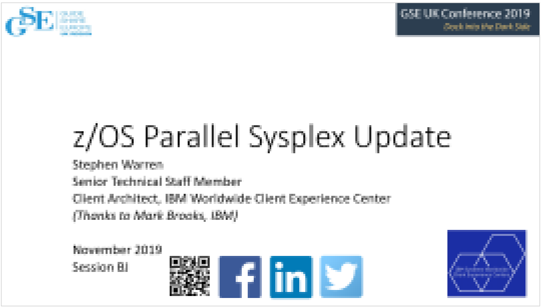

GSE UK Conference 2019
Dock into the Dark Side

z/OS Parallel Sysplex Update

Stephen Warren
Senior Technical Staff Member
Client Architect, IBM Worldwide Client Experience Center
(Thanks to Mark Brooks, IBM)

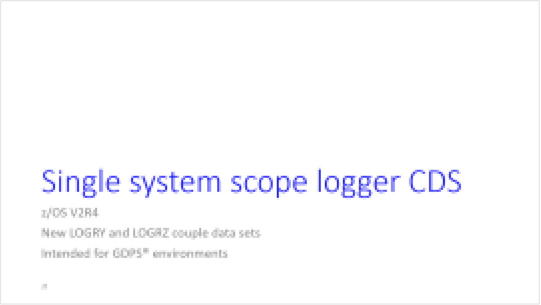
November 2019
Session BI

IBM z15

- Larger capacity for links to CF
- Faster hardware
- No support for old SR and LR hardware

Transport Class Simplification

z/OS V2R4
Eliminate the need to define size-only transport classes


Single system scope logger CDS

z/OS V2R4
New LOGRY and LOGRZ couple data sets
Intended for GDPS® environments

Sysplex Management Controls in z/OSMF

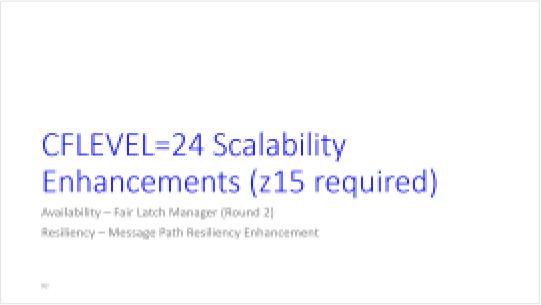
View capability available for z/OS V2R2 and V2R3
Modification of sysplex resources available in V2R3

Async duplexing for CF lock structures

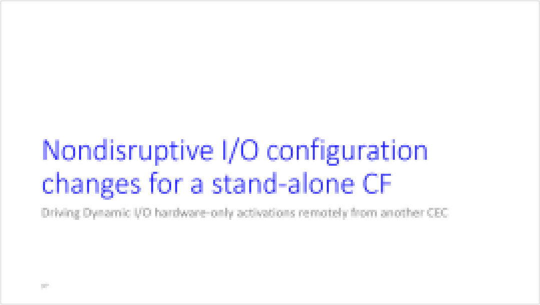
Resiliency without the overhead!

Asynchronous Cross-Invalidate for CF Cache Structures

Performance improvements for "transactional cache" processing


CFLEVEL=24 Scalability Enhancements (z15 required)

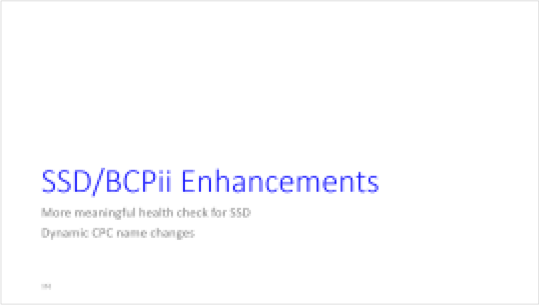
Availability – Fair Latch Manager (Round 2)
Resiliency – Message Path Resiliency Enhancement


Nondisruptive I/O configuration changes for a stand-alone CF

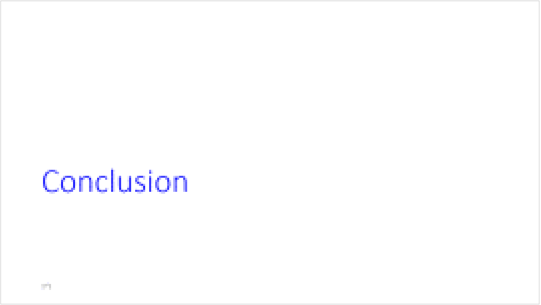
Driving Dynamic I/O hardware-only activations remotely from another CEC

CF Structure Encryption

Enhance security and data protection


SSD/BCPii Enhancements

More meaningful health check for SSD
Dynamic CPC name changes


Conclusion

z/OS Parallel Sysplex Update

Stephen Warren
Senior Technical Staff Member
Client Architect, IBM Worldwide Client Experience Center

(Thanks to Mark Brooks, IBM)

November 2019
Session BJ

IBM Systems Worldwide
Client Experience Centers

IBM z15

- Larger capacity for links to CF
- Faster hardware
- No support for old SR and LR hardware

IBM z15^{TM} from a sysplex perspective

- Up to 190 cores to configure (was 170 for z14, 141 for z13)
  - Any given CF LPAR can have at most 16 logical CP's (ICFs) (no change)
- 384 Coupling CHPIDs per CEC (was 256 for z14 & z13)
  - At most 128 can be configured for given CF LPAR
  - At most 64 internal coupling CHPIDs per CEC (was 32 on z14)
- Coupling Express LR (Long Reach)
  - Only LR option available *(see more later)*
- 96 ICA SR coupling link ports per CEC (was 80 for z14, 40 for z13)
  - Only SR option available *(see more later)*
  - 24 links per drawer (was 20 for z14)
- CFLEVEL=24 (GA1 firmware)
- CPACF – significantly faster encryption/decryption than z13; more algorithms supported
- CryptoExpress 7S – faster encryption/decryption than previous cards; more functions supported

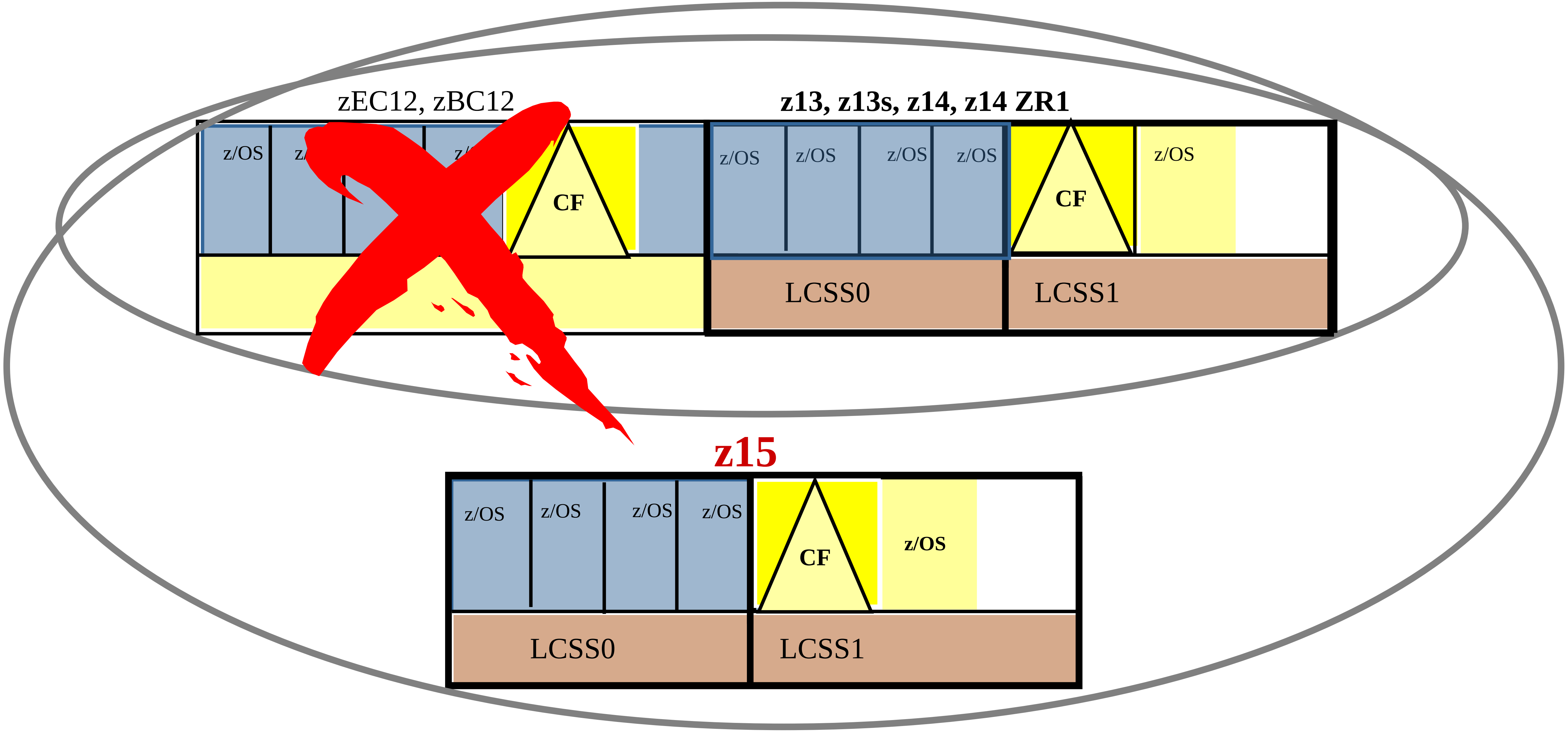
Server Participation in a Parallel Sysplex

- z15 servers support active participation in the same Parallel Sysplex with these servers:
  - IBM$_{®}$ z14^{TM} ,IBM z14 Model ZR1
  - IBM z13^{TM} IBM z13s

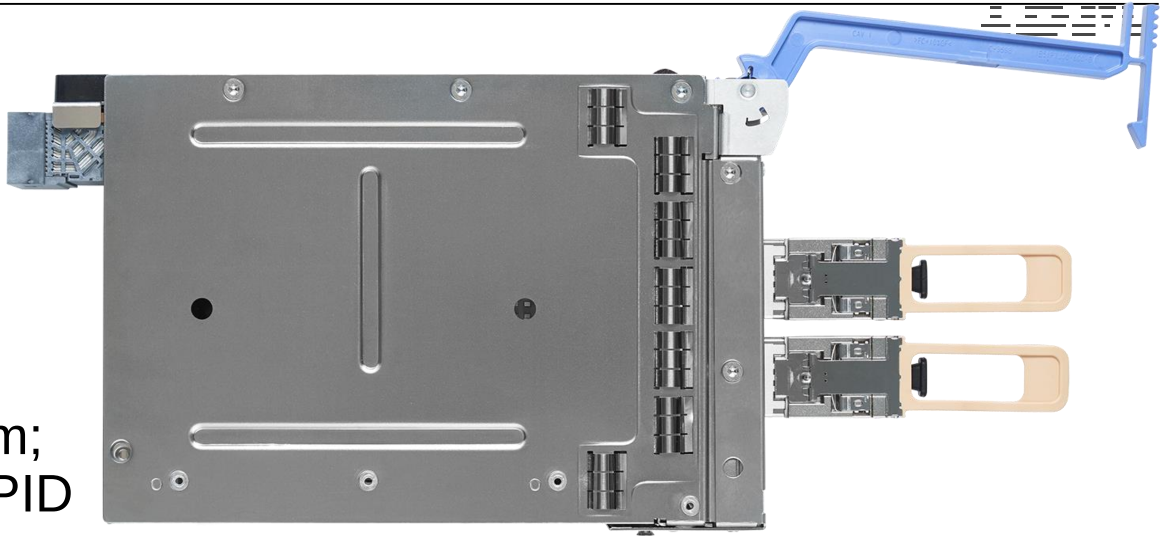
- Which means:
  - Configurations with z/OS on one of these servers can add a z15 server to their Sysplex for either a z/OS or a Coupling Facility image
  - Configurations with a Coupling Facility on one of these servers can add a z15 server to their Sysplex for either a z/OS or a Coupling Facility image

Remember: z13 and z14 servers can only connect to z15 if they have been moved off of Infiniband coupling links technology, as shown later.

Server Participation in a Parallel Sysplex …



Parallel Sysplex Coupling Links



▪ IBM Integrated Coupling Adapter (ICA SR)

- ***Coupling Connectivity into the Future (Short Distance)***
- Coupling CHPID CS5, Performance similar to Coupling over InfiniBand 12X IFB3 protocol
- PCIe Gen3, Fanout in the CPC drawer, 2-ports per fanout, 150m;
- Up to 4 CHPIDs per port, 8 buffers (i.e. 8 subchannels) per CHPID
- z13 GA1 availability

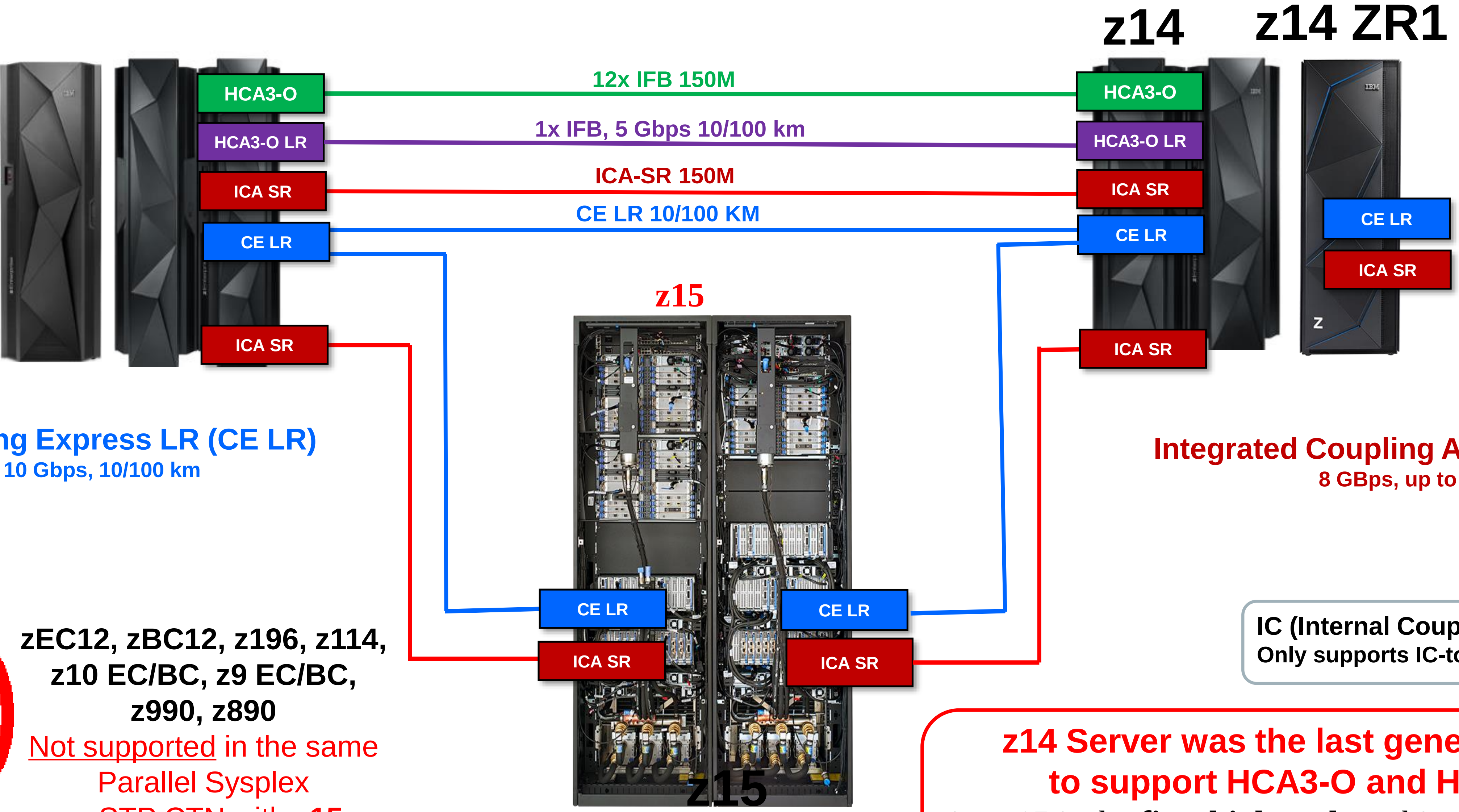
▪ Coupling Express LR (CE LR)

- ***Coupling Connectivity into the Future (Long Distance)***
- Coupling CHPID CL5, Performance similar to Coupling over InfiniBand 1x
- PCIe+ I/O drawer required for CL5 adapter
- Adapter (2-port card): same adapter as 10GbE RoCE Express but with Coupling Optics and Firmware
- 10 Gbps, Up to 4 CHPIDs per port, 32 buffers (i.e. 32 subchannels) per CHPID
- Distance: 10 KM Unrepeated; up to 100 KM with qualified DWDM
- Point-to-Point
- Retrofitted on z13 GA2
- **Last date** to MES CE LR on **z13, z13s** was June 30, 2019

Parallel Sysplex Coupling Connectivity

z13 & z13s

12x IFB3, 1x IFB3
ICA SR, CE LR



Coupling Express LR (CE LR)
10 Gbps, 10/100 km

Integrated Coupling Adapter (ICA SR)
8 GBps, up to 150M

IC (Internal Coupling Link):
Only supports IC-to-IC connectivity

zEC12, zBC12, z196, z114, z10 EC/BC, z9 EC/BC, z990, z890
Not supported in the same Parallel Sysplex or STP CTN with **z15**

z14 Server was the last generation server to support HCA3-O and HCA3-O LR

i.e. z15 is the **first high-end** machine that does not support Infiniband coupling links.

NOTE: The link data rates do not represent the performance of the links. The actual performance is dependent upon many factors including latency through the adapters, cable lengths, and the type of workload.

Firmware levels for the N-2 Parallel Sysplex CEC Connectivity

- The IBM z15 (8561 T01) can be coupled to the following servers with these MCL requirements:

- **z14 (MT 3906/MT 3907) at Driver 36**
  - CFCC Level 23 – Service Level 0.13
  - Bundle S13 / MCL P41419.003 (February 2019)

- **z13/z13s (MT 2964/MT 2965) at Driver 27**
  - CFCC Level 21 – Service Level 2.20
  - Bundle S82 / MCL P08416.008 (February 2019)

Transport Class Simplification

z/OS V2R4

Eliminate the need to define size-only transport classes

Why eliminate the need to define Transport Classes?

