Parallel Sysplex Performance: XCF Performance Considerations (Version 2)

Editor's Note:
This Washington Systems Center Flash is a total replacement for WSC Flash W9723A, MVS/ESA Parallel Sysplex Performance XCF Performance Considerations. WSC Flash W9723A will be removed from the database, and this flash should be used in all cases.

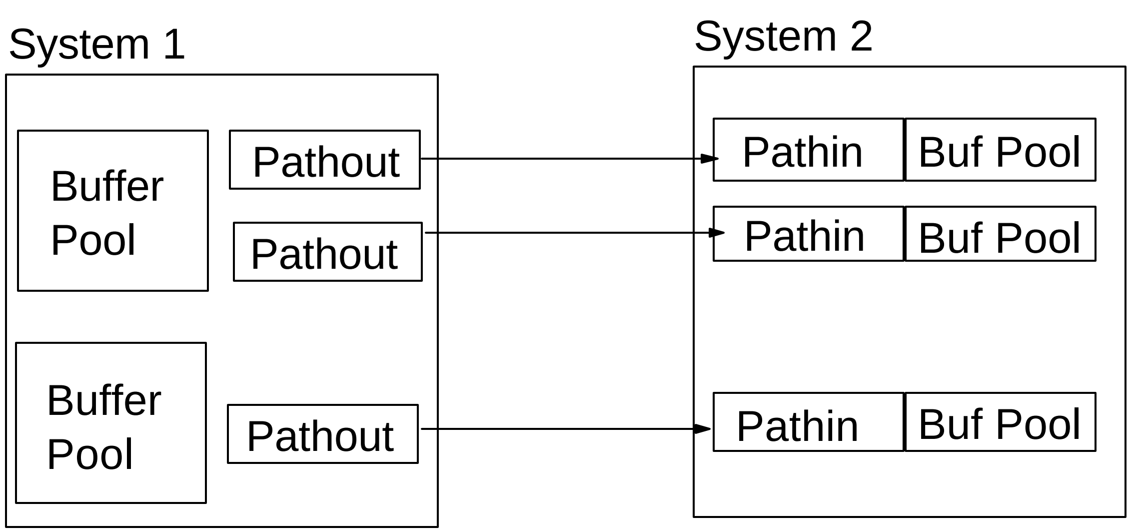
Some installations implementing parallel sysplex have seen performance issues due to XCF signaling. These performance issues are generally solved by tuning changes to the XCF transport class definitions, buffer definitions, and signaling paths. This flash is intended to review recommended XCF configurations and known performance tuning options.

Tuning XCF

XCF signaling is used to communicate between various members of a sysplex. The user of XCF signaling, usually an MVS component or a subsystem, issue messages to members within the user's group. The content and/or use of these messages are unique to the users of the group.

As XCF messages are generated, they are assigned to a transport class based on group name and/or message size. The messages are copied into a signal buffer from the XCF buffer pool.
The messages are sent over outbound paths, (PATHOUT), defined for the appropriate transport class. Messages from other systems are received by inbound paths, (PATHIN). Inbound paths are not directly assigned transport classes, although a correlation can be made about which transport class messages are received via the inbound paths based on the outbound path to which the inbound side is connected.

The following is a diagram which highlights the XCF message traffic.



Parallel Sysplex Performance: XCF Performance Considerations (Version 2)

The key to ensuring good performance for the XCF signaling service is to provide sufficient signaling resources, namely message buffers, message buffer space, and signaling paths, and to control access to those resources with the transport class definitions.

Parallel Sysplex Performance: XCF Performance Considerations (Version 2)

Transport Classes

Transport classes are used to group messages. Using the CLASSDEF parameter in the COUPLExx parmlib member you can assign messages to a transport class based on the group name, the message size, or both.

Each transport class has its own resources which consists of a buffer pool and one or more outbound signaling paths. It is recommended you keep the number of transport classes small. In most cases, it is more efficient to pool the resources and define the transport class based on message size. Some initial product documentation recommended separate transport classes for GRS or RMF. These recommendations are no longer advised. If you do have separate transport classes for specific groups based on early product recommendations you should consider changing these recommendations.

Message Buffers

XCF message buffers are managed by correctly selecting the size of the message most frequently sent from specific buffer pools and by specifying an adequate upper limit for the size of the buffer pool.

Message Buffer Size

First let's look at the individual message buffer size definitions. Message buffer size is determined by the CLASSLEN parameter on the CLASSDEF statement in the COUPLExx parmlib member. The CLASSLEN value determines the size of the most frequent message expected in this transport class. If a message could be assigned to more than one transport class, XCF selects the one with the smallest buffer which will hold the message. If the signal is larger than the CLASSLEN for any of the assigned transport classes, XCF has to choose a transport class to expand. Since APAR OW16903, XCF assigns the message to the transport class with the largest buffer size and expands the buffer size of this transport class. Prior to this APAR, the transport class named DEFAULT was chosen to be expanded, even if it had a very small class length.

