

#SHAREorg

S H A R E
Technology • Connections • Results

Sysplex Failure Management (SFM) History and Proven Practice Settings

Mark A Brooks
IBM
mabrook@us.ibm.com

4:30 p.m. Wednesday March 14, 2012
Session 10850

The bane of all clustered computing systems is sympathy sickness. Failure to deal with an unresponsive component or an unresponsive system in a timely manner can lead to mysterious slow downs in the sysplex, even outright catastrophe. Appropriate configuration of the Sysplex Failure Management (SFM) policy allows the systems in the sysplex to automatically resolve such issues before outages occur. SFM has come a long way since the early days of the sysplex. This presentation summarizes all of the options that SFM provides to help you protect your sysplex from the consequences of sympathy sickness. Particular attention will be given to the exploitation of BCPii by SFM. Use of this technology may well be the greatest thing you will ever do to protect your sysplex from sympathy sickness.

It is assumed that the reader is familiar with basic sysplex concepts and terminology.

This presentation attempts to explain the various problems that SFM is designed to address. Our attention will be focused on the key ideas and principles behind SFM, not the actual mechanics of creating and managing SFM policies. The goal is to provide a basic understanding of the risks and benefits of using (or not using) various SFM technologies. With this foundation, one can then make practical application of these principles by exploiting SFM in a way that is best suited to your own installation.

Trademarks

The following are trademarks of the International Business Machines Corporation in the United States and/or other countries.



IBM®	MQSeries®	S/390®	z9®
ibm.com®	MVS™	Service Request Manager®	z10™
CICS®	OS/390®	Sysplex Timer®	z/Architecture®
CICSPlex®	Parallel Sysplex®	System z®	zEnterprise™
DB2®	Processor Resource/Systems Manager™	System z9®	z/OS®
eServer™	PR/SM™	System z10®	z/VM®
ESCON®	RACF®	System/390®	z/VSE®
FICON®	Redbooks®	Tivoli®	zSeries®
HyperSwap®	Resource Measurement Facility™	VTAM®	
IMS™	RETAIN®	WebSphere®	
IMS/ESA®	GDPS®		
	Geographically Dispersed Parallel Sysplex™		

The following are trademarks or registered trademarks of other companies.

IBM, z/OS, Predictive Failure Analysis, DB2, Parallel Sysplex, Tivoli, RACF, System z, WebSphere, Language Environment, zSeries, CICS, System x, AIX, BladeCenter and PartnerWorld are registered trademarks of IBM Corporation in the United States, other countries, or both.
 DFSMSHsm, z9, DFSMSmm, DFSMSdtp, DFSMSdss, DFSMS, DFS, DFSORT, IMS, and RMF are trademarks of IBM Corporation in the United States, other countries, or both.
 Java and all Java-based trademarks are trademarks of Sun Microsystems, Inc. in the United States, other countries, or both.
 Microsoft, Windows, Windows NT, and the Windows logo are trademarks of Microsoft Corporation in the United States, other countries, or both.
 InfiniBand is a trademark and service mark of the InfiniBand Trade Association.
 UNIX is a registered trademark of The Open Group in the United States and other countries.
 Linux is a registered trademark of Linus Torvalds in the United States, other countries, or both.

* All other products may be trademarks or registered trademarks of their respective companies.

Notes:

Performance is in Internal Throughput Rate (ITR) ratio based on measurements and projections using standard IBM benchmarks in a controlled environment. The actual throughput that any user will experience will vary depending upon considerations such as the amount of multiprogramming in the user's job stream, the I/O configuration, the storage configuration, and the workload processed. Therefore, no assurance can be given that an individual user will achieve throughput improvements equivalent to the performance ratios stated here.

IBM hardware products are manufactured from new parts, or new and serviceable used parts. Regardless, our warranty terms apply.

All customer examples cited or described in this presentation are presented as illustrations of the manner in which some customers have used IBM products and the results they may have achieved. Actual environmental costs and performance characteristics will vary depending on individual customer configurations and conditions.

This publication was produced in the United States. IBM may not offer the products, services or features discussed in this document in other countries, and the information may be subject to change without notice. Consult your local IBM business contact for information on the product or services available in your area.

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

Information about non-IBM products is obtained from the manufacturers of those products or their published announcements. IBM has not tested those products and cannot confirm the performance, compatibility, or any other claims related to non-IBM products. Questions on the capabilities of non-IBM products should be addressed to the suppliers of those products.

Prices subject to change without notice. Contact your IBM representative or Business Partner for the most current pricing in your geography.

For a complete list of IBM Trademarks, see
www.ibm.com/legal/copytrade.shtml

History



- **MEMSTALLTIME**
 - **SSUMLIMIT**
 - **SFM with BCPii**
 - **System Default Action**
 - **XCF FDI Consistency**
 - **Critical Members**
 - **CFSTRHANGTIME**
- z/OS 1.8
 - z/OS 1.9
 - z/OS 1.11
 - z/OS 1.11
 - z/OS 1.11
 - z/OS 1.12
 - z/OS 1.12


SFM is the subcomponent within XCF that deals with the detection and resolution of sympathy sickness conditions that can arise when a system or sysplex application is unresponsive

3




SFM deals with the detection and resolution of sympathy sickness conditions that can arise when a system or a sysplex application is unresponsive. This slide summarizes the various SFM related technologies and the z/OS release in which they first appeared. I claim that exploitation of “SFM with BCPii” may well be the most significant thing you can do for your installation with regard to availability in the sysplex.

Terminology



<ul style="list-style-type: none"> • FDI • ISA • ISI • SSUM • INTERVAL • SSUM ACTION • SSUM INTERVAL • MONITOR-DETECTED STOP 	<ul style="list-style-type: none"> - <u>F</u>ailure <u>D</u>etection <u>I</u>nterval - <u>I</u>ndeterminate <u>S</u>tatus <u>A</u>ction - <u>I</u>ndeterminate <u>S</u>tatus <u>I</u>nterval - <u>S</u>ystem <u>S</u>tatus <u>U</u>pdate <u>M</u>issing = FDI = ISA = ISI = SSUM
<ul style="list-style-type: none"> • <i>actionTIME(nostatus-interval)</i> - <i>action</i> = ISA, <i>nostatus-interval</i> = ISI - ISOLATETIME(n) or DEACTTIME(n) or RESETTIME(n) 	
<ul style="list-style-type: none"> • Sysplex partitioning 	<ul style="list-style-type: none"> - Remove system from sysplex
<ul style="list-style-type: none"> • SFM 	<ul style="list-style-type: none"> - Sysplex Failure Management

4



- Each system has an FDI - set by the COUPLExx PARMLIB member (can also be updated dynamically via SETXCF operator command). Generally, one should just take the default value. If a system appears to be unresponsive for as long as the FDI, the peer systems in the sysplex assume it has failed.

- ISA/ISI is set for each system according to the rules of the SFM policy, or in some cases, the system default. If the system appears to be unresponsive for as long as the FDI, action should be taken to resolve the situation. ISI determines how long the system should wait before taking that action. ISA is the action to be taken. It can be isolate, deactivate, or reset. Isolate has wide spread use. Use of deactivate and reset is less common. Isolation will initiate partitioning and then “fence” the system so that it cannot access shared resources. Reset/Deactivate will reset/deactivate the LPAR in which the system image resides and initiate partitioning.

- Every system in the sysplex monitors every other system in the sysplex. Any system can recognize that some other system in the sysplex is experiencing a SSUM condition. If the SSUM persists longer than the FDI, one of the peer systems will take charge of dealing with the unresponsive system. Although one system is in charge, all the surviving systems have the potential to “help” (for example, they all may try to fence the failed system).

- A system that joined the sysplex is removed from the sysplex through the partitioning process. Partitioning must ensure that the target system is unable to access any shared resources, update XCF control structures to indicate that the system no longer exists, and notify sysplex applications and subsystems that the instances that used to reside on the subject system no longer exist. Normal operation for the application/subsystem might not resume until the surviving instances finish cleanup for the failed peer instance.

Sympathy Sickness



- When a system becomes unresponsive
 - It may be serializing shared resources with RESERVEs, ENQ's, Locks
 - It may stop sending responses or otherwise fail to participate in various "group" protocols
- Other work may experience delays and hangs
 - Problem compounds as the sympathy sickness spreads
- Timely intervention is needed
 - One must correctly identify the culprit
 - Take corrective action

5



Sympathy sickness is nothing new. It can raise its ugly head whenever there are two processors that need access to a common resource. Having the system take automatic corrective actions to relieve sympathy sickness is nothing new. The spin loop timeout value and the spin loop recovery actions defined in the EXSPATxx parmlib member to recover from excessive spin conditions is one example.

Single-system "sick but not dead" issues can and do escalate to cause sysplex-wide problems. A sick system typically holds resources needed by other systems and/or is unable to participate in sysplex wide processes. Other systems become impacted if they must wait for a resource to become available or for a response to be made. But the root cause of the sickness is a single system problem (contention, dispatching delays, spin loops, overlays, queue/data corruption, etc). Routing work away from the troubled system does not necessarily guarantee that other systems will not be impacted.

System/sysplex cleanup when subsystems or systems actually *terminate* is not the problem. Indeed, removal of the sick system from the sysplex generally remedies the problems. Allowing non-terminating problems to persist, where something simply becomes unresponsive, typically compounds the problem. By the time manual intervention is attempted, it is often very difficult to identify the appropriate corrective action. Appropriate SFM specifications enable systems in the sysplex to take corrective action automatically. In most cases, each parameter arose out of a real world situation that involved some sort of outage.

In general, timely corrective action is a balancing act between acting too soon (it would have recovered on its own) and acting too late (innocent parties suffered serious harm). For operations staff, the problems are further complicated by the fact that decisions must often be made under duress, and often without any certainty as to what is actually happening.

Where we are going

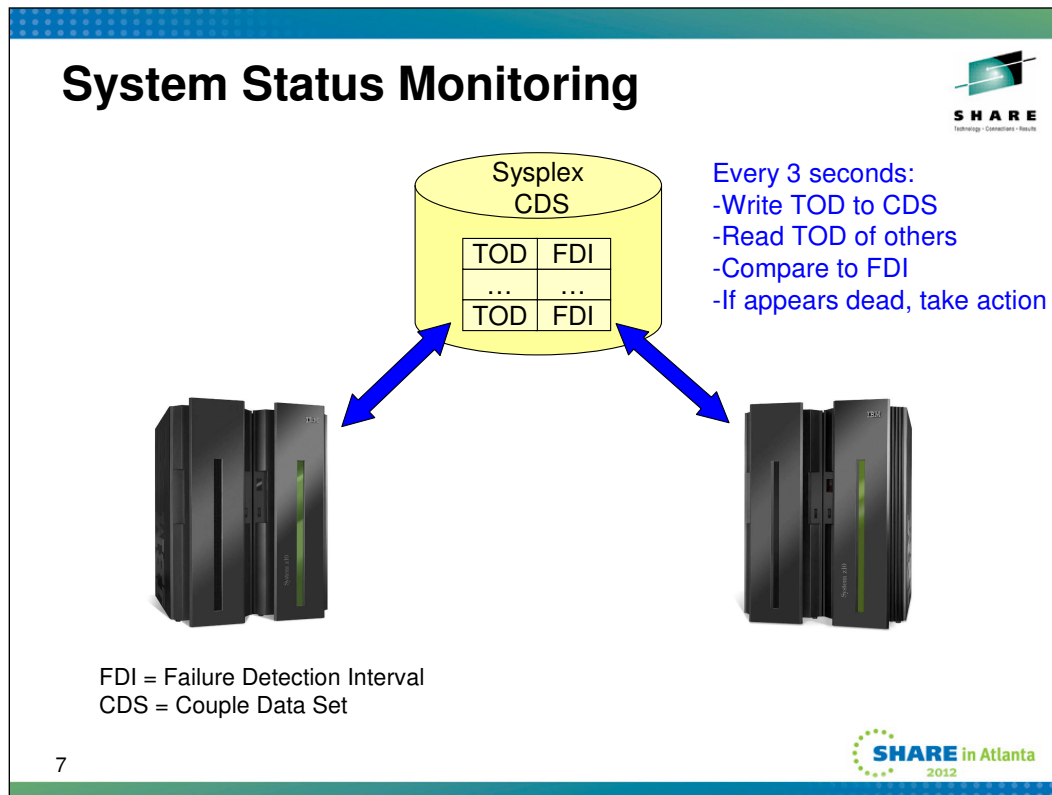


- Look back in history at early days of sysplex, moving forward in time to explore the issues encountered and solutions provided along the way
 - Though not necessarily in actual historical order
- Initially focus on unresponsive systems
- Then broaden the scope to other forms of sympathy sickness