- Simplification
  - Remove this burden from customers and all the IBM personnel that have to analyze, explain, make recommendations, etc.
  - RMF reports do not always provide clear guidance as to how signal resources should be configured; more art than science

- Avoid Outages
  - Transport Classes often not well understood, mistakes are made that lessen the resiliency of the sysplex and so permit avoidable outages
  - A Transport Class can help isolate an ill behaved member, but not really practical since you have to know in advance who is going to cause trouble
    - Such as delays and overwhelming bursts

- Self-optimization
  - Static definitions are not well suited to the dynamics of the sysplex
    - Can lead to inefficient use of resources (idle paths, excess storage)
  - System should automatically apply resources where needed most
    - For example, to handle a burst of messages or a change in signal patterns

To eliminate the need to define Transport Classes

XCF must automatically handle problems transport classes were intended to address:

- Timely message transfer
  - Maintain signal throughput and minimize signal delivery times, especially for the small messages that are typically most predominant in the sysplex
  - Provides better resiliency (minimize delays and queueing)
- Efficient utilization of signal resources
  - Helps minimize cost
  - Provides better resiliency (more capacity)

z/OS V2R4

Size segregation

Group segregation

Later*

- Isolation of ill behaved members
  - Avoid sympathy sickness so that problems with signal delivery for one member don't negatively impact signal delivery for other members
- Fair access to signal resources
  - Don't allow one member to monopolize the signal resources to the detriment of signal delivery for other members

*** Any statements regarding IBM's future direction, intent or product plans are subject to change or withdrawal without notice.**

z/OS V2R4 – XCF Transport Class Simplification

- Solution
  - XCF will internally manage the signal resources to provide timely delivery of signals in a sysplex independent of any Transport Class definitions created purely for size segregation.
  - XCF will intermix signal sizes on “XCF Managed” signal paths as it sees fit while maintaining signal throughput and timely signal transfer, especially for small signals
    - Thus eliminating the need for transport class based signal size segregation
    - Any available “XCF Managed” path can be used for any size signal
  - To make this “visible”, there is a new pseudo-transport class: _XCFMGD
  - To enable “control”, there is a new XCF FUNCTIONS switch: XTCSIZE

- Benefits
  - Simplification: You no longer need to define, monitor, tune, or manage XCF Transport Class definitions to segregate signals purely by size.
  - Simplification: You need only configure an appropriate number of signal paths.
  - Resilience: Less potential for non-optimal transport class definitions to negatively impact signal delivery.

New XCF FUNCTIONS switch XTCSIZE determines behavior

- When XTCSIZE is DISABLED:
  - XCF signal resources are managed per traditional transport class segregation rules
  - The new _XCFMGD pseudo-transport class is visible, but will not be used
- When XTCSIZE is ENABLED:
  - Traditional transport class segregation rules used if target is z/OS V2R3 (or earlier)
    - So traditional transport class definitions needed and used until z/OS V2R4 is running sysplex wide (or if you ever intend to DISABLE the XTCSIZE switch)
  - New “XCF Managed” rules apply when target is running z/OS V2R4 (or later)
    - Traditional transport classes defined purely for size segregation become “XCF Managed”
    - Signals normally sent via those traditional transport classes are instead sent via the new _XCFMGD pseudo-transport class
      - So you will not see any activity in the “XCF Managed” classes, it all moves to _XCFMGD

About the new _XCFMGD pseudo transport class

DISPLAY XCF accepts _XCFMGD as a class name.
SETXCF and COUPLExx do not.

- Implicitly defined by XCF
  - Always exists
    - But not used if XTCSIZE is DISABLED or if target system is pre-z/OS V2R4
  - Installation cannot directly control its attributes (classlen=0, XCF determines MAXMSG)
  - When XTCSIZE is ENABLED, all paths in the "XCF Managed" classes (for an up level target system) are logically reassigned to the _XCFMGD pseudo-transport class
- Uses "best fit" buffers on the send side
  - Maximizes number of signals that can be accepted for a given MAXMSG limit
    - Which is important for handling bursts and delays
  - Traditional classes generally use the "defined size" which might not be best fit
    - So could encounter send side "no buffer" condition sooner than with _XCFMGD
- Paths run at the maximum signal size
  - So any size signal can be transmitted without any additional overhead
    - Never need to re-negotiate signal size (or tune) the signal paths
  - But that implies buffers on target system are likely bigger than needed
    - Which raises "no buffer" concerns (resiliency, storage utilization, capacity, throughput)

Implications

- The only determining factors for signal performance will be:
  - The number of signal paths
  - The performance characteristics of those paths
  - The performance characteristics of the systems using those paths
- The Transport Class definitions should be completely irrelevant to performance of the sysplex workload
  - Regardless of the Transport Class definitions and how (a given set of) signal paths are assigned to them there will not be any noticeable impact on the sysplex workload
  - No matter how “good” or “bad” those specifications might be
  - Changes to class definitions will neither improve nor degrade signal delivery

Single system scope logger CDS

z/OS V2R4

New LOGRY and LOGRZ couple data sets

Intended for GDPS® environments

Overview

- Problem
  - In GDPS environments, the k-Systems must be as isolated from the rest of the sysplex as possible to ensure that GDPS can accomplish failover
  - So in some GDPS configurations, z/OS System Logger services are not available on the k-Systems:
    - Logger address space not started at all, or gets cancelled by GDPS if it does start
    - Logger CDS not accessible (either not online or ALLOWACCESS=NO specified in IXGCNFxx)
  - But logging is very useful, even on the k-Systems

- Solution
  - Logger and XCF now support two new types of logger single system-scope couple data sets (CDS) that do not interfere with GDPS ability to accomplish failover
    - LOGRY and LOGRZ

- Benefit
  - The k-Systems can make use of system logger services
  - You can extract log data using same tools/utilities as other systems in the sysplex

Usage and Invocation

- Use XCF format utility (IXCL1DSU) to create primary/alternate logger system-scope CDS
  - New CDS data types
    - DATA TYPE(LOGRY)
    - DATA TYPE(LOGRZ)
  - MAXSYSTEM(n) – use same “n” as for the sysplex-scope logger CDS
  - ITEM NAME(LSR) NUMBER(n) – max number of log streams to be defined in logger policy
  - ITEM NAME(DSEXTENT) NUMBER(n) - number of additional log stream data set directory extents to define for log stream offload data sets.
  - ITEM NAME(FMTLEVEL) NUMBER(1) – format level 1 is HBB77C0, supported by z/OS V2R4 (and up)
- Update COUPLExx parmlib member(s) to define the logger CDS to XCF
  - DATA TYPE(LOGRY) PCOUPLE(primary-dsn,volser) ACOUPLE(alternate-dsn,volser)
  - DATA TYPE(LOGRZ) PCOUPLE(primary-dsn,volser) ACOUPLE(alternate-dsn,volser)
  - You could have one common COUPLExx defining all the logger CDS, but you’ll get fewer complaint messages if you have a COUPLExx defining the one particular logger CDS that a given system is to use

Usage and Invocation (continued)

- Update IXGCNFxx parmlib members to tell logger which systems are to use which type of logger CDS
  - ALLOWACCESS(YES|NO) – can the system use the sysplex scope CDS (type LOGR)?
  - USECDSTYPE(LOGR|LOGRY|LOGRZ) – which type of logger CDS should the system use?
    - Single system-scope CDS (type LOGRx) requires ALLOWACCESS(NO)

Examples:

ALLOWACCESS(YES)	- Use sysplex scope logger CDS of type LOGR
ALLOWACCESS(YES) USECDSTYPE(LOGR)	- Use sysplex scope logger CDS of type LOGR
ALLOWACCESS(NO)	- Do not use any logger CDS
ALLOWACCESS(NO) USECDSTYPE(LOGR)	- Do not use any logger CDS
ALLOWACCESS(NO) USECDSTYPE(LOGRY)	- Use single system scope logger CDS of type LOGRY
ALLOWACCESS(NO) USECDSTYPE(LOGRZ)	- Use single system scope logger CDS of type LOGRZ

Note:

If any one system is using a single system scope logger CDS of a given type, no other system in the sysplex will be permitted to use a logger CDS of that same type

Usage and Invocation (continued)

- Update GRSRNLxx parmlib member to add the log stream related data sets to the RNL exclusion list on any system using a CDS of type LOGRY or LOGRZ

- Make the logger CDS available to the system
  - IPL with appropriate COUPLExx, or
  - Dynamically via SETXCF COUPLE command using TYPE=LOGRx

```
Example: SETXCF COUPLE,TYPE=LOGRZ,PCOUPLE=(dsn_primary cds, volser)
         SETXCF COUPLE,TYPE=LOGRZ,ACOUPLE=(dsn alternate cds, volser)
```

- Use policy utility (IXCMIAPU) to define logger policy
  - LOGRY and LOGRZ are restricted to DASD-only log streams
  - No CF structure-based log streams allowed

Log stream name conflicts

- You can define log stream with same name in a LOGRx CDS and the LOGR CDS.
- But… you must either:
  - Ensure that such log streams have a unique EHLQ/HLQ attribute in each CDS, or
  - You must ensure that the log stream resources are completely isolated from one another
    - Separate catalogs and DASD, and
    - The GRSRNLxx parmlib on the system using a LOGRx CDS must specify the following:

```
RNLDEF RNL(EXCL) TYPE(GENERIC) QNAME(SYSDSN) RNAME(IXGLOGRx)
RNLDEF RNL(EXCL) TYPE(GENERIC) QNAME(SYSDSN) RNAME(hlq.lsname)
RNLDEF RNL(EXCL) TYPE(GENERIC) QNAME(SYSDSN) RNAME(ehlq.lsname)
```

- z/OS avoids log stream data set name collisions via the following scheme:

CDS type	HLQ default	offload dataset suffix
LOGR	IXGLOGR	*cnnnnnnn*
LOGRY	IXGLOGR**Y**	*c***Y***nnnnnn*
LOGRZ	IXGLOGR**Z**	*c***Z***nnnnnn*

HLQ – High Level Qualifier used for log stream and staging data set names
ELHQ – Extended High Level Qualifier used for log stream and staging data set names

Using utilities with LOGRx CDS

- Existing utilities can be used for log stream data access for log streams defined in one of the new system logger Couple Data Set types:
  - The utilities need to be run on the same system that is using CDS of type LOGRx
  - The utility should move the log data into archive/history data set(s)
  - The archive/history data set(s) must be accessible to the other systems in sysplex in order to merge this log data with the sysplex view

- On any system in the sysplex:
  - Existing utilities can be used on any system in sysplex that has access to archive/offload data sets to merge this log data with the sysplex view.
  - If desired, use a sort program with the archive/history data sets to merge data from the system(s) that used the new system logger CDSs (LOGRZ and LOGRY).

Security

- Access authorization for new system logger policies
- New resources for the FACILITY class:
  - MVSADMIN.LOGRY
  - MVSADMIN.LOGRZ

Failover and time consistent data

- For sysplex scope logging, you need time consistent copies of the LOGR CDS, the staging data sets, and offload data sets for failover
  - In this context, “failover” amounts to using a point in time copy of the data for use by another sysplex
- For single system scope logging, you similarly need time consistent copies of the LOGRx CDS and the relevant staging and offload data sets
  - In this context, “failover” amounts to using a point in time copy of the data for use by another <u>system</u>
  - That is, if you want to successfully use a LOGRx CDS on some z/OS system other than the one that last used it, all the logger resource configuration data must be in a time consistent state for it to be useful

Migration and Coexistence

- Only a z/OS V2R4 system can actually use a single system scope LOGRx CDS
- Toleration/coexistence APARs/PTFs for z/OS V2R2 & V2R3:
  - Toleration PTFs for logger APAR → OA54815
    - Logger recommends that the APAR be installed on all down-level systems in the sysplex before bringing a LOGRx CDS into use by the sysplex
    - Enables you to format a LOGRx CDS on a down-level system (or you can STEPLIB to an up-level library)
    - Ensures that the down-level systems appropriately recognize and reject use of LOGRx CDS
  - Toleration PTFs for XCF APAR → OA57241
    - Preferred, though not required, to install the XCF APAR on all systems in the sysplex before bringing a LOGRx CDS into use by the sysplex
    - If you don't have the toleration APAR installed on the down-level systems, there is no loss of functionality, but various messages issued by XCF on those systems will not be as helpful or informative as they otherwise could be

Sysplex Management Controls in z/OSMF

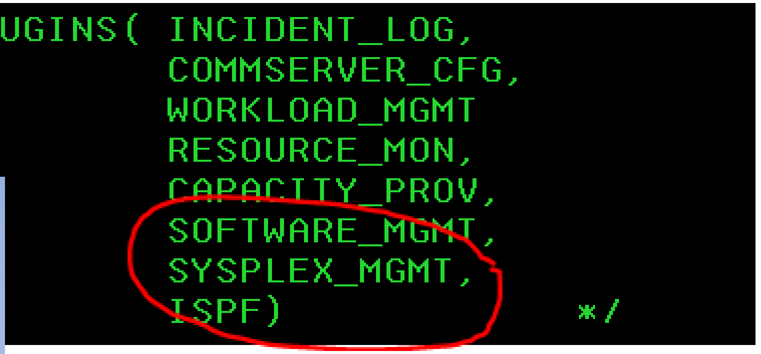
View capability available for z/OS V2R2 and V2R3

Modification of sysplex resources available in V2R3

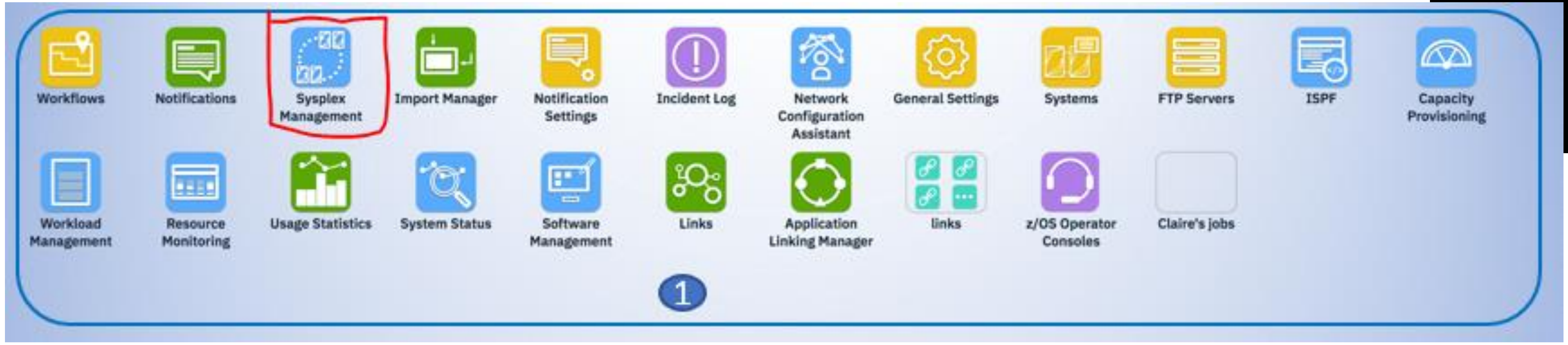
z/OSMF Sysplex Management Display-only Capabilities

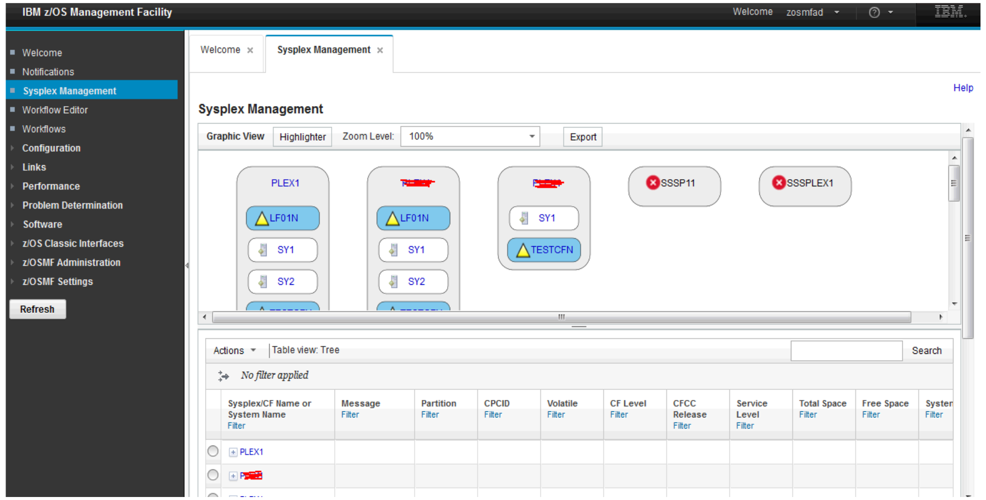
Parallel Sysplex Management Plug-In !!

- View sysplex resources such as z/OS systems, coupling facilities, coupling facility structures, programs using coupling facility structures, couple datasets and the policies they contain, coupling link connectivity resources, etc.
  - Graphical and tabular displays
  - Physical and logical views of sysplex resources
  - Visualizations and drill-downs
- Available in z/OS 2.2 and up



```
/* PLUGINS( INCIDENT_LOG,
            COMMSERVER_CFG,
            WORKLOAD_MGMT
            RESOURCE_MON,
            CAPACITY_PROV,
            SOFTWARE_MGMT,
            SYSPLEX_MGMT,
            ISPF)                  */
```





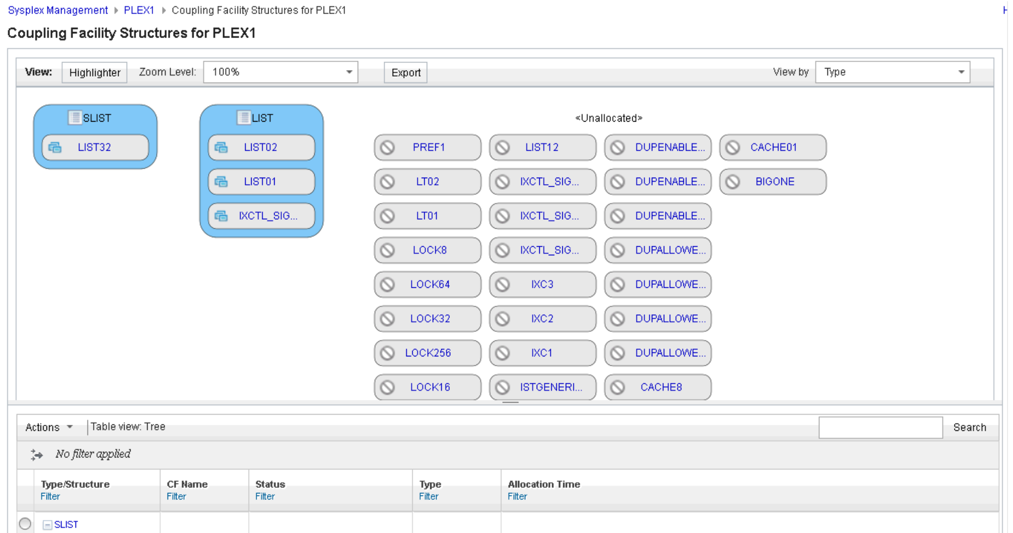
Sysplex Management

Graphic View | Highlighter | Zoom Level: 100% | Export

Actions | Table view: Tree | Search

No filter applied

	Sysplex/CF Name or System Name Filter	Message Filter	Partition Filter	CPCID Filter	Volatile Filter	CF Level Filter	CFCC Release Filter	Service Level Filter	Total Space Filter	Free Space Filter	Syster Filter
○	⊞ PLEX1										
○	⊞ [redacted]										