Expanding the message buffer entails some overhead. The PATHOUT on the sending side and the PATHIN on the receiving side must be cleared out and expanded to handle the larger buffer size. A new, larger buffer must be obtained on the PATHIN side. If no additional messages of this size are received in a short time period, XCF then contracts the PATHIN, PATHOUT, and buffer sizes. In both of these cases extra XCF internal signals are generated to communicate these changes.

The best way to eliminate the overhead of expanding and contracting the message buffers is to define transport classes based solely on the size of the message buffers. One class with the default length of 956 should handle most of the traffic. A second class can be defined to handle larger messages.

An example of this specification in the COUPLExx parmlib member is:

```
CLASSDEF CLASS(DEFSMALL) CLASSLEN(956) GROUP(UNDESIG)
  CLASSDEF CLASS(DEFAULT) CLASSLEN(16316) GROUP(UNDESIG)
```

The parameter GROUP(UNDESIG) specifies the messages should be assigned to the transport class based solely on message size. This definition makes all the resources available to all users and provides everyone with peak capacity.

There may be times when you want a separate transport class for a specific group. For instance, if you have a particular XCF user which is consuming a disproportionate amount of XCF resources, you may want to isolate this user to a separate transport class to investigate the user's behavior and protect the other XCF users. Hopefully, after you have diagnosed the problem, you can reassign this user to a transport class based on the length of the messages.

You can use an RMF XCF report to determine how well the messages fit:

```
                         XCF USAGE BY SYSTEM
-----------------------------------------------------------------------
                                             REMOTE SYSTEMS
-----------------------------------------------------------------------
                          OUTBOUND FROM JB0
-----------------------------------------------------------------------
                                        ----- BUFFER -----
TO        TRANSPORT  BUFFER        REQ   %    %    %    %      ....
SYSTEM    CLASS      LENGTH        OUT  SML  FIT  BIG  OVR
JA0       DEFAULT    16,316        189   98    1    1  100
          DEFSMALL      956     55,794    0  100    0    0
JB0       DEFAULT    16,316        176  100    0    0    0
          DEFSMALL      956     44,156    0  100    0    0
JC0       DEFAULT    16,316        176  100    0    0    0
          DEFSMALL      956     34,477    0  100    0    0    ....
                            ----------
TOTAL                          134,968
```

%SML is the % of messages smaller than the buffer length
%FIT is the % of messages which fit the buffer length
%BIG is the % of messages larger than the buffer length

In this example, the majority of the messages fit in the DEFSMALL class. A few exceeded the size of the DEFAULT class, but not enough to justify the definition of a new transport class.

Note: XCF has internal buffers of fixed size: 1K, 4K, 8K, ..64K. XCF uses 68 bytes for internal control blocks. So if you specify a length which doesn't fit one of these sizes, XCF will round up to the next largest size. For example, if you specify 1024, it will not fit into the 1K block (1024-68=956), and XCF will round up to the next largest block. If you issue a command, D XCF,CLASSDEF, it will list the CLASSLEN specified in the PARMLIB member, in this example, 1024. The RMF XCF report will show the actual buffer length, in this case, 4028.

Message Buffer Pools

Having determined the optimal size for the individual message buffer, the next thing to do is select an upper limit for the amount of virtual storage to be allocated to the message buffer pool. The message buffer space is virtual storage used by XCF to store the message buffers which are being processed, sent or received.

Most of the virtual storage used for this purpose is backed by fixed central and expanded storage. The storage to hold LOCAL buffers (for communication within the processor) is DREF storage which is backed by central storage. LOCAL buffers are used for messages within groups which are on the same MVS image. Currently APPC and JES3 are the only known IBM exploiters of local messages but OEM applications can choose to take advantage of LOCAL message processing.

XCF only uses the amount of storage it needs; but to insure there are no surprises, the installation can use the MAXMSG parameter to place an upper limit on the amount of storage which can be used for this purpose.

Storage is associated with the transport class, the outgoing paths, and the incoming paths, so MAXMSG can be specified on the CLASSDEF, PATHIN and PATHOUT definitions, or more generally on the COUPLE definition. MAXMSG is specified in 1K units. The default values are determined in the following hierarchy:

```
          OUTBOUND                                    INBOUND
-----------------------------------|--------------------------------
   PATHOUT - not specified, use    |  PATHIN - not specified, use
     CLASSDEF - not specified, use |    COUPLE
       COUPLE                      |
```

The default for MAXMSG is 500 in OS/390 R1 and prior releases. In OS/390 R2 and beyond, the MAXMSG default is 750. By not specifying the default parameter, you will automatically get the most current default size as you migrate to newer releases. If you do want a larger value than the default, specify it at the lowest level of the hierarchy as appropriate.

