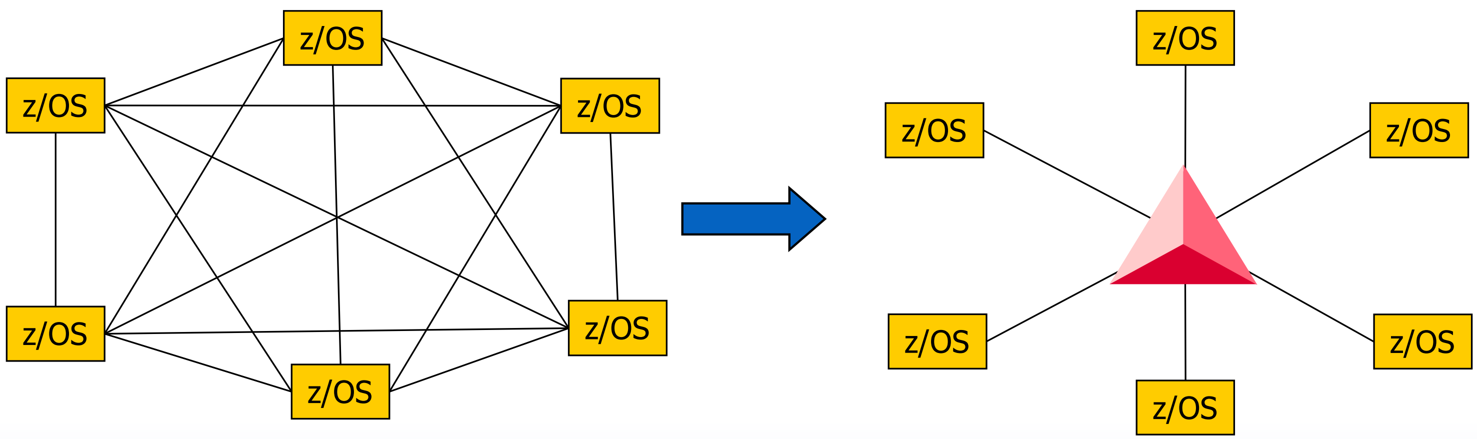
- XCF
  - Uses an un-serialized list structures to assist in message delivery
  - When System A wants to send a message to System B

- JES2 Checkpoint
  - Uses a serialized list structure for its JES2 checkpoint data set structure
  - The JES2 checkpoint function performs two separate functions:
    - Job and output queue backup to ensure ease of JES2 restart
    - Multi-access spool (MAS) member-to-member workload communication to ensure efficient independent JES2 operations

- z/OS System Logger
  - Uses un-serialized list structures for duplexing logstream log records before writes out to secondary storage
  - Examples include:
    - CICS Transaction logs (DFHLOG, DFHSHUNT, and DFHLGLOG), and User journals
    - RRS (Resource Recovery Services) for general logging and shared logs for recovery
    - OPERLOG and Logrec for merged operations log streams for all systems
    - SMF
    - Etc.

XCF Signaling in a Parallel Sysplex

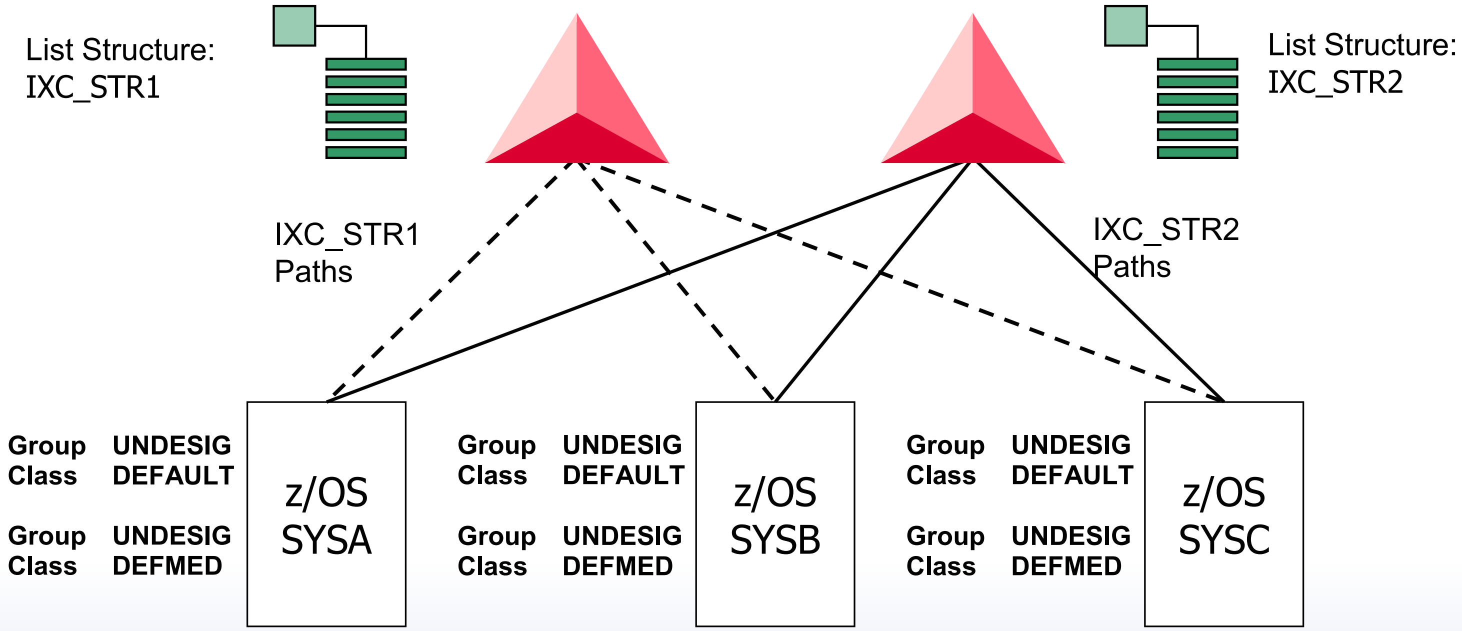
- XCF exploiters benefit from XCF usage of CF
  - XCF messages stored in CF rather than passed from each system to every other system
  - Works as well as CTCs but CF provides vastly improved scalability



