



System - Managed CF Structure Duplexing

David Raften
Raften@us.ibm.com

Table of Contents

Overview	4
Evolution of Parallel Sysplex Recovery.....	5
Anatomy of a Duplexed CF Operation	8
Asynchronous CF Structure Duplexing for Lock Structures	10
When to Configure System-Managed CF Structure Duplexing.....	10
System Logger	11
VTAM Generic Resources.....	13
IBM MQ	14
IMS	15
CICS.....	18
Db2 (IRLM) Lock Structures.....	19
BatchPipes	20
GDPS Considerations.....	20
GDPS enhanced recovery support	21
Migrating to System-Managed CF Structure Duplexing	23
Managing and Monitoring.....	26
Starting Duplexing	26
Stopping Duplexing.....	28
Operations and Displays	28
Accounting and Measurement.....	30
Performance Impact of System-Managed CF Structure Duplexing	32
Interpreting the Results.....	39
Recovery Benefits - Duplexing “Failover”	40
Hardware Configuration Considerations and Recommendations	40
Distance Considerations	40
CF-to-CF Links - Sharing and Redundancy.....	42
z/OS CPs - sharing and capacity.....	43
z/OS-to-CF Links - sharing and redundancy.....	44
Coupling Facility CP Utilization - capacity.....	46
Coupling Facility CPs - sharing	47
Coupling Facility CPs - “balanced” capacity.....	49
DUPLEX(ALLOWED) vs DUPLEX(ENABLED) Considerations.....	50
Structure Sizing Considerations for System-Managed CF Structure Duplexing	53
Summary	54
Appendix A: Calculating Simplex Resource Use	55

System-Managed CF Structure Duplexing

Host CPU Capacity.....	55
Calculate the cost of a single CF request.....	55
Calculate the CF activity rates	57
Calculate the host impact of a duplexed CF structure.....	59
CF CPU Capacity	60
Coupling Facility Link Subchannel Busy	62
Appendix B: Host effects of Technology in a Parallel Sysplex	64
Appendix C: Structure Recovery Support.....	66
Appendix D: Asynchronous Duplexing for Lock Structures.....	68

System-Managed CF Structure Duplexing

Overview

System-Managed Coupling Facility (CF) Structure Duplexing (generally referred to as “CF Duplexing” throughout this paper) is designed to provide a general purpose, hardware assisted, easy-to-exploit mechanism for duplexing CF structure data. This can provide a robust recovery mechanism for failures such as loss of a single structure or CF, or loss of connectivity to a single CF, through rapid failover to the other structure instance of the duplex pair.

Benefits of System-Managed CF Structure Duplexing can include:

- **Availability**
Faster recovery of structures by having the data already in the second CF when a failure occurs. Furthermore, if a potential IBM, vendor, or customer CF exploitation were being prevented due to the effort required to provide alternative recovery mechanisms such as structure rebuild, log recovery, etc., System-Managed CF Structure Duplexing could provide the necessary recovery solution.
- **Manageability and Usability**
Provides a consistent procedure to set up and manage structure recovery across multiple exploiters
- **Configuration Benefits**
Enables the use of internal CFs for all resource sharing and data sharing environments

As there are benefits to be derived from System-Managed CF Structure Duplexing, there are also costs associated with its exploitation. These costs are dependent upon which structures are being duplexed and how those structures are being accessed by applications executing in a particular Parallel Sysplex® environment.

Costs of System-Managed CF Structure Duplexing can include:

- *Increased z/OS® CPU utilization*
- *Increased coupling facility CPU utilization*
- *Increased coupling facility link utilization*

A cost/benefit analysis should be performed for each structure prior to enabling it for System-Managed CF Structure Duplexing. This paper presents the background information and a methodology necessary for performing such an analysis.

Evolution of Parallel Sysplex Recovery

Before System-Managed CF Structure Duplexing, z/OS had several potential mechanisms for providing recovery in hard failure scenarios, each with its own availability characteristics:

1. No Recovery

Some structures provide no recovery mechanism whatsoever for hard failures. Whatever data is placed in the CF structure, along with whatever processing is dependent on that data, is therefore unrecoverable in a failure scenario. Prior to System-Managed CF Structure Duplexing, CICS® Shared Temporary Storage structures and IBM MQ Shared Queue (non-persistent messages) structures were two examples of this. Typically, these structures contain either Read-Only or "scratch-pad" data in which recovery is not a major issue.

2. Disk Backup

Some structures recover from hard failures by maintaining another hardened copy of the data on another medium such as disk. For example, data in a directory-only cache or store-thru cache structure is hardened on disk, and is therefore recoverable from disk in the event of loss of the CF structure. System Logger's use of staging data sets to maintain a second copy of the logstream data from the time it is written to the CF until it is offloaded is another example. Such structure exploiters typically incur a substantial mainline performance cost to write their updates synchronously to disk.

3. User-managed rebuild

User-managed rebuild was introduced along with initial Coupling Facility support. This process allowed MVS™ to coordinate a structure rebuild process with all of the active connected users of the structure, in which those connectors participate in the steps of allocating a new structure instance, propagating the necessary structure data to the new structure, and switching over to using the new structure instance. User-Managed Rebuild typically uses either an in-storage copy or a log to provide the data needed for structure recovery.

User-managed rebuild provides both a planned reconfiguration capability and, in most cases, a robust failure recovery capability for CF structure data, but often requires significant amounts of support from the structure connectors. In some cases, it is either impossible to recover, or requires an elongated recovery process for the structure connectors to reconstruct the structure data. This happens when the structure is lost in conjunction with one or more of the active connectors to the structure, and the connectors' protocol for rebuilding the structure requires each of the active connectors to provide some portion of the data in order to reconstruct the complete contents of the structure that was lost.

System-Managed CF Structure Duplexing

Without System-Managed CF Structure Duplexing, these structures require "Failure Isolation," separating the structure and its connectors into different servers, perhaps requiring the presence of one or more standalone CFs in the configuration. There is a demand to be able to use an ICF-only configuration in both a Resource Sharing and Data Sharing environment. The issue of failure isolation, of being able to effectively repopulate a structure if a structure on a CF image together with one of its connectors should be lost due to an (unlikely) hardware failure, currently limits internal CF usage for certain data sharing structures. System-Managed CF Structure Duplexing resolves these failure isolation issues and can enable the use of ICF-only configurations for duplexed data sharing structures.

4. User-managed duplexing

Some structures can recover from hard failures through user-managed duplexing failover. For example, changed data in a duplexed Db2® group buffer pool (GBP) cache structure can be recovered in this way. Such structure exploiters may obtain both very good mainline performance (taking advantage of the exploiter's intimate knowledge of the nature of the data contained in the structure and its usage, and of the exploiter's pre-existing serialization protocols that protect updates to the data, as performance optimizations) and excellent availability in failure situations due to the rapid duplexing failover capability.

User-managed duplexing support allows z/OS to coordinate a duplexing structure rebuild process with all of the active connected users of the structure, in which those connectors participate in the steps of allocating a new structure instance, propagating the necessary structure data to the new structure, but then keeping both structure instances allocated indefinitely. Having thus created a duplexed copy of the structure, the connectors may then proceed to duplex their ongoing structure updates into both structure instances, using their own unique serialization or other protocols for ensuring synchronization of the data in the two structure instances.

User-managed duplexing addresses the shortcoming noted above for user-managed rebuild, in which it is impossible or impractical for the structure exploiters to reconstruct the structure data when it is lost as a result of a failure. With user-managed duplexing, the exploiter can build and maintain a duplexed copy of the data in advance of any failure, and then when a failure occurs, simply switch over to using the unaffected structure instance in simplex mode. User-managed duplexing thus provides a very robust failure recovery capability, but because it is a user-managed process, it requires significant exploiter support from the structure connectors.

In addition, user-managed duplexing is limited to cache structures only; list and lock structures are not supported.

System-Managed CF Structure Duplexing

5. System-managed rebuild

System-managed rebuild allows z/OS to internalize many aspects of the user-managed rebuild process that formerly required explicit support and participation from the connectors. z/OS allocates the new structure and propagates the necessary structure data to the new structure, then switch over to using the new structure instance.

System-managed rebuild is only able to propagate the data to the new structure by directly copying it, so that system-managed rebuild provides only a planned reconfiguration capability. It is not capable of rebuilding the structure in failure scenarios, and thus does not provide a robust failure recovery mechanism at all. However, by internalizing many of the "difficult" steps in the rebuild process into z/OS and taking them out of the hands of the connectors, system-managed rebuild greatly simplifies the requirements on the structure exploiters, drastically reducing the cost for the exploiters to provide a planned-reconfiguration rebuild capability.

6. System-Managed CF Structure Duplexing

None of the above approaches are ideal. Several of them have significant performance overheads associated with them during mainline operation (for example, the cost of synchronously hardening data out to disk in addition to the CF in a store-thru cache model); some of them compromise availability in a failure scenario by involving a potentially lengthy rebuild or log recovery process during which the data is unavailable (for example, log merge and recovery for an unduplexed Db2 group buffer pool cache). Furthermore, some of these recovery approaches involve considerable development effort on the part of the CF exploiters to provide the appropriate level of recovery, as each exploiter implements its own unique recovery mechanisms.

System-Managed CF Structure Duplexing is designed to address these problems by creating a duplexed copy of the structure prior to any failure and maintaining that duplexed copy during normal use of the structure by transparently replicating all updates to the structure in both copies, and provides a robust failure recovery capability through failover to the unaffected structure instance. This results in:

- An easily exploited common framework for duplexing the structure data contained in any type of CF structure, with installation control over which structures are/are not duplexed.*
- High availability in failure scenarios by providing a rapid failover to the unaffected structure instance of the duplexed pair with very little disruption to the ongoing execution of work by the exploiter and application.*

7. Asynchronous CF Duplexing for Lock Structures

System Managed CF Structure Duplexing relies on the Coupling Facilities exchanging

System-Managed CF Structure Duplexing

information to ensure that changes to one structure get replicated to the secondary structure in the other Coupling Facility. While this is happening, z/OS and the application is waiting for the completion of this exchange. Asynchronous CF Duplexing for Lock Structures removes this wait, speeding up the throughput rate while cutting down on CPU cost for z/OS. This function is currently supported by IRLM for Db2.

Unless otherwise stated, in this document "CF Duplexing" refers to the synchronous variety.

System-Managed CF Structure Duplexing is designed to provide the "best of all possible worlds" — robust failure recovery capability via the redundancy of duplexing, and low exploitation cost via system-managed, internalized processing. Structure failures, CF failures, or losses of CF connectivity can be handled by:

- 1. Masking the observed failure condition from the active connectors to the structure, so that they do not perform any unnecessary recovery actions,*
- 2. Switching over to the structure instance that did not experience the failure, and*
- 3. Re-establishing a new duplex copy of the structure, if appropriate, as the Coupling Facility becomes available again, or on a third CF in the Parallel Sysplex.*

System messages are generated as the structure falls back to simplex mode for monitoring and automation purposes. Until a new duplexed structure can be established, the structure operates in simplex mode, and can be recovered using whatever existing recovery techniques are supported by the exploiter (such as user-managed rebuild).

System-Managed CF Structure Duplexing's main focus is providing this robust recovery capability for structures whose users do not support user-managed duplexing and/or do not support user-managed rebuild.

Anatomy of a Duplexed CF Operation

When z/OS receives a CF exploiter's update request for a duplexed CF structure (for example, writing database data to the CF, or obtaining a lock in the CF), z/OS will first split the exploiter's request into two distinct CF requests, one destined for each of the two structure instances. The two requests are launched serially by a routine running disabled on a z/OS CP. z/OS tries to launch these two requests at as close to the same time as possible so that the CF commands can execute at the two CFs with the

System-Managed CF Structure Duplexing

maximum amount of parallelism, but (as discussed later in this document) sometimes configuration issues can interfere with z/OS's ability to do this successfully.

Once the two commands arrive at their respective CFs, they need to coordinate their execution so that the update to the duplexed structure is synchronized between the two structure instances. In effect, the commands must be made to execute in "lockstep" with one another to preserve the logical appearance to the CF exploiter that there is a single consistent copy of the data. To do this, the CFs will exchange signals with one another over a CF-to-CF coupling link, a new configuration requirement for System-Managed CF Structure Duplexing. The CF commands may suspend and later resume their execution while they wait for the arrival of the necessary duplexing signals from the "peer" duplexed command executing in the other CF. Once again (as discussed later in this document), configuration issues may interfere with the ability of these CF-to-CF signals to be exchanged efficiently, or with the ability of the CF commands to resume execution in a timely manner after the arrival of a synchronization signal for which it was waiting. Once both of the duplexed commands have exchanged all the required signals and completed execution, they each return their individual command responses to the z/OS system that originated them.

As is true for simplex requests, z/OS can process the completion of duplexed requests in either a CPU-synchronous or a CPU-asynchronous manner. z/OS can either spin synchronously on the originating CPU polling for the completion of both operations, or it can give up control to process other work and eventually observe the completion of the two CF operations asynchronously. The decision of whether requests are completed synchronously or asynchronously is largely under the control of a conversion algorithm (see Washington Systems Center Flash 10159 for additional information

<http://www.ibm.com/support/techdocs/atsmastr.nsf/WebIndex/FLASH10159>) that takes into account the observed service times of the requests in comparison with the speed of the processor that originated the requests.

Regardless of how the requests are completed, z/OS inspects each of the responses, validates that the results are consistent between the two CFs, and then merges the results of the operations into a single consolidated request response. It is this consolidated response that is presented back to the CF exploiter, whose mainline request processing is therefore completely unaware of whether the structure being

System-Managed CF Structure Duplexing

used is operating in simplex or duplex mode. (System-Managed CF Structure Duplexing is largely transparent to the exploiter, as far as the execution of mainline CF commands is concerned. However, z/OS does provide a notification to the exploiters of structure transitions into and out of the duplexed state so that the exploiter can react to this in other ways. For example, exploiters might have other mechanisms for providing data backup/redundancy that would need to operate only when their structure was operating in simplex mode.)

Asynchronous CF Structure Duplexing for Lock Structures

Asynchronous CF Lock Duplexing feature was designed to be an alternative to synchronous system managed duplexing. The goal was to provide the benefits of lock duplexing without the high performance penalty. It eliminates the synchronous mirroring between the CFs to keep the primary and secondary structures synchronized. With Asynchronous CF Lock Duplexing, the secondary lock structure updates are performed asynchronously with respect to the primary lock updates. Not only does this help when the two CFs are near each other, but it has an even greater impact as the distance between the CFs increase. Db2 (or any exploiter of this feature) can consider the command complete after the primary lock structure has been updated. Before Db2 commits a transaction, it checks that the necessary requests have been successfully written to the secondary structure. Most of the time this will be the case. If it isn't then the Db2 log write process will be suspended until the updates to the secondary structure are complete. This "sync-up" protocol ensures that the secondary structure contains all the necessary updates (those that have been committed by Db2) even though at any given moment the updates to the secondary structure are lagging behind those to the primary structure by some amount.