The total amount of storage used by XCF on a single system is the sum of:

- Sum of MAXMSG for all classes * systems in sysplex
- Sum of MAXMSG for all PATHOUTs
- Sum of MAXMSG for all PATHINs

In this example:

```
                            XCF PATH STATISTICS
------------------------------------------------------------------------
               OUTBOUND FROM JB0              INBOUND TO JB0
 -------------------------------------    --------------------------
        T FROM/TO                                 T FROM/TO
TO      Y DEVICE, OR      TRANSPORT     ...  FROM    Y DEVICE, OR
SYSTEM  P STRUCTURE       CLASS              SYSTEN  P STRUCTURE
JA0     S IXCPLEX_PATH1   DEFAULT            JA0     S IXCPLEX_PATH1
        C C600 TO C614    DEFSMALL                   C C600 TO C614
        C C601 TO C615    DEFSMALL                   C C601 TO C615
        C C602 TO C616    DEFSMALL                   C C602 TO C616
JB0     S IXCPLEX_PATH1   DEFAULT            JB0     S IXCPLEX_PATH1
        C C600 TO C614    DEFSMALL                   C C600 TO C614
        C C601 TO C615    DEFSMALL                   C C601 TO C615
        C C602 TO C616    DEFSMALL                   C C602 TO C616
```

If a MAXMSG of 1000 was specified on the CLASSDEF parameter and MAXMSG was not specified on the other parameters, the maximum storage which could be used by XCF is 22M:

- 2 classes * 3 systems * 1M = 6M
- 8 PATHOUTs * 1M = 8M
- 8 PATHINs * 1M = 8M

Note: This implies if you add additional transport classes, signaling paths or systems, you will be increasing the upper limit on the size of the message buffer pool.

Outbound Messages

For the outbound messages to a particular system if the sum of the storage for the CLASSDEF and the PATHOUTs is insufficient, the signal will be rejected. This is reported on the RMF XCF report as REQ REJECT for OUTBOUND requests. In general, any non-zero value in this field suggests some further investigation. The problem is generally resolved by increasing MAXMSG on the CLASSDEF or PATHOUT definition.

```
                           XCF USAGE BY SYSTEM
------------------------------------------------------------------------
                                            REMOTE SYSTEMS
------------------------------------------------------------------------
                         OUTBOUND FROM SYSC
------------------------------------------------------------------------
                                                       ALL
TO        TRANSPORT  BUFFER         REQ              PATHS      REQ
SYSTEM    CLASS      LENGTH         OUT            UNAVAIL   REJECT
K004      DEFAULT       956     126,255   ...            0    1,391
          DEF16K     16,316          28                  0        0
SYSA      DEFAULT       956      97,834                  0        0
          DEF16K     16,316       3,467                  0        0
                             ----------
TOTAL                           227,584
```

Inbound Messages

For the inbound messages from a particular system, if the storage for the PATHINs is insufficient, the signal will be delayed. This is reported on the RMF XCF report as REQ REJECT for INBOUND requests. If the delay causes signals to back up on the outbound side, eventually an outbound signal could get rejected for lack of buffer space. In this case, you may wish to increase the MAXMSG on the PATHIN definition.

```
                        XCF USAGE BY SYSTEM
----------------------------------------------------------------------
         REMOTE SYSTEMS                                     LOCAL
------------------------------------------------      -----------------
                        INBOUND TO SYSC                   SYSC
                     --------------------------       -----------------

         .....       FROM            REQ      REQ     TRANSPORT     REQ
                     SYSTEM           IN   REJECT     CLASS      REJECT
                     K004        117,613    1,373     DEFAULT         0
                                                      DEF16K          0
                     SYSA        101,490        0

                              ----------
                     TOTAL       219,103
```

Another indicator the storage for PATHINs is insufficient is the BUFFERS UNAVAIL count on the XCF PATH STATISTICS report. If this is high, check the AVAIL and BUSY counts: AVAIL counts should be high relative to BUSY counts. High BUSY counts can be caused by an insufficient number of paths or a lack of inbound space. First look at the inbound side of see if there are any REQ REJECTs. If so, increase the PATHIN MAXMSG. Otherwise, it is important to review the capacity of the signaling paths. The methodology for determining this is described later in this flash.

Note: The RMF Communications Device report cannot be used to determine if the CTC devices are too busy. XCF CTCs will typically always report high device utilization because of the suspend / resume protocol used by XCF.

Local Messages

Local messages are signals within the same image, so no signaling paths are required. In this case, the message buffer storage used is the CLASSDEF storage plus any storage specified on the LOCALMSG definition. If MAXMSG is not coded on the LOCALMSG statement the additional message buffer storage contributed is none, or 0 buffers.

Signaling Paths

XCF signals from each transport class are sent out on the PATHOUT path and received into the system on the PATHIN paths. Tuning is achieved by altering the number or type of paths, or both. To review the XCF path configuration use the RMF XCF Path Statistics report. Two different issues commonly reported to IBM regarding signaling paths are reviewed in this flash: no paths defined, and an insufficient number of paths defined.