6

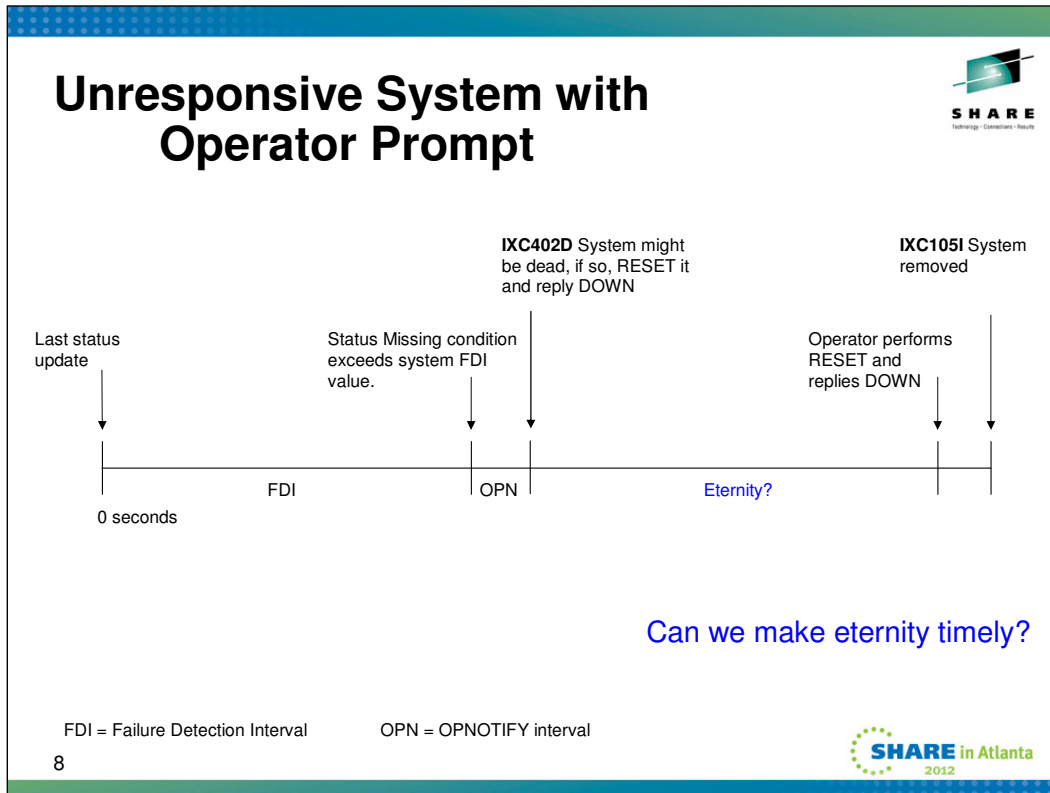


So in particular, there will be some slides whose description/content is not accurate for the way the systems behave today, though they are accurate with respect to how the system/sysplex would have behaved in the past.



The sysplex Couple Data Set (CDS) contains a status table with an entry for every possible system in the sysplex. Every few seconds, each system writes a timestamp with the current time of day (TOD) to its entry in the table, and simultaneously reads the timestamps written by every other system. Each system then inspects its in-store copy of the status table. If the interval since the last time a system updated its timestamp is greater than the Failure Detection Interval (FDI) for that system, the system is deemed to be unresponsive. Action is required. In the early days, “action” meant issue message IXC402D to ask the operator for help.

Issuing the `D XCF,S,ALL` command will show the local system's view of these timestamps.



This timeline depicts the key events that would occur when XCF engages the operator for help with handling an unresponsive system.

Scenario: Some system stops updating its status in the sysplex CDS. Eventually, some other system in the sysplex notices that the last update is older than the FDI for the system. When that happens, the system is deemed to be unresponsive. After the operator notification interval (OPNOTIFY) expires, some system issues the IXC402D message:

IXC402D sysname LAST OPERATIVE AT hh:mm:ss. REPLY DOWN AFTER SYSTEM RESET OR INTERVAL=SSS TO SET A PROMPT TIME

In response to this message, the operator is expected to either restore the subject system to normal operation, or to go reset the image and reply to the message to tell XCF to remove it from the sysplex. However, PMR after PMR tells the story of the outages that have resulted from the operator failing to respond to the message. The longer this goes on, the more likely it is for sympathy sickness to occur and the more difficult it becomes to correctly diagnose the problem. “Everything seemed to hang until we removed the system from the sysplex”. In short, manual intervention is not a reliable means of providing corrective action in a manner that is timely enough to avoid serious sympathy sickness.

Can We Shorten Eternity ?



- Sysplex Failure Management (SFM)
 - Use XCF Status Monitoring to detect unresponsive systems
 - Create a policy to specify whether XCF is to automatically remove unresponsive systems from the sysplex
- But what about the operator RESET of the system?
 - A critical step
 - Without it, data can be corrupted
 - So automating response of DOWN to IXC402D is likely bad

9



SFM can help eliminate the delays that arise when manual intervention does not occur in a timely manner. Failure to deal with the unresponsive system in a timely manner allows sympathy sickness condition to spread.

However, automatic action cannot be taken unless XCF can be certain that the unresponsive system does not have access to shared resources. Although the system appears to be unresponsive, it could still be running. If so, it could be updating data bases (for example). If so, telling the survivors that they can cleanup for their failed peer creates the potential for “rogue” I/O requests to make unserialized updates to the data base. Putting the processor in a wait-state does not prevent in-flight I/O from continuing in the channel subsystem. One must be certain that the system has undergone an I/O reset in order to ensure that there is no latent I/O. Thus the problem of isolating the unresponsive system must be solved before automatic removal of the unresponsive system can proceed.

It is rather frightening to hear of installations that have automated a reply of “DOWN” to the IXC402D, or whose operators reply “DOWN” without first ensuring that the system has been reset. They are exposing themselves to data corruption. This is not a theoretical problem. A few installations have proven that it can occur. Ouch.

Fencing

The diagram shows two mainframe systems connected to a central Coupling Facility (CF). Below the CF are three DASD units representing shared data. A red double slash indicates a disconnection or isolation of one system from the shared data.

Fencing isolates a system so that it cannot access shared data, thus making it safe for the survivors to release serialization of the shared resources.

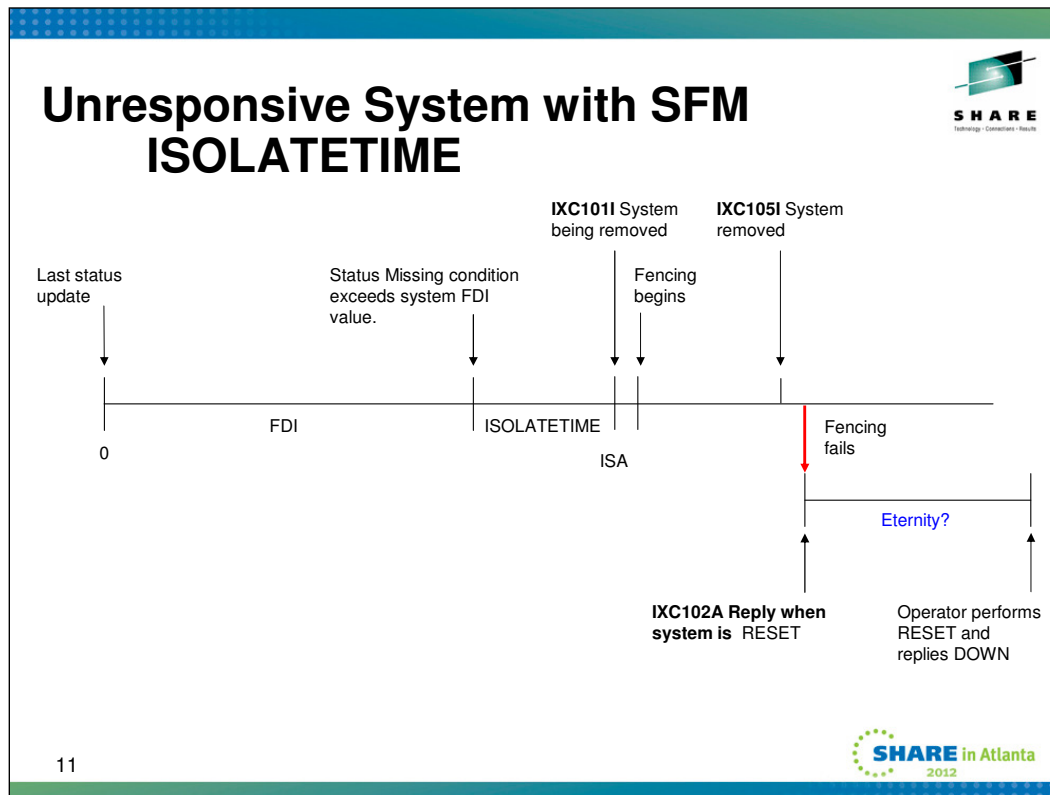
A command is sent via a CF to the target CEC. The target image will not be able to initiate any new I/O and ongoing I/O will be terminated.

10

SHARE in Atlanta 2012

System isolation allows a system to be removed from the sysplex without operator intervention, while ensuring that data integrity in the sysplex is preserved. Specifically, system isolation (sometimes called "fencing") terminates all in-progress I/O activity and coupling facility accesses, and prevents any new I/O activity and coupling facility access from starting, thus ensuring that the system is unable to access and modify shared I/O resources that the rest of the sysplex is using. System isolation therefore allows the sysplex to free up serialization resources (for example, locks and ENQs) that are held by the target system so that they may be acquired and used by the rest of the sysplex, while still preserving data integrity for all shared data.

Prior to z/OS 1.11, fencing was initiated only when the SFM policy was coded with the keyword ISOLATETIME. As of z/OS 1.11, fencing is done, as appropriate, whenever a system is removed from the sysplex. The fencing command is sent via a coupling facility. All systems will attempt to fence the system that is to be removed from the sysplex. Ganging up in this manner helps overcome connectivity issues wherein the system in charge of removing the unresponsive system from the sysplex does not have a way to get the fencing command delivered. Of course, if one is running a base sysplex (no CF's), fencing cannot be done.

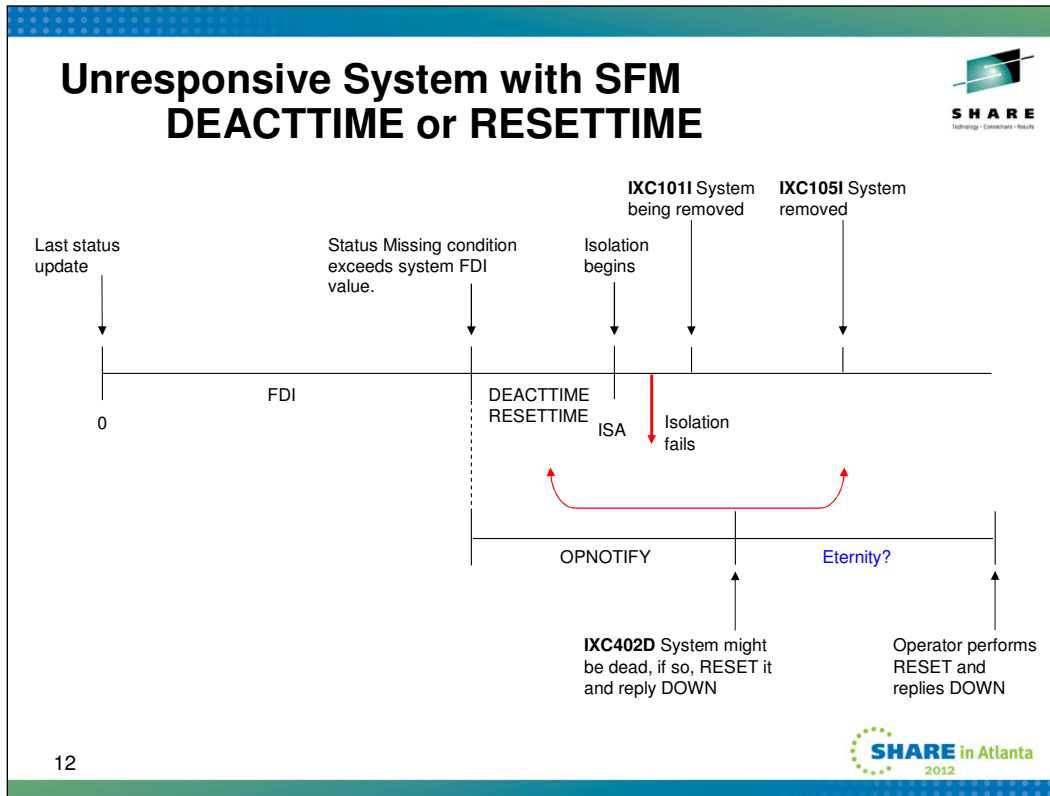


This timeline depicts the key events that would occur when an unresponsive system is to be automatically removed from the sysplex per an SFM policy that specifies ISOLATETIME.