Even though performance will be impacted by migrating from Simplex to Asynchronous duplexing for lock structures, the impact is assumed to not be significant.

When to Configure System-Managed CF Structure Duplexing

While there is value in System-Managed CF Structure Duplexing, IBM does not recommend its use in all situations, nor should it necessarily be used for all of the structures that support it. A cost/benefit evaluation must first take place. There is a performance cost associated with the additional work that z/OS needs to do to send

System-Managed CF Structure Duplexing

and receive two CF requests instead of one, and with the additional CF-to-CF communication involved in synchronizing updates to the two structure instances. These costs have to be weighed against the benefits of structure recovery and operational simplicity that duplexing can provide.

Some structures will obtain more benefit from System Managed duplexing than others.

- 1. Structures that don't support user-managed rebuild for recovery purposes should obtain the most benefit. These are the structures in Appendix C that have **NO** in the User Managed Rebuild column (highlighted in red).*
- 2. Some logging functions that currently use disk for staging datasets to duplicate data in structures. In some cases, the performance degradation when using disk is so great that staging datasets are not implemented. Using system-managed duplexing instead of staging datasets makes duplication more practical and thus improves availability. Generally, these are the structures in Appendix C that have "**Logger**" in the Structure column.*
- 3. Structures which require failure-isolation from exploiting z/OS images in order to be able to rebuild must be located in a failure-isolated (e.g. Standalone) CF when in simplex mode, but can be located on an internal CF if CF Duplexing is used for them. These are the structures in Appendix C that have FAIL-ISOL in the User Managed Rebuild column (highlighted in yellow). Note that prior to CF Duplexing, logger structures generally required either the use of staging data sets or that the structure was failure-isolated from exploiting z/OS images in order to be fully recoverable, which is why these structures are highlighted in yellow twice in Appendix C.*

A staged approach to enabling system-managed duplexing allows the installation to assess the costs and benefits at each increment. First implement duplexing for those structures that don't support user-managed rebuild at all (case 1 above), assess their performance and operational characteristics, then proceed through case 2 and 3 above, assessing the performance and capacity requirements at each step.

More details on how to determine the cost of System-Managed CF Structure Duplexing are found later in this document.

System Logger

The System Logger is designed to provide its own backup copy of log data written to a CF structure for recovery capability. The Logger keeps its backup copy of the structure data either in local buffers (data spaces associated with the IXGLOGR address space)

System-Managed CF Structure Duplexing

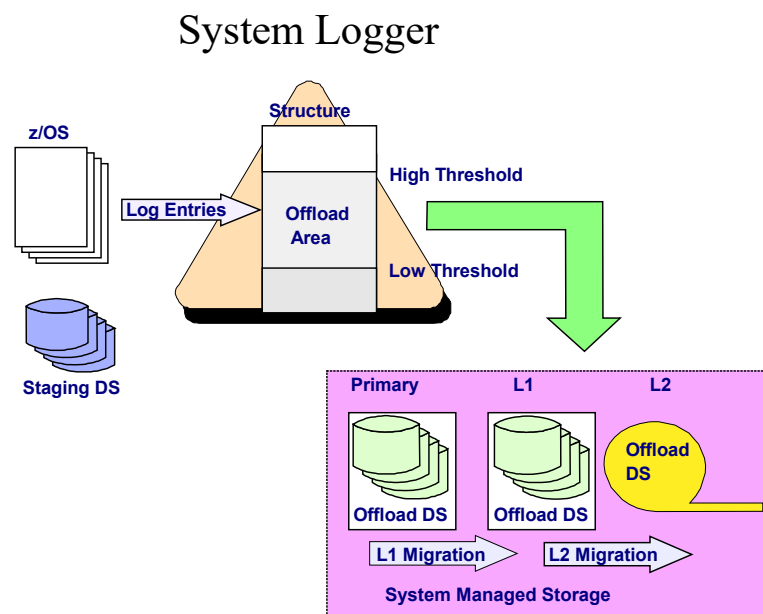
or in log stream staging data sets. This allows the Logger to provide recovery of log data in the event of many types of CF, structure, or system failures.

System-Managed CF Structure Duplexing provides an alternative to the previous means of providing recovery for log stream data. It is designed to provide greater recoverability opportunities over Logger local buffer duplexing.

The chart shows the types of storage used by the z/OS system logger. Primary storage consists of a data space in the same z/OS image as the System Logger, and a staging data set. Data is then offloaded to the Offload Data Set, then migrated to tape through HSM.

A benefit from System-Managed CF Structure duplexing from the System Logger will come from eliminating staging data sets as a requirement for availability. There are two major users of staging data sets: RRS and CICS. RRS has recommended that its data be backed by staging data sets to obtain better recoverability characteristics than using data spaces. IBM recommends a CF structure, and not DASD-Only logging, to provide the ability to restart a resource manager on a different partition.

CICS generally does not recommend the use of logger staging data sets due to the performance cost. However, there are cases when staging data sets are recommended.



System-Managed CF Structure Duplexing

These are:

- *The Coupling Facility is volatile (no internal battery backup or UPS)*
- *A Coupling Facility is not failure-isolated from the z/OS connector on different servers.*
- *A structure failure resulted in the only copy of log data being in z/OS local storage buffers*
- *DASD-only logging is implemented. For DASD-only log streams, staging data sets are the primary (interim) storage.*

STG_DUPLEX(YES) and DUPLEXMODE(COND) for the forward recovery and system log logstreams is recommended to cause the system logger to automatically allocate staging data sets if the Coupling Facility is not failure isolated.

Staging data sets must be used in a GDPS® disaster recovery environment to initiate a disk remote copy of the data to the remote site without waiting for the system logger structure to write the data to an offload dataset. More information on duplexing with GDPS can be found in a later section in this document.

VTAM Generic Resources

VTAM® Generic Resources provide a single system image to the Parallel Sysplex cluster for the end user at session logon time. A single generic name is presented to the z/OS Communication Server, where the session is then routed to one of the active session managers that has this generic name defined. Availability is improved, because when a member of the Parallel Sysplex fails, users can log on again using the same generic application name. The sessions are reestablished with another member of the sysplex.

Taken in isolation, the VTAM Generic Resource structure can be rebuilt quickly, but if there are many structures that need to be rebuilt at the same time such as in a loss of connectivity situation, it can take a couple of minutes to complete VTAM Generic

System-Managed CF Structure Duplexing

Resource structure rebuild. During this time while the Generic Resource structure is unavailable, users would be unable to log off a session or establish a session to the session manager, even if a specific name is used. LU requests are queued until the rebuild completes.

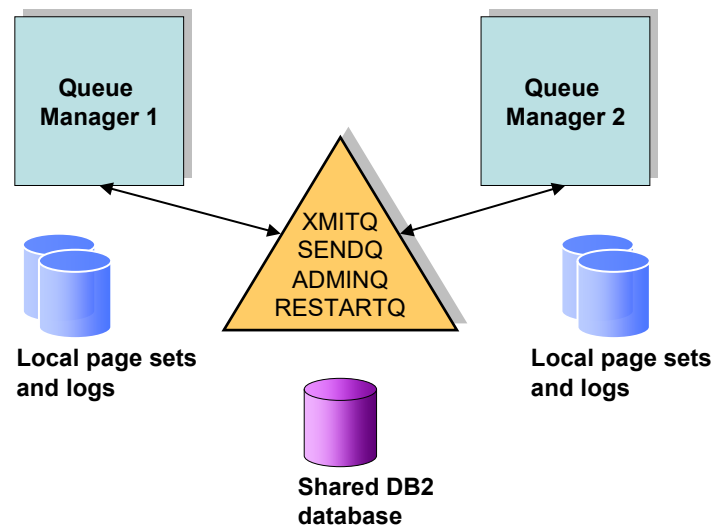
The end user impact of this depends on how VTAM generic resources are used within an installation.

Since the VTAM Generic Resource structure gets updated only when session status changes, there is no significant cost to duplexing the structure.

IBM MQ

IBM MQ has support for shared queues for messages stored in Coupling Facility structures. Applications running on multiple queue managers in the same queue-sharing group anywhere in the Parallel Sysplex cluster can access the same shared queues. This provides high availability, high capacity, and pull workload balancing. It is designed so that work can continue without interruption if an individual application instance or queue manager should fail or be recycled, as other instances of the same application accessing the same shared queues can continue to do the work. Throughput is no longer constrained by the capability of a single queue manager, as multiple queue managers can access the same shared queues. Automatic pull workload balancing can be achieved as the least constrained application instance will process the most messages.

System-Managed CF Structure Duplexing



Since non-persistent messages are not logged on disk, a loss of connectivity or a failure of the shared queue structure or the Coupling Facility would result in the loss of the shared (non-persistent) MQ message queue. In addition, all queue managers connected to the shared queue would fail together with non-persistent messages on their local queues. Similarly, if there is a loss of connectivity to the MQ administration structure or the structure or CF fails, then all queue managers using that structure would fail. System-Managed CF Structure Duplexing supplies higher availability for the non-persistent messages by providing higher availability for the CF structures containing them. Although some may choose to not duplex the shared queues due to performance impact, it is highly recommended to duplex the administration structure.

IMS

IMS™ supports many structures, depending upon the environment that is implemented. The value obtained from System-Managed CF Structure Duplexing differs for each of these environments.

System-Managed CF Structure Duplexing

IMS Full Function Data Sharing. The full function data sharing support as provided since IMS V5 requires up to three structures:

- *IRLM Lock*
- *OSAM Cache*
- *VSAM Cache*

The OSAM and VSAM Cache structures are implemented with either a "Directory-only" or a "Store-through" model. As data is written to the IMS local buffer pool, structure, it is designed to be simultaneously written to disk as well (and optionally to the OSAM structure). This is designed to make recovery easy in the loss of these structures as IMS would be able to access the data from disk and rebuild the structures as needed. Recovery of the IRLM lock structure for IMS (as well as for DB2) is fast; on the order half a minute. Because of these rapid recovery characteristics, outside of enabling the failure-isolation environment as recommended by IRLM, System-Managed CF Structure Duplexing provides little additional value in this environment.

IMS Shared Fast Path: IMS Fast Path Data Entry Data Bases (DEDB) come with the options of configuring them as DEDBs with SDEPs, and DEDBs with VSO (Sequential DEpendants, and Virtual Storage Option). While the shared DEDB/SDEP databases only require use of the (IRLM) Lock structure, DEDB/VSO does have its own structure to support data sharing with System-Managed CF Structure Duplexing implications.

IMS can manage duplexing for the DEDB/VSO structures. The VSO structure uses the "Write-Into" model. As records get updated, the data is asynchronously written into the structure, but not disk. At system checkpoints, updated data is then hardened to disk. The issue then is that if a structure or Coupling Facility should fail, the data on disk can not be assumed to be valid and a database recovery process is needed. This involves going to the last image copy and applying the logs, and can take hours. To help with this, IMS has implemented support for IMS-managed duplexing of DEDB/VSO structures on an area by area basis since IMS V6. If one structure would become unavailable, the IMS managed duplexed structure is still available for reads and updates in simplex mode. To get back to duplex mode, an IMS /VUNLOAD command is issued followed by a /STA Area command. IMS does not stop data sharing during this process. The /VUNLOAD command puts the data back to disk. The

System-Managed CF Structure Duplexing

/STA Area command puts the data back into the two structures. All data is available during this time.

Compare this time-consuming IMS recovery procedure with System-Managed CF Structure Duplexing, where the duplexed structure may be automatically reestablished as the backup Coupling Facility comes online. There are no IMS commands required with System-Managed CF Structure Duplexing. Also, the VSO structure recovery will be consistent with other IMS database and system structures which currently support rebuild. Therefore there is no need for multiple operational recovery procedures. System-Managed CF Structure Duplexing provides the benefit of much easier system management characteristics compared with IMS managed VSO duplexing.

Shared Message Queues: Shared Message Queue (SMQ) implementation requires several structures, including the SMQ list structure holding the Message Queue itself, and the System Logger structures managed by the Common Queue Server (CQS) component used for recovery of the SMQ.

Recovery of the SMQ structure is similar to a database recovery process. It involves:

- 1. Reallocating the structure*
- 2. Restoring the structure from the disk image copy in the structure checkpoint data set*
- 3. Applying changes from the System Logger structure*

Recovery of SMQ structures requires the user to take periodic "structure checkpoints" to disk. Activity to the structure is quiesced during this time, which may be several seconds, depending on the size of the structure. In the event that an SMQ structure is lost (or connectivity to it is lost), recovery is needed. The most recent structure checkpoint data set (SRDS) is used to restore the structure contents to the time of the structure checkpoint. Then the outstanding updates in the logs are applied.

This process can take a period of time, depending upon how many updates took place since the last SMQ checkpoint. For small structures it can take seconds, but for larger structures in a busy environment it can take significantly longer. Typically, it takes 1 minute to recover every 2-3 minutes of data. So, if 45 minutes of CQS log must be read after restoring the structure from the structure recovery data set, it takes a minimum of 15 minutes to read the log and finish the structure recovery. While SMQ recovery is occurring, all message activity in the IMSplex is quiesced.

System-Managed CF Structure Duplexing

System-Managed CF Structure Duplexing is designed to reduce the need (or desire) to take frequent structure checkpoints. While the chance of a loss of connectivity, a structure failure, or a CF failure is small, the impact of such a failure has to be weighed against the cost of duplexing the structure.

CICS

CICS exploits system-managed rebuild function to allow the z/OS to manage the planned reconfiguration of Coupling Facility structures used for shared temporary storage, Coupling Facility data tables (CFDTs), and named counter server pools without recycling the CICS systems using them. Without this support, you could only move these structures by using server functions to UNLOAD to a sequential data set and then RELOAD elsewhere from the data set. This switch caused errors in CICS such that restart, an unacceptable outage in some situations, was recommended.