Sysplex Management ▸ PLEX1 ▸ Coupling Facility Structures for PLEX1

Coupling Facility Structures for PLEX1

View: Highlighter | Zoom Level: 100% | Export | View by Type

Actions | Table view: Tree | Search

No filter applied

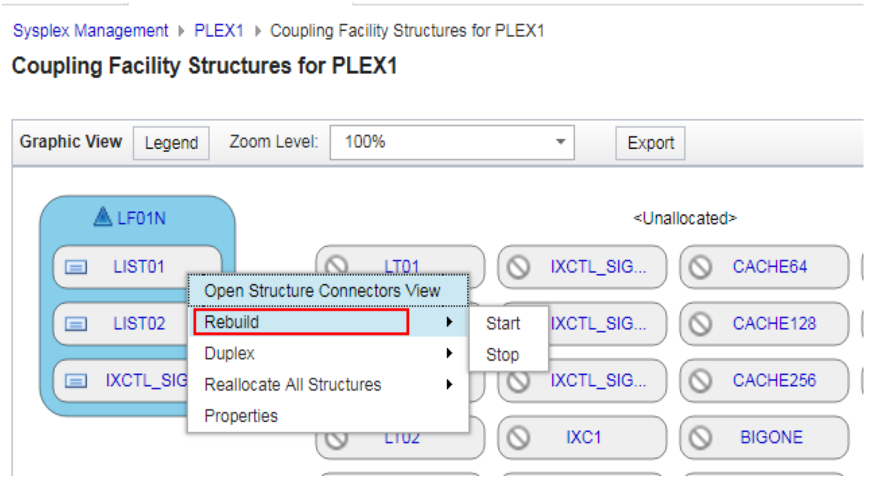
	Type/Structure Filter	CF Name Filter	Status Filter	Type Filter	Allocation Time Filter
○	⊟ SLIST				

z/OSMF Sysplex Management capabilities

- Modify Sysplex configuration
  - Sysplex-wide commands and results display
  - Command Log retained across IPL’s
  - Allows review of who took what action when (and the detailed results of each action)
  - Optionally view generated commands before issuing them

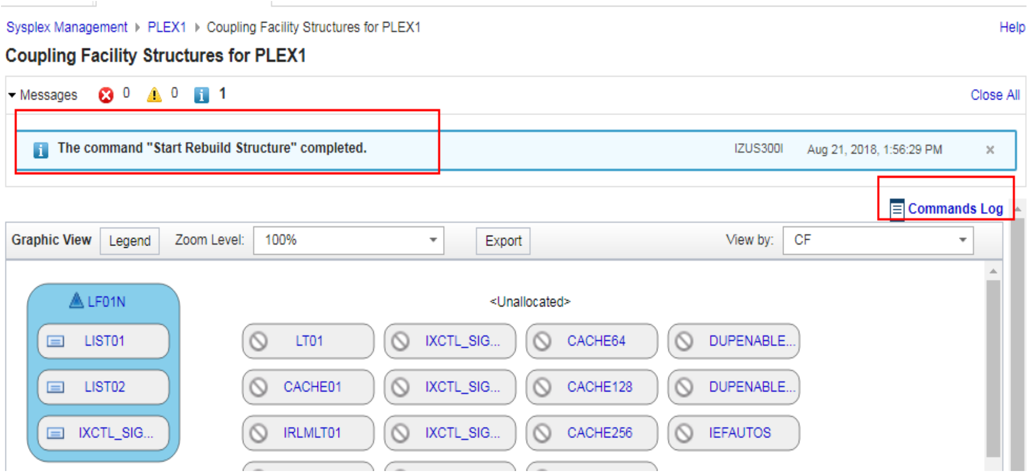
- Actions include
  - Rebuild structure(s), all structures
  - Duplex structure(s), all structures
  - Reallocate
  - Couple dataset creation, addition, switching
  - CF actions
  - CF connectivity (link and CHPID) management

z/OSMF Sysplex Management – Modify structures example



Sysplex Management ▸ PLEX1 ▸ Coupling Facility Structures for PLEX1

Coupling Facility Structures for PLEX1



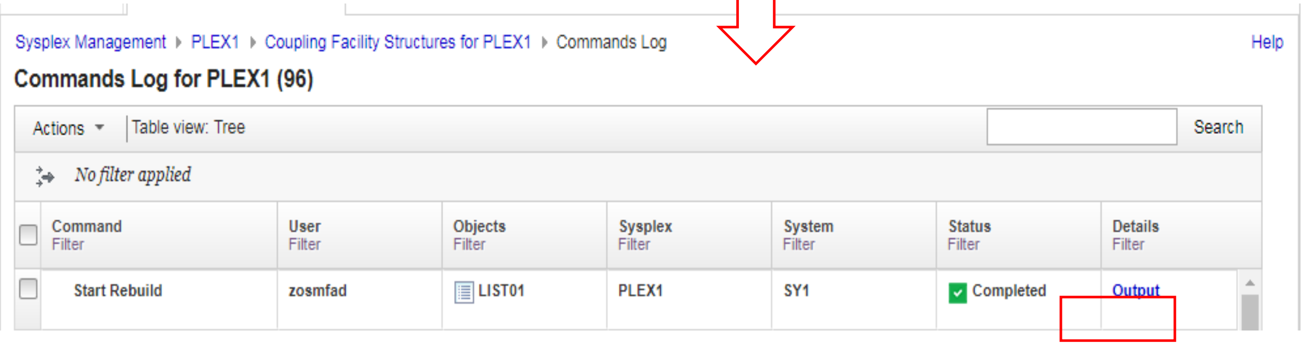
Sysplex Management ▸ PLEX1 ▸ Coupling Facility Structures for PLEX1 Help

Coupling Facility Structures for PLEX1

Messages ☒ 0 ⚠ 0 ℹ 1 Close All

ℹ The command "Start Rebuild Structure" completed. IZUS300I Aug 21, 2018, 1:56:29 PM ×

Commands Log



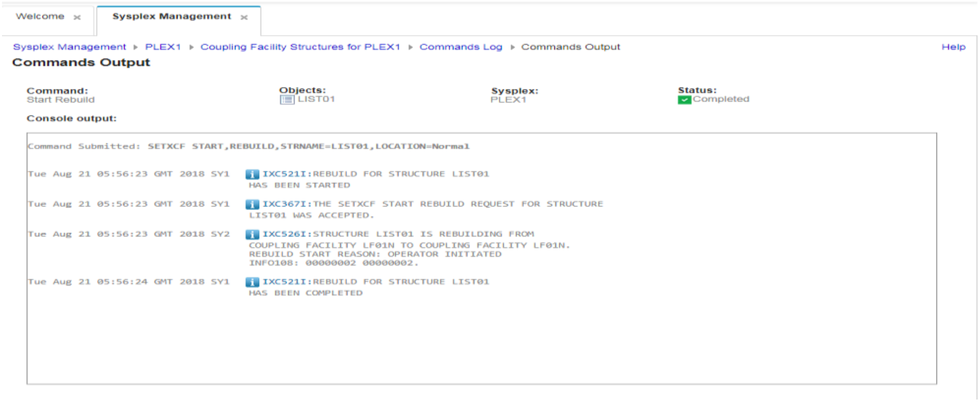
Sysplex Management ▸ PLEX1 ▸ Coupling Facility Structures for PLEX1 ▸ Commands Log Help

Commands Log for PLEX1 (96)

Actions ▾ | Table view: Tree Search

No filter applied

□	Command Filter	User Filter	Objects Filter	Sysplex Filter	System Filter	Status Filter	Details Filter
□	Start Rebuild	zosmfad	LIST01	PLEX1	SY1	✓ Completed	Output



Welcome × | Sysplex Management ×

Sysplex Management ▸ PLEX1 ▸ Coupling Facility Structures for PLEX1 ▸ Commands Log ▸ Commands Output Help

Commands Output

Command: Start Rebuild

Objects: LIST01

Sysplex: PLEX1

Status: ✓ Completed

Console output:

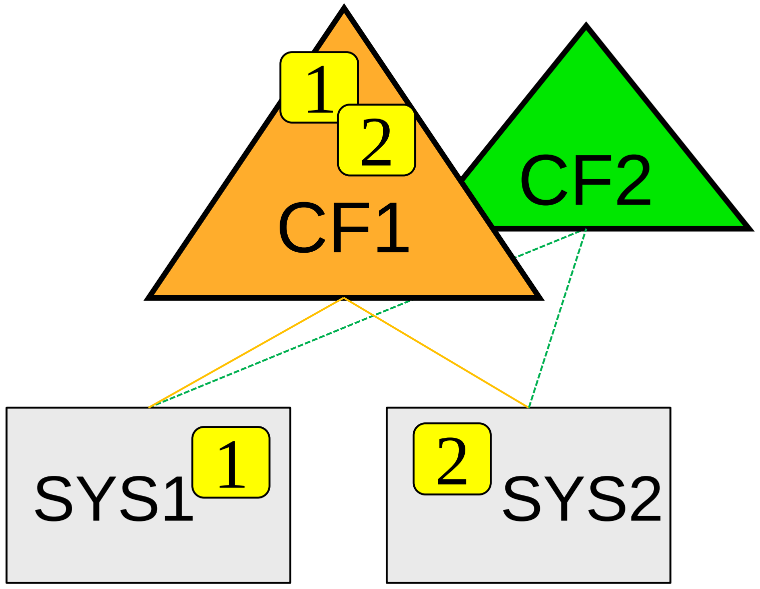
```
Command Submitted: SETXCF START,REBUILD,STRNAME=LIST01,LOCATION=Normal

Tue Aug 21 05:56:23 GMT 2018 SY1   IXC521I:REBUILD FOR STRUCTURE LIST01
                                   HAS BEEN STARTED

Tue Aug 21 05:56:23 GMT 2018 SY1   IXC367I:THE SETXCF START REBUILD REQUEST FOR STRUCTURE
                                   LIST01 WAS ACCEPTED.

Tue Aug 21 05:56:23 GMT 2018 SY2   IXC526I:STRUCTURE LIST01 IS REBUILDING FROM
                                   COUPLING FACILITY LF01N TO COUPLING FACILITY LF01N.
                                   REBUILD START REASON: OPERATOR INITIATED
                                   INFO108: 00000002 00000002.

Tue Aug 21 05:56:24 GMT 2018 SY1   IXC521I:REBUILD FOR STRUCTURE LIST01
                                   HAS BEEN COMPLETED
```

Async duplexing for CF lock structures

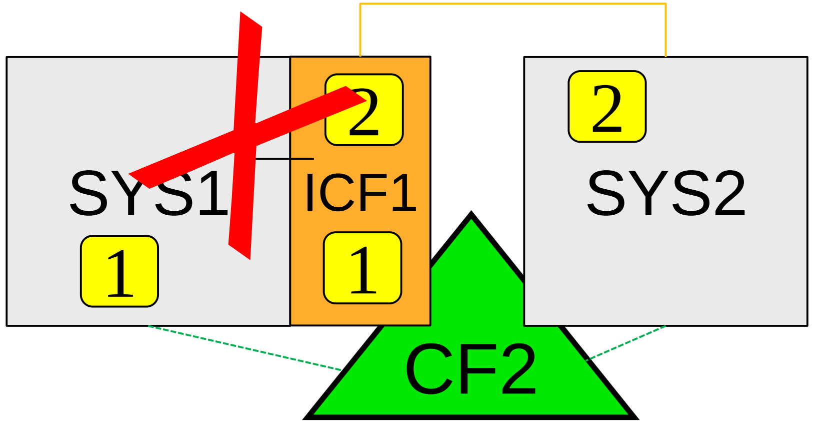
Resiliency without the overhead!

Using rebuild for failure recovery

- CF1 or ICF1 failure
  - UM rebuild into CF2 using in-store data. SM rebuild not possible.
- CF1 or ICF1 LossConn for SYS1
  - UM rebuild into CF2
  - SYS2 can SM rebuild into CF2
- CF1 or ICF1 LossConn for SYS2
  - UM rebuild into CF2
  - SM rebuild into CF2 from other system
- SYSi failure
  - Survivor continues, no impact
- CEC1 failure
  - All copies of 1 are lost
  - UM rebuild possible, but user must recover from DASD and logs. Significant effort.
  - SM rebuild not possible
- CEC2 failure
  - CEC1 continues, no impact



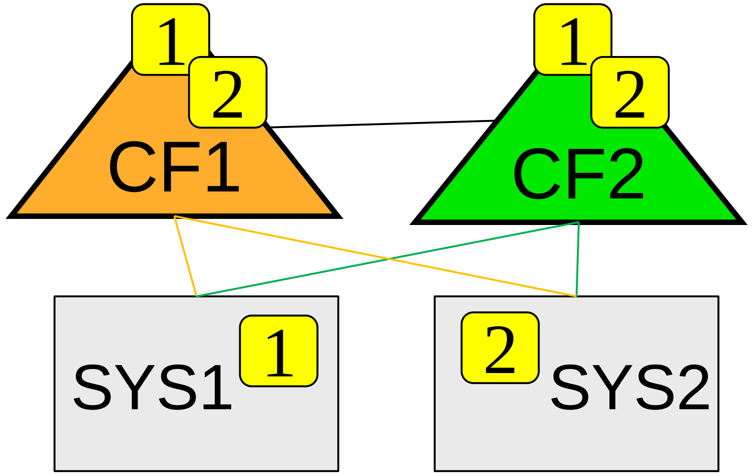
Simplex



1 2 = data

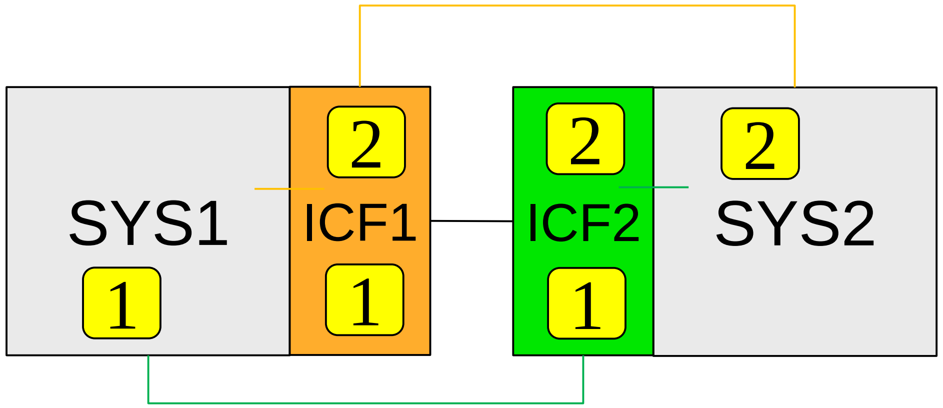
Duplexed structures provide fast, robust failure recovery

- For any of these failures:
  - Failure of any one CF
  - LossConn to any one CF from any source (one or all)
  - Failure of any one CEC
- Normal operation continues with structure in simplex mode in the surviving/accessible CF



Duplexed

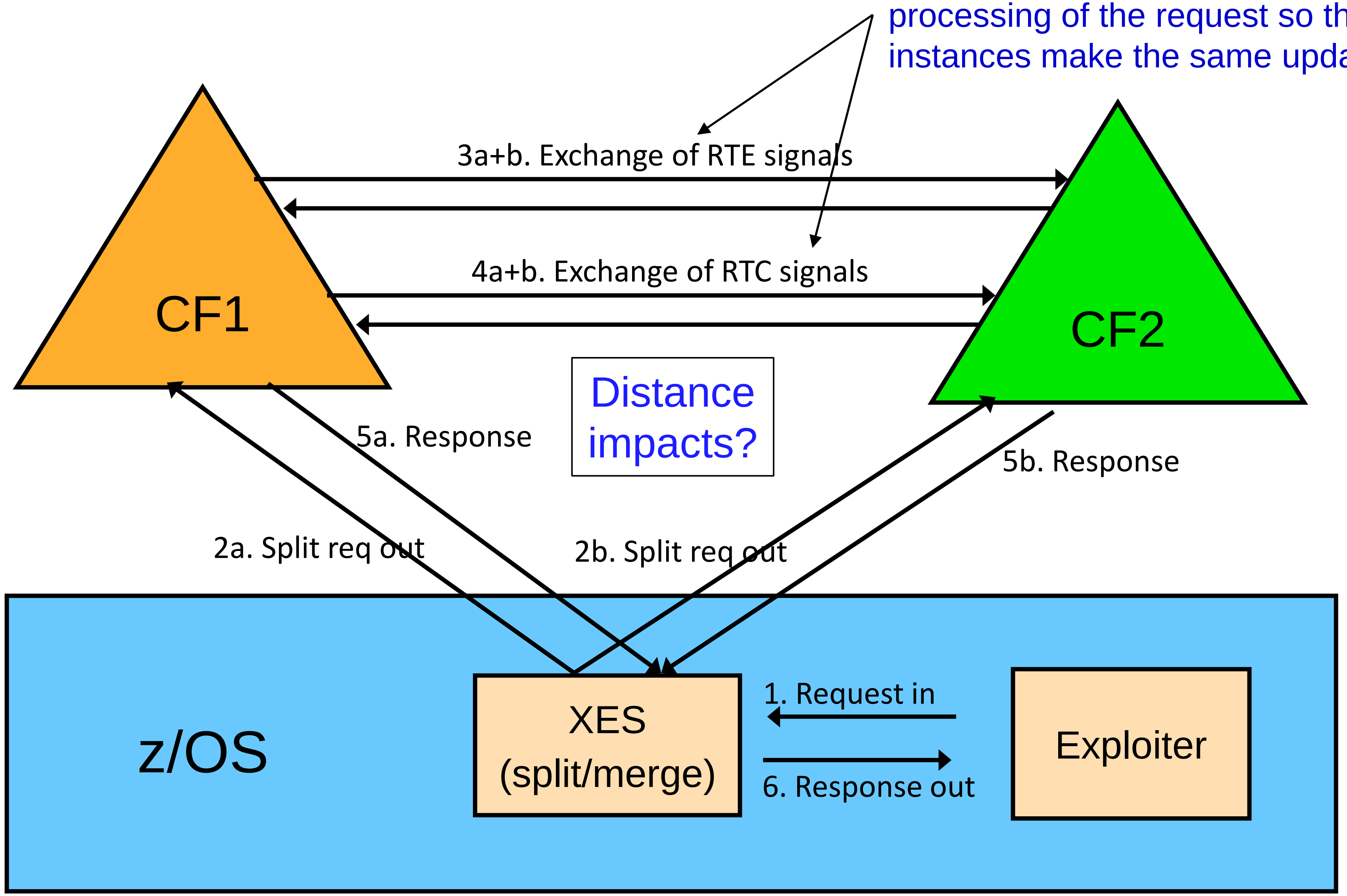
Might have more complicated lossconn scenarios if CECs are connected by "one" link