How XCF Signaling in CF Works

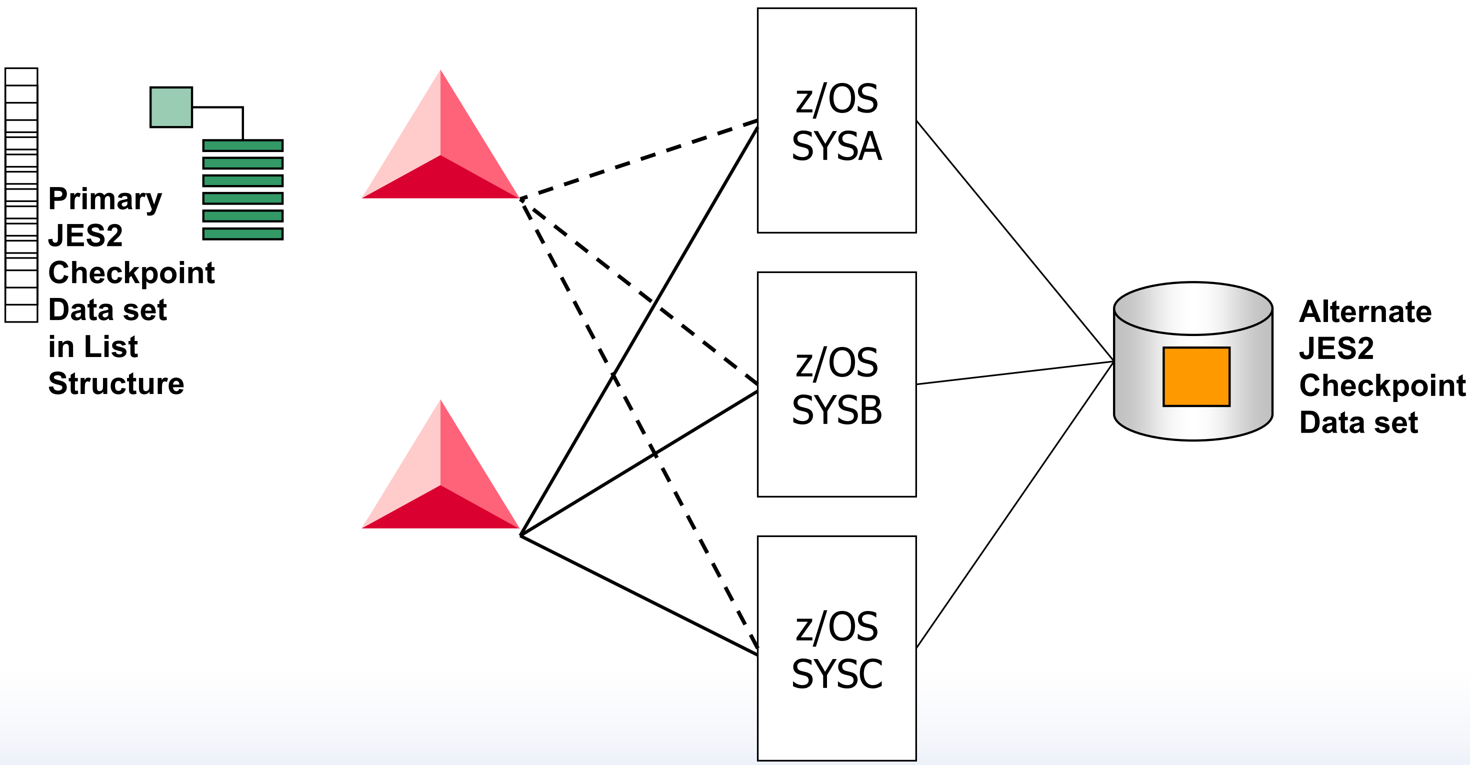
- XCF messages are stored in list structures within the CF
  - Example: Each system COUPLExx member specifies the following

```
PATHOUT STRNAME(IXC_STR1,IXC_STR2)
CLASSDEF CLASS(DEFAULT) CLASSLEN(956)   GROUP(UNDESIG)
CLASSDEF CLASS(DEFMED)  CLASSLEN(16316) GROUP(UNDESIG)
```