While z/OS is rebuilding a structure, requests for access to the structure are delayed until the rebuild is complete. The shared temporary storage and CFDT servers issue CF requests asynchronously, and therefore a rebuild simply delays such requests. Rebuild might take from a few seconds for a small structure to tens of seconds for a large structure. The only potential effect is that the transaction making the delayed request might get purged, either because of an apparent deadlock (by the DTIMOUT value) or by an operator command. The effects of this depend upon how sensitive the environment is to short stall conditions. Most environments would quickly recover. Some high volume, high stress environments may see significant backup in transactions which would take a longer time to recover. For this environment, the much faster recovery offered by System-Managed CF Structure Duplexing would greatly add to overall availability.

While System-Managed Rebuild supports the moving of structures from one Coupling Facility to another by copying data from the "old" to the "new" structure, it does not by itself support the recovering of a structure if the original is unavailable to all systems as in the case of a CF or structure failure. Although CICS provides the ability to recover a CFDT, Temporary Storage or Named Counter structure, this is time consuming and requires starting and stopping the CICS Coupling Facility Server address space several times. Because of its complexity, this function was rarely used.

It is possible to manage recovery of the Named Counters manually. For example, one method would be for CICS to read an existing database to get the "highest key value"

System-Managed CF Structure Duplexing

and using that as a starting point. One can then update the file every 100 entries. If there is a duplicate entry in the database then the application would ask for another number until it didn't get a duplicate record.

System-Managed CF Structure Duplexing support in CICS is designed to provide a rapid and simple recovery mechanism for all of these structures.

An additional benefit of System-Managed CF Structure Duplexing is application flexibility. Until now, the CFDTs and TS structures contain mostly "scratch-pad" information which are not critical to recover.

Db2 (IRLM) Lock Structures

Lock structures such as used by GRS and IRLM present a performance challenge:

1. They are generally by far the most used structures in a Parallel Sysplex.
2. Lock requests are synchronous with respect to transactions. Any response time delays slow down the transactions, and thus the throughput rate
3. Lock messages are also the shortest so are affected most by any slowdowns, especially when duplexing across distances.

With this in mind, in the past many had to carefully consider the pros and cons of duplexing Db2 lock structures.

Asynchronous Duplexing for Lock Structures removed a lot of these issues. A lock request is sent synchronously to the Primary lock structure. It gets acknowledged in the same time as if the lock structure was simplexed, and the transaction can continue. Asynchronous to this processing, the request is forwarded on to the secondary structure. Only during the "Commit" processing does Db2 asks Secondary structure to confirm the lock requests are synchronized. This is a negligible delay.

Because of the benefits of supporting an ICF-only environment and potentially speeding up Db2 rebuilds with almost no cost, Duplexing the Db2 (IRLM) lock structure is recommended.

System-Managed CF Structure Duplexing

BatchPipes

BatchPipes® offers a way to connect jobs so that data from one job can move through processor storage or, in a pipeplex, through a CF structure to another job without going to disk or tape. BatchPipes is designed to allow two or more jobs that formerly ran serially to run concurrently, helping to reduce the number of physical I/O operations by transferring data through processor storage rather than transferring data to and from disk or tape, and may also help to reduce tape mounts and use of disk.

If a BatchPipes subsystem running in a pipeplex detects a structure failure or loses connection to the coupling facility structure, the subsystem cannot service the jobs using cross-systems pipes. BatchPipes does not support automatic structure rebuild. As such, if there is a CF or SYSASFPxxxx structure failure, or a loss of connectivity to the structure, all other jobs with connections to the pipe receive an I/O error. This might cause those jobs to ABEND. Most likely it would require those jobs to be re-run from the last job that hardened the data on disk.

System-Managed CF Structure Duplexing is designed to isolate any loss of connectivity, CF failure, or structure failure incidents from the pipes connectors. This provides availability benefits through rapid failover recovery for these types of errors.

GDPS Considerations

It is not possible to guarantee consistency between a Coupling Facility structure and data on disk. Even if duplexing CF structures (either User or System managed) across two sites for disaster recovery purposes, the surviving instance of the duplexed structure in the recovery site may not be usable because it may not be consistent with the secondary copy of mirrored disk. To ensure that a subsystem did not use old data that may be residing on a CF structure after a takeover, GDPS “forces” (deletes) the structures and have the subsystems rebuild the structures using consistent data on disk.

- *It is not predictable how a disaster that affects only a single site will affect the coupling facilities and duplexing. For some connectivity failure situations, XCF/CFRM may need to revert your duplexed structures to a simplex state. This means that one instance of the duplexed structures in the subject coupling facilities will be dropped and one instance will survive. There is no guarantee that the instance in the target recovery site will be the instance that will survive the failure.*

System-Managed CF Structure Duplexing

- *You may have a disaster in the site with the primary disks and there may be surviving instances of one or more of your duplexed structures in the recovery site where the secondary disks are. Whether any surviving instance of CF structures, duplexed or not, are usable is dependent on whether the structures are consistent with the disk data that will be used. If the data on the secondary disk were frozen at time 't' but the primary disk and duplexed structures continued to be updated beyond time t, then the disk data to be used for recovery purposes will not be consistent with the CF structures and these structures can not be used for disaster recovery purposes. To be able to use any surviving instance of structures following a disaster in conjunction with mirrored copy of disk, the disk and CF data must be known to be consistent with each other.*

Having a usable instance of certain application-related structures (such as Db2 Group Buffer pools) together with mirrored disk data that is time-consistent with the structures has the potential to greatly reduce recovery times. For example, for Db2, you can eliminate the time required for GRECP recovery which can be lengthy. Refer to GDPS Family - An Introduction to Concepts and Capabilities, SG24-6374 available at <http://www.redbooks.ibm.com/abstracts/sg246374.html?Open> for further information concerning GDPS support for structures duplexed across sites for disaster recovery purposes.

Finally, if CF structures are duplexed across sites, even when used in conjunction with the GDPS HyperSwap™ capability, chances are that continuous availability may not be attainable for compound, multi-component failures such as a complete site disaster. When there is such a failure event, the sequence of events (what fails first, what happens next, and so on) will vary greatly from one failure to another. The recovery actions taken by the operating systems across the sysplex, the sysplex exploiting subsystems and GDPS, if present, will be very timing dependent. Although there can be failure scenarios where continuous availability may be attainable in conjunction with GDPS HyperSwap (HyperSwap capability is needed for continuous availability of the data), one should not count on this and set service level requirements for such failures based on the assumption of continuous application availability.

GDPS enhanced recovery support

In the event of a primary site failure, the GDPS Metro cannot ensure that the CF structure data may be time-consistent with the "frozen" copy of data on disk, so GDPS must discard all CF structures at the secondary site when restarting workloads. This

System-Managed CF Structure Duplexing

results in loss of "changed" data in CF structures. Users must execute potentially long-running and highly variable data recovery procedures to restore the lost CF data.

GDPS enhanced recovery is designed to ensure that the secondary volumes and the CF structures are time consistent, thereby helping to provide consistent application restart times without any special recovery procedures.

If you specify the FREEZE=STOP policy with GDPS Metro and duplex the appropriate CF structures, when a CF structure duplexing drops into simplex, GDPS is designed to direct z/OS to always keep the CF structures in the site where the secondary disks reside. This helps to insure the Metro Mirror volumes and recovery-site CF structures are time consistent thereby providing consistent application restart times without any special recovery procedures. This is especially significant for customers using Db2 data sharing, IMS with shared DEDB/VSO, or IBM MQ shared queues.

System-Managed CF Structure Duplexing

Migrating to System-Managed CF Structure Duplexing

As with any system change, planning must take place to help obtain a successful transition. One example of a detailed migration plan with the user tasks is shown below:

- 1. Determine which CF structures in the installation can exploit System-Managed CF Structure Duplexing, and decide, on a case-by-case basis, whether or not to enable duplexing for those structures. Note that structures can be migrated from simplex to duplex and back again one at a time, dynamically, allowing these decisions to be re-evaluated from time to time.*
- 2. Evaluate the CF configuration (storage, links, processor capacity) and make any necessary configuration changes to accommodate the new structure instances resulting from System-Managed CF Structure Duplexing. For additional details concerning CF configuration, see section "Hardware Configuration."*
- 3. Evaluate z/OS CPU requirements. Additional z/OS CPU will be required to drive two CF requests instead of one, and to reconcile the responses coming back. Some of this cost may be offset by System-Managed CF Structure Duplexing, eliminating the need for maintaining other in-storage or on-DASD copies of the CF structure data.*
- 4. Install the CF-to-CF links required for System-Managed CF Structure Duplexing*
- 5. Check for appropriate z/OS APARs listed in the CFDUPLEXING PSP bucket.*
- 6. Format a new set of CFRM Couple Data Sets with the following keywords:
ITEM NAME(SMREBLD) NUMBER(1)
Specifies that this CFRM couple data set supports the system-managed rebuild process

ITEM NAME(SMDUPLEX) NUMBER(1)
Specifies that this CFRM couple data set supports the system-managed duplexing rebuild process. Specifying SMDUPLEX implies support for SMREBLD

ITEM NAME(MSGBASED) NUMBER(1)
Specifies that this CFRM couple data set supports message-based event and confirmation processing. Specifying MSGBASED implies support for SMREBLD and SMDUPLEX

ITEM NAME(ASYNC DUPLEX) NUMBER(1)
Specifies that this CFRM couple data set supports system-managed asynchronous*

System-Managed CF Structure Duplexing

structure duplexing. Specifying ASYNCDUPLEX implies support for SMREBLD, SMDUPLEX, and MSGBASED.

Sample Couple Dataset Format Utility input:

DATA	TYPE (CFRM)	
ITEM	NAME (POLICY)	NUMBER (6)
ITEM	NAME (CF)	NUMBER (8)
ITEM	NAME (STR)	NUMBER (50)
ITEM	NAME (CONNECT)	NUMBER (32)
ITEM	NAME (SMREBLD)	NUMBER (1)
ITEM	NAME (SMDUPLEX)	NUMBER (1)
ITEM	NAME (ASYNCDUPLEX)	NUMBER (1)

Format LOGR couple data sets using the ASYNCDUPLEX(1) keyword. LOGR would then need to deallocate and reallocate the structure before it can be duplexed. You must also update the logstream definition in the System Logger policy for the structures you want to duplex with the LOGGERDUPLEX parameter.

Bring the new LOGR couple data sets into use by adding one as a new alternate, PSWITCH to make it the primary, and then bring the second into the configuration as the alternate.

7. Bring the new CDSes into use as the primary and alternate CFRM CDSes for the configuration. This can be done nondisruptively via the SETXCF COUPLE, ACOUPLE and SETXCF COUPLE, PSWITCH operator commands.
8. Note that once the above steps have been taken, it is not possible to fall back nondisruptively to a downlevel CFRM or LOGR CDS which is not System-Managed CF Structure Duplexing-capable. Doing so will require a sysplex-wide IPL of all systems using CFRM or LOGR, with the downlevel CFRM or LOGR CDSes specified for use in the COUPLExx parmlib member. This does not imply that the System-Managed CF Structure Duplexing function cannot be turned on and off nondisruptively once an uplevel, System-Managed CF Structure Duplexing-capable CFRM CDS is in use in the sysplex. The System-Managed CF Structure Duplexing function can be started and stopped for particular structure instances, in a nondisruptive manner, through either the modification of the CFRM policy DUPLEX parameter or the SETXCF START|STOP, REBUILD, DUPLEX command, while the uplevel CFRM CDS remains in use. The uplevel CFRM CDS may even be brought into use and remain in use indefinitely even when all structures are DUPLEX(DISABLED), and are thus not eligible to be duplexed.
9. Install and migrate to the new CF-exploiting product/subsystem software levels required for System-Managed CF Structure Duplexing. Each product or subsystem will define its own unique migration rules for exploiting System-Managed Duplexing.
10. Modify the CFRM policy to control the placement (via the CF preference list) and DUPLEX parameter indication, for the structures which will be duplexed via

System-Managed CF Structure Duplexing

System-Managed CF Structure Duplexing, and activate this CFRM policy. The existing DUPLEX parameter on the CFRM administrative policy STRUCTURE definition is broadened to control System-Managed CF Structure Duplexing:

- *DUPLEX(ENABLED) - z/OS will automatically attempt to start and maintain duplexing for the structure at all times*
- *DUPLEX(ALLOWED,duptopts) - Duplexing may be manually started/stopped for the structure, but z/OS will not start duplexing automatically*
- *DUPLEX(DISABLED,duptopts) - Duplexing is not allowed for the structure*

“duptopts” (Duplex Options) can be one of the following:

- *SYNCONLY – **Default.** The structure is duplexed by using synchronous duplexing only. Either user-managed duplexing or system-managed synchronous duplexing is used.*
- *ASYNCONLY - The structure is duplexed by using asynchronous duplexing only. System-managed asynchronous duplexing is used.*
- *“dupsite” – Used with CF SITE, determines if the duplexed structure can be on ANYSITE, CROSSSITE, SAMESITE, or SAMESITEONLY.*

Note that it is possible to define a CFRM policy which permits duplexing to be performed for one or more CF structures (DUPLEX(ALLOWED) or DUPLEX(ENABLED)), and activate that CFRM policy, today. This is because the DUPLEX policy keyword is already supported for User-Managed Duplexing, it is simply being broadened to pertain to System-Managed CF Structure Duplexing as well. However, the CFRM CDS itself must also be upgraded to the level that supports System-Managed CF Structure Duplexing, as described above, in order for the DUPLEX specification to pertain to and support a System-Managed CF Structure Duplexing rebuild.

11. *Monitor and evaluate actual CF structure placement and performance for duplexed structures. Make any necessary tuning changes indicated by this.*
12. *Familiarize oneself with, and make automation changes with respect to the additional information returned by the D XCF,STR,STRNAME=xxxxxx command.*
13. *Understand and document the recovery procedures and behaviors for CF structure failures, CF failures, CF losses of connectivity, and system and sysplex outages, involving CFs containing duplexed structures. This is important so that the operations staff will be familiar with the operation of the sysplex, incorporating the set of duplexed structures during failure scenarios.*
14. *Understand differences in CF monitoring and performance procedures. This includes:*

System-Managed CF Structure Duplexing

- *Managing and monitoring the additional hardware resources related to System-Managed CF Structure Duplexing (e.g. CF to CF links),*
- *Monitoring and tuning the performance of the duplexed structures themselves*
- *Additional structures being reported on in the system monitors (e.g. RMF™)*

Once the CFRM CDS and policy migration steps have been taken, duplexing can be started for the desired structures. Again, this can be staged, structure by structure, dynamically and nondisruptively, through CFRM policy changes.

More information on setting up the CFRM Policy can be found in the “MVS z/OS Setting up a Sysplex” manual.