Number of Paths

1. No paths

In the worst case, there may be NO operational paths for a transport class. This is not fatal. XCF routes the requests to another transport class but there is additional overhead associated with this operation. To determine if this condition exists, look at the RMF XCF Usage by System report. ALL PATHS UNAVAIL should be low or 0. In many cases, this is caused by an error in the path definition; in other cases, there may be a problem with the physical path.

```
                          XCF USAGE BY SYSTEM
-----------------------------------------------------------------------
                                            REMOTE SYSTEMS
-----------------------------------------------------------------------
                              OUTBOUND FROM SD0
-----------------------------------------------------------------------
                                                       ALL
TO        TRANSPORT  BUFFER         REQ                PATHS       REQ
SYSTEM    CLASS      LENGTH         OUT       ....   UNAVAIL    REJECT
JA0       DEFAULT    16,316         189                    0         0
          DEFSMALL      956      55,794               55,794         0
JB0       DEFAULT    16,316         176                    0         0
          DEFSMALL      956      44,156                    0         0
JC0       DEFAULT    16,316         176                    0         0
          DEFSMALL      956      34,477       ....         0         0
                             ----------
TOTAL                           134,968
```

In this example, the CTC links to system JA0 had been disconnected.

In the **next** example from the same system, notice for system JA0 there were no paths for the transport class DEFSMALL, so all the requests were re-driven through the DEFAULT class. This caused some queuing (see AVG Q LNGTH of 0.16).

```
                          XCF PATH STATISTICS
------------------------------------------------------------------------
                               OUTBOUND FROM SD0
------------------------------------------------------------------------
        T FROM/TO
TO      Y DEVICE, OR      TRANSPORT       REQ   AVG Q
SYSTEM  P STRUCTURE       CLASS           OUT   LNGTH   AVAIL   BUSY RETRY
JA0     S IXCPLEX_PATH1   DEFAULT      56,011    0.16  55,894    117     0
JB0     S IXCPLEX_PATH1   DEFAULT         176    0.00     176      0     0
          C C600 TO C614  DEFSMALL     16,314    0.01  16,297     17     0
          C C601 TO C615  DEFSMALL     15,053    0.01  15,037     16     0
          C C602 TO C616  DEFSMALL     15,136    0.01  15,136     20     0
```

JC0	S	IXCPLEX_PATH1	DEFAULT	176	0.00	176	0	0
	C	C600 TO C614	DEFSMALL	11,621	0.01	11,515	106	0
	C	C601 TO C615	DEFSMALL	13,086	0.01	12,962	124	0
	C	C602 TO C616	DEFSMALL	11,626	0.00	11,526	100	0

Is it necessary to correct the 'ALL PATHS UNAVAIL' condition? In most cases it is. In the example above, DEFSMALL was defined to hold small messages (956). Because there is no path, they are being re-driven through the **DEFAULT** class. The DEFAULT class is sending data in large buffers (16,316 bytes). This is certainly not an efficient use of message buffer storage to transfer a 956 byte message in a 16,316 byte buffer. Re-driving large messages through a transport class defined with small messages causes more problems. It causes the buffers in this class to expand and contract with all the extra signaling explained previously. Defining separate classes is done for a purpose. If you don't provide paths for these classes, it negates this purpose.

2. Insufficient number of paths

Signaling paths can be CTC links or Coupling Facility structures. In the example above, the TYP field indicates the connection is a CF structure (S) or a CTC link (C). Since these two types of paths operate in unique ways, different methods are used to evaluate their performance.

a. CF structures:
   For CF structures, an insufficient number of PATHOUT links could result in an increase in the AVG Q LNGTH, and BUSY counts high relative to AVAIL counts. Additional paths are obtained by defining more XCF signaling structures in the CFRM policy and making them available for use as PATHOUTs (and/or PATHINs).

   Note: RETRY counts should be low relative to REQ OUT for a transport class. A non zero count indicates a message has failed and was resent. This is usually indicative of a hardware problem.

b. CTCs
   CTCs can be configured in a number of ways. The installation can define CTC’s as unidirectional (one PATHOUT or one PATHIN per physical CTC) or bi-directional (one or more PATHOUTs and PATHINs on a physical CTC). Due to the nature of XCF channel programs, a unidirectional path definition can achieve the most efficient use of a CTC thus providing the best XCF response time and message throughput capacity. However, a unidirectional definition will also require using **at least four physical CTCs** to configure for availability. As will be noted in the capacity planning section below, two paths are sufficient for most systems, thus only those customers with very high XCF activity, (requiring >=4 paths), should consider using the unidirectional definition.