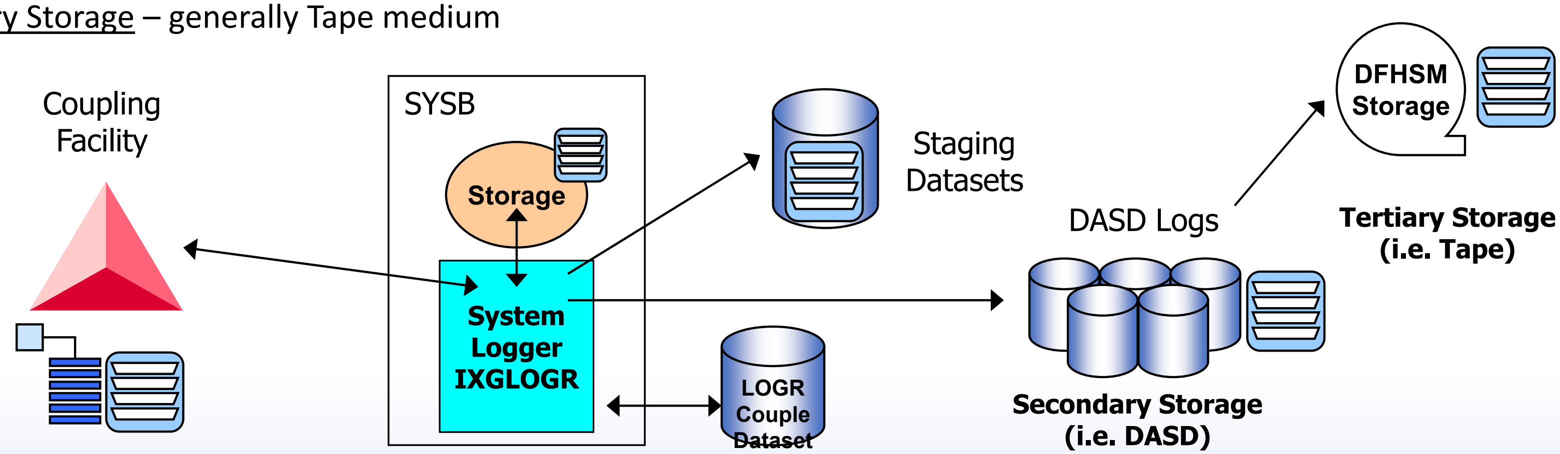
JES2 Checkpoint Data Set in Parallel Sysplex

- JES2 can optionally use a CF list structure for primary checkpoint data set
  - If use JES2 checkpoint structure, make sure alternate checkpoint is still on DASD



Introduction to z/OS System Logger

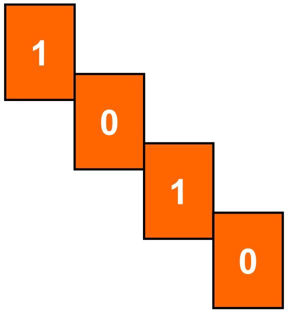
- z/OS System Logger - Component of z/OS that provides logging services
  - IXGLOGR – key system address space for logger functions
  - Interim Storage - Primary storage used to hold the log data that has not yet been offloaded
    - What 'interim storage' is depends on how the log stream has been setup
    - Examples of include central storage (via a data space), Coupling Facility or Staging data sets
  - Secondary Storage - generally DASD
  - Tertiary Storage – generally Tape medium