Managing and Monitoring

Starting Duplexing

There are two ways to start duplexing:

- *Activate a CFRM policy with DUPLEX (ENABLED) for the structure. If the "OLD" structure is currently allocated, then z/OS will automatically initiate the process to establish duplexing as soon as you activate the policy. If the structure is not currently allocated, then the duplexing process will be initiated automatically when the structure is allocated. This method attempts to reestablish duplexing automatically in the case of a failure, and also will periodically attempt to establish duplexing for structure that were not previously duplexed in accordance with the CFRM policy specification.*
- *Activate a CFRM policy with DUPLEX (ALLOWED) for the structure. This method allows the structures to be duplexed, however the duplexing must be initiated by a command - the system will not automatically duplex the structure. Duplexing may then be manually started via the existing SETXCF START,REBUILD,DUPLEX operator command or the IXLREBLD STARTDUPLEX programming interface. This method also requires that duplexing be manually reestablished in the event of a failure.*
- *Activate a CFRM policy with DUPLEX (ENABLED,ASYNCONLY) or DUPLEX(ALLOWED,ASYNCONLY) for Asynchronous duplexing for Db2 lock structures.*

System-Managed CF Structure Duplexing

For a production environment, the first method is recommended as it lends itself to a controlled change-management environment, and the change is permanent across sysplex-wide IPLs. Both methods should be used in the Test environment to build experience before going into production. A “hybrid” approach whereby DUPLEX(ENABLED) is used for some structures, and DUPLEX(ALLOWED) is used for others, is also discussed later in this document. An example of a CFRM policy entry

```
STRUCTURE NAME(DSNDB2P_SCA)
DUPLEX(ENABLED) /*SM CF Structure Duplexing Enabled*/
MINSIZE(32768) /* Minimum Size */
INITSIZE(32768) /* Initial Size */
SIZE(65536) /* Maximum Size */
ALLOWAUTOALT(YES) /* Enable Auto Alter */
FULLTHRESHOLD(80) /* Structure Full Monitoring threshold */
REBUILDPERCENT(1)
PREFLIST(CF01,CF02)
ENFORCEORDER(YES) /* Always follow PrefList. Not valid w/ ExclList */

STRUCTURE NAME(DSNDB2P_LOCK1)
DUPLEX(ENABLED,ASYNCONLY) /* Async Duplexing Enabled*/
INITSIZE(543M) /* Initial Size */
SIZE(581120K) /* Maximum Size */
REBUILDPERCENT(1)
PREFLIST(CF02,CF01)
ENFORCEORDER(YES) /* Always follow PrefList. Not valid w/ ExclList */
```

System-Managed CF Structure Duplexing

Stopping Duplexing

Duplexing may be manually stopped via the existing SETXCF STOP,REBUILD,DUPLEX operator command or IXLREBLD STOPDUPLEX programming interface.

When you need to stop duplexing structures, you must first decide which is to remain as the surviving simplex structure. Use the SETXCF STOP, REBUILD, STRNAME=xxxx,KEEP=OLD command to convert to using the primary structure as the simplex structure or with KEEP=NEW to leave the secondary as the simplex structure. z/OS will automatically stop duplexing when:

- *The CFRM policy changes to DUPLEX(DISABLED)*
- *Needed to allow a connect request to succeed*
- *Duplexing is "broken" as indicated by duplexed request response information*
- *A failure affects one of the structure instances*

When duplexing is stopped as a result of a failure affecting one of the structures (e.g. structure failure, CF failure, loss of connectivity), XES completely hides these failures from connectors so that stopping duplexing is the only recovery action taken. No loss of connectivity or structure failure events are presented to connectors. What happens next depends on the current setting of the DUPLEX policy parameter. If DUPLEX(ENABLED) is specified, the system will automatically try to re-establish duplexing the structure into the same CF with some caveats, or another CF if one is available. If DUPLEX(ALLOWED) is specified, re-duplexing does not occur automatically but must be done manually. If DUPLEX(DISABLED) is specified, then re-duplexing will not occur.

Operations and Displays

Several different command responses are affected by duplexing. One example includes the

"D XCF,STR,STRNAME=str_name" command. An example of the output is shown below. This example is for a System-Managed CF Structure Duplexed JES2 Checkpoint. The changed responses are highlighted.

The "OLD" structure (or "Primary") is the first structure of the duplexed pair that got allocated. The "NEW" structure (or "Secondary") is the second structure of the duplexed pair that got allocated. All reads go only to the OLD/Primary structure, while all updates are reflected in both structure instances.

System-Managed CF Structure Duplexing

The D XCF,CF,CFNM=* command shows the duplexed structure in both Coupling Facilities.

```
D XCF,STR,STRNAME=JES2_CKPT1
IXC360I 12.10.16 DISPLAY XCF 931
STRNAME: JES2_CKPT1
STATUS: REASON SPECIFIED WITH REBUILD START:
        POLICY-INITIATED
        DUPLEXING REBUILD
        METHOD          : SYSTEM-MANAGED
        AUTO VERSION   : B648FDD4 55CCED64
        REBUILD PHASE  : DUPLEX ESTABLISHED
POLICY INFORMATION:
POLICY SIZE      : 60000 K
POLICY INITSIZE : N/A
POLICY MINSIZE  : 0 K
FULLTHRESHOLD  : 80
ALLOWAUTOALT   : NO
REBUILD PERCENT: N/A
DUPLEX         : ENABLED
PREFERENCE LIST: CF01      CF02
ENFORCEORDER   : NO
EXCLUSION LIST IS EMPTY

DUPLEXING REBUILD NEW STRUCTURE
-----
ALLOCATION TIME: 11/28/2001 14:29:31
CFNAME        : CF01
COUPLING FACILITY: 002064.IBM.02.000000010A8B
                PARTITION: A  CPCID: 00
ACTUAL SIZE   : 60160 K
STORAGE INCREMENT SIZE: 256 K
PHYSICAL VERSION: B648FDF4 B3A04FA4
LOGICAL  VERSION: B644170C B7F3E740
SYSTEM-MANAGED PROCESS LEVEL: 8
XCF GRPNAME   : IXCLO002
DISPOSITION   : KEEP
ACCESS TIME   : NOLIMIT
MAX CONNECTIONS: 18
# CONNECTIONS : 4

DUPLEXING REBUILD OLD STRUCTURE
-----
ALLOCATION TIME: 11/28/2001 14:29:31
CFNAME        : CF02

CONNECTION NAME  ID VERSION  SYSNAME  JOBNAME  ASID  STATE
-----
JES2_W02        05 00050062 W02      JES2     0031 ACTIVE NEW,OLD
JES2_W03        06 0006004D W03      JES2     0032 ACTIVE NEW,OLD
JES2_W04        03 0003009B W04      JES2     0031 ACTIVE NEW,OLD
```

System-Managed CF Structure Duplexing

Additional D CF command output information is provided to show additional information on the CF-to-CF connectivity and links. This includes CHPID number and type for each receiver and sender CF link.

For example: (Note: This example shows only a single sender and receiver between the CFs, so there is a single point of failure in this configuration)

```
REMOTELY CONNECTED COUPLING FACILITIES

CFNAME  COUPLING FACILITY
-----  -----
CF02    SIMDEV.IBM.EN.ND0200000000
        PARTITION: 0  CPCID: 00
CHPIDS ON CF01 CONNECTED TO REMOTE FACILITY
RECEIVER:  CHPID  TYPE
           F1    CFR

          SENDER:  CHPID  TYPE
           E1    CFS

CF03    SIMDEV.IBM.EN.ND0100000000
        PARTITION: 0  CPCID: 00
CHPIDS ON CF01 CONNECTED TO REMOTE FACILITY
RECEIVER:  CHPID  TYPE
           F0    CFR

          SENDER:  CHPID  TYPE
           E0    CFS

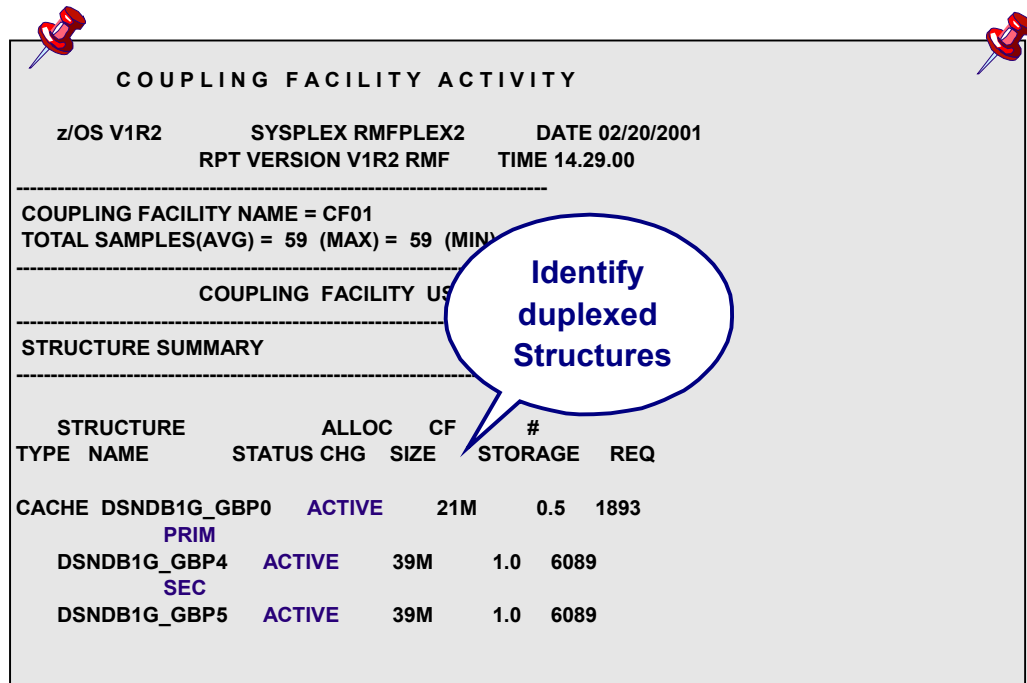
NOT OPERATIONAL CHPIDS ON TESTCF FF
```

Accounting and Measurement

RMF provides support to monitor and evaluate the actual CF structure placement and performance for duplexed structures, and thus for making any necessary tuning changes. This support is for both System-Managed and User (Db2) Managed duplexed structures.

The STATUS column of the Structure Summary Section indicates the duplexing attributes for a specific structure:

System-Managed CF Structure Duplexing



The screenshot shows a report titled "COUPLING FACILITY ACTIVITY" with the following details:

- z/OS V1R2 SYSPLEX RMFPLEX2 DATE 02/20/2001
- RPT VERSION V1R2 RMF TIME 14.29.00

COUPLING FACILITY NAME = CF01
TOTAL SAMPLES(AVG) = 59 (MAX) = 59 (MIN)

COUPLING FACILITY USE

STRUCTURE SUMMARY

STRUCTURE TYPE	NAME	ALLOC STATUS	CF CHG	SIZE	# STORAGE	REQ
CACHE	DSNDB1G_GBP0	ACTIVE		21M	0.5	1893
	PRIM					
	DSNDB1G_GBP4	ACTIVE		39M	1.0	6089
	SEC					
	DSNDB1G_GBP5	ACTIVE		39M	1.0	6089

A callout bubble with the text "Identify duplexed Structures" points to the "STRUCTURE SUMMARY" section of the report.

- *ACTIVE PRIM*
The structure is the rebuild-old (primary) structure in a duplexing rebuild process
- *ACTIVE SEC*
The structure is the rebuild-new (secondary) structure in a duplexing rebuild process

System-Managed CF Structure Duplexing

The new duplexing related delay reasons are shown in Structure Activity Section:

- *PR WT*: The amount of time that the system was holding one subchannel while waiting to get the other subchannel, to launch the duplexed operation.
- *PR CMP*: One of the two duplexed operations has completed, but the completed subchannel remains unavailable for use until the other operation completes

New Delay Reasons

- ▶ peer subchannel wait contention
- ▶ waiting for peer completion

COUPLING FACILITY STRUCTURE ACTIVITY												

STRUCTURE NAME = DSND81G_GBP0 TYPE = CACHE STATUS = ACTIVE PRIMARY												
# REQ REQUESTS DELAYED REQUESTS												
SYSTEM	TOTAL	#	% OF	-SERV TIME(MIC)-			REASON	#	% OF	AVG TIME(MIC) ----		
NAME	AVG/SEC	REQ	ALL	AVG	STD_DEV		REQ	REQ	/DEL	STD_DEV	/ALL	
RMF1	8621	SYNC	89	0.3	101.9	34.6	NO SCH	1	0.1	82.2	111.3	0.1
	4.79	ASYNC	8532	26.3	376.4	355.6	PR WT	4	0.1	285.8	440.2	0.2
		CHNGD	0	0.0			INCLUDED IN ASYNC	12	0.3	46.5	345.4	0.4
							DUMP	0	0.0	0.0		0.0

The CF-to-CF Activity Section shows a summary of basic counts for duplexing related operations only. In addition, the CF-to-CF section contains the information about the specific CF link types. This enhancement, the display CF link type information, has also been made in the existing subchannel activity section.

New CF to CF Activity Section

COUPLING FACILITY ACTIVITY													

z/OS V1R2			SYSPLEX RMFPLEX2			DATE 02/20/2001			INTERVAL 001.00.000				
RPT VERSION V1R2 RMF			TIME 14.29.00			CYCLE 01.000 SECONDS							

COUPLING FACILITY NAME = ICF1													

CF TO CF ACTIVITY													
# REQ REQUESTS DELAYED REQUESTS													
PEER	TOTAL	-- CF LINKS --	#	-SERVICE TIME(MIC)-			#	% OF	AVG TIME(MIC) ----				
CF	AVG/SEC	TYPE	USE	REQ	AVG	STD_DEV	REQ	REQ	/DEL	STD_DEV	/ALL		
ICF2	86830	CFR	2	SYNC	86830	53.0	33.8	SYNC	19	0.2	4.5	2.7	14.5
	86.2												

CF Link Type Information

System-Managed CF Structure Duplexing

Performance Impact of System-Managed CF Structure Duplexing

A Parallel Sysplex provides numerous benefits for the business. Like many things, there is a cost. This cost can be broken down into its factors:

1. *Software costs (pathlength) to manage a multi-system environment. This includes the additional work done by GRS serialization, JES2 shared Checkpoint, shared catalogs, etc.*
2. *Software costs to initiate and receive messages to and from a coupling facility.*
3. *Processor time needed to wait for synchronous messages to make the round trip to and from the CF. This time is a lost opportunity to process other instructions. This can be broken down into its components:*
 - *Time spent while the message is being prepared in the host hardware*
 - *Time spent while the message is traveling in the coupling link*
 - *Time spent while the message is being processed in the coupling facility*

System-Managed CF structure duplexing will impact these costs in a variety of ways.