   What indicators should be used to determine if there are enough CTCs for a particular transport class? First of all, the AVG Q LEN on the RMF XCF report is **not a good indicator.** In the case of CTCs, queued requests are added to the CCW chain which can increase efficiency. A better indicator to use instead is the Display XCF command. This command was updated by XCF APAR OW38138 to provide the path response time (as seen by XCF).

```
D XCF,PI,DEVICE=ALL,STATUS=WORKING
 IXC356I  12.02.12  DISPLAY XCF 901
 LOCAL DEVICE    REMOTE   PATHIN      REMOTE                   LAST     MXFER
 PATHIN          SYSTEM   STATUS      PATHOUT RETRY  MAXMSG  RECORD   TIME
 C200            JA0      WORKING     C200       10     500    3496     339
 C220            JA0      WORKING     C220       10     500    3640     419
```

The MXFER TIME is the mean transfer time in microseconds for up to the last 64 signals received within the last minute. If the MXFER TIME is acceptable, less than 2 milliseconds, (or 2000 microseconds), there is probably enough CTC capacity. To insure capacity for heavier or peak workloads, also check the channel utilization for the CTCs, as reported on an RMF Channel Activity report. In laboratory testing, acceptable XCF message response times were observed even at channel utilization of 70% (or 90% when there were multiple CTCs per transport class). Beyond this threshold, response time degenerated rapidly.

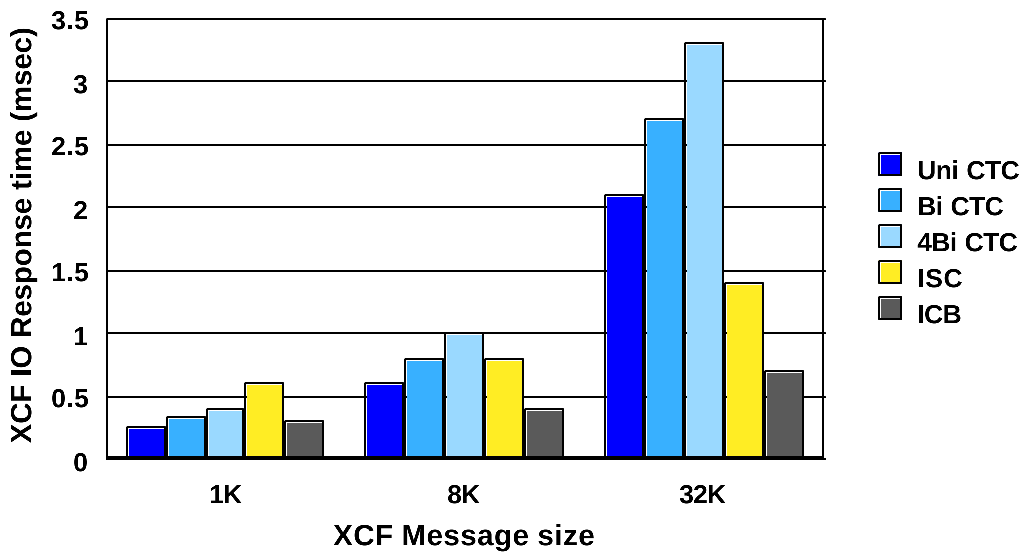
RMF, with APAR OW41317 installed, will store the MXFER TIME as observed in the last minute before the end of the RMF interval in the RMF SMF 74 subtype 2 record.

TYPE OF SIGNALING PATH

A CTC provides a direct path between two systems, while sending a message through a CF is a two step, push-pull process. Thus, depending on message size and the type of CF link, CTCs are sometimes faster than using CF structures.

These are examples of XCF response time, (MXFER TIME), from controlled experiments in a test environment. The unidirectional CTCs have a single PATHIN or PATHOUT per physical CTC. The bi-directional CTCs have a pair of PATHIN AND PATHOUT defined for physical CTC. The 4 bi-directional CTCs have 4 pairs of PATHIN and PATHOUT per physical CTC.

Parallel Sysplex Performance: XCF Performance Considerations (Version 2)



A comparison of these examples shows unidirectional CTCs are the fastest option for 1K messages, although ICBs are close behind. The bi-directional CTCs are **somewhat** slower, but perfectly adequate for most installations. For larger messages, ICBs are the faster option. This results from the higher bandwidth associated with ICB, (and ISC), coupling links compared to CTCs, (ESCON).

XCF internally times the various signals and gives preference to the faster paths. In the following example, compare the number of requests for DEFSMALL which were sent through the structure to the number which were sent through the CTCs. It should be noted XCF does not attempt to balance the workload across paths; once it finds a fast path, it continues to use it. APAR OW38138 describes changes which improves the path distribution.

XCF PATH STATISTICS

OUTBOUND FROM JA0

TO SYSTEM	T Y P	FROM/TO DEVICE, OR STRUCTURE	TRANSPORT CLASS	REQ OUT	AVG Q LNGTH	AVAIL	BUSY	RETRY
JC0	S	IXCPLEX_PATH1	DEFAULT	1,744	0.00	1,176	0	0
	S	**IXCPLEX_PATH2**	**DEFSMALL**	**8,582**	**0.01**	**8,362**	**220**	**0**
	C	**C600 TO C614**	**DEFSMALL**	**20,223**	**0.01**	**20,160**	**63**	**0**
	C	C601 TO C615	DEFSMALL	23,248	0.01	23,229	19	0
	C	C602 TO C616	DEFSMALL	23,582	0.01	23,568	14	0

In many environments, the difference in response time between CTCs and CF structures is indiscernible and using CF structures certainly simplifies management of the configuration.