Some system stops updating its status in the sysplex CDS. Eventually, some other system in the sysplex notices that the last update is older than the FDI for the system. When that happens, the system is deemed to be unresponsive. After the Indeterminate Status Interval (ISI) expires, the Indeterminate Status Action (ISA) is taken. In this case, ISI is the number of seconds indicated by the ISOLATETIME keyword, and ISA is “initiate partitioning and try to fence the system”.

XCF initiates partitioning for the system, issuing message IXC101I to so indicate. It then initiates fencing of the target system. Fencing is successful if some system in the sysplex was able to send a fencing command via some CF to the target CEC, and the target CEC was able to reply that the target image had in fact been fenced. If successful, XCF finishes removing the system from the sysplex and issues message IXC105I to so indicate.

If fencing is not successful, XCF cannot finish partitioning because it cannot be certain that the target system has been safely isolated from shared resources. Fencing could fail as the result of a loss of connectivity which prevents successful transmission of the fencing command or its response. In practice, fencing often fails because the token used to validate the command does not match (you wouldn't want the wrong image or instance of the image to be fenced). For example, reIPLing the image invalidates the fence token. Unfortunately, in the absence of positive confirmation of a successful fence, XCF cannot assume that the system has been isolated. When that happens, XCF falls back to asking the operator for help. Message IXC102A is issued to have the operator indicate when the system has been reset. When the operator responds, partitioning can complete. Until the operator responds, sympathy sickness can spread.



This timeline depicts the key events that would occur when an unresponsive system is to be automatically removed from the sysplex per an SFM policy that specifies either DEACTTIME or RESETTIME.

Some system stops updating its status in the sysplex CDS. Eventually, some other system in the sysplex notices that the last update is older than the FDI for the system. When that happens, the system is deemed to be unresponsive. After the Indeterminate Status Interval (ISI) expires, the Indeterminate Status Action (ISA) is taken. In this case, ISI is the number of seconds indicated by the DEACTTIME (or RESETTIME) keyword, and ISA is “deactivate (or reset) the system and if successful, initiate partitioning”. If the deactivation (or reset) is successful, the system has been successfully isolated from shared resources. XCF initiates partitioning for the system, issuing message IXC1011 to so indicate. When XCF finishes removing the system from the sysplex, message IXC1051 is issued.

Unlike fencing, the deactivate (or reset) must be done from a peer system that resides in the same CEC as the target system. A very simple algorithm is used to make that happen. Since every system in the sysplex is monitoring the status of every other system in the sysplex, all the operational peers will eventually notice that the target system has become unresponsive. Thus each system in turn will consider whether it is able to do the deactivate (or reset). If a peer system resides on the appropriate CEC, it will attempt to deactivate (or reset) the target system. If successful, partitioning proceeds.

Now it could be the case that no peer system is running on the appropriate CEC, or it could be the case that the deactivate (or reset) fails. If so, XCF cannot initiate partitioning because it cannot be certain that the target system has been safely isolated from shared resources. When that happens, XCF must fall back to asking the operator for help. Thus, in parallel with the ISA, one of the systems will take responsibility for issuing message IXC402D to notify the operator. If partitioning is not initiated before the operator notification interval expires (OPNOTIFY), the message is issued. The message is deleted if partitioning is initiated after the message is issued. So if no system is able to successfully deactivate (or reset) the unresponsive system, XCF must rely on the operator to indicate when the system has been reset. When the operator responds, partitioning can be initiated. Until the operator responds, sympathy sickness can spread.

The ISI and the OPNOTIFY intervals start when the status missing condition exceeds the FDI. Generally, the ISI will be less than the OPNOTIFY, and the deactivate or reset will work, in which case, there is no need to issue the IXC402D. But if the OPNOTIFY is less than the ISI, the message may be issued before the isolation begins. If the isolation actions fail, the message will also be issued. This point is what the bowl shaped (red) arrows are trying to depict in the slide.

z/OS 1.11 System Default Action



- SFM Policy defines how XCF is to deal with an unresponsive system
- Each system “publishes” in the sysplex couple data set the action that is to be applied by its peers
- A system publishes “default action” if:
 - The policy does not specify an action for it, or
 - There is no SFM policy active
- “Default action” is:
 - PROMPT prior to z/OS 1.11
 - ISOLATETIME(0) as of z/OS 1.11

13



With z/OS V1R11, the default action for dealing with an unresponsive system is now in accordance with best practice. In the past, the default action was “ask the operator”. Now the default action is to proceed as if there was an SFM policy that specified ISOLATETIME(0).

If you want to preserve past default behavior of PROMPT for a z/OS 1.11 (or later) system, you must define and activate an SFM policy that explicitly specifies PROMPT. However, this is not recommended since best practice is ISOLATETIME(0). If you already explicitly specify PROMPT, consider changing to use the best practice specification of ISOLATETIME.

If you are not comfortable with ISOLATETIME(0) which implies that the system is to immediately partition an unresponsive system, then I suggest you consider ISOLATETIME(n) with some reasonable value for “n” instead of PROMPT. The idea is to choose an “n” that gives time for your automation or operational procedures to resolve the problem, yet is not so long that sympathy sickness conditions become severe. The key principle, which is true for most SFM policy specifications, is to have an automatic action in place as a backstop to save the day in case the automated/manual intervention fails to resolve the problem in a timely manner. XCF does not currently issue a specific “system is not responsive” message when ISOLATETIME is specified, so a potential glitch is that one would need some independent trigger to initiate the automated/manual intervention before the nonzero ISOLATETIME expired.

z/OS 1.11 System Default Action ...



- The resulting “default action” depends on who is monitoring who:
 - z/OS 1.11 will isolate a peer z/OS 1.11
 - z/OS 1.11 will PROMPT for lower level peer
 - Lower level system will PROMPT for z/OS 1.11
- D XCF,C shows what the system *wants*
 - *But it may not get that in a mixed sysplex*
- Note: z/OS 1.11 always tries fencing whenever it is needed
 - Lower level releases performed fencing only when an SFM policy was active

14



The SFM Policy defines how XCF is to deal with an unresponsive system (not updating status and not sending signals). Each system “publishes” in the sysplex couple data set the action that is to be applied by its peers. The system “default action” is published if either (a) the policy does not specify an action for it, or (b) there is no SFM policy active. Prior to z/OS 1.11, the “default action” was PROMPT, which causes the system to prompt the operator (message IXC402D) when the system appears to be unresponsive.

IBM suggests specifying or defaulting to ISOLATETIME(0) to allow SFM to immediately partition and fence an unresponsive system without operator intervention. As of z/OS V1R11, the system default will be in accord with this suggestion.

If a system becomes unresponsive and there is no active SFM policy, the monitoring system will take the “default action” against the failed system. This means if the monitoring system is a pre-z/OS V1R11 system, it will use the old default and prompt the operator. If the monitoring system is a z/OS V1R11 system, it will use the “default action” expected by the failed system. The D XCF,C command shows the ISA the system expects, but the monitoring system may use a different action if no action is specified in the SFM policy or if an SFM policy is not active.

Default action of ISOLATETIME(0) is not guaranteed. z/OS 1.11 observing a peer z/OS 1.11 knows the new default should be ISOLATETIME(0). But a z/OS 1.10 or 1.9 observing a z/OS 1.11 will continue to treat the default as PROMPT. A z/OS 1.11 observing a z/OS 1.9 or 1.10 system knows that their default action was PROMPT, and continues to honor that old behavior. In order to actually do an ISOLATETIME(0), one either needs to be able to fence the system or use BCPii to reset the system. So if SFM is unable to do either of those, the operator will be prompted.

Historically, there were problems

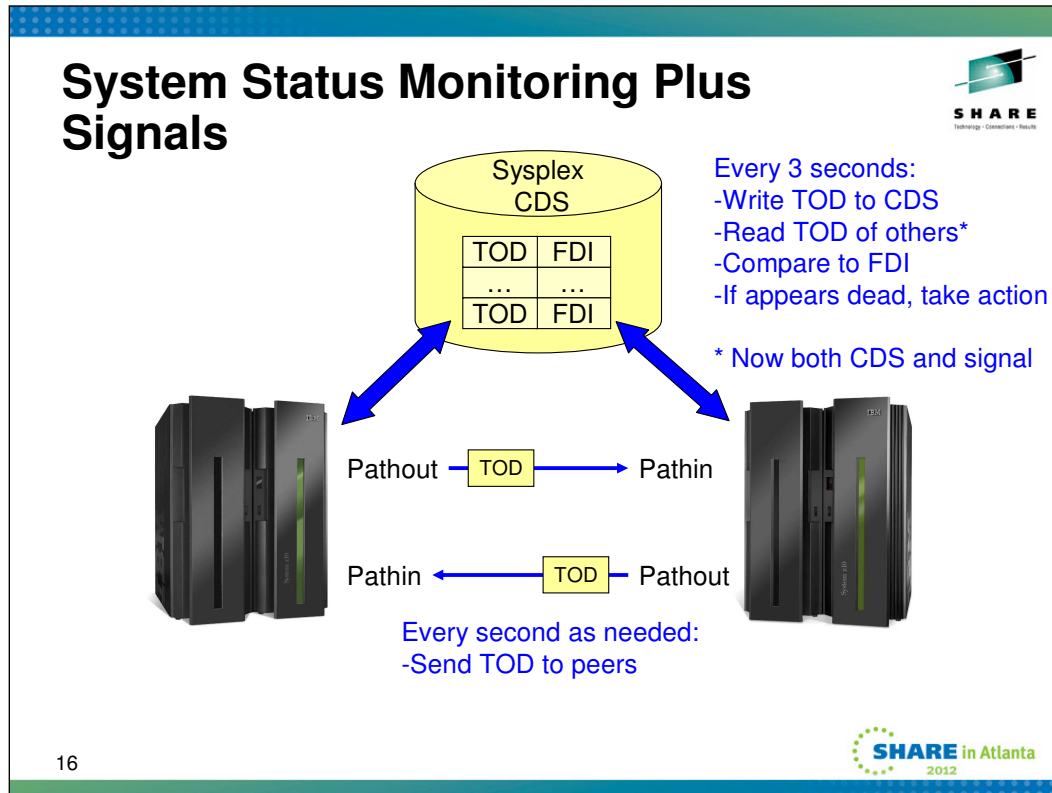


- Systems were being “needlessly” removed by SFM
- Failure to update status in the sysplex CDS was not a sufficiently reliable indicator of system failure
- So the system monitor was enhanced to watch for XCF signal traffic as well

15



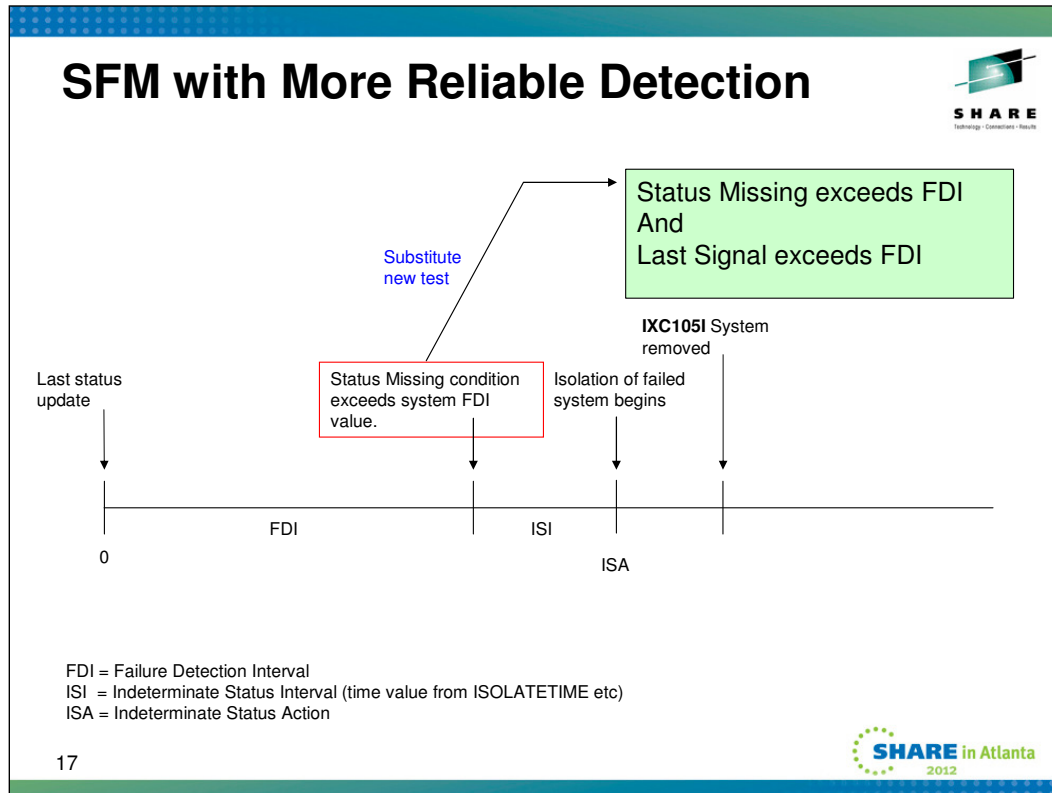
When SFM first released many years ago, it quickly developed a bad reputation because it was perceived to be taking systems out of the sysplex needlessly. That is, SFM too often removed systems that were still performing useful work. However, this was not so much a problem with SFM per se as it was with the mechanism used to determine that a system was unresponsive. Far too often there were circumstances where XCF was unable to write status to the sysplex CDS but the system continued to perform work, often with little or no impact on the peer systems. To resolve the problem, XCF enhanced its status monitor to increase the likelihood of it being able to correctly determine that a system was truly unresponsive. After changing the definition of unresponsive to be “not updating status and not sending signals”, the detection mechanism became much more reliable. Problems related to “needless removal” largely disappeared (though the process of restoring SFM’s “good name” is an ongoing effort).