Major drawback of (synchronous) SM duplexing is its overhead …

Synchronous System Managed CF Structure Duplexing Protocol

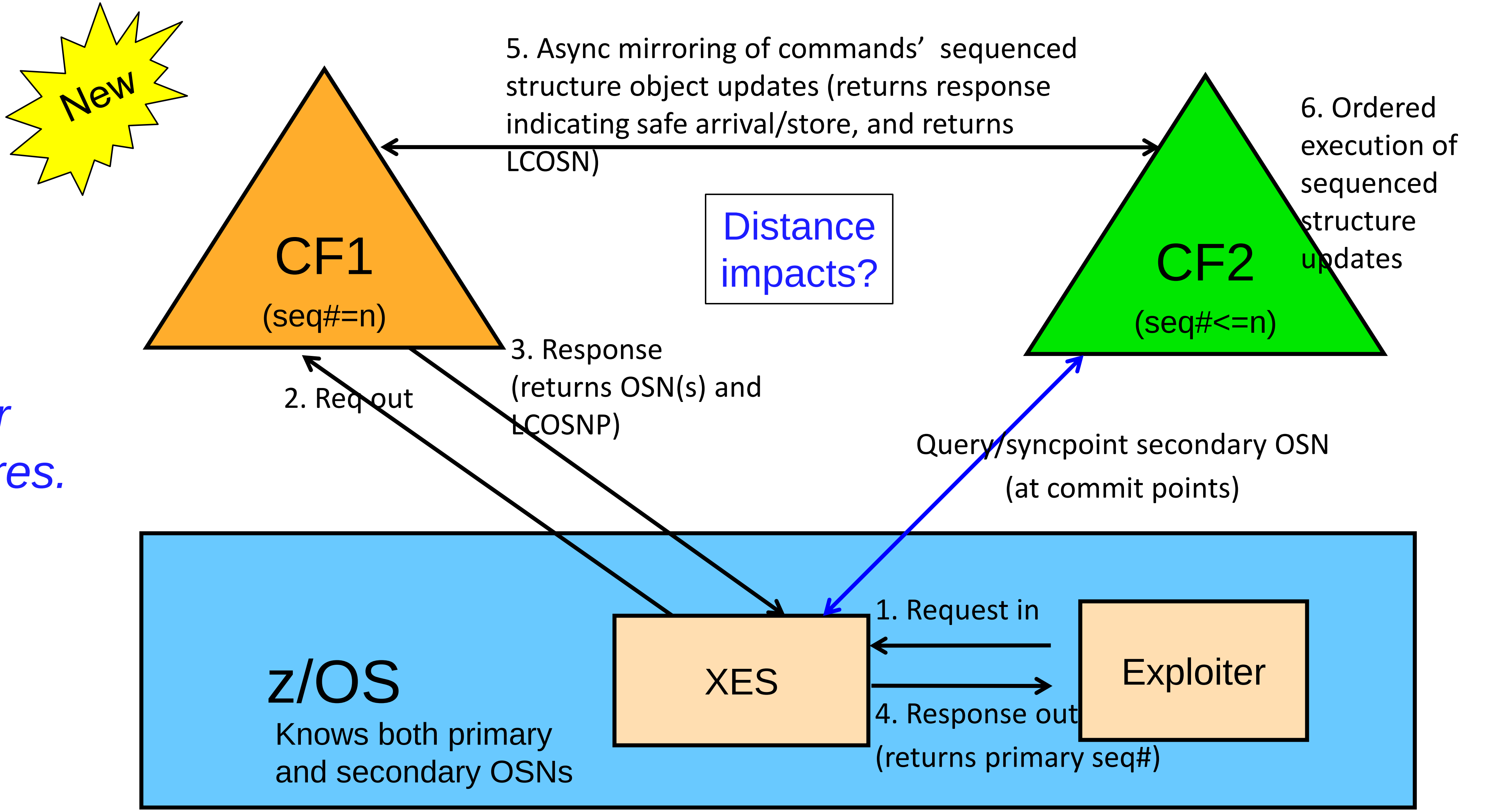
The coupling facilities communicate to coordinate processing of the request so that both structure instances make the same update.



RTE – ready to execute
RTC – ready to complete

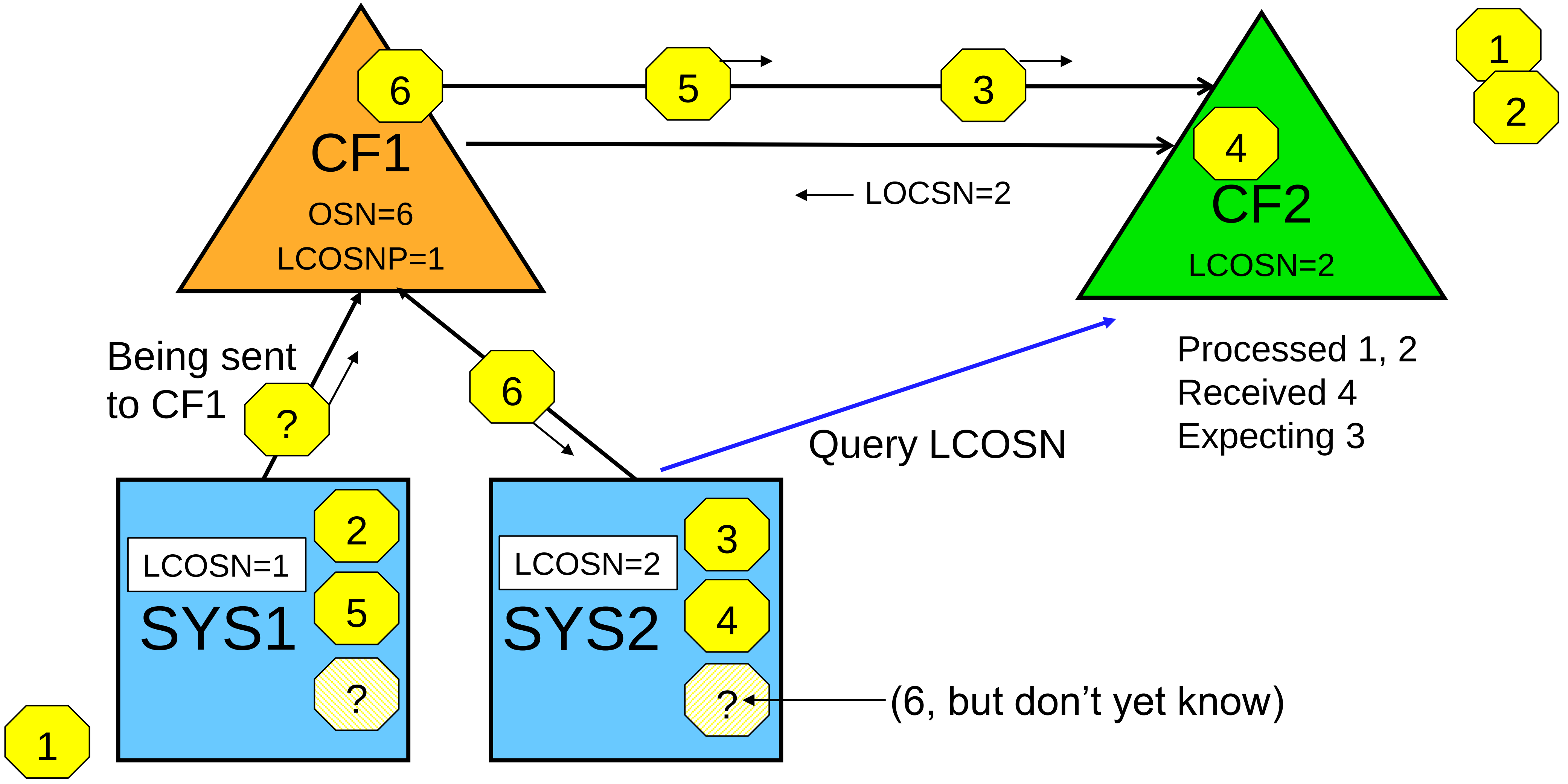
Asynchronous System Managed CF Structure Duplexing Protocol

Available for lock structures.


New
5. Async mirroring of commands' sequenced structure object updates (returns response indicating safe arrival/store, and returns LCOSN)
CF1
(seq#=n)
CF2
(seq#<=n)
6. Ordered execution of sequenced structure updates
Distance impacts?
2. Req out
3. Response (returns OSN(s) and LCOSNP)
Query/syncpoint secondary OSN (at commit points)
z/OS
Knows both primary and secondary OSNs
XES
1. Request in
Exploiter
4. Response out (returns primary seq#)

OSN	= Operation Sequence Number
LCOSN	= Last OSN completed by secondary
LCOSNP	= LCOSN known to primary

Secondary structure instance normally lags the primary



OSN = Operation Sequence Number
LCOSN = Last OSN completed by secondary structure
LCOSNP = LCOSN known to primary structure

Exploiters must be cognizant of the lag

- A request will have executed in the primary structure instance, but might not have been hardened in the secondary instance

- Exploiter must ensure that such hardening has completed before proceeding with any actions that might compromise timely recovery from failures
  - At a minimum, want a fast, easy way to verify. But…
  - Might have to query the secondary CF to find out. And…
  - In worst case, might need to wait until request is processed by secondary.

- Potentially an additional overhead for asynchronous duplexing
  - Protocol loses efficiency as the frequency of queries or waiting increases (so there are measurements and reports)

Asynchronously duplexed structure may require "sync up" when primary fails since secondary instance normally lags the primary

- z/OS may need to make sure the secondary instance gets caught up before it can allow traditional failure processing to proceed.

- So each system maintains a Secondary Update Recovery Table (SURT) to log local in-flight updates not yet known to have been hardened in the secondary structure instance.

- If sync up is needed, a sysplex wide coordinator is nominated to:
  - Gather the logs (SURTs) and use them to…
  - Reconstruct the final result of uncommitted in-flight requests, and
  - Update secondary instance to match the reconstructed results

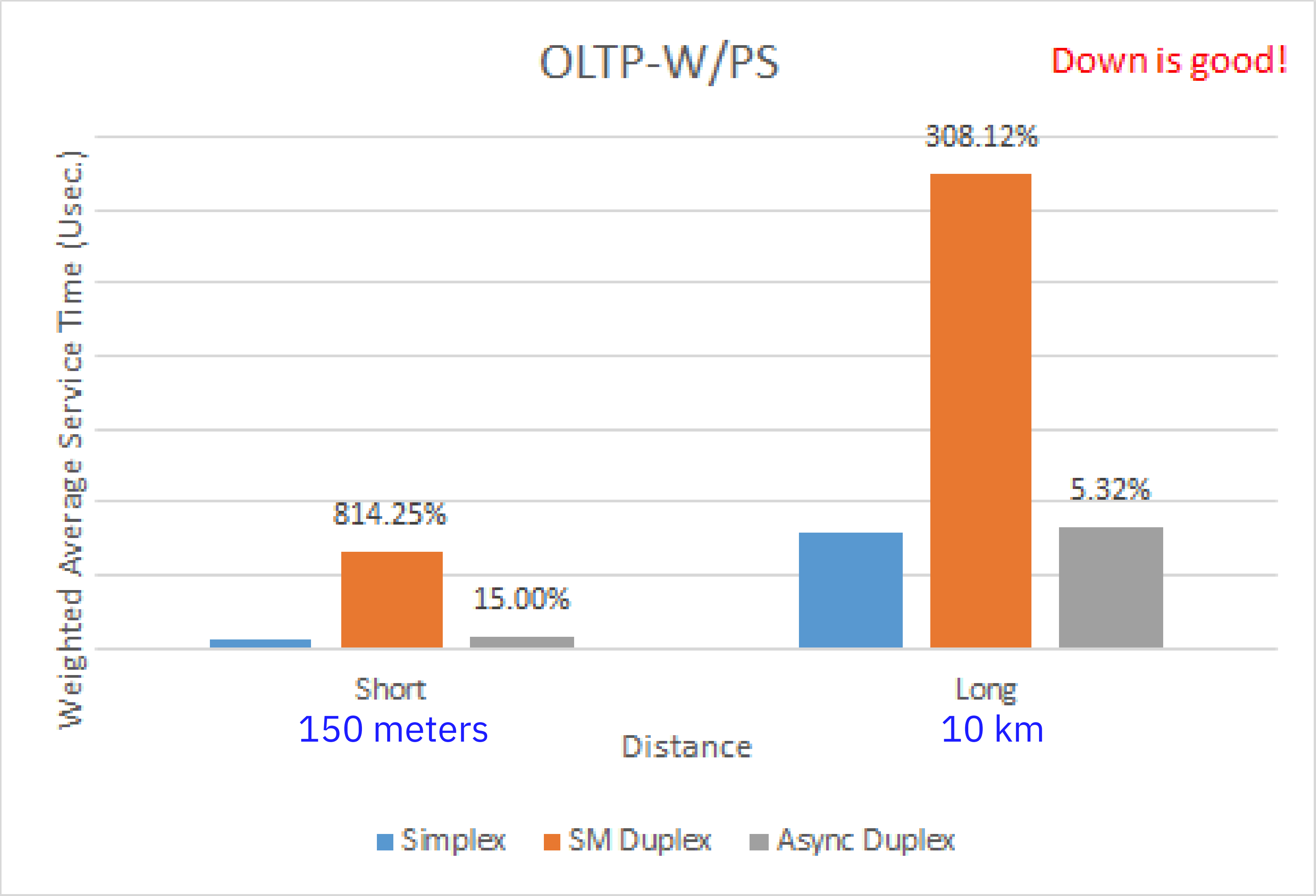
- Connectors:
  - Might have requests held/delayed until sync up is completed
  - Might need to back out uncommitted transactions related to requests with ADupReqSeqNum values higher than the highest hardened request

Failover for async duplex might be a little slower than for sync duplex.

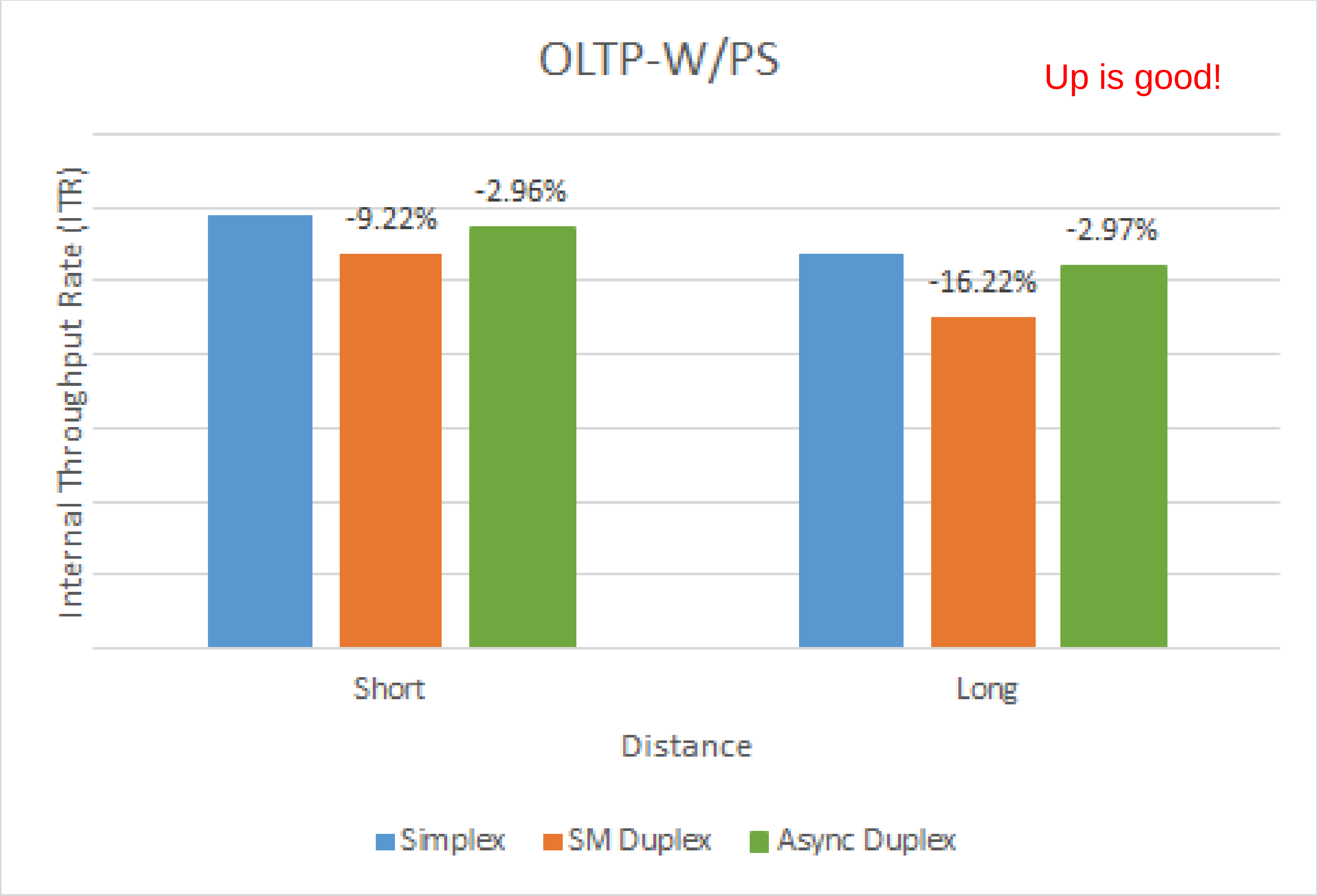
Applies when "lose" SURT for some connector

Is it really worth all this effort? …

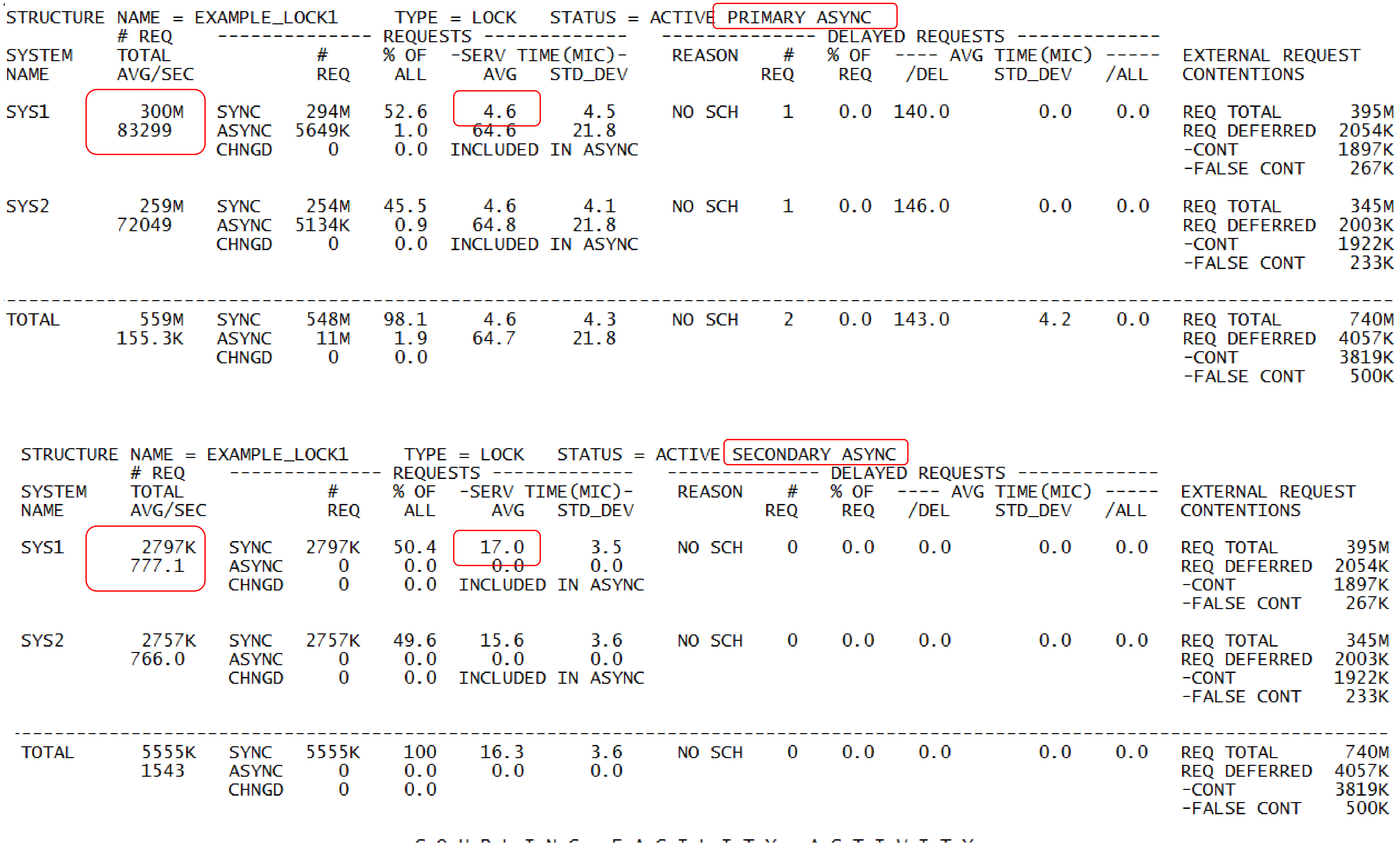
Near simplex service times !