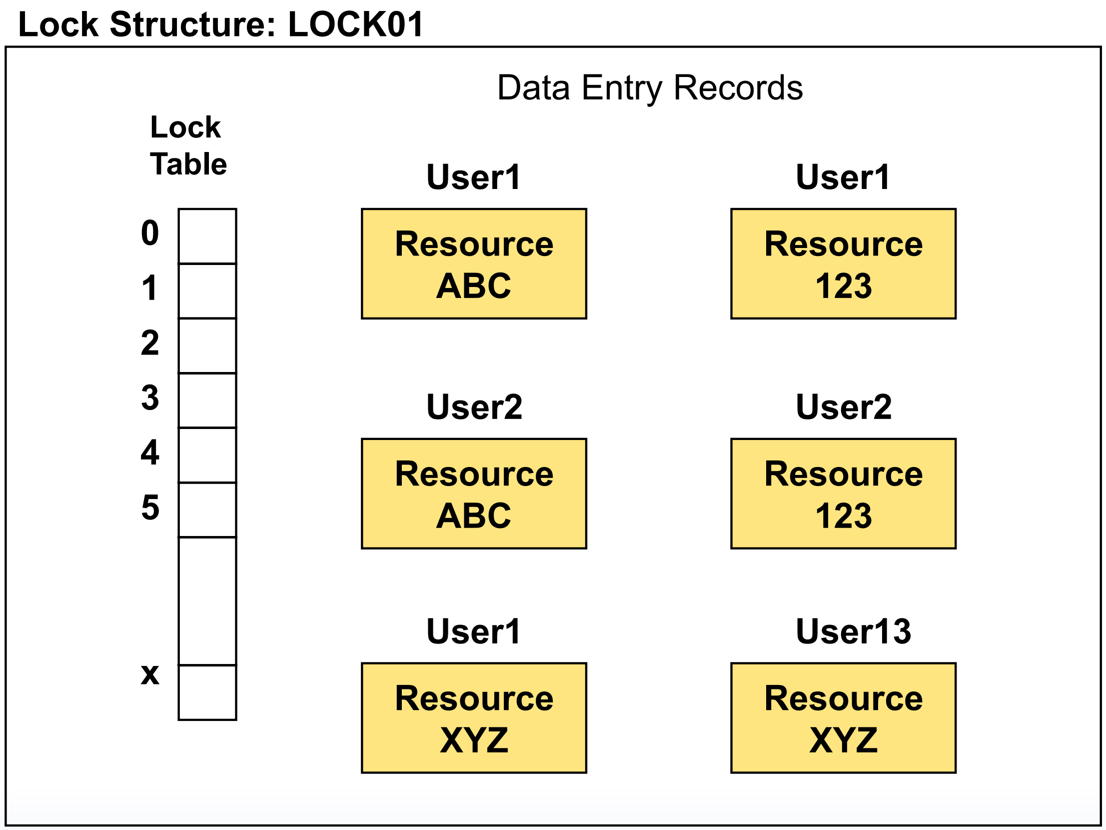
Introduction to Lock Structures

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

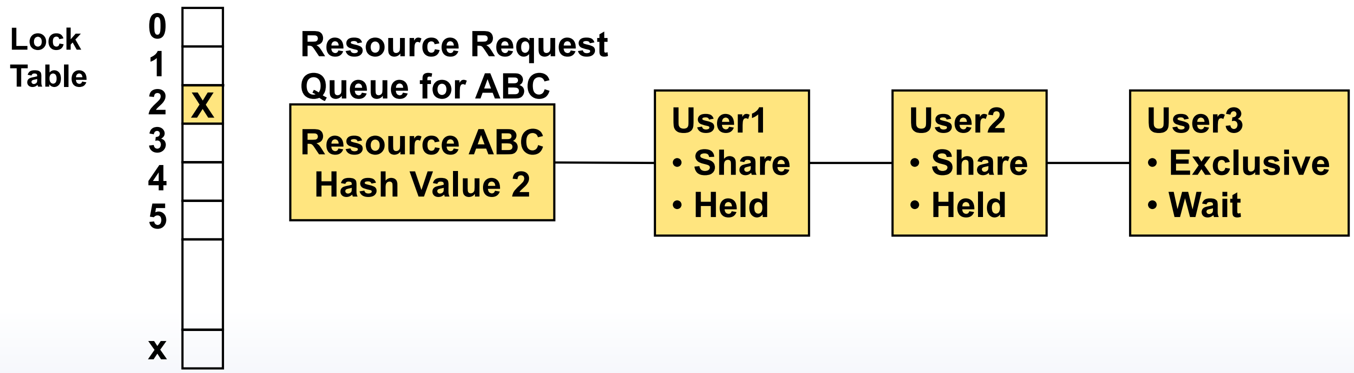


Types of Lock Contention

- Real Lock Contention
  - Contention caused by multiple units of work attempting to serialize on the same resource

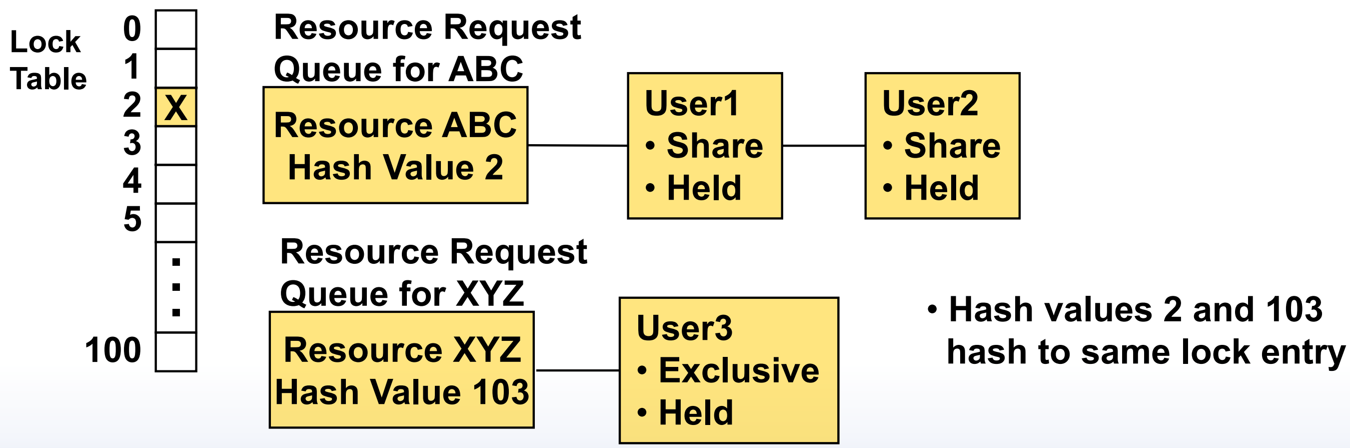
  - Factors that influence real lock contention
    - How the locks are being used
    - Amount of time locks are held
    - Degree of data sharing