Traditional system status monitoring, wherein every few seconds, each system writes its status to the sysplex CDS and reads the status of every other system in the sysplex continues as in the past. Independently of that processing, the XCF signalling service records TODs extracted from signals received from each system in the sysplex. Most every signal contains the TOD when the message it contains was created. As signals are received over an inbound signalling path, XCF records the most recent such TOD. Note that this is not the TOD when the signal was received, as there could have been transfer delays. The TOD extracted from the signal is a TOD that is known to have been taken by the sending system, and is therefore an indication of a time when the system was known to have been running. Though not likely, it might be the case that a pair of systems have no exploiter initiated signal traffic. If so, the XCF signal service ensures that a signal is sent to every other system at least once a second. Thus the status monitor should normally find recent signal activity from all operational systems.

When the monitor inspects the status TOD of its peer systems, and finds that the interval since the last update exceeds the FDI, it then inspects the TODs that have been recorded by the signalling service. If the interval since the most recent signal TOD also exceeds the FDI, the monitor concludes that the subject system is unresponsive. If the system is not updating status but still sending signals, the system might be having some issues, but it remains healthy enough to be capable of sending signals.

Having the status monitor consider both status updates and signals made the detection of unresponsive systems much more reliable.



In the distant past, the XCF status monitor deemed a system to be unresponsive if it stopped updating its status in the sysplex CDS for an interval that exceeded the system's FDI. That test proved to be an unreliable indicator of whether a system was healthy enough to do useful work. Today, the XCF status monitor deems a system to be unresponsive only if both (1) the interval since the last status update in the sysplex CDS exceeds the FDI and (2) the interval since the last signal TOD received from the system exceeds the FDI. This substitute definition of unresponsive proved to be a much more reliable indicator of whether the system was healthy enough to perform useful work.

Improving the reliability of the detection mechanism does not change how SFM works. That is, the SFM timeline remains the same. However, problems wherein SFM was perceived to be needlessly removing systems largely disappeared after this change was made.

However, we note two things for future consideration: (1) Even if a system is not updating status and not sending signals, it might still be running. (2) A system that stops updating status and stops sending signals is likely dead, but the system waits the entire FDI before the condition is recognized.

More Reliable Detection



- Use of status update TODs along with the TODs from the signals proved to be a much more reliable indicator
 - When both TODs stop making progress, there is a real good chance that the system failed
 - Issues with respect to “needless” removal largely disappeared
- But what if only one of the indicators is moving?
 - Signal TODs stop but status updates continue, or
 - Status updates stop, but signal TODs continue

18



So we are in a much better spot since we have a mechanism to reliably detect unresponsive systems. However, it does raise some new issues. What does it mean if one of the indicators is working but the other is not? Let us consider each of them in turn.

Signals Stop but Status Updates Continue



- Most likely an issue with signalling paths
 - Status updates imply XCF timer DIE is running
 - Which implies signal monitor is running
 - And it would be sending fresh TODs as needed
- System status monitor ignores this case
 - Signal monitor will deal with path problems
 - Restart or stop of inoperative paths may lead to loss of signalling connectivity
- If a pair of systems does not have signal connectivity, one of them must be removed from the sysplex

19

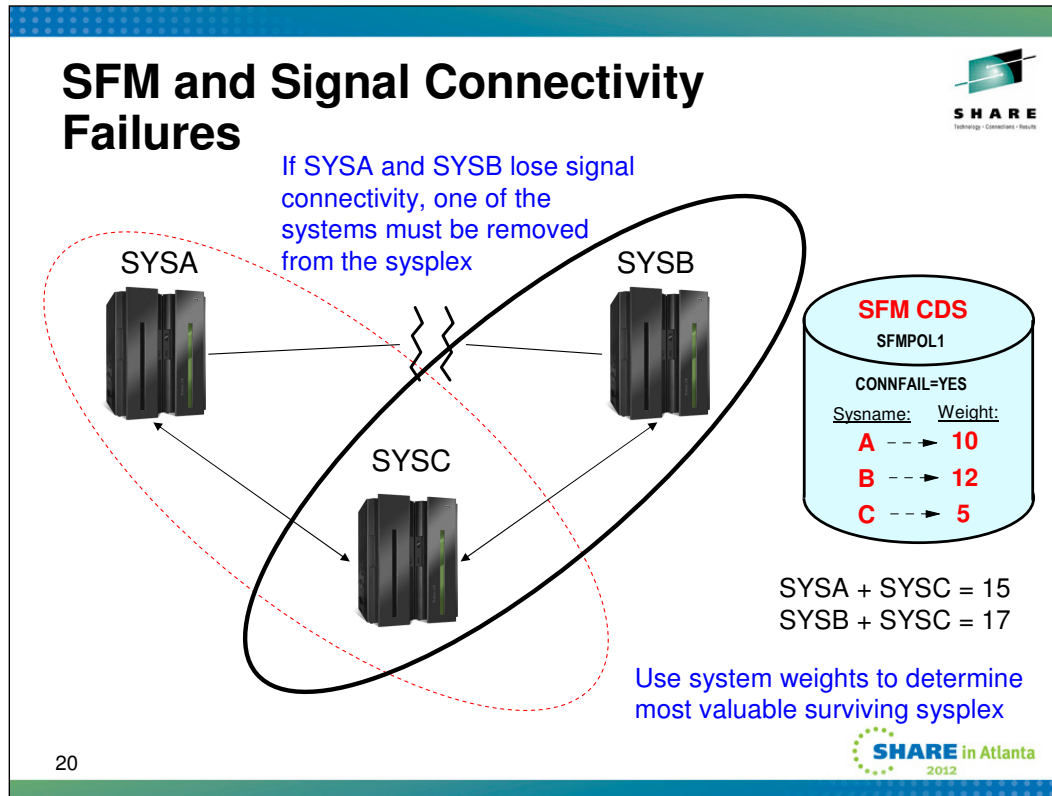


Let us consider the case where a system continues to update status in the sysplex CDS but stops sending signals.

Although the status update processing and the signal monitor (that ensures the sending of signals at least once per second) are independent processes, the two monitors do have a common timer DIE that triggers them. So the fact that status updates continue implies that the trigger mechanism is operational. It is unlikely that exploiters on the system would fail to generate XCF signals. Even if that were so, given that the timer DIE is running, it is unlikely that the signal monitor would fail to generate at least one signal per second.

So the most likely explanation for the lack of signals would be some sort of issue with the signalling paths. Given that the signal monitor is running, it would also be monitoring signal delivery across the signalling paths. If the signal traffic stopped flowing, the monitor would take corrective action to restore the signalling paths to normal service. That is, the paths would be restarted or stopped. If not even one path can be restored to service, the system loses signal connectivity. When that happens, one of the systems needs to be removed from the sysplex. This can be done automatically if the SFM policy so permits (CONNFAIL keyword), or manually (message IXC409D).

For future consideration, note that a lack of inbound buffers on the target system could be the reason that signals stop flowing. In such cases, the signal path may successfully re-establish connectivity between the systems, yet be incapable of performing signal transfers. The exchange of signals that allows the path to re-establish connectivity causes the target system to get a new signal TOD, so for a moment at least, the target system will see recent signal activity. If the lack of inbound buffers persists, the most likely reason is that exploiters on the target system are consuming XCF signal buffers. The SFM policy MEMSTALLTIME parameter can be used to enable the system to automatically relieve this form of sympathy sickness.



The CONNFALL parameter indicates whether SFM is to handle signaling connectivity failures for the sysplex. You specify CONNFALL on the DEFINE POLICY statement of the SFM policy.

If you specify CONNFALL(YES) (the default), and there is a signaling connectivity failure, SFM automatically determines which systems to keep and which to remove from the sysplex and then attempts to implement that decision by system isolation. In handling a system connectivity failure, SFM attempts to maximize the aggregate value of the surviving sysplex to the installation. The WEIGHT parameter of the system statement allows you to indicate the relative value of each system in the sysplex so that SFM can base its decision on installation-specified values.

If a system that is being removed can be successfully isolated, SFM can reconfigure the sysplex without operator intervention. Successful isolation requires the ability to fence the system or the ability to reset it via BCPii services. If isolation cannot be performed, message IXC102A prompts the operator to reset the system manually before it is removed from the sysplex.

CONNFAIL(NO) indicates that in a connectivity failure situation, MVS is to prompt the operator with message IXC409D to retry or remove the affected system(s). Operator intervention is not desirable because it leaves the sysplex exposed to sympathy sickness conditions if the problem is not resolved in a timely manner.

System weights

System weights are a way for you to communicate the relative importance to your business of the various members of your sysplex to SFM. If a failure results in SFM having to decide to partition some member from the sysplex, it will use the weights of all systems in the sysplex to determine which system should be removed from the sysplex in order to ensure the health of the group of systems with the highest aggregate weight.

System weights are assigned via the SFM policy. It is quite common to see sysplexes where every system has the same weight. But, in reality, it is likely that some systems are more important to your business than others, either because of their size, or the applications that run on that system, or the fact that they play a critical role (a GDPS Control system, for example). So it is generally a good idea to try to assign appropriate weights to all the systems in your sysplex. It is not always easy to do because a single static definition of weights may not reflect the dynamic nature of how the sysplex can behave with respect to the customer's particular workload and particular configuration. The problem of being able to reliably detect "doing useful work" and recognizing something as "important to my business" in the context of these very dynamic systems remains open.

Status Updates Stop but Signals Continue



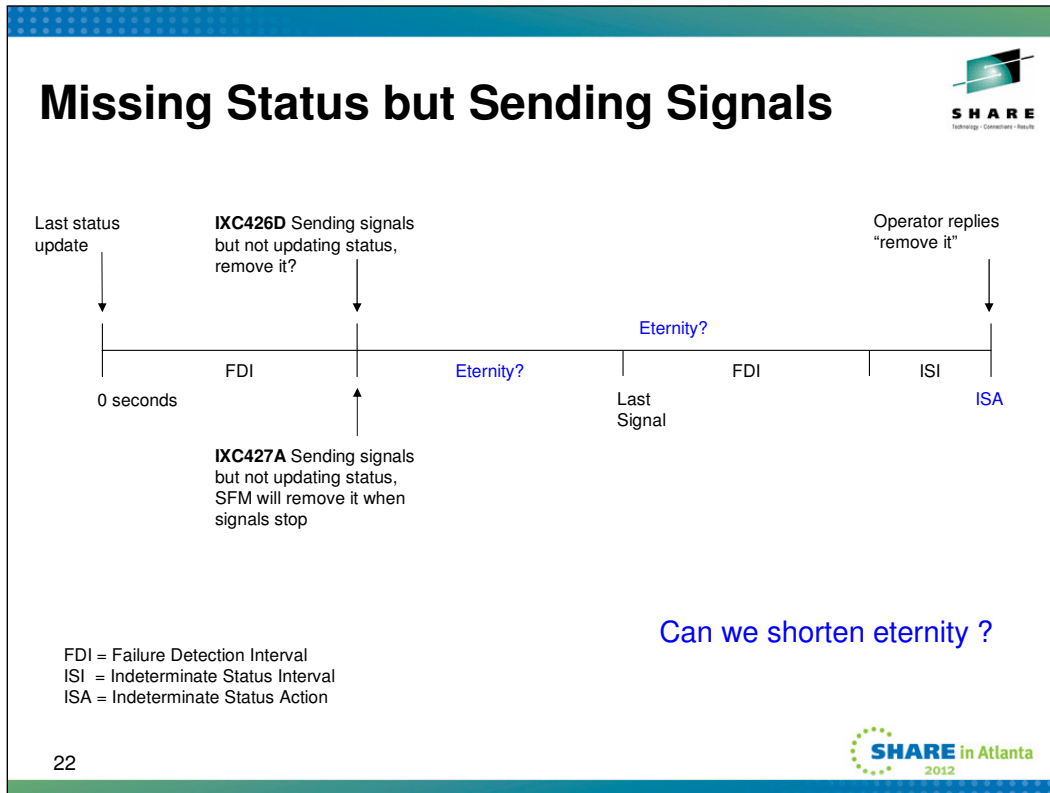
- System is not healthy
 - Often result of paging issues
 - Could be sysplex CDS issues (contention, performance, reserves, ...)
- Does not meet definition of failed
 - Not subject to automatic removal since signals imply system is still alive
 - So XCF engages the operator

21



Now we consider the case where the status monitor sees that a system has stopped updating its status in the sysplex CDS, but is still sending signals. Such a system no longer satisfies the conditions for the more reliable definition of “unresponsive”. So the status monitor will not tell SFM that the system is no longer responsive. Thus the system will not be automatically removed by SFM (at least in our story so far).