With less overhead !



RMF: Report of Coupling Facility Structure Activity



```
STRUCTURE NAME = EXAMPLE_LOCK1     TYPE = LOCK   STATUS = ACTIVE PRIMARY ASYNC
          # REQ    -------------- REQUESTS -------------    -------------- DELAYED REQUESTS -------------
SYSTEM    TOTAL               #     % OF   -SERV TIME(MIC)-    REASON     #    % OF   ---- AVG TIME(MIC) -----    EXTERNAL REQUEST
NAME      AVG/SEC            REQ    ALL      AVG   STD_DEV                REQ   REQ    /DEL      STD_DEV   /ALL     CONTENTIONS

SYS1        300M   SYNC    294M    52.6      4.6      4.5    NO SCH     1    0.0  140.0         0.0     0.0    REQ TOTAL       395M
           83299   ASYNC   5649K    1.0     64.6     21.8                                                     REQ DEFERRED   2054K
                   CHNGD      0     0.0   INCLUDED IN ASYNC                                                    -CONT          1897K
                                                                                                               -FALSE CONT     267K

SYS2        259M   SYNC    254M    45.5      4.6      4.1    NO SCH     1    0.0  146.0         0.0     0.0    REQ TOTAL       345M
           72049   ASYNC   5134K    0.9     64.8     21.8                                                     REQ DEFERRED   2003K
                   CHNGD      0     0.0   INCLUDED IN ASYNC                                                    -CONT          1922K
                                                                                                               -FALSE CONT     233K

----------------------------------------------------------------------------------------------------------------------------------
TOTAL       559M   SYNC    548M    98.1      4.6      4.3    NO SCH     2    0.0  143.0         4.2     0.0    REQ TOTAL       740M
          155.3K   ASYNC    11M     1.9     64.7     21.8                                                     REQ DEFERRED   4057K
                   CHNGD      0     0.0                                                                       -CONT          3819K
                                                                                                               -FALSE CONT     500K


STRUCTURE NAME = EXAMPLE_LOCK1     TYPE = LOCK   STATUS = ACTIVE SECONDARY ASYNC
          # REQ    -------------- REQUESTS -------------    -------------- DELAYED REQUESTS -------------
SYSTEM    TOTAL               #     % OF   -SERV TIME(MIC)-    REASON     #    % OF   ---- AVG TIME(MIC) -----    EXTERNAL REQUEST
NAME      AVG/SEC            REQ    ALL      AVG   STD_DEV                REQ   REQ    /DEL      STD_DEV   /ALL     CONTENTIONS

SYS1       2797K   SYNC   2797K    50.4     17.0      3.5    NO SCH     0    0.0    0.0         0.0     0.0    REQ TOTAL       395M
           777.1   ASYNC      0     0.0      0.0      0.0                                                     REQ DEFERRED   2054K
                   CHNGD      0     0.0   INCLUDED IN ASYNC                                                    -CONT          1897K
                                                                                                               -FALSE CONT     267K

SYS2       2757K   SYNC   2757K    49.6     15.6      3.6    NO SCH     0    0.0    0.0         0.0     0.0    REQ TOTAL       345M
           766.0   ASYNC      0     0.0      0.0      0.0                                                     REQ DEFERRED   2003K
                   CHNGD      0     0.0   INCLUDED IN ASYNC                                                    -CONT          1922K
                                                                                                               -FALSE CONT     233K

----------------------------------------------------------------------------------------------------------------------------------
TOTAL      5555K   SYNC   5555K     100     16.3      3.6    NO SCH     0    0.0    0.0         0.0     0.0    REQ TOTAL       740M
            1543   ASYNC      0     0.0      0.0      0.0                                                     REQ DEFERRED   4057K
                   CHNGD      0     0.0                                                                       -CONT          3819K
                                                                                                               -FALSE CONT     500K

                          C O U P L I N G   F A C I L I T Y   A C T I V I T Y
```

RMF: Asynchronous CF Duplexing Summary

```
ASYNCHRONOUS CF DUPLEXING SUMMARY
-------------------------------------------------------------------------------------------------------------------------

                             --------- ASYNC DUPLEX CF OPERATIONS ---------    --- ASYNC DUPLEX SYNC_UP REQUESTS ---
       STRUCTURE              TOTAL     --TRANSMIT TIME--   --SERVICE TIME--     TOTAL  #SUSPEND    --SUSPEND TIME—
  TYPE  NAME                                  AVG   STD_DEV      AVG   STD_DEV                            AVG   STD_DEV

  LOCK  EXAMPLE_LOCK1        740516K     10.3        5.8      12.9       8.8      8701K       117     138.5      50.3
```

Information about cost to inquire about (and wait for) progress in secondary CF.

Provides a sense of how long it is taking for request information to arrive at the secondary structure.

Provides a sense of how much the secondary tends to lag the primary (transmit plus processing time).

Lock structure duplexing is now practical

- Performance is very similar to simplex operation
  - Even at distance !
- With all the benefits of robust duplex failure recovery
  - Though failover is a bit more work since secondary is not identical copy
  - And so recovery process is not necessarily transparent to exploiter
- Exploiter participation?
  - Not needed for lock structures without record data if connectors support system managed processes (XES can do it all) (if any out there)
  - Is needed for structures with record data (all the ones I know of)
    - Exploiter must be prepared to roll back a transaction initiated by a failed connector that could not be committed because updates are missing in the surviving secondary

So what’s needed?

Requirements for Asynchronous Duplexing of CF Lock Structures

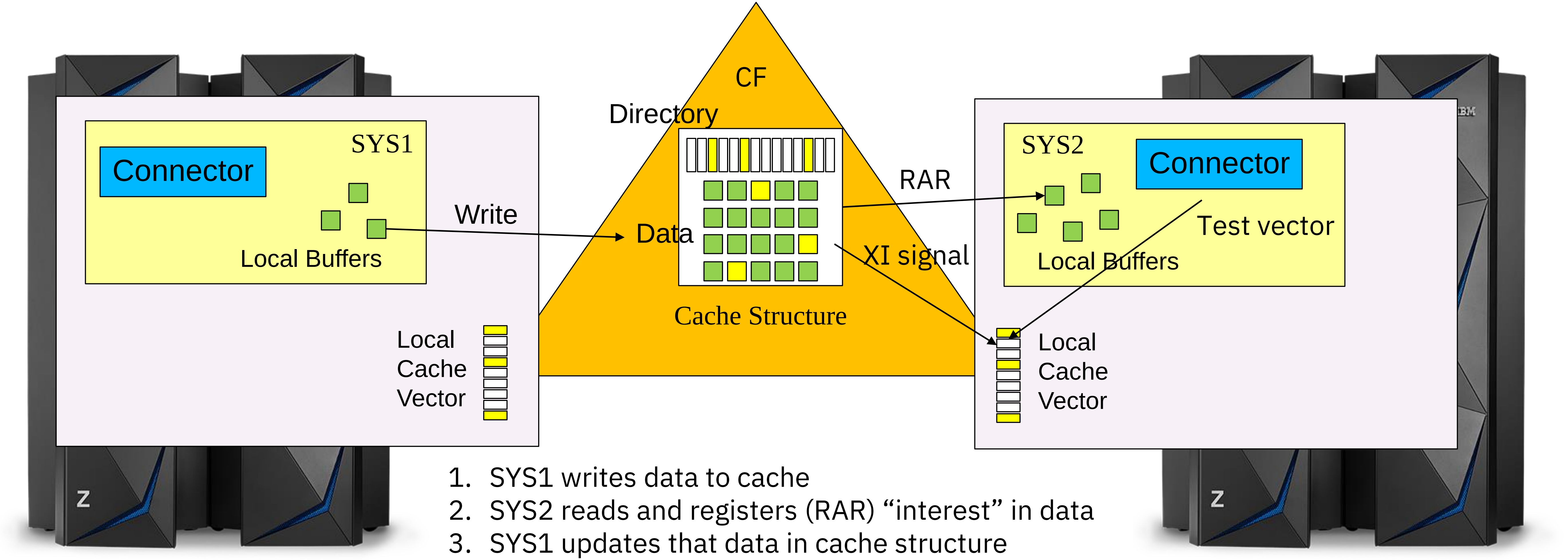
- At least two peer connected coupling facilities
  - Either CFLEVEL=21 (at minimum service level 02.16) or CFLEVEL=22
  - z13 GA2+, z13s, z14, z14 ZR1
- z/OS V2.2 with APARs:
  - OA47796, OA49148, OA51945, OA52015, and OA52618
- z/OS V2.3 with APARs:
  - OA52618
- Db2® 12 with enabling APAR PI66689
- IRLM 2.3 with APAR PI68378

All systems in the sysplex must be capable of doing async duplex protocols.

Asynchronous Cross-Invalidate for CF Cache Structures

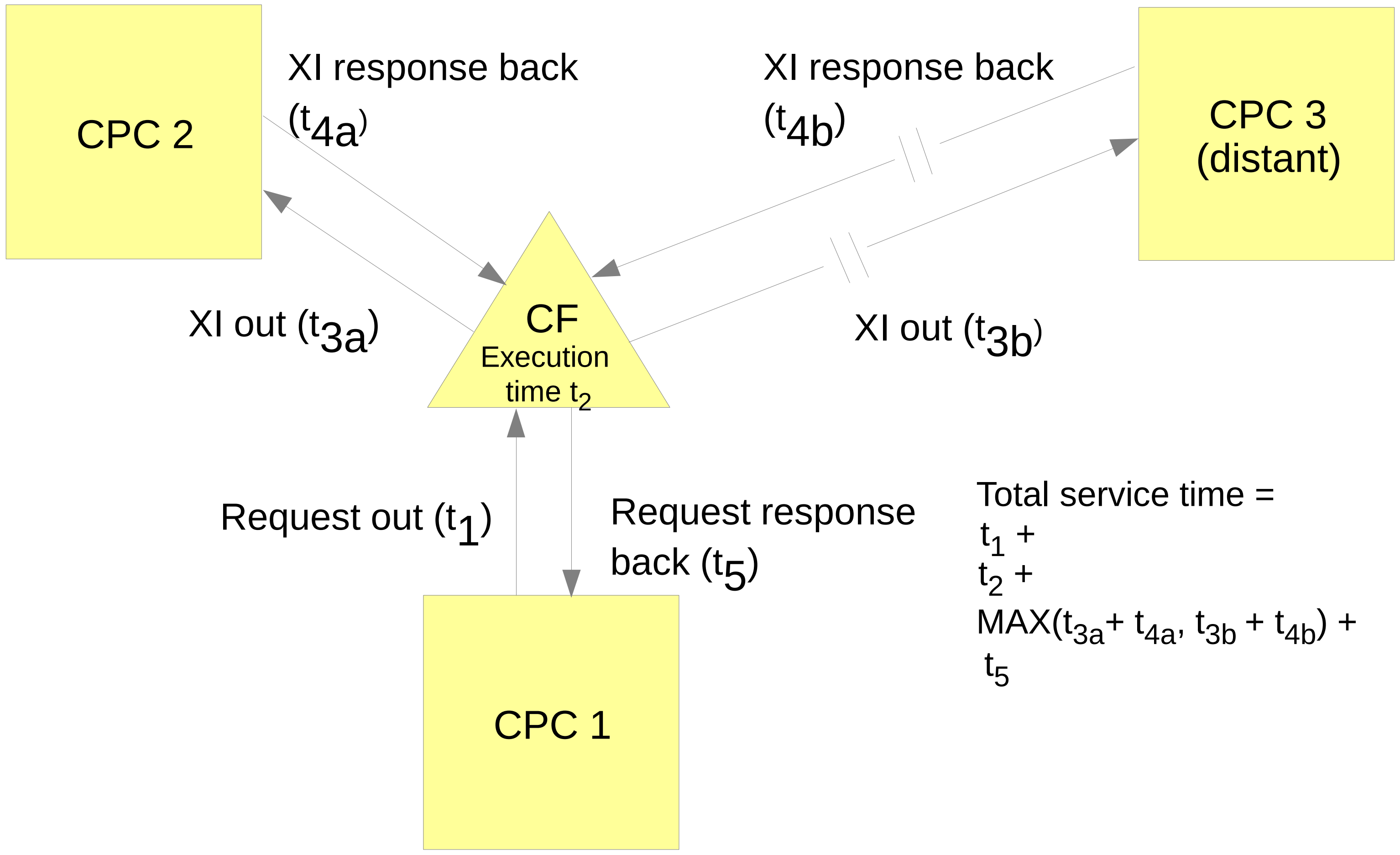
Performance improvements for “transactional cache” processing

Background: Cache Structures and Cross Invalidates (XI)



1. SYS1 writes data to cache
2. SYS2 reads and registers (RAR) “interest” in data
3. SYS1 updates that data in cache structure
4. CF sends XI signal to update local cache vector on SYS2 to indicate that the copy of the data in the local buffers from step 2 is no longer current
5. Before using the local copy, SYS2 tests the local vector, discovers the need to refresh its local copy before proceeding

Service time for cache request that spawns (synchronous) XI signals

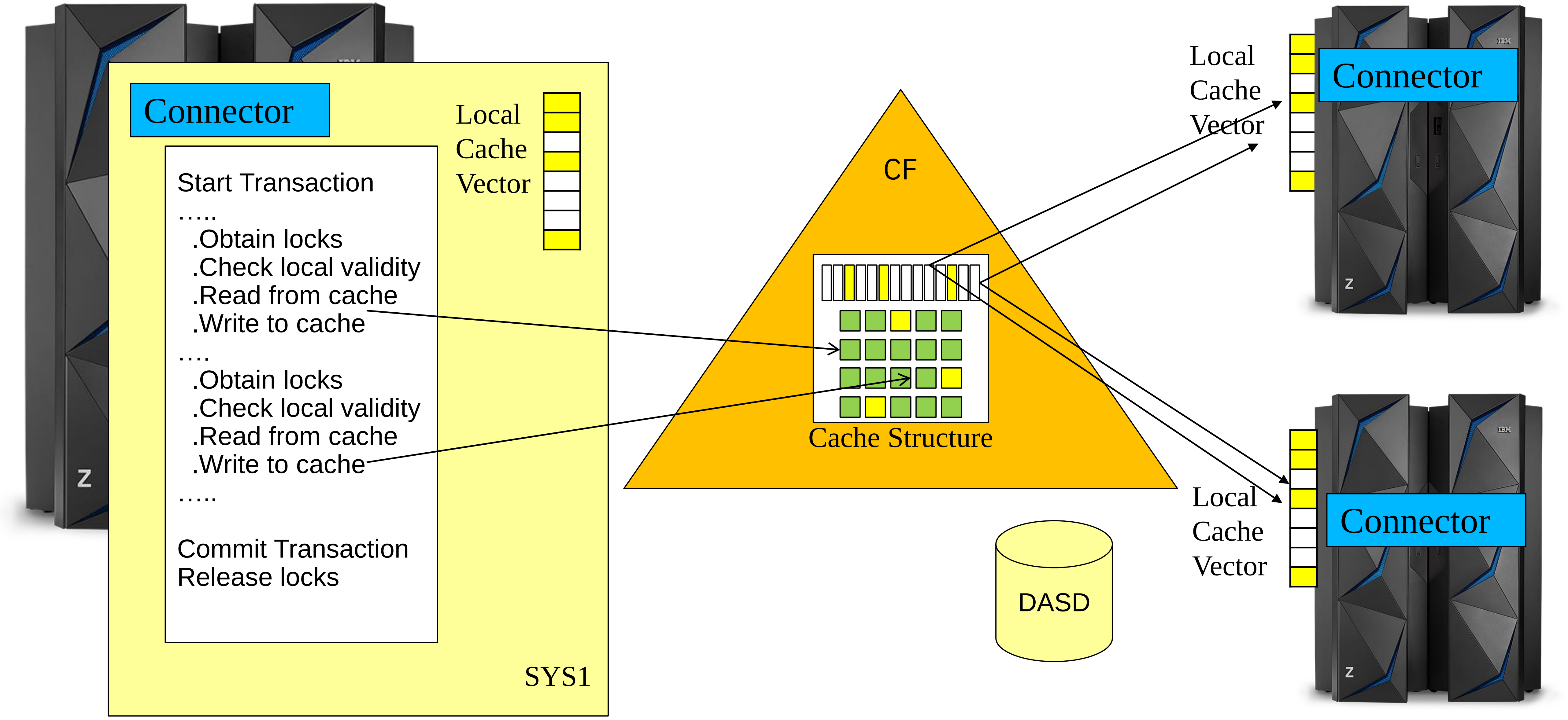


Total service time =
t_1 +
t_2 +
MAX(t_{3a}+ t_{4a}, t_{3b} + t_{4b}) +
t_5

(XI’s run in parallel)

In short, cache request will not complete until all the cross-invalidates complete.