Note, this does not apply to Asynchronous duplexing for Lock structures as described earlier.

Software costs to initiate and receive messages will increase. Simplex read operations will cost the same since z/OS will send a message to only one CF as before. The cost of Write operations (and any type of operation that modifies a structure) will increase, as the CF request has to be split, two messages sent, two messages received, and the results reconciled.

The CF response times from each duplexed (update) request will increase. Although the messages going to each of the two coupling facilities are overlapped for performance, there is additional time as each coupling facility coordinates the updates with its peer CF. For requests that are synchronous, a CP (engine) on the host system waits for both responses to come back before continuing processing. For example, a CP issuing 1000 CF requests/second with an average response time of 50 microseconds per request results in a CP cost of $(1000 \text{ req/sec}) * (.000050 \text{ sec/req}) = .05$ or 5%. If the synchronous response time went up to 75 microseconds, this coupling cost would increase to 7.5%. Note that z/OS has a heuristic algorithm to convert long synchronous requests to asynchronous, to limit this cost. It is based on observations of actual CF service times, and considers the speed of the sending CEC

System-Managed CF Structure Duplexing

to determine for each request whether it would be more CPU-efficient to process it synchronously or asynchronously.

The Read/Write ratios are different for each exploiter. Projecting the cost of System-Managed CF Structure Duplexing in terms of host effect (coupling cost), CF utilization, and CF Link utilization will therefore require an analysis of each structure based on the cost currently being paid for that structure in simplex mode and the read/write ratio. The appropriate duplexing costs can then be applied to reflect the duplexed write (update) requests. Many customers have tools that calculate these costs for their simplex case today. For those that need to estimate their current simplex costs, Appendix A contains a methodology that can be followed.

Depending upon which structures are being duplexed and how they are being invoked by the applications, System-Managed CF Structure Duplexing impacts include:

- *z/OS CPU utilization:*
For those operations that update the structure ...
 1. *Instead of paying the software cost to send and receive one CF message, two messages are being sent and two responses received with the results reconciled*
 2. *Additional pathlength is required to overlap the sending of the messages.*
 3. *For synchronous messages, the host has to wait until both messages return. This has the effect of increasing the synchronous response time, directly affecting host CPU utilization. Some requests may be converted to asynchronous to reduce the impact*
- *Coupling Facility CPU utilization:*
For those operations that update the structure ...
 1. *The coupling facility containing the original ("old," or "Primary") structure continues to process requests as when running in simplex mode*
 2. *The coupling facility containing the new structure now has to process requests that update the duplexed structure. The impact of processing these requests may have already been planned for to handle a CF rebuild situation in a simplex environment.*
 3. *Additional CF usage for both CFs is incurred to handle the CF-to-CF communication to coordinate the updates. This communication is done to ensure that both images of a structure remain synchronized with each other.*
- *Coupling Facility link usage:*
For those operations that update the structure ...

System-Managed CF Structure Duplexing

1. *There will be additional traffic on the links due to the additional requests to the new (or "secondary") structure.*
2. *The CF-to-CF communication requires CF links.*
3. *Since the z/OS-to-CF response times increase due to the CF to CF communication, the z/OS CF link subchannel utilization will increase*

It should be noted that CF storage requirements need not increase. Although a new structure is now required on the second CF, this space should have already been planned for to handle the CF rebuild situation.

It needs to be stressed again that the impact of these effects is on a structure by structure basis and is directly related to the amount of duplexing being done within each structure. For example, a cache structure may have 80% read accesses. Therefore, the System-Managed CF Structure Duplexing costs will apply to only the other 20% of its accesses that update the structure. It is up to each installation to determine the cost/benefit relationship before enabling System-Managed CF duplexing for a particular structure.

Structure Type	Host CPU Busy	CF CPU Busy	CF Link Subch Busy	Percent update
User Managed Duplexed GBPs	1.2x	2x	2x	1% to 100%, avg. 20%
(Synchronous) SM Lock	4x	5x	8x	100%
SM List	3x	4x	6x	Near 100%

Since User-Managed CF structure duplexing for Db2 Group Buffer Pool duplexing is similar to System-Managed CF Structure Duplexing in terms of both what it is used for and its external interface, data for projecting GBP duplexing is included. The cost of User-Managed CF Structure Duplexing for Db2 Group Buffer Pools is generally less because all of the synchronization costs are already paid through IRLM locking that also occurs in simplex mode, whereas for System-Managed CF Structure Duplexing the synchronization is obtained through additional CF-to-CF communication that does not occur in simplex mode.

An example of applying these numbers can be seen below:

Here, the host CPU capacity cost for the SIMPLEX version of the UM GBPs is shown to be 5%. To calculate the cost impact of User-Managed (Db2 Group Buffer Pool)

System-Managed CF Structure Duplexing

Duplexing one would take 20% of the simplex cost (reflecting the portion of activity to the structure that is updated and must be duplexed) and multiply that portion by 2. This would result in a total overhead equal to 1.2 times the SIMPLEX cost. In this example, this results in a total cost of 6% (thus, a growth of 1% above the simplex cost). Similarly, if a lock structure has a simplex cost of 2%, the duplexing impact would be found by multiplying 100% of the simplex cost by 4, yielding a total after-duplex cost of 8%.

Host CPU Capacity	Simplex	CF Duplex Multiplier	Duplex
UM GBPs	5%	1.2x	6%
SM Lock	2%	4x	8%
SM List	1%	3x	3%
Non-Duplexed	2%	n/a	2%
Total	10%		19%

System-Managed CF Structure Duplexing

It is very important to note that when using the simplex CPU cost calculation as a basis for determining what the projected CPU cost of CF Duplexing will be, care must be taken to apply the appropriate multipliers to the different components of the total CPU cost. For example:

- *The cost of handling asynchronous operations is increased by a factor of 2 - 2.5x by CF Duplexing.*
- *The cost of handling lock contention is not increased at all by CF Duplexing (that is, a factor of 1x).*
- *The cost of handling synchronous operations is increased by a factor of 3 - 4x, as shown in the table above.*

In the above example, it was assumed that essentially the entire lock structure and list structure simplex cost contribution came from synchronous operations, and that the contribution of asynchronous operations and of handling lock contention were negligible. Therefore, the multiplier of 3 - 4x was applied to the entire simplex cost. However, if a significant portion of the simplex cost had come from asynchronous operations or from handling lock contention, this would not have been appropriate; we would have had to separately apply a multiplier of 3 - 4x to the component of the simplex cost from synchronous operations, a multiplier of 1x to the component of the simplex cost from handling lock contention, and a multiplier of 2 - 2.5x to the component of the simplex cost from asynchronous operations.

As mentioned before, the CPU cost of Asynchronous Duplexing for (Db2) Lock Structures is near zero compared to simplexed IRLM lock structure.

The duplexing impact on CF CPU Busy and CF Link Subchannel Busy can be estimated in an analogous manner.

Avg CF CPU Busy	Simplex	Impact	Duplex
UM GBPs	15%	1.2x	18%
(Synchronous) SM Lock*	5%	5x	25%
SM List	4%	4x	16%
Non-Duplexed	6%	n/a	6%
Total	30%		65%

System-Managed CF Structure Duplexing

* This applies to when NOT using asynchronous duplexing for lock structures with Db2 (IRLM) locks. The CPU impact for Asynchronous duplexing is approximately "1." In other words, no impact.

For example, if CF01 required 5% of the CPU to process a simplexed Lock structure, then if it is duplexed, both CF01 and CF02 will each require 25% of the CPU to manage the structure.

Avg CF Link Sub ch Busy	Simplex	Impact	Duplex
UM GBPs	5%	1.2x	6%
SM Lock	2%	8x	16%
SM List	1%	6x	6%
Non-Duplexed	3%	n/a	3%
Total	10%		31%

System-Managed CF Structure Duplexing

Interpreting the Results

The increase in CF CPU usage is split between each of the two coupling facilities. Similarly, the increase in CF Link Subchannel usage is split between the links going to each of the coupling facilities. After System-Managed CF Structure Duplexing is turned on, some relocation of CF structures may be necessary to rebalance the coupling facility CPU and CF link subchannel utilization.

In both simplex and duplexed environments, the best performance occurs when the coupling facility is less than 50% CPU busy. At utilizations higher than this, significant queuing within the CF can occur, elongating response time and increasing z/OS CPU usage for synchronous activity. With this guideline in mind, some installations may have already planned spare CF CPU capacity (white space) for recovery situations. In these cases, less white space may be planned for after duplexing. The CPU capacity white space that would have been needed during a failure scenario is already being used on an ongoing basis with System-Managed CF Structure Duplexing.

In the simplex environment it is recommended that average Coupling Facility storage usage not exceed 50%. When planning for a failure situation, this is to allow a CF to rebuild all the structures on the other CF. With duplexed structures, the backup structures are preallocated, so the 50% rule is no longer applicable. There is no increase in CF storage required due to having duplexed structures, provided that the 50% rule was followed. The storage capacity white space that would have been needed for rebuild purposes is already being used on an ongoing basis with System-Managed CF Structure Duplexing.

The link subchannel utilization and response times, affecting z/OS CPU requirements, are directly dependent upon the coupling technology employed. This includes coupling facility technology as well as link technology. As seen in customer studies, (See Appendix B: Host Effects of Technology in a Parallel Sysplex), significant performance benefits to the z/OS partitions can be obtained by migrating to more efficient technologies. This holds true whether structures are Simplex or Duplexed.

Utilizing newer technology increases the coupling efficiency by improving the response times. It should be remembered that with System-Managed CF Structure Duplexing you can only go as fast as your slowest CF partition and link, because each operation waits for its peer to complete. For sysplexes with a mixture of coupling

System-Managed CF Structure Duplexing

facility CPU and link technologies, duplexing efficiency will be determined by the slowest component.

More information on link types can be found in "Coupling Facility Configuration Options" available linked off of the Parallel Sysplex home page at:

<https://www.ibm.com/it-infrastructure/z/technologies/parallel-sysplex>

Recovery Benefits - Duplexing “Failover”

One of the major benefits of System-Managed CF Structure Duplexing is its ability to rapidly recover from situations where there is a loss of connectivity to a Coupling Facility or a CF failure. In an environment was configured to illustrate this, a CICS/Db2 workload was run with 24 structures spread across two coupling facilities. A loss of connectivity to one of the CFs resulted in all structures in that CF being rebuilt or recovered via duplexing failover. The results were:

Configuration	Time	Ratio
Simplex structures	117 sec	1
Simplex structures, UM duplexed GBPs	84 sec	0.72
SM duplexed structures, UM duplexed GBPs	28 sec	0.24

This experiment did not include the case where the Db2 Group Buffer Pool structures had to be rebuilt from logs as they would have if they failed while in simplex mode). Experience suggests that the ratio time for log recovery would be 10 times longer than the rebuild case, or about 40 times slower than the SM+UM case above. It should be noted as well that recovery times are affected by the number and size of structures that must be recovered, thus larger configurations would result in bigger differences between the scenarios.

Hardware Configuration Considerations and Recommendations

This section describes specific configuration options and considerations associated with System-Managed CF Structure Duplexing and provides specific recommendations for each of those configuration options.

Distance Considerations

System-Managed CF Structure Duplexing

As the distances between two coupling facilities increase, the response time needed for the CFs to exchange signals also increases. At some point, CF communications are converted from synchronous to asynchronous to limit the CPU overhead for z/OS.

The CF link and subchannel remain allocated for the interval as reported by the response time, increasing the distances between host and CF causes the link and subchannel utilization to increase. Increased subchannel and link utilization would also affect the path busy conditions for shared links and subchannel queuing. These effects are true for simplex as well as duplexed structures, for z/OS to CF links and CF to CF links.

For a GDPS Metro configuration, System-Managed CF Structure Duplexing can still provide the same benefits in terms of improved structure data availability that it does in a non-GDPS configuration. However, in a GDPS configuration, System-Managed CF Structure Duplexing may incur substantial distance effects as described above, especially when one of the coupling facilities is located in one site and the other coupling facility is located in the other site (that is, when the CF-to-CF links are inter-site long links). This distance effect is undesirable, and in a GDPS site failover situation, if GDPS "Enhanced Recovery Support" is not configured, the duplexed copies of the CF structure data are not preserved for use in any event, so GDPS really derives no value from having one copy of the CF structure data resident in each of the two sites.

Recommendation: As z/OS to CF and CF-to-CF distances increase, monitor the coupling facility link/subchannel and path-busy status. If more than 10% of all requests are being delayed on the CF link due to subchannel or path busy condition, either migrate to peer mode links to increase the number of subchannels from two to seven for each link, or configure additional link(s).

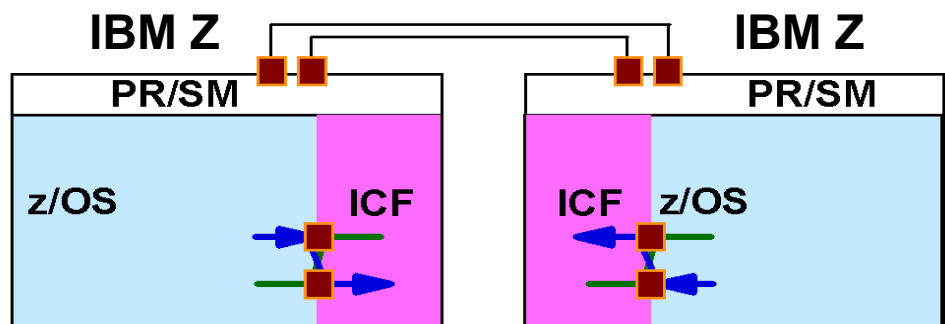
Recommendation: In a GDPS Metro multi-site configuration, do not duplex CF structure data between coupling facilities located in different sites (other than with Asynchronous Duplexing for Lock structures. More on that later). If desired, duplex the structures between two coupling facilities located at the same site. CF structure data is not preserved in GDPS site failover situations, regardless of duplexing.

System-Managed CF Structure Duplexing

A connectivity requirement for System-Managed CF Structure Duplexing is that there must be bi-directional CF-to-CF connectivity between each pair of CFs in which duplexed structure instances reside. This connectivity can be provided by a single bi-directional CF link (two with redundancy) between IBM Z processors.