However, the fact that XCF cannot update status in the sysplex CDS does imply that something is wrong. Experience suggests that this state is often caused by paging issues. But it could also be due to contention or performance issues with respect to the sysplex CDS. Still there does seem to be a problem. Thus XCF issues a message to ask the operator for help. Depending on the SFM policy specification, message IXC426D or IXC427A is issued.



This timeline depicts the key events that would occur when a system stops updating its status in the sysplex CDS, but continues to send signals.

Message IXC426D is issued if there is no SFM policy or if the SFM policy specifies PROMPT as the action. In response to this message, the operator is expected to either restore the subject system to normal operation, or to go reset the image and reply to the message to tell XCF to remove it from the sysplex.

Message IXC427A is issued if the SFM policy permits automatic action (ISOLATETIME, RESETTIME, DEACTTIME). The system continues to run. If there should come a point where the interval since the last signal exceeds the FDI (at which point, both indicators indicate that the system is unresponsive), SFM will wait the Indeterminate Status Interval (ISI) and then initiate the Indeterminate Status Action (ISA) when it expires.

Thus at the right edge of the timeline, we reach a point where the subject system can be removed from the sysplex. But will we ever reach that point? Regardless of the policy specification, the timeline has a potentially infinite interval. In the case of the IXC426D message, the operator may fail to deal with the problem. In the case of the IXC427A message, the system may continue to send signals and thus never come to a point where the system is deemed to be unresponsive. So in either case, we could have an indefinite period where the system is not updating status yet sending signals.

Even though the system does not meet the definition of unresponsive and so is not subject to automatic isolation, remaining in this state will eventually lead to trouble. Sympathy sickness will eventually occur.

Can We Shorten Eternity?



- We have a sick but not dead situation
- Sympathy sickness will eventually occur
 - Lack of status updates suggests that sysplex CDS is not readily accessible
 - Join and Leave processing likely impacted
 - Some members record status and control info in CDS
 - Could also prevent systems from being removed from the sysplex
 - Thus preventing systems from getting back in as well

23



Experience shows that the sysplex can sometimes survive a long time with a system that is not updating status but sending signals. However, sympathy sickness will eventually occur. Failure to update status suggests that XCF is not able to access the sysplex CDS. Thus it may be the case that exploiters will not be able to join their XCF group. Member termination processing (leave, quiesce, EOT, EOM) may not be able to complete, which in turn could prevent other members from finishing the recovery for their failed peer and/or prevent the failed peer from being able to restart. Some exploiters record status information in the sysplex CDS. If the system was the monitor in charge of cleaning up for a failed system, it might not be able to finish removing the system from the sysplex, which in turn could prevent the failed system from being able to reIPL back into the sysplex. Eventually the rest of the sysplex will be impacted. Thus we really don't want the system to persist in this state forever.

z/OS 1.9 SSUMLIMIT



- SSUMLIMIT indicates how long a system is allowed to persist in the “not updating status but sending signals” state
 - Allows the installation to “bound” the amount of time that a sick system might impact the remainder of the sysplex
 - When the SSUMLIMIT interval expires, the system will be partitioned from the sysplex
- Not too aggressive, perhaps 15 minutes
 - Zero would be equivalent to the original status monitoring that led to “needless” removal of systems

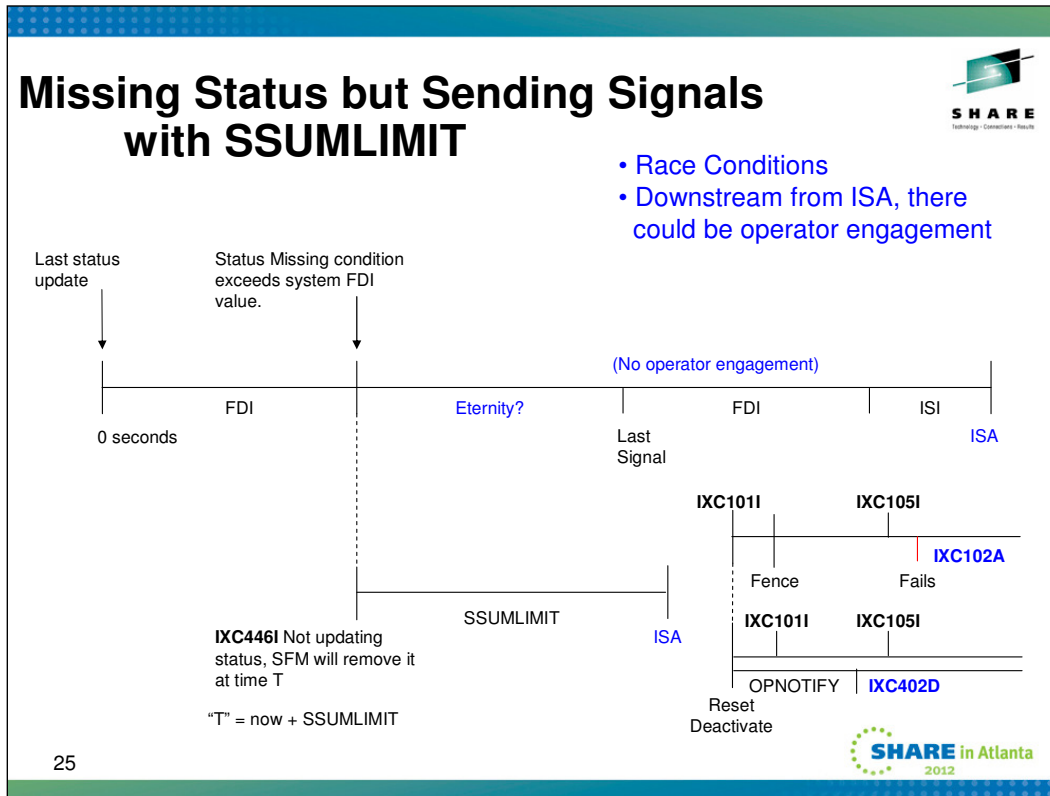
24



When SFM was first introduced, it took automatic action to remove a system from the sysplex if a system was not updating its status in the sysplex CDS. This led to many false positives wherein systems were removed from the sysplex even though they were still performing useful work. Thus the monitor was extended to incorporate both status updates and signalling activity. A system would be deemed unresponsive if it was neither updating the sysplex CDS nor sending signals. This proved to be a much better indicator of system responsiveness and has worked well for many many years.

However, there are sysplex wide processes that do require access to the sysplex CDS. So even though a system may be sending signals (which suggests that it is “alive”), the failure to update the sysplex CDS does suggest the existence of a problem. If the condition persists, there can be a sympathy sickness impact. So in z/OS 1.9, a new SSUMLIMIT keyword was added to the SFM policy. This keyword indicates the amount of time the sysplex should allow a system to persist in a “system status update missing” condition despite the fact that it is continuing to send signals. If the condition persists and the indicated time limit expires, SFM will remove the system from the sysplex.

Note that SSUMLIMIT should not be too small. Setting SSUMLIMIT to a small value in effect returns your sysplex to the days where only the system status updates get used to determine whether a system is responsive. That is, you expose yourself to the original SFM behavior that led to many “needless” removals. A value equivalent to 15 minutes seems to be reasonable. It provides enough time for systems to overcome the problem on their own and/or enough time for manual intervention to resolve the problem, yet does not let the condition persist for so long that the sympathy sickness impact becomes severe.



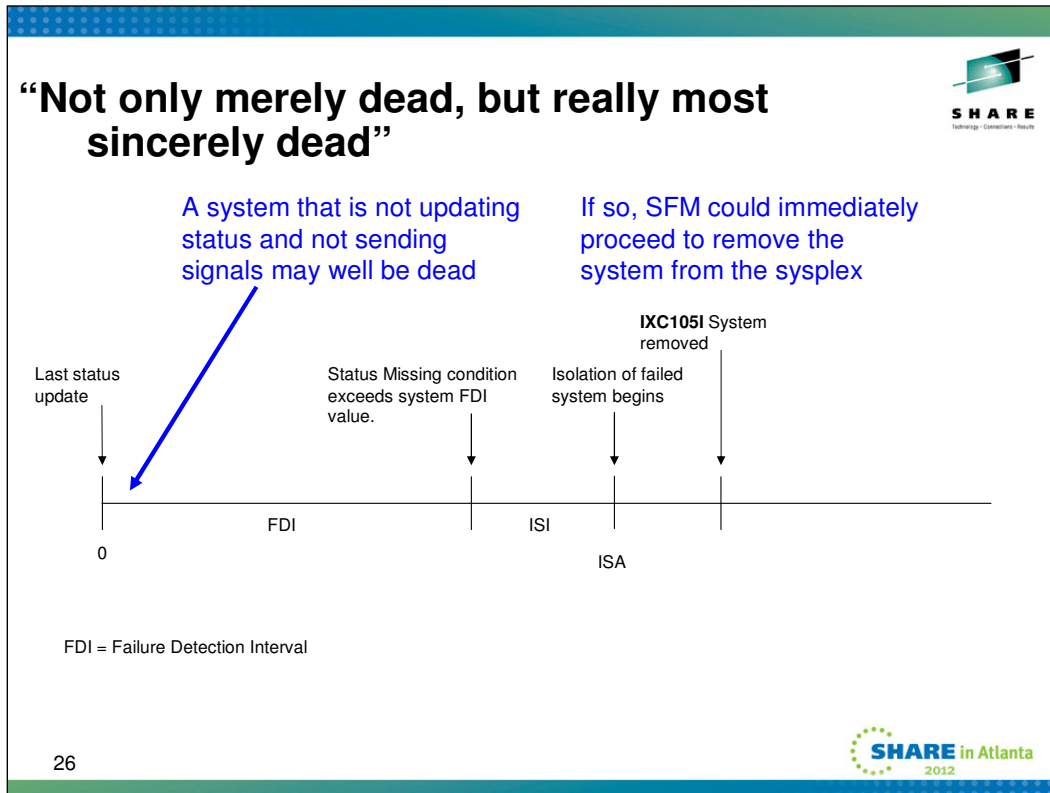
This timeline depicts the key events that would occur when a system stops updating its status in the sysplex CDS, but continues to send signals, and the SFM policy specifies SSUMLIMIT. Note that SSUMLIMIT can only be specified if ISOLATETIME, RESETTIME, or DEACTTIME is also specified for the system. In particular, PROMPT does not apply.

When the interval since the last status update exceeds the FDI, message IXC446I is issued to indicate that the system is sending signals but not updating its status. At this point we have two processes running in parallel. First, as depicted in the top timeline, normal monitoring continues. If there comes a time where the interval since the last signal exceeds the FDI, the system will be deemed unresponsive. SFM waits the ISI and then initiates the ISA to isolate the system. Second, as depicted in the bottom timeline, SFM waits the SSUMLIMIT interval. If SSUMLIMIT expires, it initiates the ISA to isolate the system (it does not wait for the ISI to expire).

In effect, we have a race condition, whichever thread finishes first will initiate the ISA to deal with the problem (and the other thread will terminate).