Typical Transactional Cache Processing

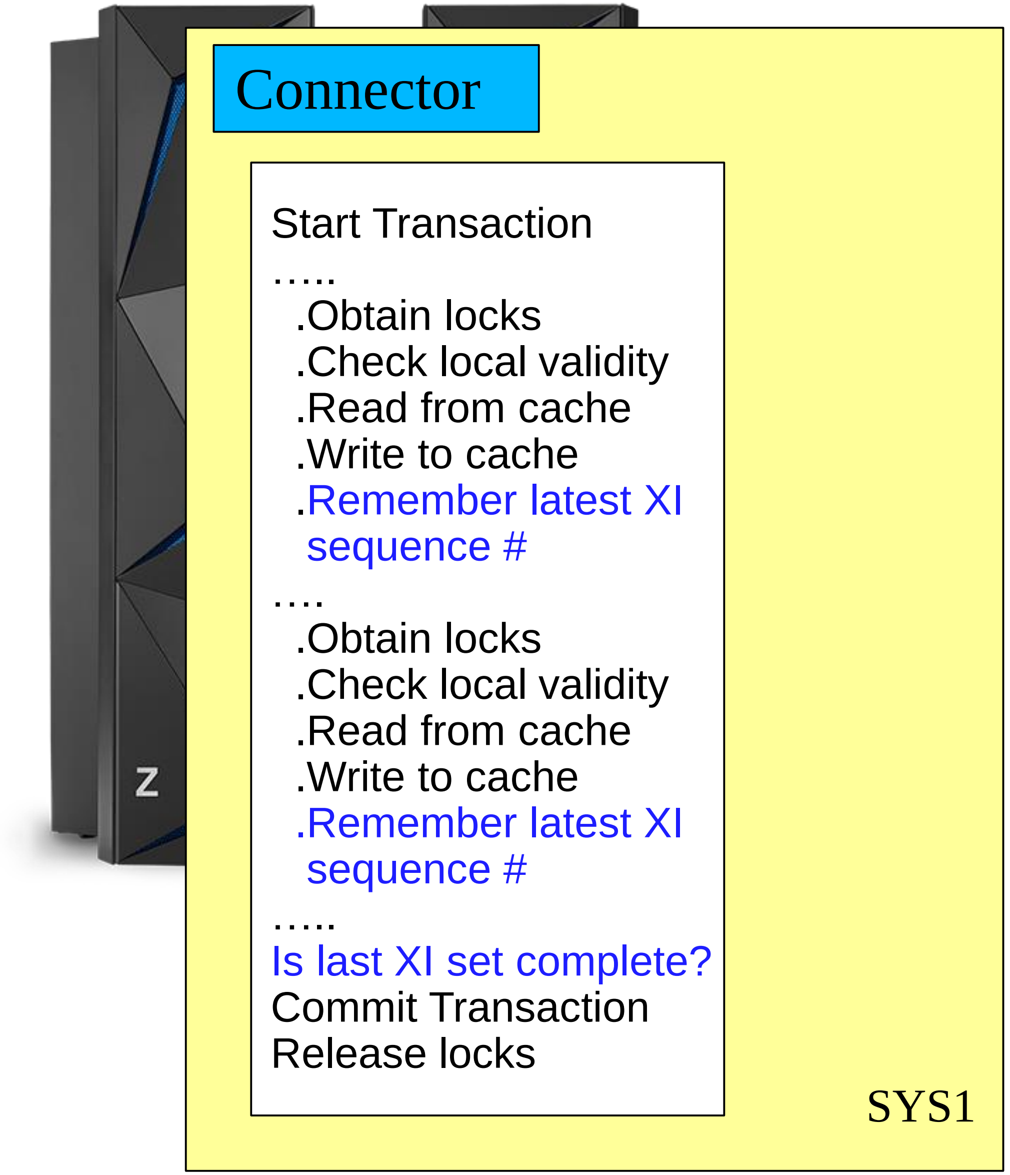


Start Transaction
…..
.Obtain locks
.Check local validity
.Read from cache
.Write to cache
….
.Obtain locks
.Check local validity
.Read from cache
.Write to cache
…..

Commit Transaction
Release locks

Transactional cache processing with async XI

- Exploiter knows the set of cache requests associated with a given transaction
- As each cache request completes, z/OS returns to the exploiter, the sequence number assigned by the CF to the set of XI's spawned by that request
- The exploiter tracks the highest such sequence number associated with the transaction
- Prior to committing the transaction, the exploiter must verify that this last set of XI's has completed in the CF
  - If set N is complete, then so are all sets < N
  - May have to wait for said completion.



Connector

Start Transaction
…..
.Obtain locks
.Check local validity
.Read from cache
.Write to cache
.Remember latest XI sequence #
….
.Obtain locks
.Check local validity
.Read from cache
.Write to cache
.Remember latest XI sequence #
…..
Is last XI set complete?
Commit Transaction
Release locks

SYS1

Asynchronous Cross-Invalidate for CF Cache Structures

- Enables improved efficiency in CF data sharing by adopting a more transactional behavior for cross-invalidate (XI) processing which is used to maintain coherency/consistency of data managers' local buffer pools across the sysplex
  - Instead of performing XI signals synchronously on **every cache update request that causes them,** data manager exploiters will be able to "opt in" for the CF to perform these XIs asynchronously (and then sync them up with the CF at or before transaction completion)
  - As long as all asynchronous XI signals complete by the time transaction locks are released, data integrity is preserved
- Benefits
  - Provides faster completion of cache update CF requests that generate XIs, *especially with cross-site distance involved*
  - Provides improved cache structure service times and coupling efficiency
  - Shortens elapsed time of the transaction

Performance Evaluation

- Asynchronous XI is expected to yield service time improvement for individual commands since XI time no longer included
  - Applies to both synchronous and asynchronous command execution
    - Note that command execution mode is distinct from XI delivery mode
  - CPU time / cost reduction for synchronous command execution
  - Expect less frequent heuristic conversion to asynchronous execution
- No new SMF records or RMF data
- No new accounting and measurement (IXLMG) data
- **Transaction throughput improvement is the real measure of success**

Requirements

- z/OS V2R3 (HBB77B0) or z/OS V2R2 (HBB77A0) with APAR OA54688
  - APAR must be applied to every exploiting system in the sysplex
- Z14 GA2 (CFLEVEL 23) Coupling Facility
  - CFLEVEL 23 toleration APAR OA54985 required on all systems
- Requires explicit **data manager exploitation/participation**
  - Not transparent to the data manager
  - Initially, DB2 V12-based exploitation
    - Exploitation by other data managers is possible

CFLEVEL=22 Scalability Enhancements

Work management changes – more throughput

Synchronous SM duplexing protocol – fewer instances of duplexing deadlocks

List notification enhancements – better responsiveness with less CPU overhead

CFLEVEL=22 Scalability Enhancements

z14 GA1

- Refactored CF work management and dispatching algorithms to improve efficiency and scalability for CF images
  - Exploitation of new z14 instructions, compiler options
  - Elimination of ordered work queues allows CF to optimize its processing for simpler FIFO processing of requests
  - Increases throughput !
- For System Managed (SM) synchronously duplexed structures:
  - In cooperation with z/OS, the potential for duplex requests to deadlock is reduced (thereby avoiding the inefficiencies they induce)
- If a z14 CF is to host a SM synchronously duplexed structure, the z/OS images connected to that structure must be running:
  - z/OS 2.3, or
  - z/OS 2.2 or z/OS 2.1, with APAR OA52058 installed
- Otherwise, the potential for duplex request deadlock goes UP!
  - An issue for “heavy“ duplex request workloads

CFLEVEL=22 List Notification Enhancements for CF list structures

In cooperation with z/OS, the z14 CF offers several “list notification” enhancements to better meet the needs of list structure exploiters:

- Aggressive notification (for every new list entry)
- Round robin notification (for list and key range monitoring)
- List full / not full transition notification
  - Alternative to empty / not empty transition notification

z/OS systems must be running either:

- z/OS 2.3, or
- z/OS 2.2 with new function APAR OA51862 installed

CFLEVEL=22 Full to not full list transition notification

- When writing to a list structure, the request might fail with a “list full” condition
  - So the write request must be re-driven when the list becomes “not full”.
  - How is the exploiter to know when that happens?
  - Prior to this support the exploiter might:
    - Keep retrying the write until it works
    - Keep retrying the write with a pause between attempts
    - Periodically test to see if the list is empty (?), and retry the write when so
  - But there are tradeoffs between responsiveness and system overhead

- Full to not full list transition notification is the answer !
  - Exploiter list transition exit can be driven when the list becomes “not full”, or
  - Exploiter can test list vector for “not full”

XCF exploits this capability
for Signal Structures in z/OS V2R4

CFLEVEL=23 Scalability Enhancements

Resiliency - CF task hang detection

Availability - Granular latching for structure alter processing

CFLEVEL=23 Scalability Enhancements

z14 GA2
z14 zR1

CF task hang detection

- CFCC dispatcher monitors dispatched tasks for “hangs”
  - “Hang” = task does not return to dispatcher in a timely manner
- Prior to CFLEVEL=23
  - Timely = 60 seconds
  - When a hang is detected, the CF will abort, dump, and reboot the CF image
- With CFLEVEL=23
  - Timely = 2 seconds
  - When a hang is detected, the CF will (in most cases):
    - Confine scope of failure to “structure damage” for the specific CF structure whose command was hung
    - Capture diagnostics with a non-disruptive CF dump (and continue operating without aborting or rebooting the CF image)
- Benefits:
  - Faster detection and recovery from hangs
  - Failure scope more limited (structure vs CF) and therefore less disruptive to the sysplex workload
  - Still have FFDC

No z/OS support is needed.

CFLEVEL=23 Scalability Enhancements

Granular latching for structure alter processing

- Structure Alter processing dynamically modifies in-use structures to expand, contract, or re-apportion structure object ratios (entries, elements, EMCs)
- Prior to CFLEVEL=23
  - Alter serialized by structure-wide latches that temporarily prevents execution of all mainline commands
  - Could be long running, so the CF "time slices" alter processing to create gaps where mainline commands can execute
  - However, the delays for mainline commands awaiting gaps during structure alter does induce transient performance impacts (which can last for minutes at a time)
- With CFLEVEL=23 (but only for list and lock structures, not cache)
  - Alter now serialized by same structure object latches used by all mainline commands
  - A small number of "edge conditions" still need structure-wide latching for serialization
- Benefits:
  - Eliminates performance degradation caused by structure-wide latching

No z/OS support is needed.

CFLEVEL=24 Scalability Enhancements (z15 required)

Availability – Fair Latch Manager (Round 2)

Resiliency – Message Path Resiliency Enhancement

CFLEVEL 24 Exploitation

- Structure and Coupling Facility Storage Sizing with CF Level 24
  - May increase storage requirements when moving from:
    - CF Level 23 (or below) to CF Level 24
    - CFSizer Tool recommended
    - http://www.ibm.com/systems/z/cfsizer
  - As in prior CF Levels, ensure that the CF LPAR has at least 512 MB storage for CFCC µcode
- CF Enhancements:
  - CF Fair Latch Manager 2
    - Intended to improve work management efficiency to contribute to better CF scaling as well as arbitration for internal CF serialization of resources.
  - Message Path SYID Resiliency Enhancement
    - Intended to improve resiliency of message path connectivity through new transparent recovery processing for certain types of link initialization errors that can occur as z/OS images are IPLed into the sysplex.

CFLEVEL 24 Fair Latch Manager

- Improves CF work management efficiency
  - Global suspend queue used much less frequently
  - Latch-specific waiter queues used for the exact instance of the latch they are requesting, and in exact order
  - **Less global contention / cache misses in the CF**

- Improves “fairness” of arbitration for internal CF resource latches across tasks
  - When latch is released, the queue transfers ownership of the latch to the next request in-line
  - No possibility of unfairness or “cutters” in line between the time the latch is released vs. when it is re-obtained

z/OS support is needed.

CFLEVEL 24 Resiliency Enhancement

- Current problem
  - When a z/OS system IPLs, message paths are supposed to be deactivated via system reset, and their SYIDs are supposed to be cleared in the process.
  - During the IPL, z/OS will then re-activate the message paths with a new SYID that represents the new instance of z/OS that is currently using the paths.
  - On rare occasions, a message path may not get deactivated during system reset / IPL processing, leaving the message path active with the z/OS image’s OLD, now-obsolete SYID.
  - Since the path erroneously remained active, z/OS does not see any need to re-activate it with a new, correct SYID.
  - **From the CF’s perspective, the incorrect SYID persists, and prevents delivery of signals to the z/OS image currently using that message path.**
- Solution
  - New resiliency mechanism that will transparently recover for this “missing” message path deactivate (if and when that ever happens).
  - CF will provide additional information to z/OS about every message path that appears active: the current SYID with which the message path is currently registered in the CF.
  - Whenever z/OS interrogates the state of the message paths to the CF, z/OS will check this SYID information for currency and correctness. If there is an obsolete or incorrect SYID in the message path for any reason, z/OS will:
    - Request non-disruptive gathering of diagnostic information for the affected message paths and CF image
    - **Re-activate the message path with the correct SYID for the current z/OS image, to seamlessly correct the problem**

z/OS support is needed.

CFLEVEL 24 Requirements

- Software coexistence requirements:
  - Install the preconditioning and exploitation PTFs for CFLEVEL 17-23 throughout the sysplex
  - As with all required service, use FIXCAT value **IBM.Device.Server.z15-8561.RequiredService**
    - PTFs available for V2R2 and higher

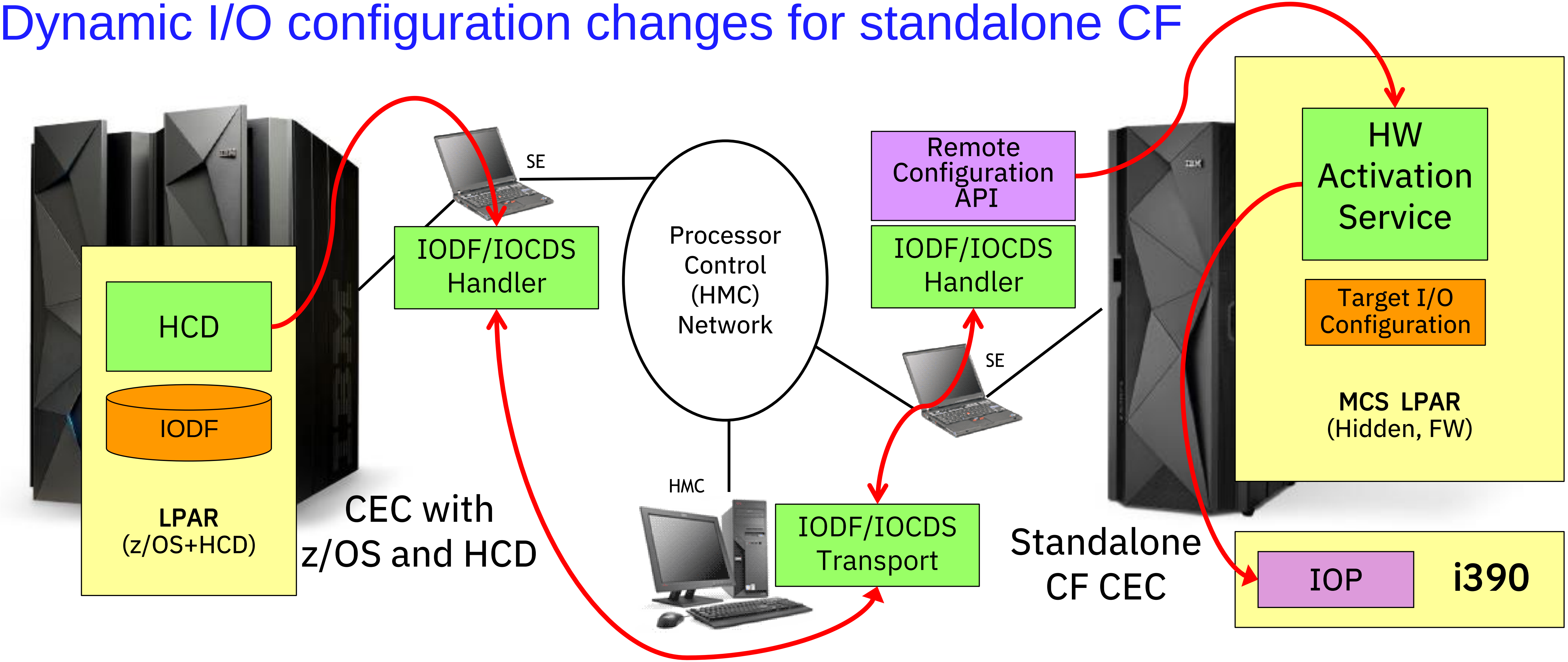
Nondisruptive I/O configuration changes for a stand-alone CF

Driving Dynamic I/O hardware-only activations remotely from another CEC

Dynamic I/O configuration changes for standalone CF

- Problem
  - Standalone CF does not have co-resident z/OS image that can run the Hardware Configuration Dialog (HCD) to make hardware only Dynamic I/O configuration changes on behalf of the CF partition(s)
  - Therefore, such changes require disruptive IML or POR of the standalone CF CEC
  - Which increases complexity, impacts sysplex availability, and reduces redundancy
- Solution: Drive Dynamic I/O hardware-only activations remotely from another CEC
  - New support enables the “driving” HCD running on a CEC with z/OS to:
    - Send target configuration from modified IODF to the standalone CF
    - Drive the Dynamic I/O activate and all associated recovery/management functions
  - On the standalone CF CEC, a firmware defined “MCS LPAR” acts as the local agent to accomplish the Dynamic I/O changes without disruption
    - CF image reacts to these changes just as if they’d been driven through a co-resident z/OS-based HCD
- Benefits
  - Simple, Dynamic I/O changes for standalone CF without disruptive IML or POR

Dynamic I/O configuration changes for standalone CF



HCD running on CEC with z/OS LPAR uses the HMC processor control network to remotely drive Dynamic I/O changes on the standalone CF CEC.