CF-to-CF Links - Sharing and Redundancy

CF-to-CF links may either be dedicated or shared via Multiple Image Facility (MIF). They may be shared with z/OS-to-CF links between z/OS and CF images in the pair of CECs they connect. When the same physical link is shared as both a z/OS-to-CF link and a CF-to-CF link, the usage of the link as a CF-to-CF link may cause a small amount of path busy conditions to occur for the z/OS-to-CF usage of the link (even when that physical link is "dedicated" in so far as it is serving only one z/OS-to-CF link). This amount of path busy is generally not significant and causes no noticeable performance concerns. Multiple CHPIDs can also be shared using the same physical link.



System-Managed CF Structure Duplexing

Recommendation: Provide two or more physical CF-to-CF links between each pair of CFs participating in duplexing. The physical CF-to-CF links may be shared by a combination of z/OS-to-CF links and CF-to-CF links.

z/OS CPs - sharing and capacity

When z/OS drives duplexed requests to a duplexed CF structure, z/OS launches the two requests to the two coupling facilities serially, by a routine running disabled on a z/OS CP. When the z/OS processor is dedicated, these two duplexed requests will both be launched in as expeditious a fashion as possible, allowing for maximum parallelism in executing the duplexed request and yielding optimal CF duplexing performance.

When the z/OS processor is shared, there is a small chance that PR/SM™ will take control away from the z/OS partition on the processor during the window of time between launching one duplexed request and the other. The second request will not be launched until PR/SM re-dispatches the z/OS partition. When this occurs, one duplexed request may reach one of the CFs well before the other request has even been launched, causing a delay between the execution of the two duplexed commands. This results in less parallelism for the duplexed request, less optimal use of CF resources, and less optimal System-Managed CF Structure Duplexing performance overall. In general, the chances of a duplexed request falling into this window are small, as is the penalty associated with sharing of z/OS CPs for System-Managed CF Structure Duplexing.

The overall synchronous service time for a duplexed request, and the corresponding z/OS CP utilization, is significantly higher for duplexed requests as compared to simplex requests. The sync/async conversion algorithm (see Washington Systems Center Flash 10159 for additional information) tends to react to this by preferentially converting the slower duplexed operations to asynchronous execution, more so than it converts simplex operations. Converting these requests to be processed asynchronously limits the increase in CPU consumption that would otherwise result from the z/OS processor waiting for synchronous completion of these requests but at the expense of further elongating the response time for these requests since asynchronously processed requests are subject to additional z/OS polling latencies. Given these considerations, a modest increase in the z/OS CPU utilization will be seen when operation is converted from simplex to duplexed. Increased CF service times and increased conversion of CF requests from synchronous to asynchronous

System-Managed CF Structure Duplexing

execution will also be seen. In general, the increased CF service times have little effect on overall transaction response time for the workload, but is something that should be watched carefully.

Recommendation: You may provide either dedicated or shared z/OS CPs when using System-Managed CF Structure Duplexing. Generally, it is not necessary to provide dedicated z/OS CPs to obtain good performance with CF Duplexing. Be prepared to provide additional z/OS CPU capacity when the workload's CF operations become duplexed.

z/OS-to-CF Links - sharing and redundancy

As in simplex mode, at least one (or two for redundancy) z/OS-to-CF link must be defined between each z/OS image using a coupling facility, and each CF image that it is using. Additional z/OS-to-CF links may be required for additional capacity.

In simplex mode, the sharing of z/OS-to-CF links between multiple LPARs via MIF can result in path busy conditions. These occur when there is contention for the physical resources of the link, between the various z/OS LPARs that are sharing the link. Each LPAR will handle these path busy conditions by redriving the requests over and over again until they succeed. This can substantially increase the CPU cost of accessing the coupling facility for the sharing z/OS partitions. For this reason, it is recommended that path busy conditions be limited to at most 10-20% of the total requests to a CF. If the occurrence of path busy conditions exceeds this amount, it should be addressed either by providing more shared links for use by the z/OS partitions, or preferably by dedicating links to the z/OS partitions using them. Path busy conditions do not occur with dedicated links.

The considerations for z/OS-to-CF links are similar for System-Managed CF Structure Duplexing. Sharing of links between multiple z/OS partitions can similarly cause contention for CF link resources and result in path busy conditions which must be redriven by z/OS. However, there are several factors which make this even more expensive in a duplexed environment. First, duplexed requests in general have longer service times and are thus active on a link for longer periods of time compared with a simplex request. Other things being equal, they are much more likely to experience path busy conditions and need to be redriven more frequently before the operation is successfully launched. Second, since a duplexed request starts two operations

System-Managed CF Structure Duplexing

instead of just one, the likelihood of experiencing a path busy condition on at least one of them is substantially increased compared with a simplex request. Third, once the first CF request of a duplexed pair is successfully launched, any path busy conditions that occur during the starting of the second CF request of the pair will delay the second request from being started, causing a "skew" between the arrival and execution of the two duplexed commands at the CFs. This in turn results in less parallelism for the duplexed request, less optimal use of CF resources, and less optimal System-Managed CF Structure Duplexing performance overall.

System-Managed CF Structure Duplexing

Recommendation: For redundancy, provide two or more z/OS-to-CF links from each system to each CF. Provide dedicated z/OS-to-CF links if possible. If z/OS-to-CF links are shared between z/OS partitions, the occurrence of path busy conditions should be limited to at most 10-20% of total requests. If path busy exceeds this guideline, either provide dedicated links, or provide additional shared links, to eliminate or reduce the contention for these link resources. Use peer links whenever possible.

Coupling Facility CP Utilization - capacity

CF requests get the best response times when the coupling facility CP utilization is limited to 50% or less. At higher utilizations, it becomes less likely that the CF processor(s) will be available to process incoming requests in a timely manner, degrading the CF request service time.

The recommendations for coupling facility CP utilization are similar for System-Managed CF Structure Duplexing. However, there are several factors which make high coupling facility CP utilization even more expensive in a duplexed environment. First, in order to obtain optimal CF duplexing performance, both CF commands must be received by their respective CFs and begin execution in a timely fashion. Any delay in executing either of the requests may ultimately result in delayed completion of the duplexed pair of requests as well as less optimal usage of CF resources. Second, as duplexed CF requests exchange signals in order to coordinate and synchronize their execution across the pair of CFs, the duplexed commands may need to suspend their execution while awaiting a signal from the other CF and then resume execution when the signal arrives. Any delay in resuming execution caused by excessive coupling facility CP utilization will ultimately result in delayed completion of the duplexed pair of requests and less optimal usage of CF resources.

Recommendation: Provide sufficient coupling facility CP resources so that coupling facility CP utilization remains below 50% in all CF images. If coupling facility CP utilization exceeds this guideline, then

- *Provide additional CF processor resources to reduce the overall utilization.*
- *Try to rebalance or redistribute duplexed structures between CFs to reduce the utilization of an overutilized one.*
- *If you already have three or more CF engines on the server, try to limit the number of structures enabled for System-Managed CF duplexing.*
- *Select structures based on the hierarchy described on the section: "When to Configure System-Managed CF Structure Duplexing."*

System-Managed CF Structure Duplexing

Coupling Facility CPs - sharing

In simplex mode, running with shared coupling facility CPs can cause longer CF request service times compared to running CFs with dedicated engines. The reason for this is that by default the CF is not interrupt driven, it is a polling engine that is continuously looking for requests to process. When the CF is using a processor that is shared with other images, there are periods in time where the CP is unavailable to poll for and process any incoming work for that CF image simply because the shared CP happens to be doing work for another image at the time. This latency is what drives the average CF service times up with shared CPs. Since some requests experience this latency while others do not, another effect of shared Coupling Facility CPs is a large standard deviation on the response times as reported by RMF.

For System-Managed CF Structure Duplexing, the latency effect of shared coupling facility CPs is magnified even further. Processing of a duplexed operation requires several exchanges of coordination signals between the two CFs containing the duplexed structure instances. Each exchange requires the attention of the CF at the other end to perform certain activities and respond. If the CFs involved in duplexing are using shared engines, then the kind of shared CP latency described above can occur not only at the beginning of each request, but also at several points during the processing of each duplexed request. This makes the performance of shared-CP coupling facilities for CF duplexing much worse than with dedicated-CP coupling facilities. In extreme cases, the duplexed structures may even have difficulty maintaining their duplexed state. When this occurs, one or more structures may "break duplexing" spontaneously and revert to simplex mode.

To resolve this issue with sharing CPs we recommend using Dynamic CF Dispatching using "Coupling Thin Interrupts." When the Coupling Facility is defined using the "DYNDISP=THIN" keyword, the CF will give up the processor as soon as it has no work to do and become eligible for dispatching once an interrupt is received indicating that it has work to do. Since the time the CF might not be dispatched when a request arrives is virtually eliminated by the use of the interrupt, the CF service times are improved and generally less variable. This option is more aggressive about giving up use of the processor when it does not have work to do so it allows for a more efficient sharing of the process or among the multiple CF images.

System-Managed CF Structure Duplexing

An additional benefit of Coupling Thin Interrupts is when z/OS images are sharing the same CP. In this case, Thin Interrupts helps get the z/OS dispatched quickly when it receives an interrupt that an asynchronous CF operation completed.

Recommendation: When performance is critical, ensure all production CFs participating in System-Managed CF Structure Duplexing always have at least one (physical) CP which is available 100% of the time.

Recommendation: If shared engines must be used, then define the CF images involved with duplexing using DYNDISP=THIN.

More information on Coupling Facility Thin Interrupts can be found at:

<https://www.ibm.com/support/pages/coupling-thin-interrupts-and-coupling-facility-performance-shared-processor-environments>

System-Managed CF Structure Duplexing

Coupling Facility CPs - "balanced" capacity

Another issue related to coupling facility CPs for System-Managed CF Structure Duplexing is that of "balanced" capacity. Given the "lockstep" nature of the synchronization of the execution of duplexed commands, the completion of the duplexed operation is gated by the execution speed of the slower of the two operations.

When there is a mismatch in either raw processor speed (e.g. machine type of the CF), or overall coupling facility CPU capacity (e.g. number of coupling facility CPs, CF LPAR weights, shared vs. dedicated coupling facility CPs, etc.), the slower of the two CFs will tend to become a bottleneck. This can cause the overall service time for the duplexed requests to degrade and the duplexed requests may utilize the task and processor resources of the faster CF inefficiently because the duplexed commands are not able to execute at the "full potential" of the faster CF engines on which they are executing. These effects can be minimized by avoiding such imbalances in CF processor speed and capacity and by carefully following the recommendations above regarding overall "Coupling Facility CP Utilization."

System-Managed CF Structure Duplexing

Recommendation: Provide "balanced" coupling facility CP capacity between CF images participating in CF Duplexing, if possible. Avoid such significant imbalances as one CF with shared CPs and the other CF with dedicated CPs, CFs with wildly disparate numbers of CPs, CFs of different machine types with very different raw processor speed, etc. In general, re-balancing of coupling facility CP capacities may be necessary for any processor upgrade affecting a CF, and should be part of the installation planning prior to production use of the new/upgraded processor.

If such imbalances are unavoidable, it may be possible to compensate for them in other ways. For example, consider providing more CPs in the slower CF than there are in the faster CF to compensate for the inherent difference in engine speed. Another option is to consider redistributing the simplex structures so as to push more of the simplex workload at the faster CF, leaving the more constrained, slower CF free to devote more of its capacity to servicing the duplexed requests.

DUPLEX(ALLOWED) vs DUPLEX(ENABLED) Considerations

When defining the CFRM policy for duplexed structures, one must define either DUPLEX(ALLOWED) or DUPLEX(ENABLED) to indicate that a structure is to be duplexed. Specifying or defaulting to DUPLEX(DISABLED) for a structure makes it ineligible to be duplexed. At least two CF images that are connected to one another with CF-to-CF links must also be specified in the preference list (PREFLIST) for the structure.

Specifying DUPLEX(ENABLED) is often desirable from an operational standpoint because it minimizes the amount of operator intervention required for duplexing a structure and makes sure that it remains duplexed whenever possible. z/OS will automatically start duplexing for these structures and attempt to keep them duplexed (re-duplexing if needed) as best it can. On the other hand, DUPLEX(ALLOWED) requires manual intervention via the SETXCF command to decide when to start or stop duplexing a particular structure, giving the installation finer control over duplexing (and re-duplexing) the structure.

There are some workload availability considerations that come into play in deciding how to specify this attribute. If DUPLEX(ENABLED) is specified with three or more CFs in each structure's preference list, then in the event of a CF failure or loss of CF connectivity all structures in the affected CF will revert to simplex mode and then immediately re-duplex into the other available CF in the preference list. If there are

System-Managed CF Structure Duplexing

only two CFs in the structure's preference list, then in the event of a CF failure or loss of CF connectivity, the structure will first revert to simplex mode and then as soon as the second CF is restored, it will re-duplex back into that restored CF. While reverting to simplex mode requires no propagation/copying of structure data and is generally quite fast and non-disruptive, re-duplexing the structure requires that z/OS copy the structure content into a newly-allocated secondary structure instance. Reduplexing the structure can be somewhat disruptive to the ongoing workload that is using the structure as structure activity is temporarily quiesced during this copy process.

An installation can choose to specify DUPLEX(ENABLED) for a particular structure and the structure will automatically maintain its duplexed state as best it can without manual intervention, but in the event of a failover to simplex mode there will be a temporary disruption in workload processing when the structure automatically tries to reestablish duplexing at the earliest possible opportunity. Alternatively, an installation can choose to specify DUPLEX(ALLOWED) for a particular structure, which will require more operator intervention to start and maintain duplexing over time, but in the event of a failover to simplex mode the installation can plan for and control when it will undertake the temporary workload disruption associated with re-duplexing the structure (also taking into consideration the risks associated with leaving the structure in simplex mode for some period of time, before it is eventually re-duplexed).

To summarize, the choice of DUPLEX(ALLOWED) vs. DUPLEX(ENABLED) amounts to an installation making the tradeoff between three competing considerations: avoiding manual intervention to start and maintain duplexing, avoiding unnecessary disruption caused by re-duplexing structures at inopportune times (such as during a workload peak), and avoiding risks associated with having structures remain in simplex mode any longer than necessary.

System-Managed CF Structure Duplexing

Recommendation: In general, the use of DUPLEX(ALLOWED) is recommended. With this option, minimal disruption to the workload occurs at the time of an unplanned failure (e.g. CF failure, loss of CF connectivity) which causes structures to revert to simplex mode.

If DUPLEX(ALLOWED) is used, the installation must carefully document the protocols for reestablishing duplexing once the CF-related failure has been recovered including determining *when* to undertake this processing so as to minimize the disruption of the workload caused by the re-duplexing process while at the same time considering that the structures are exposed to the possibility of even more disruptive recovery actions should a second failure occur while the structures are still operating in simplex mode.