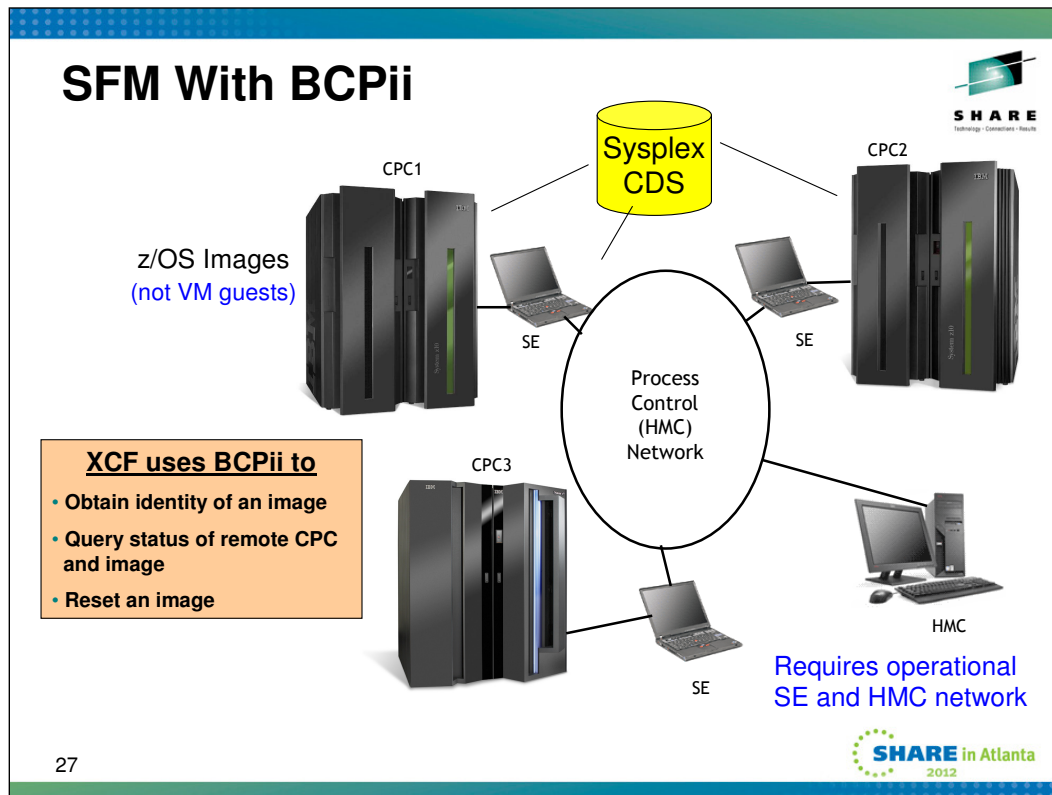
As is always the case, the ISA may not be successful. Thus there is always the potential for subsequent prompting of the operator.

The lower right corner of the diagram depicts the ISA, ISOLATETIME on the top, DEACTTIME and RESETTIME on the bottom. Previous slides showed these diagrams with more detail.



In some cases (many? most?), a system that is not updating status and not sending signals is in a wait-state. In such cases, waiting for the FDI to expire is a needless waste of time.

How wonderful it would be if the system could reliably detect not just that a peer system appeared to be dead, but that it was in fact certainly dead. There would not be any need to wait for it to come back to life of its own accord, nor for manual intervention to resolve the problems the system might be having. If a system is known to be dead, there is no need for policies and intervals and the like. The course is certain. Remove it from the sysplex.



CPC – Central Processor Complex containing images (LPARs)

SE – Support Element

HMC – Hardware Management Console

The Base Control Program internal interface (BCPii) allows authorized z/OS applications to have HMC-like control over systems in the process control HMC network. Note that there is complete communication isolation of existing networks (internet/intranet) from the process control (HMC) network, and communication with the System z support element is completely within base z/OS. BCPii provides a set of authorized APIs to enable communications between z/OS applications and the local support element, as well as between other support elements connected to other CPCs routed by the HMC. The BCPii query services can provide information about the operational state of any CPC connected to the HMC network, as well as the operational state of any image on the CPC.

As each z/OS image IPLs into the sysplex, XCF sets an IPL token in the hardware to uniquely identify the image. The IPL token is also published in the sysplex couple data set so that each system in the sysplex can ascertain the IPL token for every other system. If a system appears to be unresponsive, XCF uses BCPii query services to inquire as to the state of the subject system. If the system is down, it will be removed from the sysplex. As needed, XCF will use BCPii services to reset the system. For example, a system reset might be needed to ensure that the system has been successfully isolated from the sysplex. The IPL token is used when doing such resets, as it ensures that the reset is applied to the intended instance of the system image.

Note: The SE must be operational in order for it to detect, report, and reset an image. If the SE is down, BCPii will not be able to provide information to XCF. If XCF is unable to determine that the image is down, XCF and SFM proceed as in the past. In particular, note that an Emergency Power Off (EPO) of the CEC shuts down the SE.

z/OS 1.11 SFM with BCPii



- Expedient removal of unresponsive or failed systems is essential to high availability in sysplex
- XCF exploits BCPii services to:
 - Detect failed systems
 - Reset systems
- Benefits:
 - Improved availability by reducing duration of sympathy sickness
 - Eliminate manual intervention in more cases
 - Potentially prevent human error that can cause data corruption

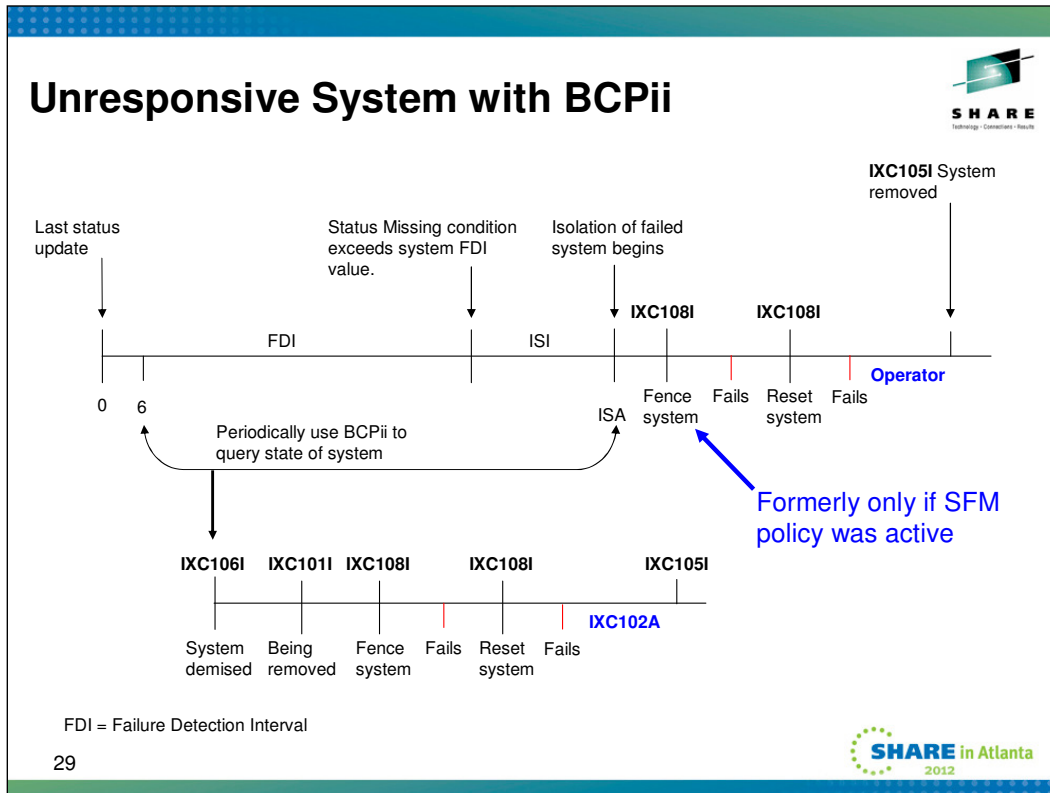
28



The sysplex failure management (SFM) component of XCF, which is used to manage failures and reconfiguration in the sysplex, has been enhanced in z/OS V1.11. It is now designed to use new Base Control Program internal interface (BCPii) services to determine whether an unresponsive system has failed, expedite sysplex recovery by bypassing delay intervals when possible, and automatically reset failed systems without manual intervention. This function allows SFM to avoid waiting for a period of time before assuming that systems have failed, improves the responsiveness of failure management, avoids operator intervention, and helps limit or avoid sysplex-wide slowdowns that can result from single-system failures.

The Base Control Program Internal Interface (BCPii) component of z/OS provides a set of programming interfaces to allow authorized programs to perform Hardware Management Console (HMC) functions for System z servers within an attached HMC network. These operations include obtaining information about servers and images (LPARs), issuing commands for certain hardware and software-related functions, and listening for certain hardware and software events. BCPii communication to HMCs and Support Elements (SEs) uses internal communication protocols and does not require communication on an IP network. Therefore, it is isolated from other network traffic.

Through the use of BCPii, XCF can detect that a system has entered a non-restartable wait-state, or that it has been re-IPLed. XCF can also perform a system reset on other systems in the sysplex. Thus XCF now has the ability to ascertain with certainty that a system is no longer operational. With this certain knowledge, XCF can ensure that the system is safely isolated from shared resources and remove the failed system from the sysplex – all without operator involvement. Furthermore, since XCF need not wait for the system failure detection interval to expire (to conclude that the system has no signs of life), the isolation of the failed system can occur sooner, which in turn reduces the amount of time that other systems in the sysplex will experience sympathy sickness.



This timeline depicts the key events that occur when a system fails to update its status and BCPii services are usable. Some system stops updating its status in the sysplex CDS and stops sending signals (we'll just say "not updating status"). Along the top of the timeline, we see that if the condition persists longer than the FDI for the system, other systems will wait for the Indeterminate Status Interval (ISI) to expire, and then some system will take the Indeterminate Status Action (ISA) to remove the system from the sysplex. The top timeline depicts attempts to fence and reset the system as needed. In the past, fencing was attempted only when an SFM policy was active. The reset via BCPii is new.

Along the bottom of the timeline we see that after 6 seconds, the systems that have BCPii capability will use it to determine whether the system is demised. A system image is **demised** if it can be immediately removed from the sysplex because it is in a non-restartable disabled wait state, underwent a LOAD operation, was RESET, or experienced any other equivalent action (e.g. system reset, checkstop, power-down, etc.). If the system is not known to be demised, then so long as it fails to update its status, XCF continues to use BCPii services every 3 seconds to ask about the state of the system. In the diagram, the curved arrow is an attempt to depict these periodic queries, which continue until the system updates its status, is found to be demised, or is otherwise removed from the sysplex. If the system is truly down, we normally expect its demise to be detected quickly.

If the system is demised, message IXC106I is issued to so indicate and partitioning is immediately initiated (message IXC101I). In some cases, the status reported by BCPii allows XCF to conclude that the image has been safely isolated from shared resources. If not, an attempt is made to fence the system (message IXC108I). If the system cannot be fenced, an attempt is made to reset the system using BCPii services (message IXC108I). If the system cannot be reset, message IXC102A is issued to have the operator take action. More likely, the system is successfully isolated and message IXC105I is issued to indicate that it was removed from the sysplex.

z/OS 1.11 SFM with BCPii



- With BCPii, XCF can know system is dead, and:
 - Bypass the Failure Detection Interval (FDI)
 - Bypass the Indeterminate Status Interval (ISI)
 - Bypass the cleanup interval
 - Reset the system even if fencing fails
 - Avoid IXC102A, IXC402D and IXC426D manual intervention
 - Validate “down” to help avoid data corruption

Helps improve availability

30



Unresponsive systems must be partitioned from the sysplex in a timely manner to avoid sympathy sickness. But, without BCPii, XCF does not really *know* a system’s operational state. At best, systems can monitor each other for signs of activity (updates to the sysplex couple data set, signals being exchanged). When the monitored activity stops, XCF waits the Failure Detection Interval (FDI) to try to avoid falsely removing an operational system. One would not want to remove a system that was suffering a temporary problem. But the penalty for this caution is that the sysplex may suffer workload processing interruptions/delays if the system has truly failed.

A partitioned system must be isolated from the rest of the sysplex to avoid corruption of shared data. In a parallel sysplex, XCF relies on CF Isolate command to fence a system from the channel subsystem. If the installation does not have a CF, or if the isolation fails, manual system reset is required. Manual intervention elongates the partitioning process, which elongates sympathy sickness.

But with BCPii services, XCF can now detect that a system has failed, and if so, whether it has been reset or otherwise isolated from shared resources. With this certain knowledge, XCF can safely remove a failed system from the sysplex without waiting for the failure detection interval to expire. If the failed system has not been isolated, XCF need not wait for the indeterminate isolation interval (ISI) to expire before it attempts to fence the failed system. If the fencing fails (or cannot be performed due to a lack of a CF), XCF can use BCPii services to appropriately reset the failed system. In cases where the failed system has been appropriately reset, XCF need not perform any isolation actions.

If the operator issues `VARY XCF,sysname,OFFLINE` to remove an active system from the sysplex, XCF will normally give the system time to do cleanup processing before entering a wait-state. But if the system is already down, this CLEANUP interval (which is specified in the COUPLExx parmlib member) is irrelevant. In the past, the system on which the VARY XCF command was issued would not be able to tell whether the subject system was still active (and thus entitled to its cleanup time), or already down (and cleanup is not needed). Through the use of BCPii services, the sysplex can detect that the system is down and bypass the CLEANUP interval.