  - Alleviate real lock contention by tuning the workload
    (not by tuning the Sysplex or CF structures)



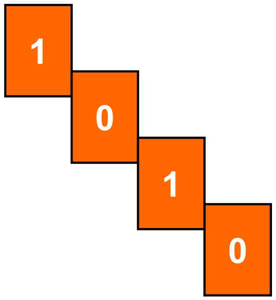
Types of Lock Contention

- 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

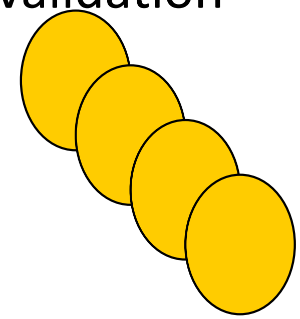


Introduction to Cache Structures

How Data Sharing Works!

Cache Structures

- CF can be used as a high-speed caching facility and buffer validation

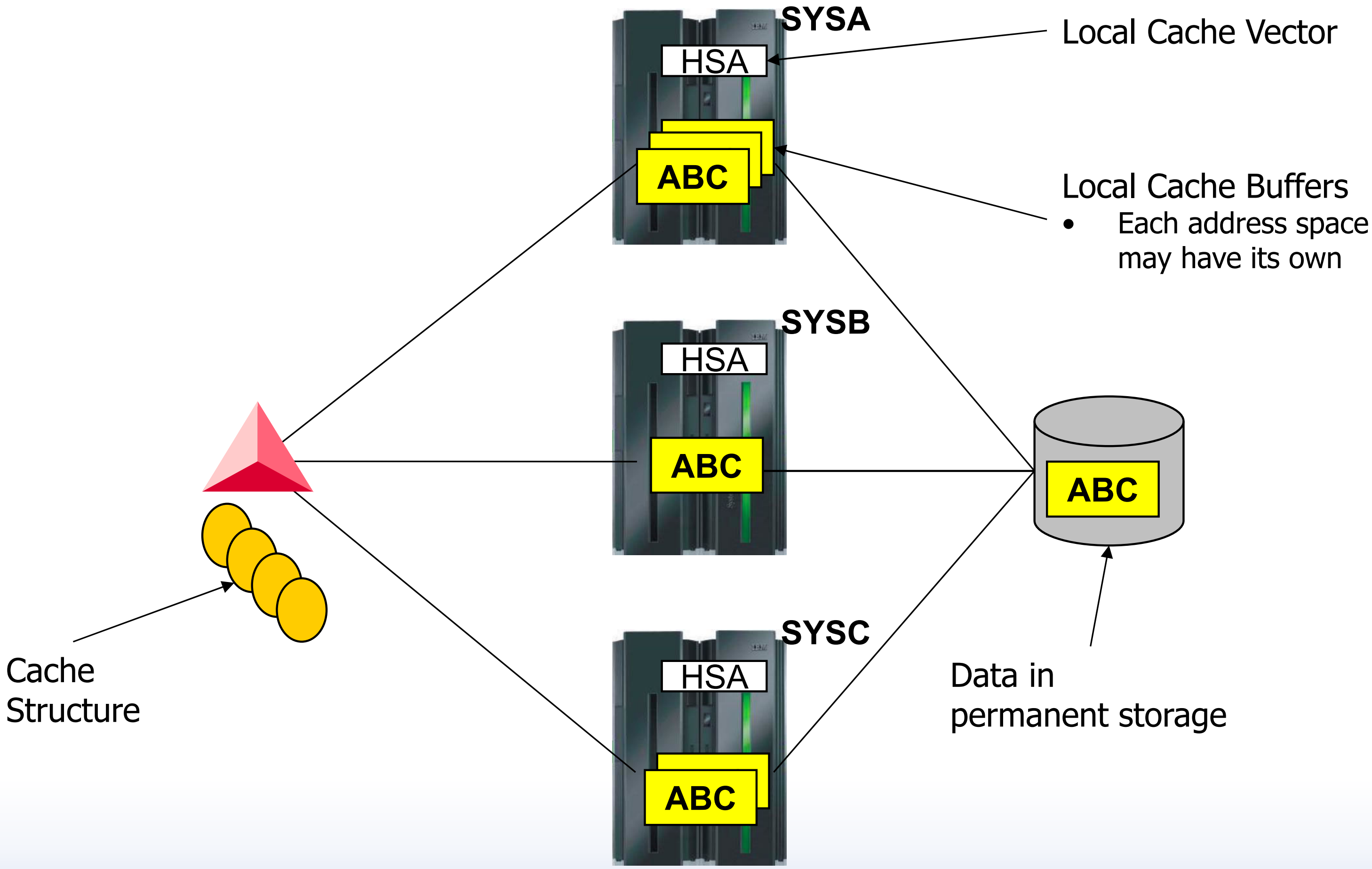


- Cache structure made up of
  - directory to keep track of registered data elements
  - optionally, data elements

- 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

- Local Cache Buffers
  - Buffers in private area storage of cache structure exploiting subsystems