An installation that wishes to minimize this “double failure” risk should make use of DUPLEX(ENABLED) as this option will tend to ensure that the structures remain duplexed as much of the time as possible without requiring manual intervention to do so. An installation using DUPLEX(ENABLED) should also maintain an alternative copy of the CFRM policy that is defined with DUPLEX(ALLOWED) for use during planned CF reconfiguration/maintenance actions.

A hybrid approach is also recommended. Specify DUPLEX(ENABLED) for those structures which *do not* support user-managed rebuild processing (or which have failure-isolation considerations that make user-managed rebuild impossible in certain failure scenarios), and specify DUPLEX(ALLOWED) for those structures which *do* support user-managed rebuild (and which do not have any special failure-isolation considerations). With this approach, if a failure occurs the only structures that will automatically and immediately re-duplex themselves are those for which duplexing provides the *only* robust structure recovery mechanism. The structures that are not as dependent on duplexing as a recovery mechanism will remain in simplex mode until the installation chooses a convenient time to manually re-duplex them. This alternative is a good compromise in terms of reducing the workload disruption of re-duplexing immediately after a failure while also mitigating the risks associated with leaving some structures in simplex mode for an extended period of time.

System-Managed CF Structure Duplexing

Structure Sizing Considerations for System-Managed CF Structure Duplexing

Each Coupling Facility release usually will have increased storage requirements relative to the same structure allocated in a previous level Coupling Facility. The amount of increase varies, dependent on new function in that release as well as the specific structure attributes, so there is no simple "rule of thumb" that can be used to estimate the amount of storage increase that may occur. The CFSIZER utility may be used to calculate the appropriate structure size for a given type of structure. Each of the two duplexed structure instances will require this amount of CF space. The CFSIZER and the SIZER utilities are available under the "Tools" navigation option on the Parallel Sysplex home page at <https://www.ibm.com/it-infrastructure/z/technologies/parallel-sysplex> .

When defining the CFRM policy size parameters (SIZE, INITSIZE, and MINSIZE) for a duplexed structure, the size values specified should define the size of one of the structure instances, not both added together.

Recommendation: Perform the appropriate structure sizing/re-sizing for duplexed structures.

Recommendation: Since *each* of the duplexed structure instances will require the calculated amount of CF space, you must plan for that amount of space in *each* of the CF instances in which a copy of the duplexed structure may reside. However, the CFRM policy values that you specify for structure size (e.g. SIZE, INITSIZE, and MINSIZE) should define the size of *one* of the structure instances, not both added together.

System-Managed CF Structure Duplexing

Summary

System-Managed CF Structure Duplexing is the next evolutionary step in the Parallel Sysplex architecture, providing value through:

- *Eliminating Staging Data Sets for the System Logger as a requirement for availability if a CF or Logger structure fails. This is expected to provide both CPU and response time benefit.*
- *System Management benefits by simplifying recovery procedures. A major benefactor of this is IMS Shared DEDB/VSO structures.*
- *Asynchronous duplexing for (IRLM for Db2) lock structures is always recommended.*
- *Basic recovery for failed structures, failed CFs, and losses of CF connectivity. Without System-Managed Duplexing, many structures have no simple means to recover data. This includes BatchPipes, CICS Data Tables, Temporary Storage, and Named Counters structures.*
- *Faster recovery for structures that would otherwise require reading log data to recover. The IMS Shared Message Queue is one example of this.*

In addition, System-Managed CF Structure Duplexing is designed to help provide direct financial savings by potentially enabling the use of an ICF instead of a standalone model Coupling Facility to simplify the Parallel Sysplex configuration.

Consistent with z/OS's design points of Reliability/Availability/Serviceability, once the system is enabled for duplexing, migrating in and out of duplexing mode on a structure by structure basis is dynamic. It can be as simple as issuing an operator command or making a CFRM policy update. In addition, RMF reports on the performance and service characteristics of the environment for monitoring and tuning.

As with any technology, a cost / benefit analysis of System-Managed CF Structure Duplexing needs to be done for each exploiter to determine the value in each environment.

Appendix A: Calculating Simplex Resource Use

Host CPU Capacity

The Parallel Sysplex technology provides many benefits to the z Systems environment, including high availability, workload balancing, scalable growth, reduced cost of computing, ease of use, and investment protection of current applications. System-Managed CF Structure Duplexing enhances this by providing a general-purpose, hardware-assisted, easy-to-exploit mechanism for duplexing CF structure data, and providing a robust recovery mechanism for that duplexed CF structure data during failover situations. But there is an impact on the host (z/OS) CPU utilization.

To measure the effect on host CPU capacity of each CF structure, one would need to

1. Calculate the CPU time spent on each individual CF request
2. Calculate the rate of each CF request
3. Calculate amount of CPU time spent by the system on behalf of each structure. This is (CF Access Rate) * (Software + Hardware time)
This is then divided by the amount of engines in the server to get the percent of time spent managing that structure.
4. Multiplied this by the impact of duplexing the structure as described earlier in the document. Subtract the value obtained from step (3) from this to get the cost of duplexing that structure. Remember, there is approximately no CPU cost on the z/OS side for duplexing a Db2 (IRLM) lock structure with Asynchronous Duplexing.

Calculate the cost of a single CF request

Approximate software times for each type of request can be found in the table below. As these are machine dependent, sample times (in microseconds) are shown for mid-sized models in each family of servers. The values are scaled by the single processor speed of each model (total model capacity divided by number of CPs). Performance ratios for the processors shown and any other models are available in IBM

System-Managed CF Structure Duplexing

ResourceLink <https://www.ibm.com/servers/resourceLink/>. From the home page, go to the Planning tab and go to “Large Systems Performance Reference (LSPR) data.”

Host software times (in microseconds) for various types of requests to the coupling facility are estimated as follows. This is **NOT** the total response time as that includes hardware time in the links, as well as time going to and from the CF, and time in the CF.

Model	LSPR MIPS (PCI)	Engine Speed	Synchronous Lock	Lock Contention	Synchronous List/Cache	Asynchronous
zBC12						
2828-D04	266	66.5	36.3	1142	61.6	288.5
z13s						
2965-D04	386	96.5	25	787	42.4	198.8
z15 T01						
8561-510	6571	657	3.7	115.6	6.2	29.2
8561-725	35,436	1,417	1.7	53.6	2.9	13.5
Z15 T02						
8562-D05	666	133	18.2	571	35.8	144.3
8562-R05	3310	662	3.6	114.7	6.2	29.0
Z14						
3906-505	3358	672	3.6	113.0	6.1	28.5
3906-725	31,084	1,243	1.9	61.1	3.3	15.4
Z14 ZR1						
3907-D04	462	116	20.8	654.7	35.3	171.7
3907-R04	2372	593	4.1	128.1	6.9	32.4

The fields are calculated using the methodology of:

- Engine Speed = LSPR MIPS / Number of CPs
- Compare the engine speed with a base server, such as the z12BC D04. This server is here to minimize errors in calculation since the engine speed value is low. Get the ratio in the difference of engine speeds.
- Divide the numbers for the z12BC D10 in the other fields by this ratio to get the values for the new servers.

For example, looking at the **z15 T01 – 510** in the table above:

System-Managed CF Structure Duplexing

Model	(PCI)	Engine Speed	Synch Lock	Lock Contention	Synchronous List/Cache	Asynchronous
z12BC D04 266	66.5	36.3	1142	61.6	288.5	
z15 T01 510	6571	657	3.7	115.6	6.2	29.2

6571 is the PCI rating from the LSPR chart.

657 is the engine speed. Since the 510 is a 10-way server, $6571/10 = 657$

Since 66.5 is the engine speed of the z12BC D04, the z15 T01-510 is 9.89x faster than the z14 BC D04 ($657/66.5 = 9.89$).

Divide the CF request types in the z12BC D04 by this ratio (9.89) to get the values for the z15 T01 510.

Synchronous Lock = $36.3 / 9.89 = 3.7$
 Lock Contention = $1142 / 9.89 = 115.6$
 Synchronous List/Cache = $61.6 / 9.89 = 6.2$
 Asynchronous = $288.5 / 9.89 = 29.2$

If you are interested in a model not listed, calculate the Processor Capacity Index (PCI) rating per CP for that model and one in the table. For example:

z15 T01	CPs	PCI	Engine Speed
8561-711	11	18,369	1,670
8561-750	50	63,594	1,271
8561-775	75	88,312	1,177

Again, the PCI ratings are available in IBM ResourceLink

<https://www.ibm.com/servers/resourceLink/>. From the home page, go to the Planning tab and go to "Large Systems Performance Reference (LSPR) data." Note that there are multiple LSPR tables, depending upon the z/OS level being run. Be sure to use values within the same table for better accuracy.

Calculate the CF activity rates

The CF access rates and hardware times are easily determined from an RMF Structure Activity Report. For example, consider this hypothetical structure:

Interval = 15.00.00

STRUCTURE NAME	NAME	# REQ	TYPE	STATUS	REQUESTS	-SERV TIME (MIC) -
SYSTEM NAME	TOTAL AVG/SEC	-----	# REQ	% OF ALL	AVG	STD_DEV
SYS2	12345K	SYNC	12M	100	19.7	1.1
	13717	ASYN	325K	0.0	139.1	59.8

System-Managed CF Structure Duplexing

```

CHNGD      0      0.0  INCLUDED IN ASYNC
SUPPR      0      0.0

```

The synchronous hardware response time is 19.7 microseconds for 12M synchronous requests over 900 seconds, or 13,333 per second in this 15 minute interval. The asynchronous hardware response times are not used in a CPU time calculation as the host processor does not wait for these requests to finish. However, there are larger software costs (compared to synchronous operations) associated with starting and completing these requests.

Another component of CPU cost for coupling functions involves resolving lock contention (note the software cost of lock contention in the table above). Lock contention rates are identified at the far right of the lock structure activity report in RMF. Consider the extracted example below:

STRUCTURE NAME = DSND BAP_LOCK1		TYPE = LOCK		STATUS = ACTIVE		
SYSTEM NAME	# REQ TOTAL AVG/SEC	----- REQUESTS	# REQ	% OF ALL	-SERV TIME (MIC)- AVG	STD_DEV
MXP2	3262K	SYNC	3206K	98.3	19.4	1.5
	3625	ASYNC	56K	1.7	67.3	25.1
		CHNGD	0	0.0	INCLUDED IN ASYNC	
		SUPPR	0	0.0		

CONTENTIONS	
# REQ	5326K
# REQ DELAYED	25K
- CONT	119K
- FALSE CONT	8796

Using the "-CONT" field and assuming a 15 minute interval, this results in $(119,000 / 900) = 132$ contention events per second.

System-Managed CF Structure Duplexing

Using the data from the structure activity report and the software table, the simplex host cost for each structure can be determined. Take, for example, an IBM Z server connected to a CF with three structures. The RMF report for the structures might look like the following:

STRUCTURE NAME = DSNDB0P_LOCK1							TYPE = LOCK		STATUS = ACTIVE	
	# REQ	-----			REQUESTS		-----			
SYSTEM	TOTAL		#	% OF	-SERV TIME (MIC) -					
NAME	AVG/SEC		REQ	ALL	AVG	STD_DEV				
SYS2	3262K	SYNC	3206K	98.3	19.4	1.5				
	3625	ASYN	56K	1.7	67.3	25.1				
		CHNGD	0	0.0	INCLUDED IN ASYN					
		SUPPR	0	0.0						
STRUCTURE NAME = LISTSTR							TYPE = LIST		STATUS = ACTIVE	
	# REQ	-----			REQUESTS		-----			
SYSTEM	TOTAL		#	% OF	-SERV TIME (MIC) -					
NAME	AVG/SEC		REQ	ALL	AVG	STD_DEV				
SYS2	180K	SYNC	150K	83.3	4.3	2.4				
	200.3	ASYN	30K	17.7	82.7	9.5				
		CHNGD	0	0.0	INCLUDED IN ASYN					
STRUCTURE NAME = CACHE_STR							TYPE = CACHE		STATUS = ACTIVE	
	# REQ	-----			REQUESTS		-----			
SYSTEM	TOTAL		#	% OF	-SERV TIME (MIC) -					
NAME	AVG/SEC		REQ	ALL	AVG	STD_DEV				
SYS2	12849K	SYNC	12M	93.4	19.7	1.1				
	14277	ASYN	847K	6.6	139.1	59.8				
		CHNGD	0	0.0	INCLUDED IN ASYN					
		SUPPR	0	0.0						

Calculate the host impact of a duplexed CF structure

From the example report above, we can construct a worksheet for each structure using software times from the z15 T01 - 510 row of the host software time table, and frequencies and hardware times from the RMF reports. Note that "CPU Cost" column has been converted from microseconds to seconds.

System-Managed CF Structure Duplexing

Structure	Frequency * (per sec)	(Software + (mics)	Hardware) (mics)	=	CPU Cost (sec)
Lock_Str	3625 *	(3.7 +	19.4)	=	0.073
Lock Contention	132 *	(115.6)		=	0.015
List_Str synch	167 *	(6.2 +	4.3)	=	0.002
List_Str asynch	35 *	(29.2)		=	0.001
Cache_Str synch	13335 *	(6.2 +	19.7)	=	0.345
Cache_Str asynch	942 *	(29.2)		=	0.027
					====
			Total		0.463 Seconds

To calculate the CPU cost for coupling as a percentage of host capacity, one must divide the CPU time by the number of engines in the host. Thus, for this 10-way processor example, the total time is divided by 10 engines and then transformed into a percentage (by multiplying by 100). Thus, we have $(0.463/10) = 0.046$, or 4.6% simplex host coupling cost for the total of all three structures. Individually, the structure impact on host capacity cost would be calculated similarly yielding in this example 0.9% for Lock_Str, 0.3% for List_Str, and 3.7% for Cache_Str.

The final step is to multiply the host impact of a simplex structure by the factor as described in this table shown earlier:

Host CPU Capacity	CF Duplex Multiplier
UM GBPs	1.2x
SM Lock	4x
SM List	3x

Using our example, CPU time duplexing the calculate the list structure will be $3 \times 0.3 = .9\%$. Subtract the original 0.3% gives a cost of .6% CPU.

Again, the table shown assumes you are not using Asynchronous Duplexing. There is no impact for that, so the multiplier will be just 1x.