If an operator (or automation) replies “DOWN”, XCF will use BCPii services to inquire as to the state of the system. If BCPii confirms that the system has not been reset, the prompt will be re-issued. When it works, this check will help prevent data integrity problems that can arise if the system is not truly isolated from shared resources. But if BCPii provides no answer, the system proceeds with partitioning regardless – so not all danger is eliminated.

Thus the certain knowledge that a system has failed enables XCF to immediately partition failed systems from the sysplex, all without operator intervention.

z/OS 1.11 SFM with BCPii



- SFM will automatically exploit BCPii and as soon as the required configuration is established:
 - Pairs of systems running z/OS 1.11 or later
 - BCPii configured, installed, and available
 - XCF has security authorization to access BCPii defined FACILITY class resources
 - z10 GA2 with appropriate MCL's, or z196, or z114
 - **New version of sysplex CDS is primary in sysplex**
 - Toleration APAR OA26037 for z/OS 1.9 and 1.10
 - Does NOT allow systems to use new SSD function or protocols

May need
MCL
Fixes !

31



- Refer to the “BCPii Setup and Installation” topic in *MVS Programming: Callable Services for High Level Languages* for information on installation and configuration steps and SAF authorization requirements to enable BCPii to invoke z/Series Hardware APIs.
- See topic "Assigning the RACF TRUSTED attribute" in *MVS Initialization and Tuning Reference* for information on using RACF to assign the TRUSTED attribute to the XCF address space.
- A system running z/OS V1R11 on down-level hardware is only eligible to target other systems that are enabled to exploit the full functionality of the System Status Detection (SSD) partitioning protocol. A system not running on the requisite hardware can not be the target of SSD partitioning protocol functions.
- A new version of the sysplex couple data set (CDS) is required. It has bigger records to make room for the tokens that uniquely identify the z/OS images in the sysplex. Install toleration PTFs for OA26037 on V1R10 and V1R9 systems in the sysplex to use the newly formatted sysplex couple data set required by the protocol.
- By default, the SYSSTATDETECT function is enabled in V1R11. The current setting of the SYSSTATDETECT function can be determined by issuing a DISPLAY XCF,COUPLE command. SYSSTATDETECT is the name of the XCF FUNCTIONS switch that enables or disables use of SSD. For more information on enabling or disabling the SYSSTATDETECT function in V1R11, see *MVS Initialization and Tuning Reference* for information on specifying SYSSTATDETECT in the COUPLExx parmlib member and *MVS System Commands* for information on enabling and disabling the SYSSTATDETECT function via the SETXCF FUNCTIONS command

“Sick But Not Dead” Refinements



- For cases where it can be known that system is dead, the FDI, ISI, and SSUMLIMIT intervals are irrelevant
- Remain relevant for “sick but not dead” cases
 - Including cases where BCPii cannot ascertain the state of the system
- We will now explore additional refinements that:
 - Reduce “needless” removal
 - Improve “needed” removal

32



With BCPii, XCF can know with certainty that a system is dead. In these cases, the various intervals that delay SFM actions until it becomes highly likely that the system is dead are no longer relevant. With BCPii, a large class of problems can now be automatically resolved.

However, there could be cases where BCPii is unable to provide the required information. Or more likely perhaps, there could be “sick but not dead” cases where BCPii reports that the system is not dead, yet the XCF status monitor detects that the system is unresponsive. In these cases, the intervals remain relevant. Presumably these “sick but not dead” problems will get more attention now that BCPii provides a mechanism for resolving the cases where the system is in fact dead (the most prevalent case).

So in the remainder of the presentation, we will look at some of the support that SFM provides for dealing with the sympathy sickness that can arise from “sick but not dead” situations. But first we briefly touch on a case of “needless removal” that led to some reworking of the FDI.

Refinement to Avoid “Needless” Removal



- FDI needs to be short enough to recognize unresponsive systems before sympathy sickness gets too severe
- Yet long enough to allow the system to overcome “normal” stalls and hangs
- Historically, $FDI = 2 * spintime + 5$
 - Want to allow time for system to recover from an excessive spin condition
 - “2” worked pretty well since first action of ABEND was usually sufficient to break out of the spin
 - But not always ...

33



The system FDI is a rather critical parameter. If too short, systems might be removed from the sysplex for recoverable problems. If too long, sympathy sickness may become excessive because the failing system is not removed quickly enough.

Historically, it was suggested that the FDI be set to 5 seconds more than twice the excessive spin loop timeout value (as defined in the EXSPATxx parmlib member). The default spin loop timeout is 10 seconds if z/OS is running on a machine with dedicated processors, and 40 seconds if running on a machine with shared processors (which is the vast majority of installations). The idea behind this suggestion was that a spin loop might make a system appear to be unresponsive. But since that is a recoverable situation, removal of a system before spin loop recovery had acted would be a “needless” removal. It takes one spin loop (either 10 or 40 seconds) to recognize that a problem might exist. The system then allows another interval to expire to confirm that the problem appears to be persistent. When that second interval expires, the first spin loop recovery action is taken. By default, the recovery actions are ABEND, TERM, and ACR. In many cases, the initial ABEND is enough to break out of the loop. Thus $2 * spintime + 5$ was effective for a long time. But it was proven in the real world that there are cases where the entire gamut of spin recovery actions was needed to break out of the spin loop. Failure to allow for all of those actions to transpire before removing the system was a “needless” removal. So the suggested formula for setting FDI was changed.

z/OS 1.11 XCF FDI Consistency




- Enforces consistency between the system Failure Detection Interval (FDI) and the excessive spin parameters
 - $FDI = (N+1) * spintime + 5$
- Allows system to perform full range of spin recovery actions before it gets removed from the sysplex
- Avoids false removal of system for a recoverable situation

34



In z/OS V1.11, XCF automatically adjusts the failure detection interval (FDI) when the excessive spin parameters are changed. The **spin FDI** is the FDI value that is obtained using the formula $(N+1) * spintime + 5$ where N is the number of spin recovery actions specified in the EXSPATxx parmlib member. The **user FDI** is the FDI value explicitly set by the installation. The **effective FDI**, which is the FDI value actually used by the system, will either be the spin FDI or the user FDI, whichever is greater. The idea is that the effective FDI should be greater than or equal to the spin FDI to ensure that the system has a chance to complete the entire gamut of spin recovery actions before it gets removed from the sysplex because it appears to be unresponsive.

z/OS 1.11 XCF FDI Consistency ...



```

D XCF,C
IXC357I 15.12.46 DISPLAY XCF          E  SYS=D13ID71
SYSTEM D13ID71 DATA
  INTERVAL  OPNOTIFY  MAXMSG  CLEANUP  RETRY  CLASSLEN
    165      170      3000    60        10     956

  SSUM ACTION  SSUM INTERVAL  SSUM LIMIT  WEIGHT  MEMSTALLTIME
    PROMPT          165          N/A         N/A     N/A

  PARMLIB USER INTERVAL:    60
  DERIVED SPIN INTERVAL:    165
  SETXCF  USER OPNOTIFY: +  5

< - - - snip - - - >
OPTIONAL FUNCTION STATUS:
FUNCTION NAME      STATUS      DEFAULT
DUPLXCF16         ENABLED    DISABLED
SYSSTATDETECT     ENABLED    ENABLED
USERINTERVAL      DISABLED   DISABLED
    
```

Effective Values

User FDI
Spin FDI
User OpNotify
 - Absolute
 - Relative

Switch

This slide shows relevant output from the DISPLAY XCF,COUPLE command. The effective FDI and OpNotify value are reported as in past releases. A new section reports user specified FDI, derived spin FDI, and user specified OpNotify value as well as the source from which the current value was derived (COUPLExx parmlib, SETXCF command, system default). The FUNCTION section now reports the state of the new USERINTERVAL switch.

The **Failure Detection Interval (FDI)**, is the amount of time that a system can appear to be unresponsive before XCF is to take action to resolve the problem. The **User FDI** is an FDI value explicitly specified by the user, either directly (via COUPLExx parmlib member or SETXCF COUPLE command) or indirectly (through cluster services interfaces – IXCCROS macro). The **Spin FDI** is an FDI value derived by XCF from the excessive spin parameters (spin loop timeout value and the number of excessive spin actions (such as SPIN, TERM, ACR)). The **Effective FDI** is the FDI value that is being used for the system. If the USERINTERVAL switch (FUNCTIONS) is DISABLED, the effective FDI = max(user FDI, spin FDI). If USERINTERVAL is ENABLED, the effective FDI = user FDI. By default, the effective FDI will be the larger of the user FDI and the spin FDI. If the installation really wants the smaller user FDI to be the effective FDI, the USERINTERVAL switch must be ENABLED to force XCF to use it.

User OpNotify is the operator notification interval explicitly specified by the user, either directly (via COUPLExx parmlib member or SETXCF COUPLE command). OpNotify determines when XCF should alert the operator about a system that is unresponsive. OpNotify can now be **relative** to the effective FDI (ie, a delta). In the past, it was always an **absolute** that had to be greater than or equal to the (effective) FDI. So if one wanted to change one of the values, one might in fact have to change them both (and in a particular order) so as to maintain the required relationship. With a relative OpNotify value, the system automatically maintains the relationship. If the effective FDI changes, the effective OpNotify value changes as well. The **Effective OpNotify** is the OpNotify value being used by the system. The system ensures that effective FDI is always less than or equal to the effective OpNotify value.

If the installation changes the excessive spin parameters, sets a new user FDI value, or changes the USERINTERVAL switch, the effective values are recomputed and then written to the sysplex CDS to make them visible to the rest of the sysplex (via status update processing).

Is Spin FDI too long?

- If system is truly dead
 - If detected via BCPii, FDI is irrelevant
 - If BCPii cannot ascertain, detection is elongated
- If system is sick but not dead
 - No status updates, sending signals
 - SSUMLIMIT is key, FDI is “irrelevant”
 - No status updates, not sending signals
 - For spin loops, spin FDI is the desired value
 - If not spin loop, detection is elongated

Even without BCPii, likely OK. But watch and adjust as needed or if concerned. Once BCPii set up, should be rare.

SSUMLIMIT is tens of minutes. Dominates time to resolution

Seldom goes beyond ABEND

Probably Rare

36

SHARE in Atlanta 2012

Making the FDI be consistent with the excessive spin recovery parameters generally causes the effective FDI to be 165 seconds instead of the 85 seconds that most installations are used to seeing. Many are concerned that this larger value will increase their risk of suffering from sympathy sickness conditions. Point taken. However, the appropriate setting of FDI is a balancing act between values that lead to “needless” removal and values that allow sympathy sickness to become too severe. Let us consider some scenarios.

Often a system becomes unresponsive is because it is dead. In this case, the expectation is that XCF will detect that the system is dead via BCPii services and initiate partitioning long before the FDI (old or new) expires. So it does not matter whether the FDI is longer than in the past. If BCPii is not available, or is otherwise unable to determine the state of the system, then yes the longer FDI might be an issue, but only if the policy allows the system to take automatic action. If the policy specification is PROMPT, then the fact that manual intervention is required suggests that the installation is willing to tolerate elongated resolution. For the case of automatic action, I suggest (1) getting the SFM with BCPii support enabled as soon as possible, and (2) in the mean time, run with the new spin FDI as the default (after duly considering the discussion on smaller FDI values below).

If the system is not actually dead, we have a “sick but not dead” situation. For FDI to apply, it must be the case that the system is not updating status in the sysplex CDS. Then it must either be the case that the system is or is not sending signals. If the system is sending signals, I claim that the longer FDI is immaterial. This is the case where SSUMLIMIT applies. Since SSUMLIMIT should be on the order of 15 minutes or more, and past experience suggests that a sysplex can tolerate this state for quite a while, an extra 80 seconds before initiation of the SSUMLIMIT timer is not particularly worrisome. If SSUMLIMIT is not coded, this case reverts to manual intervention. If the system is neither updating status nor sending signals, a spin loop is probably the root cause. Past experience suggests that the system breaks out of the spin loop after the first spin recovery action is taken, in which case the longer FDI is not material because the system resumes normal operation. If it does not resume normal operation after the first action, then this is precisely the case for which we want spin recovery to complete its full course. If the system is not in a spin loop, we would seem to have a case where some sort of situation is preventing XCF from doing any work. It must be fairly serious since XCF should be running at system priority. Further, it would seem that none of the XCF interrupt routines or timer DIE's are running. This is probably rare.

The choice of an FDI is a balancing act between “needless” removal and “should have acted sooner”. The longer spin FDI does create the potential for longer sympathy sickness conditions, with the possible benefit of avoiding a “needless” removal of a system that might have recovered. Using the shorter FDI increases the likelihood of a “needless” removal, with the benefit of a shorter sympathy sickness condition. You must determine what is best for your shop.