  - Required and allocated by every exploiter of cache structures
  - Contains copies of shared data
  - Populated by disk or CF cache structure
  - Used to refresh CF cache structure or disk copy

- Permanent Storage
  - Final and permanent repository for shared data - usually disk
  - Used to populate local cache buffers

- Local Cache Vectors
  - User defined vector in HSA
  - Allows connectors of a cache structure (i.e. those sharing data) to determine if their local cache buffers contain the latest copy of the data

Cache Structure Terminology

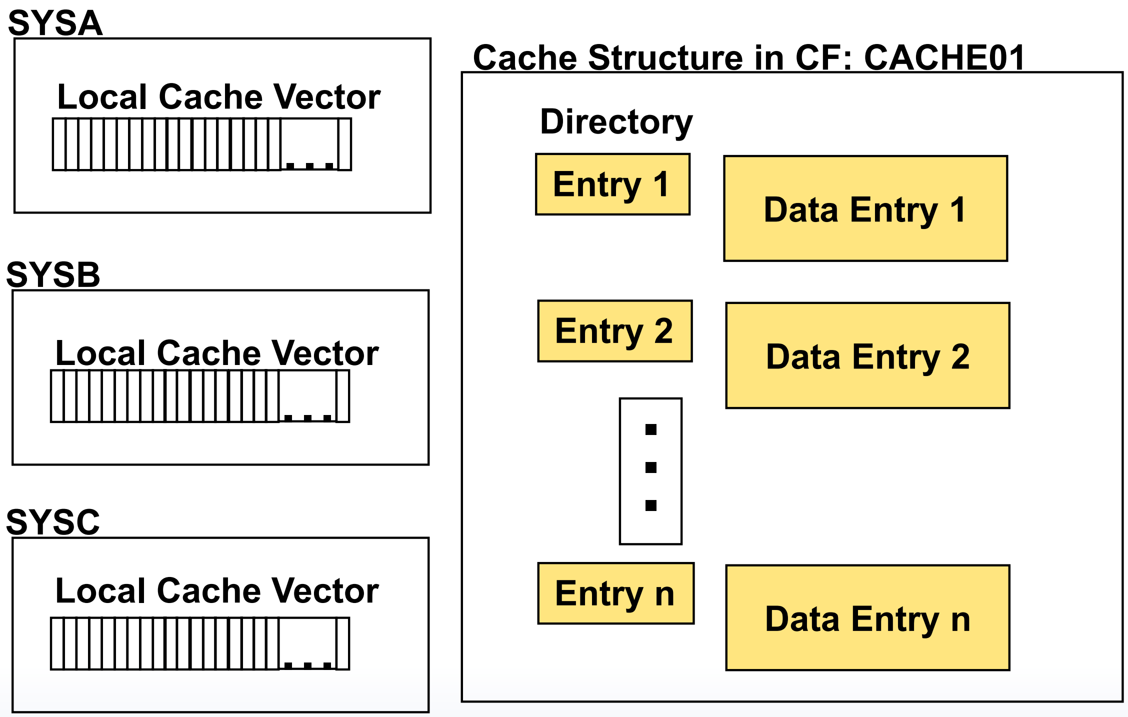
- The cache structure in the coupling facility has two primary components