Capacity Planning

For availability, a minimum of two physical paths must be provided between any two systems. This can be accomplished with two physical CTCs, structures in each of two different CFs, or a combination of CTCs and CF structures.

Most environments will find the rate of XCF traffic can be handled by the two paths which were configured for availability. Only for environments with very high rates of XCF traffic would additional paths be required.

The XCF message rate capacity of a path is affected by many factors:
1. The size of the message
2. How the paths are defined
3. If the path is also used for other (non-XCF) functions?

Based on these factors, message rates (XCF IN+OUT), have been observed from 1000/sec to 5000/sec on a CTC, up to 9000/sec via an ICB and up to 4000/sec per HiPerLink. The adage "Your mileage may vary" is certainly true here.

When using CF structures for XCF messaging, there is also a cost in CF CPU utilization to plan for. As an example, running 1000 XCF messages/sec through an R06 CF would utilize approximately 10% of one CF processor. Additionally, if you use CF structures as XCF paths, make sure the structure size is adequate. You can use the CF sizer available on the Parallel Sysplex website, www.s390.ibm.com/products/pso to obtain an initial estimate for the structure size. If the structure is too small, you will see an increase in the number of REQ REJECT and AVG Q LNGTH, and these events will definitely affect response time.

CTC Configuration Planning

When configuring CTCs for large volumes of XCF traffic some additional configuration planning needs to be done. CTC I/O will use SAP capacity, and large XCF environments can generate I/O rates much higher than traditional DASD and Tape workloads.

The SAP acts as an offload engine for the CPUs. Different processor models have different numbers of SAPs, and a spare 9672 PU can be configured as an additional SAP processor. SAP functions include:

- Execution of ESA/390 I/O operations. The SAP (or SAPs) are part of the I/O subsystem of the CPC and act as Integrated Offload Processor (IOP) engines for the other processors.
- Machine check handling and reset control
- Support functions for Service Call Logical Processor (SCLP)

In high volume XCF environments planning should be done to ensure the CTC configuration is defined so the CTC I/O load is spread across all available SAPs. Information on channel to SAP relationships can be found in the *IOCP User's Guide and ESCON CTC Reference*, GC38-0401-11. Additional

Information on 9672 SAP performance and tuning can be found in WSC Flash 9646E at www.ibm.com/support/techdocs.

Case Study:

This is a case study which illustrates some of the items discussed.

An application was invoked which was changed to use CF signaling. When the workload was increased XCF delays increased. This was evident from messages like ERB463I which indicated the RMF Sysplex Data Server was not able to communicate with another system because the XCF signaling function was busy.

Looking at RMF Monitor III it showed:

RMF 1.3.0 XCF Delays

Samples: 120 System: J90 Date: 02/07/97 Time: 13.03.00

		Service	DLY	------------ Main Delay Path(s)					
Jobname	C	Class	%	%	Path	%	Path	%	Path
WLM	S	SYSTEM	87	87	-CF-				
MASTER	S	SYSTEM	10	10	-CF-				
RMFGAT	S	SYSSTC	3	3	-CF-				
JESXCF	S	SYSTEM	1	1	C601				

Comparing the RMF XCF reports to some earlier reports, it was noticed the amount of XCF traffic had quadrupled and the increase was in the class with the larger CLASSLEN (DEFAULT on this system).

In order to protect other XCF users and to investigate what was happening, a decision was made to separate these messages into their own transport class. A new transport class, NEWXCF, was defined using the GROUP keyword to specifically assign messages from the new application to this class. Since it was known the messages were bigger than the transport class with the smaller CLASSLEN (DEFSMALL), using guess work it was decided the messages might fit into a 4K(-68) buffer. This report was generated:

					---- BUFFER -----			ALL	
TO	TRANSPORT	BUFFER	REQ	%	%	%	%	PATHS	REQ
SYSTEM	CLASS	LENGTH	OUT	SML	FIT	BIG	OVR	UNAVAIL	REJECT
JA0	DEFAULT	20,412	2,167	92	8	<1	100	0	0
	DEFSMALL	956	29,730	0	100	0	0	0	0
	NEWXCF	4,028	106,018	0	0	**100**	0	0	0
JB0	DEFAULT	20,412	6,132	97	3	<1	100	0	0
	DEFSMALL	956	82,687	0	100	0	0	0	0
	NEWXCF	4,028	18,085	0	0	**100**	0	0	0

Since all the NEWXCF messages were too big, the CLASSLEN was increased.