CF CPU Capacity

Information needed to calculate the current contribution to CF CPU utilization of each structure in a coupling facility can be found in the Coupling Facility Usage Summary of

System-Managed CF Structure Duplexing

RMF's CF Activity Report. An example is shown below, showing details of some of the structures on a coupling facility:

STRUCTURE TYPE	NAME	STATUS CHG	ALLOC SIZE	% OF CF STOR	# REQ	% OF ALL REQ	% OF CF UTIL
LOCK	ISGLOCK	ACTIVE	9M	0.0	330441	5.1	0.4
	DSNDBOP_LOCK1	ACTIVE	4G	10.7	2967K	36.6	8.3
LIST	DFHXQLS_PLXPSQ0	ACTIVE	350M	0.9	160255	2.0	0.7
	MQG0PCQ_ADMIN	ACTIVE	20M	0.1	13210	0.2	2.3
CACHE	DSNDBOP_GBP18	ACTIVE	2G	6.6	1795K	27.5	16.2
	SYSIGGCAS_ECS	ACTIVE	3M	0.1	68900	0.8	0.2

and on the next page of the report we see:

COUPLING FACILITY	8561	MODEL 702	CFLEVEL 23	DYNDISP OFF
AVERAGE CF UTILIZATION (% BUSY)			17.7	
LOGICAL PROCESSORS:		DEFINED	2	EFFECTIVE 2.0

Note the field, “% OF CF UTIL”. From the RMF Report Analysis manual, “This is the percentage of CF processor time used by the structure. The structure execution time is related to the total CF-wide processor busy time. The sum of the values in this column is less than 100%, because not all CF processor time is attributable to structures.” This number must be multiplied by the % busy the CF is to see how much CPU on the Coupling Facility is used to manage this structure.

Using the example above where the **% BUSY** is 17.7%, one can calculate the contribution of each structure as:

Structure	% of CF Util (total 100%)	CF % Busy Contribution (total 17.7% in example)
ISGLOCK	0.4	.08
DSNDBOP_LOCK1	8.3	1.5
MQG0PCQ_ADMIN	2.3	.41
DSNDBOP_GBP18	16.2	2.9

System-Managed CF Structure Duplexing

Coupling Facility Link Subchannel Busy

Coupling links connect the server running z/OS and the coupling facility, as well as the CF to CF for system-managed duplexing. Each link has either seven or 32 buffers available to send and receive signals. These buffers are associated with subchannels. When a message is being sent from a z/OS to a CF, the link is busy for the time needed to send the signals and responses over the connection, but the subchannel is allocated for the entire duration from the beginning of the first signal to the final response back. In an asynchronous message, the subchannel is busy for the entire duration of the request as well. The link subchannel utilization can be calculated as the product of the request rate per second and the service time per request.

Information needed to calculate the current contribution of each structure to CF link subchannel utilization can be found in the Structure Activity Report of RMF's CF Activity Report. As an example, consider the report shown earlier for Cache_Str. Using data extracted from that report, we can make the following calculations:

System-Managed CF Structure Duplexing

Total link subchannel busy time for Cache_Str
= (synch ops/sec * serv time) + (async ops/sec * serv time)
= ((14,277 * .934) * 19.7) + ((14,277 * .066) * 139.1)
= 262,694 + 131,071 Microseconds = .263 + .131 microseconds
= .394 seconds

If there were 2 links to the CF containing this structure, and each link had 8 subchannels, then the average simplex link subchannel utilization would be:

Average link subchannel utilization
= 100 * subchannel busy time per second / number of subchannels in use
= 100 * .394 / 16
= 2.5%

It is recommended to keep the average subchannel utilization below 30% busy. Going beyond this value may cause elongation of response times and therefore increased host CPU usage for the system.

System-Managed CF Structure Duplexing

Appendix B: Host effects of Technology in a Parallel Sysplex

The table below shows the effects that various servers, coupling facilities, and coupling links have on the host z/OS capacity cost of coupling. The host technologies are listed across the top, and the coupling technologies are listed down the side. The value at each intersection gives the approximate percentage of host capacity that is consumed for coupling functions. For example, a value of 10% would indicate that approximately 10% of the host capacity (or host MIPS) is consumed by the subsystem, operating system and hardware functions associated with coupling facility activity. The values in the chart are based on customer experiences where their major applications are involved in data sharing. Your actual results may vary depending on the amount of data sharing your applications are doing. All structures are assumed to be in Simplex mode.

This chart is based on 9 CF operations per MIPS. One can calculate your activity by simply summing the total req/sec of the two CFs and dividing by the used MIPS of the attached systems (that is, MIPS rating times CPU busy). Then, the values in the table would be linearly scaled. For example, if the customer was running 4.5 CF ops per MIPS then all the values in the table would be cut in half.

CF	Host	z13s	z13	z14 ZR1	z14	z15	
z13s CL5		20	20	21	21	23	
z13s 1x IFB		20	20		21		
z13s 12x IFB		17	17		17		
z13s 12x IFB3		12	12		12		
z13s CS5		11	11	11	11	12	
z13 CL5		20	20	21	21	23	
z13 1x IFB		20	20		21		
z13 12x IFB		16	17		17		
z13 12x IFB3		12	12		12		
z13 CS5		11	11	11	11	12	
z14 ZR1 CL5		20	20	21	21	23	
z14 ZR1 CS5		11	11	11	11	12	
z14 CL5		19	20	21	21	23	
z14 1x IFB		19	20		21		
z14 12x IFB		16	16		17		
z14 12x IFB3		11	11		12		
z14 CS5			10	11	11	12	
z15 T02 CL5			19	20	21	23	
z15 T02 CS5			10	11	11	12	
z15 T01 CL5			19	20	21	22	
z15 T01 CS5			10	10	11	11	

System-Managed CF Structure Duplexing

Note 1: Assumes 9 CF requests / MI for production workload

Note 2: The table does not take into consideration any extended distance effects or system managed duplexing.

Note 3: For 9 CF requests/MI, host effect values in the table may be considered capped at approximately 18% due to z/OS 1.2 feature Synchronous to Asynchronous CF Message Conversion. Configurations where entries are approaching 18% will see more messages converted to asynchronous. As synchronous service times degrade relative to the speed of the host processor, the overhead % goes up. This could happen, for example, where the CF technology stays constant but you upgrade the host technology. This can be seen in the table by the % value increasing. z/OS converts synchronous messages to asynchronous messages when the synchronous service time relative to the speed of the host processor exceeds a breakeven threshold. At this point it is cheaper to go asynchronous. When all CF operations are asynchronous, the overhead will be about 18%. By the time you have reached $\geq 18\%$ in the table, that corresponds to the time z/OS must have been converting almost every operation asynchronous.

More information on using this table can be found in “CF Configuration Options” ZSW01971USEN.

System-Managed CF Structure Duplexing

Appendix C: Structure Recovery Support

The following table summarizes various structures and their recovery support.

Subsystem	Structure	Structure Type	User Managed Rebuild	System Rebuild / SM Duplex
Allocation	Shared tape	List	Yes	No
BatchPipes	Multi-system pipes	List	Yes	Yes
Catalog	Enhanced Catalog Sharing	List	Yes	No
CICS	DFHLOG	List	Fail-Isol	Yes
CICS	DFHSHUNT	List	Fail-Isol	Yes
CICS	Fwd Recovery	List	Fail-Isol	Yes
CICS	Temp Storage	List	No	Yes
CICS	Shared Data Tables	List	No	Yes
CICS	Named Counter	List	No	Yes
Db2	SCA	List	Fail-Isol	Yes
Db2	GBP	Cache	Yes	No. Supports User Managed Duplexing
Db2	IRLM Lock	Lock	Yes Fail-Isol	Yes
GRS	Star	Lock	Yes	No
IMS	IRLM Lock	Lock	Yes Fail-Isol	Yes
IMS	VSO	Cache	No	Yes
IMS	OSAM	Cache	Yes	No
IMS	VSAM	Cache	Yes	No
IMS	CQS	List	Yes	Yes
IMS	CQS Logger	List	Fail-Isol	Yes
IMS	CQS Logger (EMH)	List	Fail-Isol	Yes
IBM MQ	Administration	List	No	Yes
IBM MQ	Application	List	No	Yes
JES2	Checkpoint	List	No	Yes
z/OS Operlog	Logger	List	Fail-Isol	Yes
z/OS Logrec	Logger	List	Fail-Isol	Yes
RACF	Shared DB	Cache	Yes	No

System-Managed CF Structure Duplexing

RRS	Logger	List	Fail-Isol	Yes
DFSMS	HSM Common Recall Queue	List	No	Yes
DFSMS	RLS Cache	Cache	Yes	No
DFSMS	RLS Lock IGWLOCK00	Lock	Fail-Isol	Yes
VTAM	Generic Resources	List	Fail-Isol	Yes
VTAM	MNPS	List	Fail-Isol	Yes
WLM	IRD	Cache	No	Yes
WLM	Enclaves	List	No	Yes
XCF	Signaling	List	Yes	No

Appendix D: Asynchronous Duplexing for Lock Structures

Running some or all of the SCA, lock, and group buffer pool structures in duplex mode is one way to achieve high availability for these structures across many types of failures, including lost connections and damaged structures.

With asynchronous CF duplexing, multi-site data sharing groups that implement duplexing all Db2 CF structures can have continuous availability without significant performance impact after site failures.

CF Asynchronous Duplexing for Lock Structures:

1. z/OS sends command to primary CF only with Operation Sequence Number (OSN)
2. Primary CF processes command and returns result with last OSN completed
3. Primary CF forwards description of required updates to secondary
4. Secondary CF updates secondary structure instance asynchronously and responds back to primary CF that it completed successfully with last OSN completed
5. At Db2 commit, z/OS verifies with secondary CF it completed the last OSN. May need to delay commit until secondary CF is caught up.

Asynchronous CF duplexing for lock structures requires:

- z/OS V2.2 with PTFs
- Db2 V12 with PTFs
- CFCC level 22 is delivered on the z14 servers with driver level D32
- CF-to-CF connectivity through coupling links

You can enable asynchronous CF duplexing by setting the DUPLEX option in the coupling facility resource management (CFRM) policy to one of the following values:

- DUPLEX(ENABLED,ASYNCONLY)
- DUPLEX(ENABLED,ASYNCR)
- DUPLEX(ALLOWED,ASYNCONLY)
- DUPLEX(ALLOWED,ASYNCR)

CPU impact of Asynchronous duplexing on the z/OS system was measured as 5 – 15%. For the sake of this document, we are assuming approximately no impact

System-Managed CF Structure Duplexing

References

- Parallel Sysplex home page
<https://www.ibm.com/it-infrastructure/z/technologies/parallel-sysplex>
- Coupling Facility Configuration Options White Paper
<https://www.ibm.com/downloads/cas/JZB2E38Q>
- *MVS Setting Up a Sysplex (SA23-1399)*
<https://www.ibm.com/servers/resourcelink/svc00100.nsf/pages/zosInternetLibrary>
Go to **z/OS Internet Library** and click on “z/OS MVS” field. Look for “Setting up a Sysplex.”
- GDPS home page
<https://www.ibm.com/it-infrastructure/z/technologies/gdps>

System-Managed CF Structure Duplexing



©Copyright IBM Corporation 2021
IBM Corporation
New Orchard Road
Armonk, NY 10504
U.S.A.
06/21

IBM, ibm.com, the IBM logo, eServer, e-business logo, BatchPipes, CICS, Db2, GDPS, GDPS, IMS, MQSeries, MVS, MVS/ESA, OS/390, Parallel Sysplex, PR/SM, RACF, RMF, S/390, System z9, System z10, VTAM, z9, z10 EC, z/OS, and zSeries.. are trademarks or registered trademarks of the International Business Machines Corporation.

A current list of IBM trademarks is available on the Web at <https://www.ibm.com/legal/us/en/copytrade.shtml>, and select third party trademarks that might be referenced in this document is available at https://www.ibm.com/legal/us/en/copytrade.shtml#section_4.

Adobe, the Adobe logo, PostScript, and the PostScript logo are either registered trademarks or trademarks of Adobe Systems Incorporated in the United States, and/or other countries.

Cell Broadband Engine is a trademark of Sony Computer Entertainment, Inc. in the United States, other countries, or both and is used under license therefrom.

InfiniBand and InfiniBand Trade Association are registered trademarks of the InfiniBand Trade Association.

Intel, Intel logo, Intel Inside, Intel Inside logo, Intel Centrino, Intel Centrino logo, Celeron, Intel Xeon, Intel SpeedStep, Itanium, and Pentium are trademarks or registered trademarks of Intel Corporation or its subsidiaries in the United States and other countries.

Java and all Java-based trademarks and logos are trademarks or registered trademarks of Oracle and/or its affiliates.

The registered trademark Linux® is used pursuant to a sublicense from the Linux Foundation, the exclusive licensee of Linus Torvalds, owner of the mark on a worldwide basis.

Microsoft, Windows, Windows NT, and the Windows logo are trademarks of Microsoft Corporation in the United States, other countries, or both.

OpenStack is a trademark of OpenStack LLC. The OpenStack trademark policy is available on the [OpenStack website](#).

Red Hat®, JBoss®, OpenShift®, Fedora®, Hibernate®, Ansible®, CloudForms®, RHCA®, RHCE®, RHCSA®, Ceph®, and Gluster® are trademarks or registered trademarks of Red Hat, Inc. or its subsidiaries in the United States and other countries.

RStudio®, the RStudio logo and Shiny® are registered trademarks of RStudio, Inc.

TEALEAF is a registered trademark of Tealeaf, an IBM Company.

UNIX is a registered trademark of The Open Group in the United States and other countries.

Worklight is a trademark or registered trademark of Worklight, an IBM Company.

Zowe™, the Zowe™ logo and the Open Mainframe Project™ are trademarks of The Linux Foundation.

All statements regarding IBM's future direction and intent are subject to change or withdrawal without notice, and represent goals and objectives only.

The information contained in this documentation is provided for informational purposes only. While efforts were made to verify the completeness and accuracy of the information contained in this documentation, it is provided "as is" without warranty of any kind, express or implied. In addition, this information is based on IBM's current product plans and strategy, which are subject to change by IBM without notice. IBM shall not be responsible for any damages arising out of the use of, or otherwise related to, this documentation or any other documentation. Nothing contained in this documentation is intended to, nor shall have the effect of, creating any warranties or representations from IBM (or its suppliers or licensors), or altering the terms and conditions of the applicable license agreement governing the use of IBM software.

References in these materials to IBM products, programs, or services do not imply that they will be available in all countries in which IBM operates. Product release dates and/or capabilities referenced in these materials may change at any time at IBM's sole discretion based on market opportunities or other factors and are not intended to be a commitment to future product or feature availability in any way.

ZSW01975-USEN