- 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



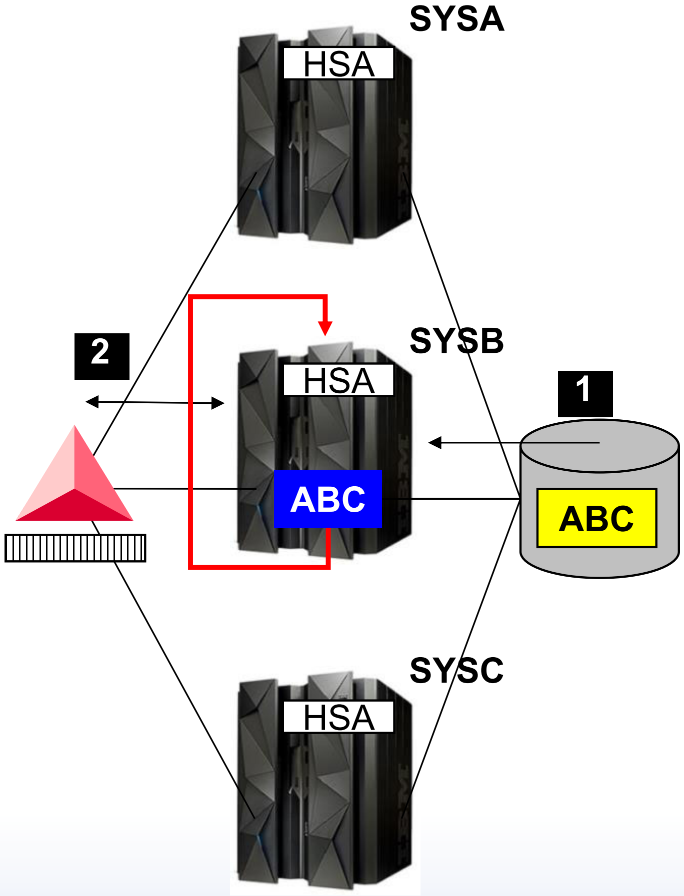
Parallel Sysplex Data Sharing – High Level

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.



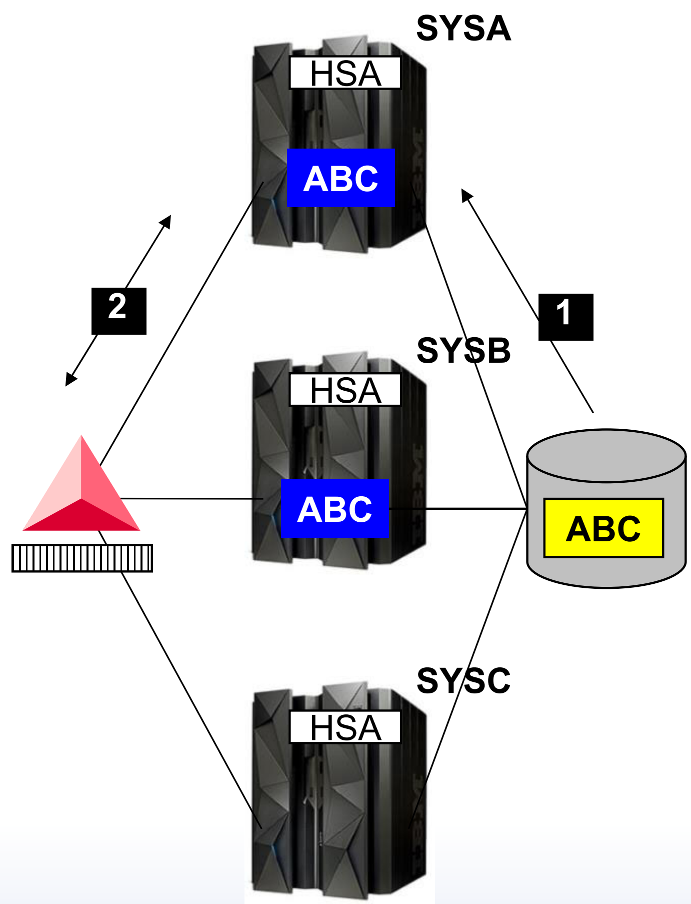
Parallel Sysplex Data Sharing – High Level

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:

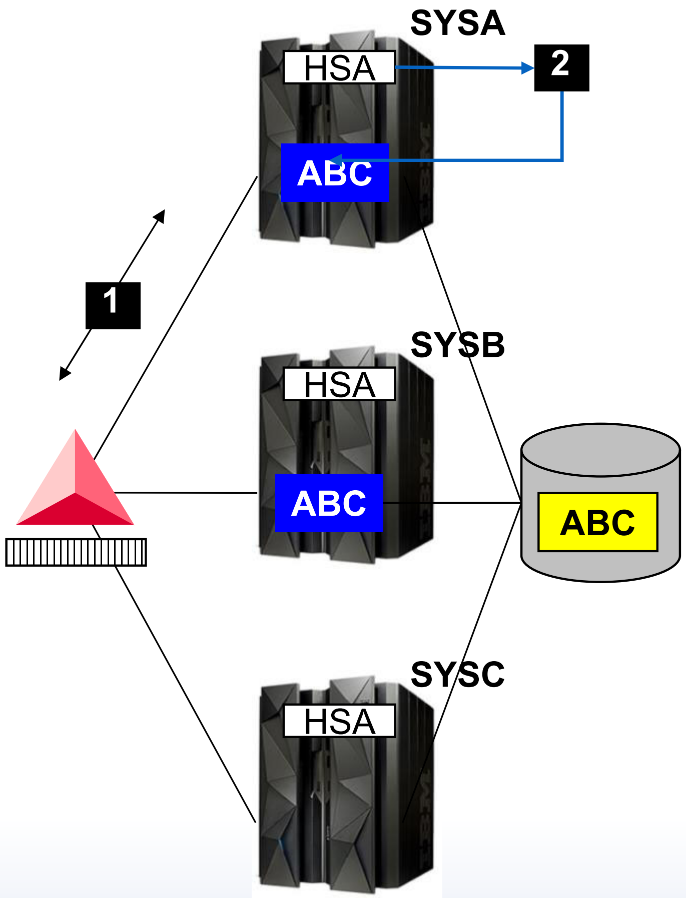
Current state: Both systems SYSA and SYSB have read ABC into local memory and has registered their interest in ABC in the coupling facility