				----- BUFFER	-----			ALL	
TO SYSTEM	TRANSPORT CLASS	BUFFER LENGTH	REQ OUT	% SML	**% FIT**	% BIG	% OVR	PATHS UNAVAIL	**REQ REJECT**
JA0	DEFAULT	20,412	1,715	90	10	0	0	0	0
	DEFSMALL	956	37,687	0	100	0	0	0	0
	NEWXCF	**8,124**	103,063	0	**100**	0	0	0	**3,460**
JB0	DEFAULT	20,412	2,075	92	8	0	0	0	0
	DEFSMALL	956	38,985	0	100	0	0	0	0
	NEWXCF	**8,124**	117,727	0	**100**	0	0	0	**195**

Now all the messages fit, but some are being REJECTed. This suggests message buffer space for the outbound path is no longer large enough. The XCF path statistics confirm outbound messages are queuing up.

TO SYSTEM	Y P	DEVICE, OR STRUCTURE	TRANSPORT CLASS	REQ OUT	**AVG Q LNGTH**	AVAIL	BUSY
JA0	S	IXCPLEX_PATH1	DEFAULT	1,715	0.00	1,715	0
	S	IXCPLEX_PATH2	DEFSMALL	486	0.00	486	0
	S	IXCPLEX_PATH3	NEWXCF	103,063	**1.42**	102,818	245
	C	C600 TO C584	DEFSMALL	13,644	0.00	13,644	0
	C	C601 TO C585	DEFSMALL	13,603	0.00	13,603	0
	C	C602 TO C586	DEFSMALL	12,610	0.00	12,610	0
JB0	S	IXCPLEX_PATH1	DEFAULT	2,075	0.00	2,075	0
	S	IXCPLEX_PATH2	DEFSMALL	737	0.00	737	0
	S	IXCPLEX_PATH3	NEWXCF	117,727	**1.26**	117,445	282
	C	C610 TO C584	DEFSMALL	16,391	0.00	16,391	0
	C	C611 TO C585	DEFSMALL	12,131	0.01	12,131	0
	C	C612 TO C586	DEFSMALL	12,294	0.00	12,294	0

Increasing the MAXMSG on the PATHOUT for the NEWXCF transport class from 1000 to 2000 clears up the queuing delays.

				----- BUFFER	-----			ALL	
TO SYSTEM	TRANSPORT CLASS	BUFFER LENGTH	REQ OUT	% SML	% FIT	% BIG	% OVR	PATHS UNAVAIL	REQ REJECT
JA0	DEFAULT	20,412	2,420	93	7	0	0	0	0
	DEFSMALL	956	41,215	0	100	0	0	0	0
	VTAMXCF	8,124	133,289	0	100	0	0	0	0
JB0	DEFAULT	20,412	2,362	93	7	0	0	0	0
	DEFSMALL	956	39,302	0	100	0	0	0	0
	VTAMXCF	8,124	143,382	0	100	0	0	0	0

The BUSY conditions are reduced, and more importantly the AVG Q LNGTH has been greatly reduced. Since the pathout with the contention is a coupling facility structure AVG Q LNGTH is an appropriate metric to use when tuning.

TO SYSTEM	TYP	FROM/TO DEVICE, OR STRUCTURE	TRANSPORT CLASS	REQ OUT	**AVG Q LNGTH**	AVAIL	BUSY
JA0	S	IXCPLEX_PATH1	DEFAULT	2,420	0.00	2,420	0
	S	IXCPLEX_PATH2	DEFSMALL	361	0.00	361	0
	S	IXCPLEX_PATH3	NEWXCF	133,289	**0.08**	133,117	2
	C	C600 TO C584	DEFSMALL	12,700	0.00	12,700	0
	C	C601 TO C585	DEFSMALL	16,421	0.00	16,421	0
	C	C602 TO C586	DEFSMALL	14,173	0.00	14,173	0
JB0	S	IXCPLEX_PATH1	DEFAULT	2,362	0.00	2,362	0
	S	IXCPLEX_PATH2	DEFSMALL	1,035	0.00	1,033	2
	S	IXCPLEX_PATH3	NEWXCF	143,382	**0.09**	143,086	296
	C	C610 TO C584	DEFSMALL	12,647	0.00	12,646	1
	C	C611 TO C585	DEFSMALL	15,944	0.00	15,944	0
	C	C612 TO C586	DEFSMALL	12,183	0.00	12,182	1

When determining how to tune the application to limit the number of XCF messages, a DEF8K transport class for UNDESIG messages was created and the NEWXCF class assigned to this application was eliminated.

Note: In this case study, the messages were being queued because the message buffer space was too small. If, instead of REJECTS, there was a high percentage of messages marked as BUSY, then increasing the number of signaling paths would have been appropriate.

Incidentally the path associated with the NEWXCF was a CF structure which used the new HiPerLinks available on the G3 server. The structure was chosen since it was quicker and easier to implement. Since the structure was receiving over 500 req/sec, it was unclear if the structure could handle the traffic. As can be seen from the queue lengths, it was capable of handling this rate.

Special Notices

This publication is intended to help the customer manage an OS/390 Parallel Sysplex environment. The information in this publication is not intended as the specification of any programming interfaces provided by OS/390. See the publication section of the IBM programming announcement for the appropriate OS/390 release for more information about what publications are considered to be product documentation.. Where possible it is recommended to follow-up with product related publications to understand the specific impact of the information documented in this publication.