The Master Control Services (MCS) LPAR

- A firmware based appliance that provides the “hardware activation” service
  - Licensed Machine Code (LMC) partition
  - Fully managed by the z14 GA2 firmware
  - Included with the base firmware; no need to order a feature code
- You will need to do a POR to establish the MCS LPAR on the standalone CF CEC before this new capability can be used
  - After this “last” POR, all subsequent Dynamic I/O changes can be done dynamically without disruption
- This is a PR/SM based solution, does not require Dynamic Partition Manager (DPM) mode

Dynamic I/O for a standalone CF - Requirements

- Both the CF CEC and the “driving” z/OS CEC need to be running z14 GA2 firmware
  - The CF CEC will need (one last) POR to activate the MCS LPAR
  - The driving z/OS CEC needs the firmware (but an IML/POR is not required)
- The SE and HMC needs to be at the z14 GA2 service level as well
  - Version 2.14.1
- The “driving” z/OS will need PTFs:
  - IO25603 (HCM)
  - OA53952 (IOS)
  - OA54912 (HCD)
  - OA55404 (IOCP)

CF Structure Encryption

Enhance security and data protection

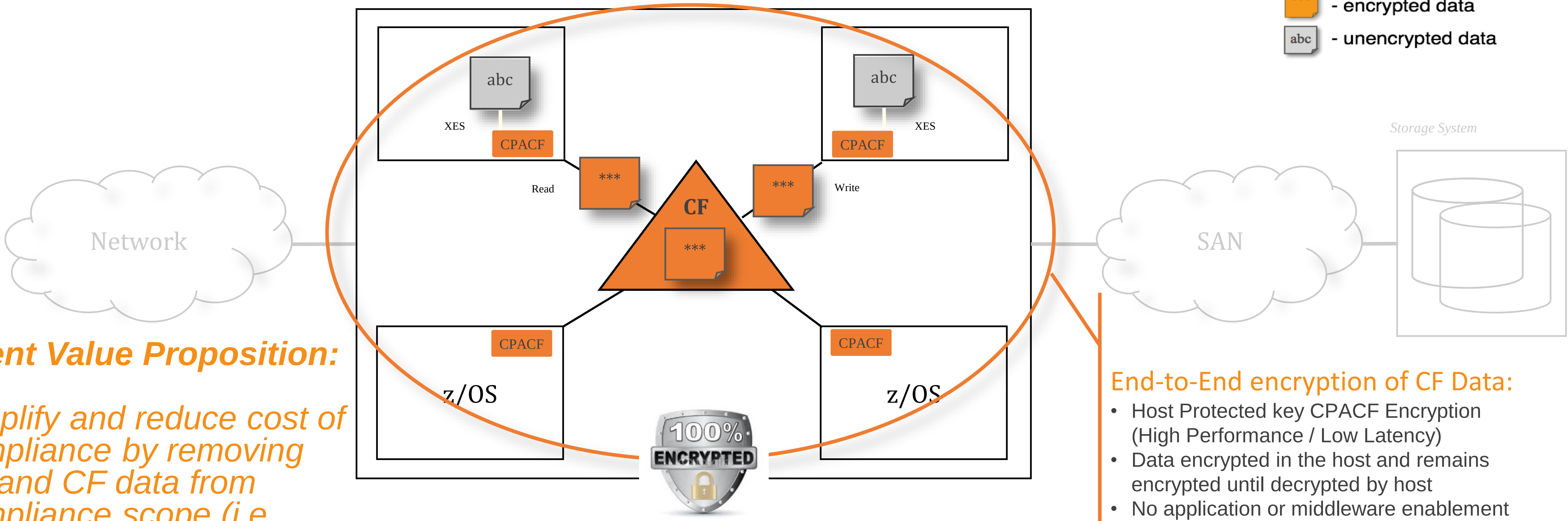
Coupling Facility Data Encryption in a Parallel Sysplex

Protection of Data at-rest and in-flight (CF)

Legend:

*** - encrypted data

abc - unencrypted data



Client Value Proposition:

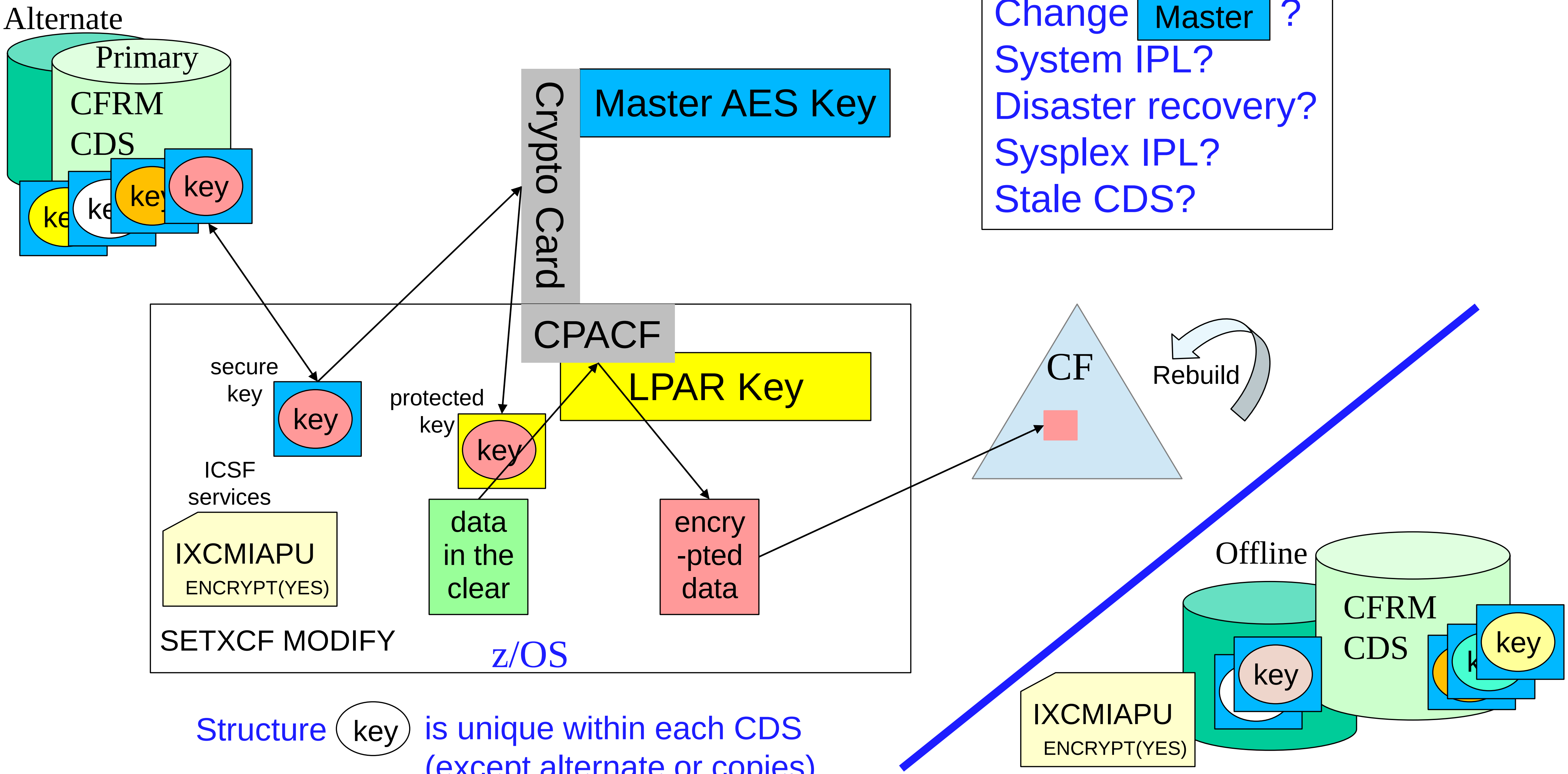
Simplify and reduce cost of compliance by removing CF and CF data from compliance scope (i.e. ability to encrypt all CF data)

Especially in cases where CF and/or links are outside the data center.

End-to-End encryption of CF Data:

- Host Protected key CPACF Encryption (High Performance / Low Latency)
- Data encrypted in the host and remains encrypted until decrypted by host
- No application or middleware enablement required
- List & Cache Structures only – *No Lock!*

CF Structure Encryption – The Big Picture



Change key ?
Change Master ?
System IPL?
Disaster recovery?
Sysplex IPL?
Stale CDS?

Structure key is unique within each CDS
(except alternate or copies)

Software Dependencies

- z/OS 2.3 is required for CF structure encryption
  - If a down level system is connected to a structure, the structure cannot be encrypted
  - If a structure is encrypted, a down level system cannot connect to the structure

- **All systems in the sysplex must have the same AES Master Key**
  - In general, implies all systems in the sysplex must share the same ICSF CKDS
    That is, all systems in the sysplex must be in the came CKDS cluster
    - Ensures that the AES Master Key is activated
    - Ensures that all systems have the same AES Master Key
    - Enables sysplex wide coordinated changes to the master key so that the “same master key for all systems” requirement is preserved
      - **ICSF notifies XCF of AES Master Key change, which spurs XCF to rewrap encryption keys in CFRM CDS**
  - So ICSF services must be activated on all systems in the sysplex

The zSecure audit tool does not yet support encrypted CF structures.

SSD/BCPii Enhancements

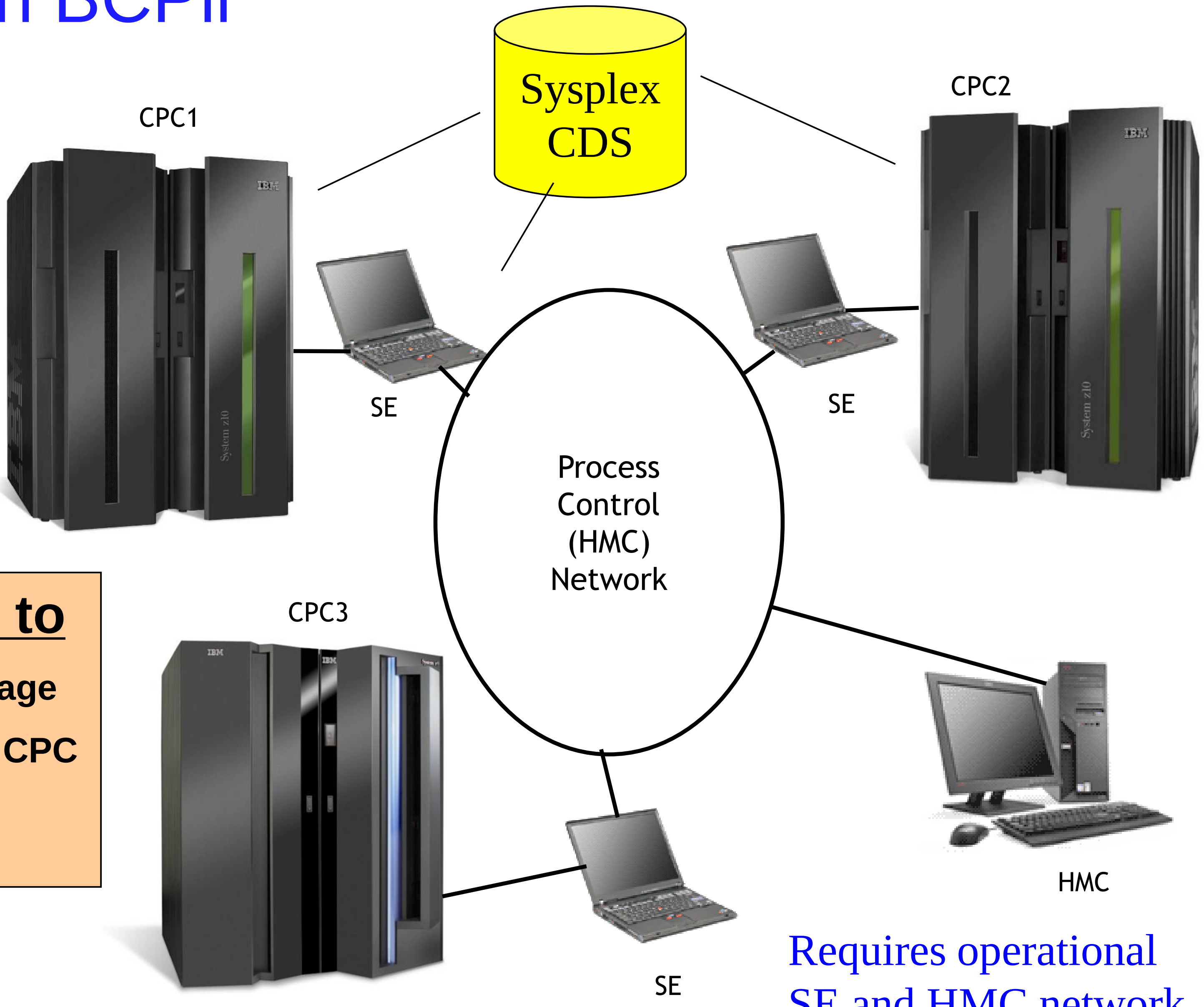
More meaningful health check for SSD

Dynamic CPC name changes

z/OS V1R11 - XCF with BCPii

z/OS Images

(not VM guests)



XCF uses BCPii to

- Obtain identity of an image
- Query status of remote CPC and image
- Reset an image

Requires operational SE and HMC network

XCF Health Check - XCF_SYSSTATDET_PARTITIONING

Old behavior

- The check only looked at environmental conditions related to XCF configuration
  - Sysplex CDS appropriately configured?
  - XCF FUNCTIONS switch ENABLED?
- You could pass the check even though the function was incapable of being used !

New behavior

- If you pass the check, the function is fully operational
  - BCPii services are available; have appropriate connections to CPC’s; XCF has necessary SAF authority
- Health check is dynamically triggered in response to changes that might affect the result of the check
- Additional data provided in verbose mode

Conclusion

Just in case you missed some key functions in z/OS V2R2

- **XCF message isolation**
- **CFRM policy site preferences**
- Health based routing
- SMT exploitation
- Need new ARM CDS
- **CF Gain Ownership protection**
- IXCQUERY for CDS and policies

See appendix for details.

Summary

- z/OS V2R4
  - Transport Class Simplification
  - Single system scope LOGRx CDS for GDPS
- IBM z14 GA1
  - More: cores, coupling connectivity
  - STP stratum level 4
  - CFLEVEL=22 list notification enhancements
- IBM z14 GA2
  - Dynamic I/O configuration changes for standalone CF
  - STP enhanced support for splitting and merging CTNs
  - CFLEVEL=23 enhancements to reduce impact of structure alter processing for list and lock structures
- Asynchronous XI for cache structures to improve Db2 transaction performance
- Async duplexing of (Db2) lock structures
  - Robust failure recovery with simplex like response times, even at distance
- CF Structure Encryption
  - Provides better protection against potential breaches that could otherwise expose sensitive data

IBM Systems Worldwide Client Experience Centers

IBM Systems Worldwide
Client Experience Centers

IBM Systems Worldwide Client Experience Centers *maximize IBM Systems competitive advantage in the Cloud and Cognitive era by providing access to world class technical experts and infrastructure services to assist Clients with the transformation of their IT implementations..*

9 Worldwide Locations (* also Infrastructure Hubs):

Austin TX , *Poughkeepsie NY, Rochester MN, Tucson AZ, *Beijing CHINA, Boeblingen GERMANY, Guadalajara MEXICO,*Montpellier FRANCE, Tokyo JAPAN

ROCHESTER
TUCSON
POUGHKEEPSIE
AUSTIN
GUADALAJARA
BOEBLINGEN
MONTPELIER
BEJING
TOKYO

Client Experience

Tailored, in-depth technology
Innovation Exchange Events
Relationship building
Demonstrations
Meetups
Solution workshops
Remote options

(Inbound & Outbound)

Architecture & Design

Advise clients, "Art of the Possible"
Discovery & Design
Workshops, Consulting,
Showcases, Reference Architectures, Co-Creation of assets

(Inbound & Outbound)

Infrastructure Solutions

Benchmarks, MVP & Proof of Technology
"Test Drives"
Demonstrations
Infrastructure Services
Certify ISV solutions
Hosting
Cloud Environment

(Inbound to Centers)

Content

Content Development
IBM Redbooks
Training Courses
Video courses
"Test Drives"
Demonstrations

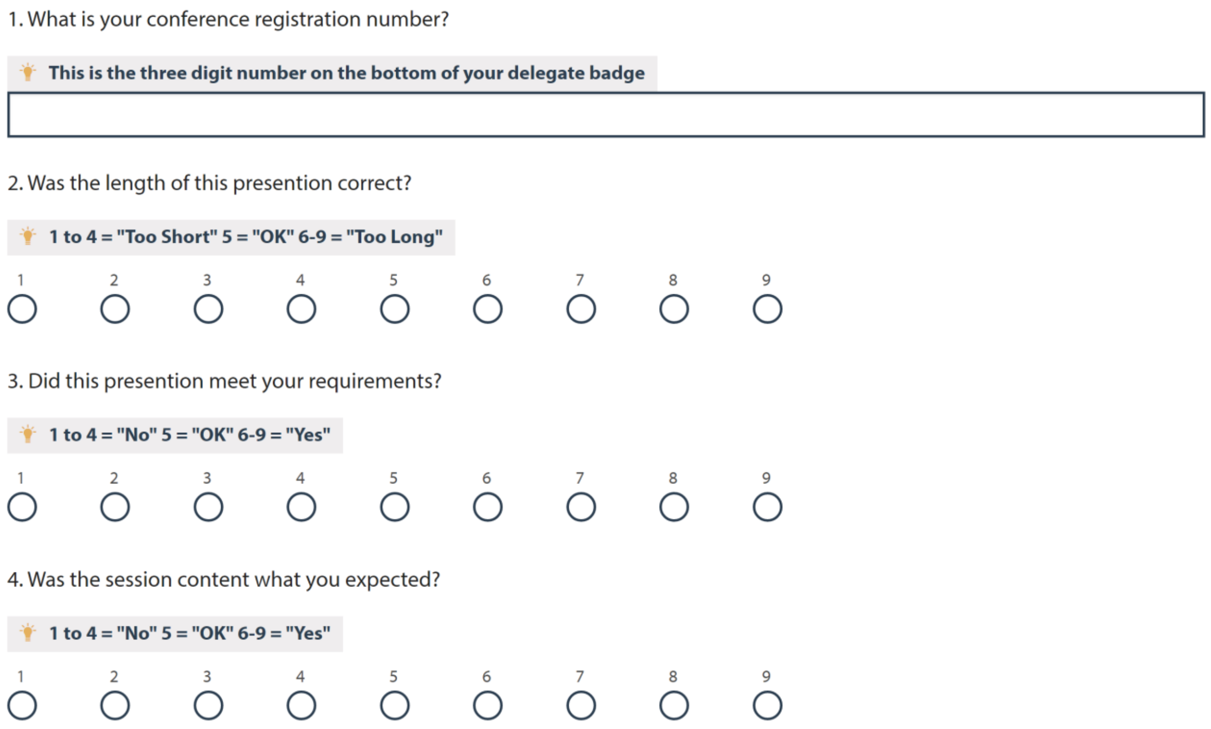
NEW: *Co-Creation Lab; CEC Cloud; RedHat Center of Competency*

For further information, please contact the Centers via email at:
ccenter@us.ibm.com

Please submit your session feedback!

- Do it online at http://conferences.gse.org.uk/2019/feedback/bj

- This session is BJ



1. What is your conference registration number?

This is the three digit number on the bottom of your delegate badge

2. Was the length of this presention correct?

1 to 4 = "Too Short" 5 = "OK" 6-9 = "Too Long"

1	2	3	4	5	6	7	8	9
○	○	○	○	○	○	○	○	○

3. Did this presention meet your requirements?

1 to 4 = "No" 5 = "OK" 6-9 = "Yes"

1	2	3	4	5	6	7	8	9
○	○	○	○	○	○	○	○	○

4. Was the session content what you expected?

1 to 4 = "No" 5 = "OK" 6-9 = "Yes"

1	2	3	4	5	6	7	8	9
○	○	○	○	○	○	○	○	○

For more information

z/OS Publications

- *MVS Setting Up a Sysplex*
- *MVS Initialization and Tuning*
- *MVS Systems Commands*
- *MVS Diagnosis: Tools and Service Aids*
- *z/OS V2R3 Migration*
- *z/OS Planning for Installation* (GA32-0890)
- *z/OS MVS Programming: Callable Services for High Level Languages*
  - Documents BCPii Setup and Installation and BCPii APIs

Sysplex-related Redbooks

- System z Parallel Sysplex Best Practices, SG24-7817
- Considerations for Multi-Site Sysplex Data Sharing, SG24-7263
- Server Time Protocol Planning Guide, SG24-7280
- Server Time Protocol Implementation Guide, SG24-7281
- Server Time Protocol Recovery Guide, SG24-7380

- Exploiting the IBM Health Checker for z/OS Infrastructure, REDP-4590

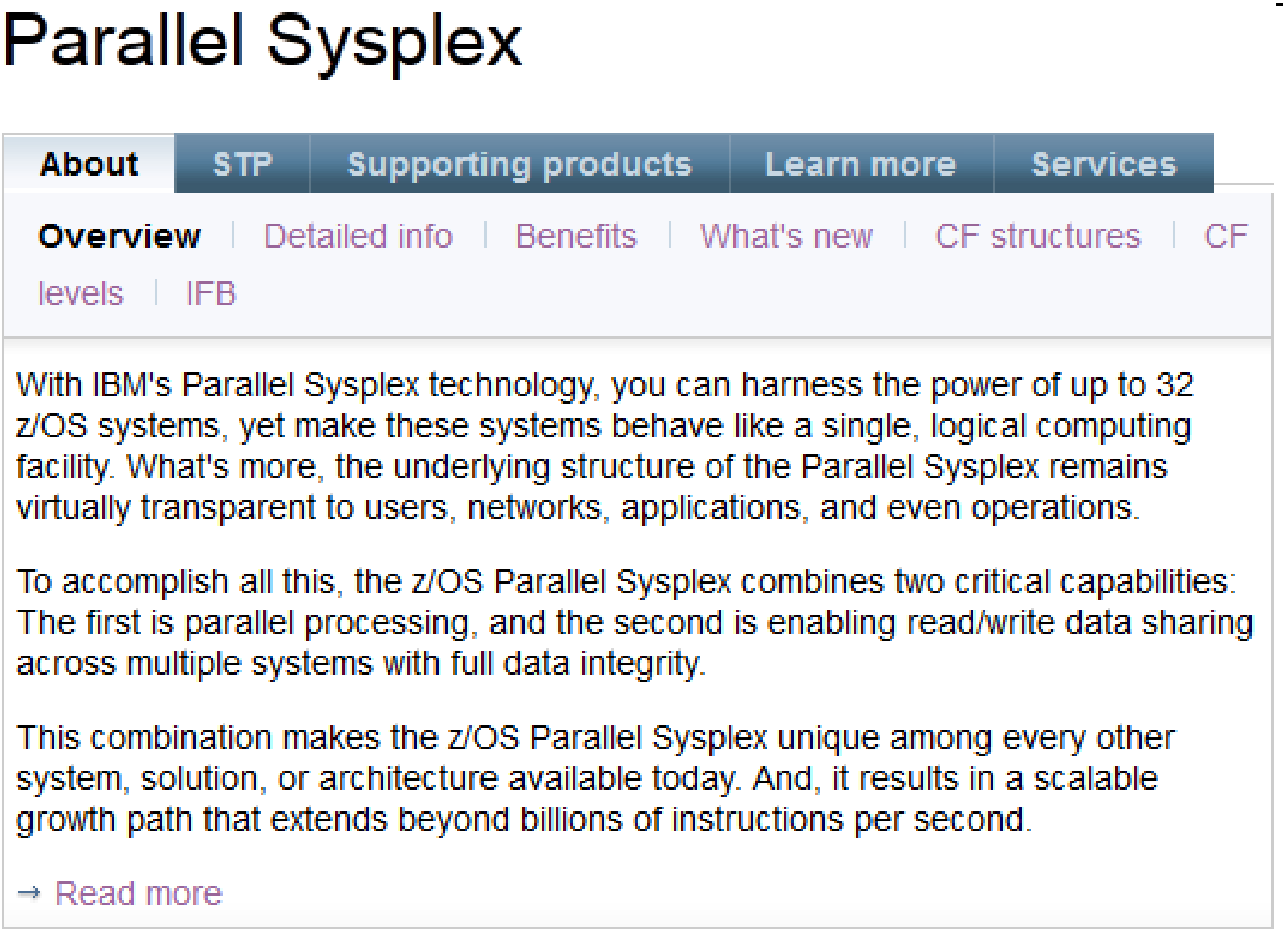
Available at www.redbooks.ibm.com

Available on Resource Link

- *IBM z Systems Planning for Fiber Optic Links (FICON/FCP, Coupling Links, and Open System Adapters)*, GA23-1408.