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

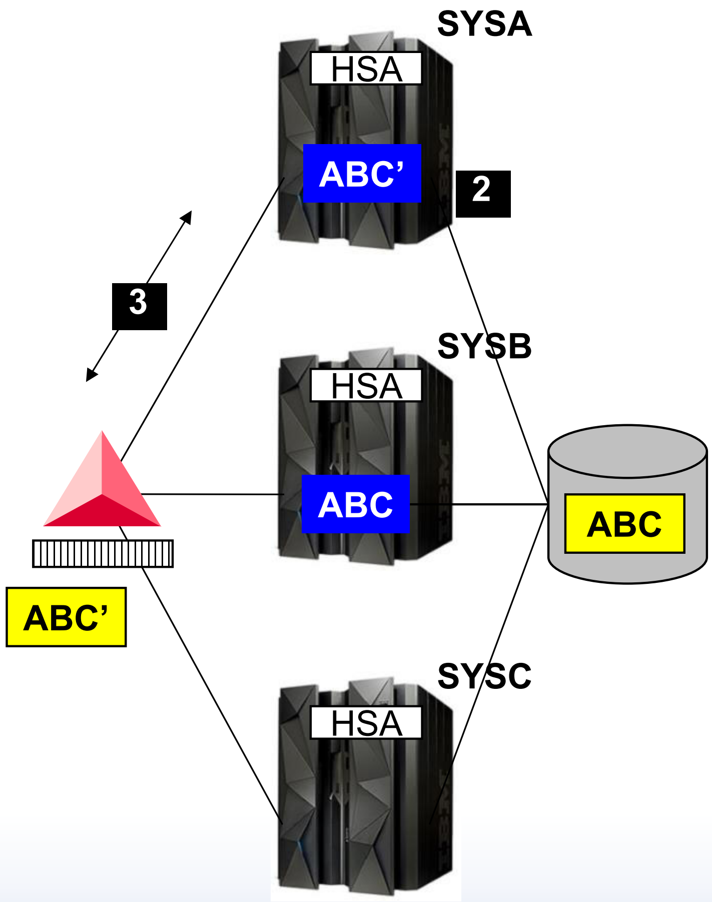


Parallel Sysplex Data Sharing – High Level

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...

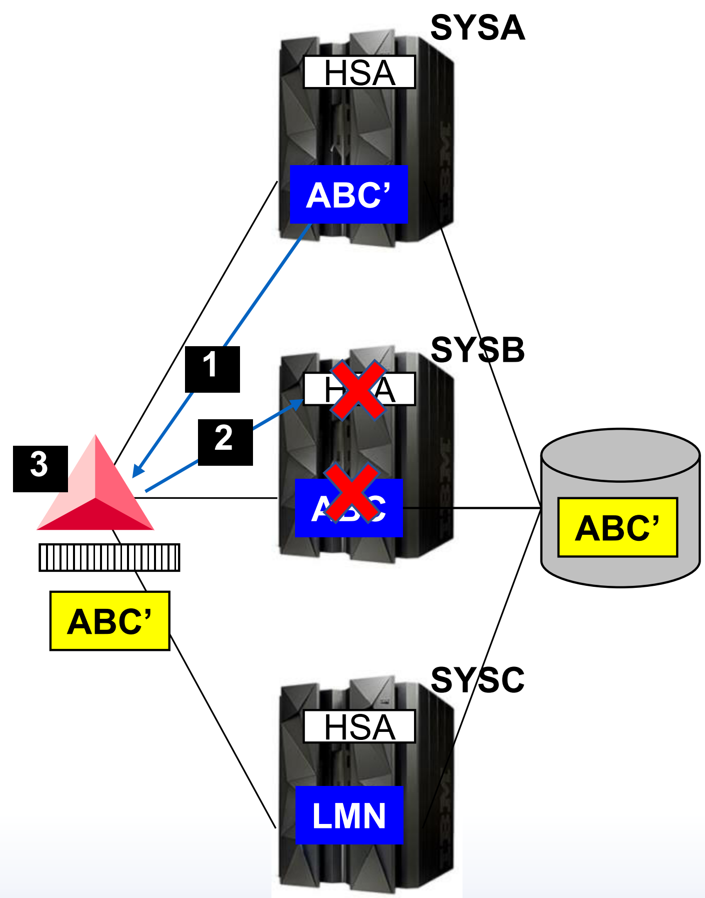
1) Signal sent by database manager to CF to indicate that record ABC has been updated
   - CF cache structure updated so all systems will know ABC has been updated

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?