The information contained in this document has not been submitted to any formal IBM test and is distributed on an "as is" basis without any warranty either expressed or implied. The use of this information or the implementation of any of these techniques is a customer responsibility and depends on the customer's ability to evaluate and integrate them into the customer's operational environment. While each item may have been reviewed by IBM for accuracy in a specific situation, there is no guarantee the same or similar results will be obtained elsewhere. Customers attempting to adapt these techniques to their own environments do so at their own risk.

Performance data contained in this document was determined in a controlled environment; therefore the results which may be obtained in other operating environments may vary significantly. No commitment as to your ability to obtain comparable results is any way intended or made by this release of information.

Appendix

APARS

The following APARs are directly related to XCF performance and/or RMF reporting of XCF performance:

1. OW10662 - %BIG is always 0 on RMF XCF report
2. OW13190 - %SML is always 0 on RMF XCF report
3. OW13418 - C * UNK on XCF path reports
4. OW14617 - Excessive XCF internal signals
5. OW16903 - XCF expands largest class (rather than one named DEFAULT)
6. OW19913 - *COUNTS RESET in RMF XCF path report for structures
7. OW21327 - RMF Mon III never shows XCF delays for XCF structures
8. OW22065 - AVG Q LENGTH for structures is always 0
9. OW38138 - XCF Path Selection Enhancements
10. OW41317 - RMF records XCF MXFER time in RMF 74.2 records

XCF users (List will change as new exploiters are added):

GROUP	OWNER
AOFSMGRP	AOC
ASFBGRP1	AOC
ATRRRS	* RRS
BBGROUP	CPSM
COFVLFNO	* VLF
DFHIR000	CICS
DSNDB1G	DB2
DXRDBZG	DB2
EJESEJES	EJES
ESCM	ESCOM MGR
EZBTCPCS	BatchPipes
IDAVQUIO	VSAM
IGWXSGIS	VSAM RLS
IRLMGRP1	IRLM
IRRXCF00	RACF
ISTCFS01	VTAM
ISTXCF	VTAM
IXCLOxxx	*# XES
JES2xx	$JES2 MAS
JES3xx	@JES3 Cmplx
POKUTC58	NJE-JES2
SYSATBxx	APPC
SYSDAE	* DAE
SYSENF	* ENF
SYSGRS	* GRS
SYSIGW00	DF/SMS - PDSE
SYSMCS	* CONSOLES
SYSMCS2	* CONSOLES
SYSRMF	RMF
SYSWLM	* WLM

```
* denotes MVS component
# one for each lock and serialized list
  structure
```

JES2xx - Local node name
JES3xz - Node name on NJERMT init stmt

Sample COUPLExx PARMLIB member

This PARMLIB member defines two transport classes:
DEFSMALL - used for messages <= 956, defined with 4 PATHOUTs:

- 1 CF structure named IXCPLEX_PATH2
- 3 CTC connections

for each of the 10 systems in the SYSPLEX.

DEFAULT - used for messages >956, defined with 1 PATHOUT
1 CF structure named IXCPLEX_PATH1.

Since this is an OS/390 R2 system, the MAXMSG default of 750 is used for everything except the PATHIN and PATHOUT paths which use structures.

```
CLASSDEF CLASS(DEFAULT)  CLASSLEN(16316) GROUP(UNDESIG)
CLASSDEF CLASS(DEFSMALL) CLASSLEN(956)   GROUP(UNDESIG)

LOCALMSG MAXMSG(500) CLASS(DEFSMALL)

PATHOUT  CLASS(DEFSMALL) MAXMSG(1000) STRNAME(IXCPLEX_PATH2)
PATHOUT  CLASS(DEFAULT)  MAXMSG(1000) STRNAME(IXCPLEX_PATH1)
PATHIN   MAXMSG(1000)    STRNAME(IXCPLEX_PATH1,IXCPLEX_PATH2)

PATHOUT CLASS(DEFSMALL) DEVICE(C400,C410,C580,C590,C600,C610)
PATHOUT CLASS(DEFSMALL) DEVICE(C620,C630,C640,C650)
PATHIN                  DEVICE(C404,C414,C584,C594,C604,C614)
PATHIN                  DEVICE(C624,C634,C644,C654)

PATHOUT CLASS(DEFSMALL) DEVICE(C401,C411,C581,C591,C601,C611)
PATHOUT CLASS(DEFSMALL) DEVICE(C621,C631,C641,C651)
PATHIN                  DEVICE(C405,C415,C585,C595,C605,C615)
PATHIN                  DEVICE(C625,C635,C645,C655)

PATHOUT CLASS(DEFSMALL) DEVICE(C402,C412,C582,C592,C602,C612)
PATHOUT CLASS(DEFSMALL) DEVICE(C622,C632,C642,C652)
PATHIN                  DEVICE(C406,C416,C586,C596,C606,C616)
PATHIN                  DEVICE(C626,C636,C646,C656)
```