Parallel Sysplex Web Site

http://www.ibm.com/systems/z/advantages/pso/index.html


About
STP
Supporting products
Learn more
Services
Overview | Detailed info | Benefits | What's new | CF structures | CF levels | IFB

Parallel Sysplex

With IBM's Parallel Sysplex technology, you can harness the power of up to 32 z/OS systems, yet make these systems behave like a single, logical computing facility. What's more, the underlying structure of the Parallel Sysplex remains virtually transparent to users, networks, applications, and even operations.

To accomplish all this, the z/OS Parallel Sysplex combines two critical capabilities: The first is parallel processing, and the second is enabling read/write data sharing across multiple systems with full data integrity.

This combination makes the z/OS Parallel Sysplex unique among every other system, solution, or architecture available today. And, it results in a scalable growth path that extends beyond billions of instructions per second.

→ Read more

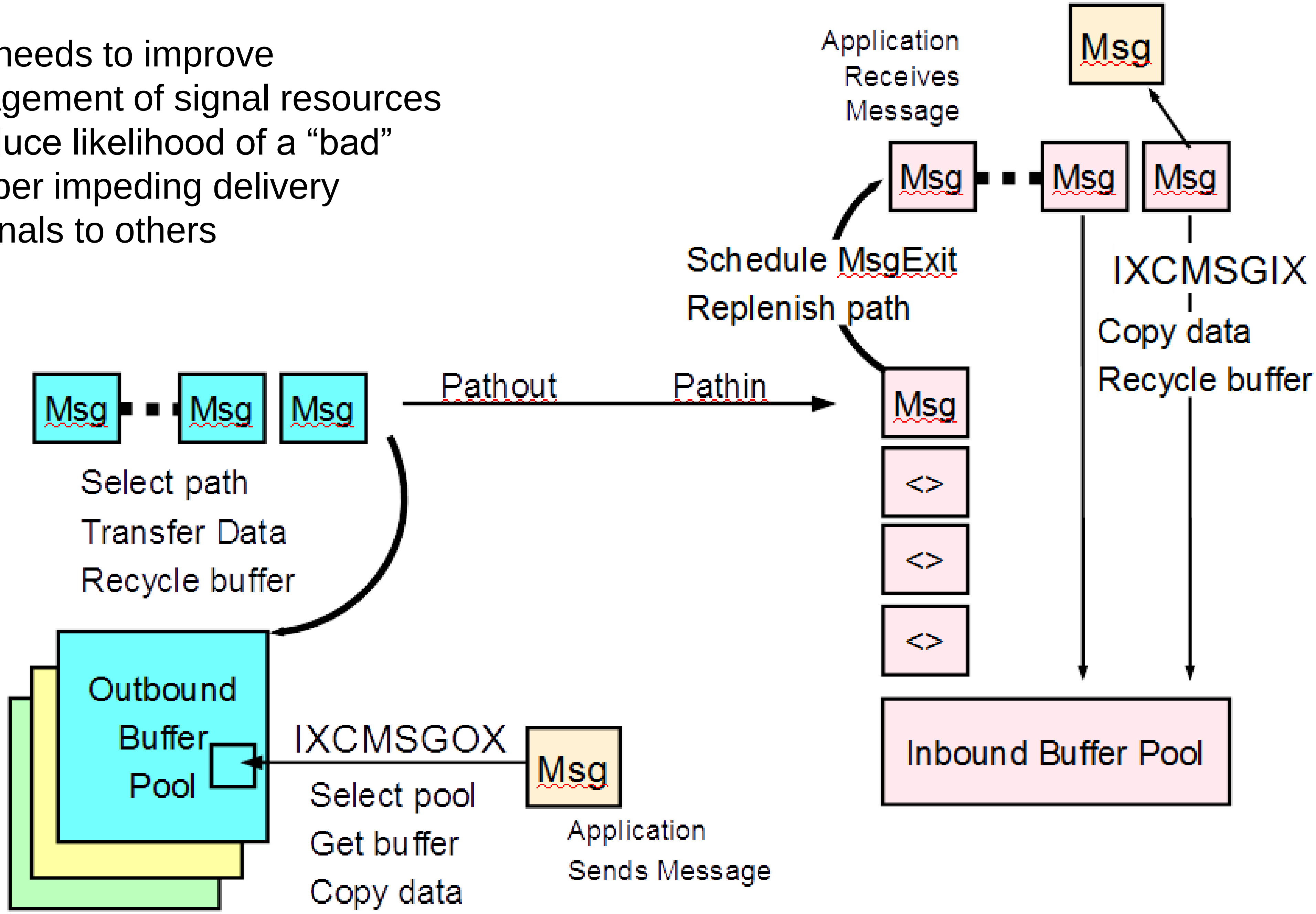
z/OS 2.2

z/OS V2R2 Summary

- XCF message isolation
- CFRM policy site preferences
- Health based routing
- SMT exploitation
- Need new ARM CDS
- CF Gain Ownership protection
- IXCQUERY for CDS and policies

XCF Signal Service

XCF needs to improve management of signal resources to reduce likelihood of a “bad” member impeding delivery of signals to others



Message Isolation

- Goal: Prevent a target member from consuming so much signal resource that it impedes the delivery of signals to other members
  - Primary issue is consumption of inbound signal buffers
  - Secondary issue is consumption of common storage

- Once a signal is accepted by XCF, it must be delivered
  - Receiving system cannot reject or discard incoming signals
  - The sending system must be the one to refuse messages targeted to a “bad” member

- The inbound side identifies “bad” members and “isolates” them
  - Tells sending systems to stop accepting signals for the bad member
  - (and when to resume sending)

Consequences of being Message Isolated

- XCF either delays or rejects signals targets to an isolated member
- By default, the reject reason is “no buffer”
  - “*no buffer for you*”. Signals sent to non-isolated members are still accepted.
  - Exploiters can optionally request unique reason code of “isolated”
    - Specify MSGISO=MSGORSN when issuing IXCJOIN to join group
- Delayed signals are held by sending system until the target member becomes “not isolated” or the message completes (timeout, cancel, …)
- If XCF accepts a signal, but the target member is isolated before the signal can be queued to a signal path, the signal is held until the member becomes “not isolated”, terminates, or message completes

Consequences of being “Message Isolated” …

- So impacted senders may see more rejects, delays, or timeouts
  - But this is the intended behavior
  - If a target member is not able to participate effectively, there will likely be impacts to its peers
- Goal is to limit scope of impact to the offending application (group)
  - In the past, all group members had potential for impact due to no buffer conditions or transfer delays
  - Signals for unrelated applications (groups) should flow freely
- Note! There can still be impact to other applications
  - Others may depend on the services of the offending application
  - There may be (are) scenarios that could defeat the XCF algorithms

New XCF Messages

- IXC638I – documents isolation window for given member
- IXC6371 – documents impact window for given system

By default, these messages are issued to hardcopy log.
Issued at start and end of a window.
Periodically reissued if window persists long enough.

- IXC645E – alerts operator to existence of isolated members
- IXC440E – alerts operator to existence of impacted members

These are issued as highlighted messages.
Persist until no members on the issuing system qualify.
Investigate with D XCF,G and/or review messages IXC637I, IXC638I.

Coexistence, Migration, Exploitation

- New behaviors only apply when sender and target reside on a system running z/OS V2R2 or later
  - Does apply to local message traffic, though we seldom see issues there
  - So not likely to see any new behavior until there are at least two systems running z/OS V2R2 in the sysplex
- Simply IPLing system with z/OS V2R2 activates the new behavior
  - Can be disabled via new XCF FUNCTIONS switch MSGISO
- When communicating with down level system, the old behaviors apply (and so derive no benefit)
  - Members on down level system will not be isolated
  - Signals from down level sender to isolated target member will be sent
- Down level systems do not require any compatibility support

Exploiter messages regarding “no buffer” may be inaccurate !

- On z/OS V2R2 systems, XCF might now selectively indicate “no buffer” for signals targeted to an isolated member
- Some XCF exploiters issue messages to complain when their msgout request is rejected for a “no buffer” condition
  - In the past, you might then go look at your MAXMSG specifications
  - But with z/OS V2R2, those exploiter messages might be the result of the target member being “message isolated”
  - So with z/OS V2R2, you must first look to see whether message isolation might apply
- XCF query services (and therefore measurement products such as RMF) only indicate “no buffer” for true MAXMSG constraints

z/OS V2R2 Summary

- XCF message isolation
- CFRM policy site preferences
- Health based routing
- SMT exploitation
- Need new ARM CDS
- CF Gain Ownership protection
- IXCQUERY for CDS and policies

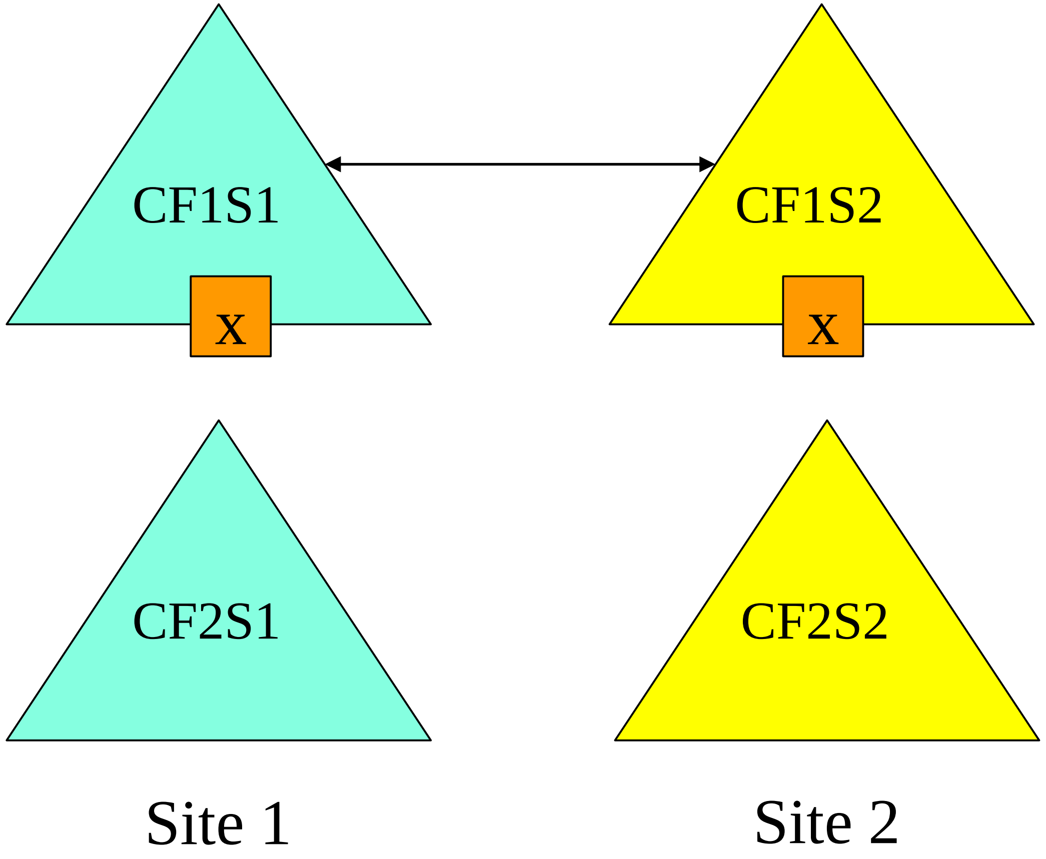
PREFLIST might not achieve desired placement for duplexed structures

- Want structures duplexed across sites for DR or availability

CFRM Policy

```
CF NAME(CF1S1) SITE(SITE1) …
CF NAME(CF2S1) SITE(SITE1) …
CF NAME(CF1S2) SITE(SITE2)
CF NAME(CF2S2) SITE(SITE2)

STRUCTURE STRNAME(x)
  DUPLEX(ENABLED)
  PREFLIST( CF1S1, CF1S2, CF2S1, CF2S2 )
```

But what if CF1S2 is down for maintenance?
Structure likely duplexed across the site 1 CF’s.
Fail to achieve desired cross site availability.



New CFRM Policy Specifications

- For a structure that is eligible for duplexing, can now specify a site preference
  - DUPLEX(ALLOWED,*site_preference*)
  - DUPLEX(ENABLED,*site_preference*)
- Where *site_preference* is one of:
  - ANYSITE – default, ignore site (same as today)
  - CROSSSITE – prefer host CF’s be in different sites
  - SAMESITE – prefer host CF’s be in the same site
  - SAMESITEONLY – host CF’s must reside in the same site

You may want to revise PREFLIST when specifying site preference

- Objective
  - Prefer structure be duplexed across sites for availability
  - Want primary instance in site 1 for locality to workload
- Current structure definition might be:
  - ENFORCEORDER(NO)
  - DUPLEX(ENABLED)
  - PREFLIST(CF1SITE1,CF1SITE2,CF2SITE1,CF2SITE2)

List first pair of CF's in site 1, site 2 order to encourage CFRM to allocate primary in site 1 and secondary in site 2. But if CF1SITE1 not available, primary instance is placed in in site 2 CF – fail to get desired locality.

- New structure definition might be:
  - ENFORCEORDER(NO)
  - DUPLEX(ENABLED,CROSSSITE)
  - PREFLIST(CF1SITE1,CF2SITE1,CF1SITE2,CF2SITE2)

List both site 1 CF's first to maintain locality for primary. CROSSSITE will get site 2 CF's used for secondary instance.

z/OS V2R2 Summary

- XCF message isolation
- CFRM policy site preferences
- Health based routing
- SMT exploitation
- Need new ARM CDS
- CF Gain Ownership protection
- IXCQUERY for CDS and policies

CF Gain Ownership

- Problem Statement / Need Addressed
  - Configuration errors can cause multiple sysplexes to attempt to use the same coupling facility (CF).
  - The operator must decide whether a sysplex should use the CF or not – with little provided information.
  - Incorrectly deciding to use the CF can cause a sysplex outage.
- Solution
  - Provide new COUPLExx CFRMTAKEOVERCF keyword to control whether the operator is prompted or XCF rejects use of a CF that may be in use by another sysplex. New default is NO.
- Benefit / Value
  - Avoid operator errors by forcing the installation to reactivate a CF in order to pass it from one sysplex to another

CFRMOWNEDPROMPT(YES)

- When sysplex is re-IPLed, scrub the sysplex name portion of all the authority values found in the CFRM policy
- Later when looking to gain ownership of a CF:
  - If CF authority is zero, take ownership of the CF
  - If CF authority is nonzero, and:
    - Matches CFRM policy, take ownership of CF
    - Matches CFRM policy except for the zero sysplex name, prompt the operator
    - Differs from CFRM policy then:
      - Prompt operator if CFRMTAKEOVERCF is PROMPT
      - Reject use of CF if CFRMTAKEOVERCF is NO

CFRMOWNEDPROMPT(NO)

- When sysplex is re-IPLed, the authority values found in the CFRM policy are not changed
- Later when looking to gain ownership of a CF:
  - If CF authority is zero, take ownership of the CF
  - If CF authority is nonzero, and:
    - Matches CFRM policy, take ownership of CF
    - Differs from CFRM policy then:
      - Prompt operator if CFRMTAKEOVERCF is PROMPT
      - Reject use of CF if CFRMTAKEOVERCF is NO

Usage & Invocation

- SYS1.PARMLIB(COUPLExx)
  - COUPLE
    - CFRMTAKEOVERCF(NO)
      …

- Statements for the “safest” configuration
  - CFRMOWNEDCFPROMPT(YES)
  - CFRMTAKEOVERCF(NO) z/OS V2R2 new default behavior

- Statements with most automatic gain ownership of CF (susceptible to more configuration/operator errors)
  - CFRMOWNEDCFPROMPT(NO) default
  - CFRMTAKEOVERCF(PROMPT) pre-z/OS V2R2 behavior

Migration & Coexistence Considerations

- Migration action
  - Update COUPLExx to specify CFRMTAKEOVERCF(PROMPT) if new CFRMTAKEOVERCF(NO) behavior is not desired
- Coexistence consideration when using CFRMOWNEDCFPROMPT(YES)
  - A downlevel system clears CF authorities in the CFRM CDS when initializing the sysplex (sysplex-wide IPL)
  - This may cause a z/OS V2R2 system using CFRMTAKEOVERCF(NO) to reject use of a CF when the desired behavior might have been to PROMPT
  - To avoid the strange behavior, do one of the following
    - Create a new COUPLExx for z/OS V2R2 with CFRMTAKEOVERCF(PROMPT)
    - Update COUPLExx to use CFRMOWNEDCFPROMPT(NO)
- No toleration/coexistence APARs/PTFs