Refinements for “Sick But Not Dead”



- Signalling Sympathy Sickness
- Unresponsive Critical Members
- Unresponsive CF structure connectors

37

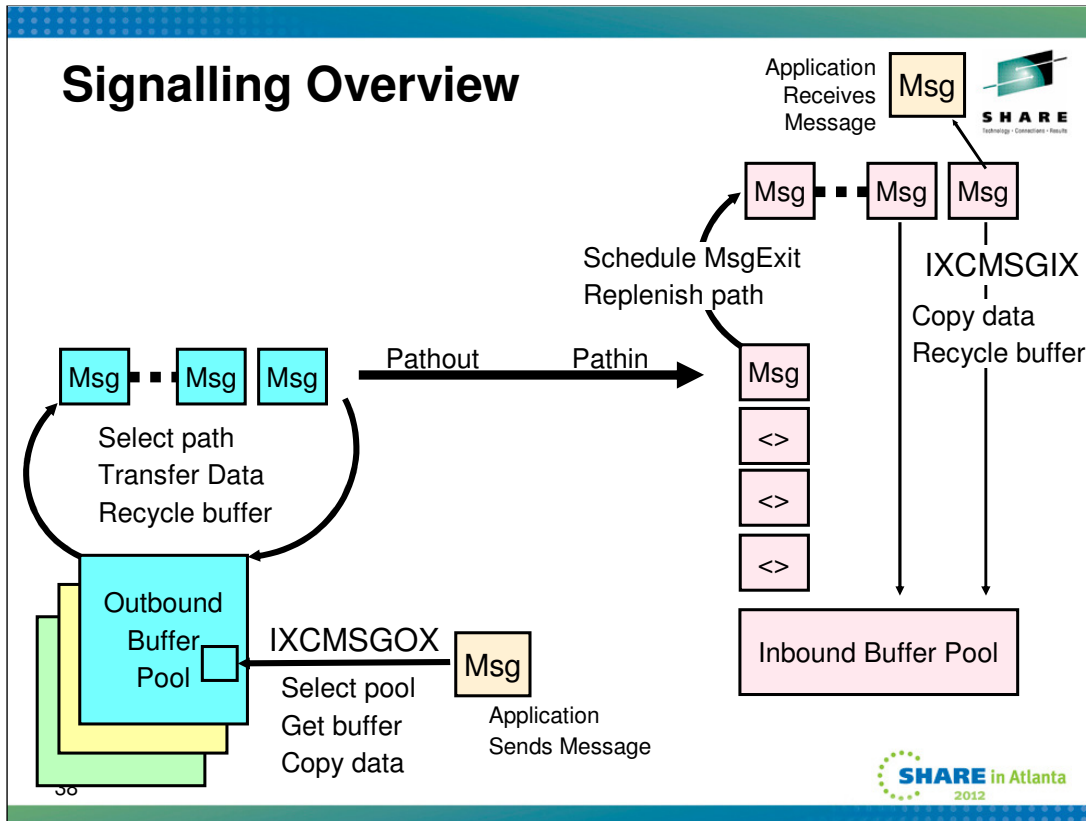


Over the years, SFM has been enhanced to deal with various conditions for which sysplex applications are experiencing problems that can lead to sympathy sickness. The systems themselves are not unresponsive, but the failure of the application to make progress leads to problems for others.

Signalling sympathy sickness occurs when a stalled XCF group member fails to process its signals in a timely manner. In the worst case, the stalled member consumes inbound signal buffers, which prevents signal transfers across signalling paths, which prevents other members from being able to receive signals.

Critical members is the term used for an XCF group member that performs services that are critical to the normal operation of the system and by extension, sysplex. GRS is an example. If the critical member becomes unresponsive, it is unable to provide its services and those that need the services suffer.

Connectors to a coupling facility structure are required to participate in various sysplex wide protocols. If a connector fails to provide the expected responses, the protocol hangs. Other connectors suffer as a result.



This diagram illustrates the key steps in delivering a message between two different systems in the sysplex.

At the bottom center of the chart, an authorized application invokes the XCF Message-out service (either the IXCMGOX or the older IXCMGO macro) to initiate the send of a message. The message out service receives control via a stacking, space-switch PC. The service routine obtains a message buffer from a suitable buffer pool, copies the user message data into the buffer, then queues the signal for transfer over an outbound signalling path (we assume inter-system communication for this example). In most cases, I/O transfer for the signal is initiated when the signal is queued to the path for transfer. Upon successful transfer of the signal to the target system, the outbound buffer is recycled whereupon it becomes available for use by some other send request.

On the target system, the message is received via the inbound path that is connected to the outbound path on the sending system. The message is received into an inbound message buffer. The message is scheduled to the target member's address space. In some cases, perhaps due to volume or perhaps due to the attributes of the signal, the message will be queued for delivery. In either case, as soon as the inbound signalling path dispenses with the signal, it replenishes the path. The specifics of this replenishment vary according to the type of path (CTC or list structure), but the intent is the same. Namely, to provide another inbound buffer with which to receive the next incoming message (if any).

When the SRB is dispatched, an XCF stub routine is given control, which in turn, calls the message exit routine provided by the target member when it joined the group. The message exit routine may or may not invoke the XCF Message-in service (macro IXCMGIX or the older IXCMGI) to receive the message data. If the exit routine does invoke the message-in service, the XCF Message-in service routine receives control via a stacking, space switch PC. The service copies the message data from the XCF message buffer into storage designated by the member. The message buffer is then recycled, whereupon it can be used to receive the next message. If the message exit routine does not receive the message, the message buffer is recycled by XCF when the exit routine returns to the XCF stub.

After the message exit routine returns to XCF, the XCF stub routine looks to see if there are any messages queued for delivery to the member. If so, the stub routine plucks the next message from the queue and calls the message exit routine to process the message. The XCF stub routine will give up control if there are no messages queued for delivery, or if the number of messages it has presented to the message exit routine exceeds some threshold (since failure to give up control could prevent any other work from being processed in the address space). If there are still messages queued for delivery when the SRB ends, a new instance of the SRB will be soon be scheduled to continue processing the queue.

z/OS 1.8 Signalling Sympathy Sickness



- XCF detects and surfaces inter-system signalling sympathy sickness caused by stalled group member(s)
- SFM policy MEMSTALLTIME specification determines how long XCF should wait before taking action to resolve the problem
- After expiration, the stalled member is terminated
 - For GRS, XCF, or Consoles, implies system termination
- Provides a backstop that can take automatic action in case your automation or manual procedures fail to resolve the issue

39




Each pair of systems in the sysplex cooperate to detect signalling sympathy sickness caused by XCF group members that fail to process signals in a timely fashion. Sysplex Failure Management (SFM) for Stalled Members allows the system to automatically terminate such members to alleviate the sympathy sickness. Signalling sympathy sickness occurs when a system has signals to send, but signal traffic has stopped flowing across the signalling paths because the target system has no I/O buffers available. Furthermore, the I/O buffers are unavailable because they contain messages that have yet to be delivered to an XCF group member that is not processing its signals in a timely manner.


If signalling sympathy sickness is detected, both systems issue messages to identify the problem. Information about the stalled member and the sympathy sickness impact can be obtained from either system by issuing appropriate DISPLAY XCF commands. If the SFM policy indicates that automatic action is to be taken to resolve the problem (MEMSTALLTIME parameter is not "NO"), the system will terminate the stalled member. Terminating the stalled member allows XCF to reclaim signalling resources being consumed by the member, which in turn allows signal transfers to resume, which in turn eliminates the signalling sympathy sickness. Even if automatic action is not enabled, manual resolution of the problem should be less error prone because the sympathy sickness problem and the culprit are clearly identified.

The MEMSTALLTIME parameter determines whether a system can take automatic action to alleviate the sympathy sickness, and if so how quickly. In general, it is recommended that some time interval be specified for MEMSTALLTIME so that there is always an automatic "backstop" to take action in case manual intervention does not (or cannot) resolve the problem. The MEMSTALLTIME specification will generally be the amount of time the installation wants to allow for manual intervention (or automation) to resolve the problem.

Signalling Sympathy Sickness Indicators



Impacted System	Culprit System
<ul style="list-style-type: none"> • D XCF,G... shows stalls • IXC467I Restart stalled I/O <p style="text-align: center; color: blue;">Stalled Members</p>	<ul style="list-style-type: none"> • D XCF,G... shows stalls • IXC431I member stalled • ABEND 00C 020F0006 • IXC430E stalled members
<ul style="list-style-type: none"> • IXC440E impacted <p style="text-align: center; color: blue;">Sympathy Sickness</p>	<ul style="list-style-type: none"> • IXC631I mem causing SS • IXC640E if/when to act • ABEND 00C 020F000C
<p style="color: blue;">If SFM allowed to take action</p>	<ul style="list-style-type: none"> • ABEND 00C 020F000D • IXC615I terminating <ul style="list-style-type: none"> – ABEND 00C 00000160 – Wait State 0A2 rsn 160

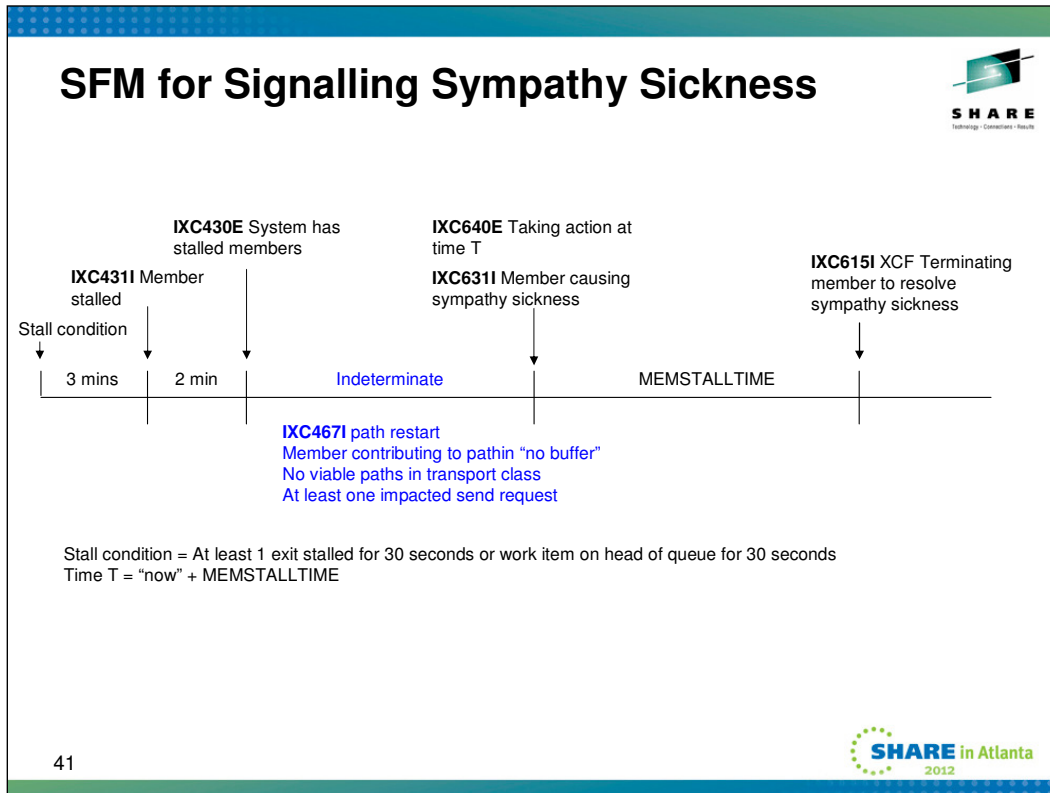
40


This slide summarizes the various external indications that will appear as XCF detects and reacts to a signalling sympathy sickness problem. The left column shows the indicators on the impacted system, the right column shows the indicators on the culprit system.

The indicators in the “**Stalled Members**” box existed prior to z/OS 1.8. The various flavors of DISPLAY XCF, GROUP highlight stalls with an asterisk after roughly 30 seconds. Roughly 3 minutes after the stall starts, hardcopy message IXC431I is issued. The stalled member now becomes eligible for consideration as a sympathy sickness culprit. Roughly 15 seconds later, XCF issues ABEND 00C rsn 020F0006 to cut a logrec entry and verify internal XCF control structures. Roughly 5 minutes after the stall, message IXC430E is issued to alert the operator. Signal path restarts for stalled I/O could be occurring throughout.

If the impacted and culprit system are both running with this support, the **Sympathy Sickness** box shows the external indicators that might appear. The timing for detection of signalling sympathy sickness between systems depends on the dynamics of the sysplex. When detected, hardcopy message IXC631I indicates the member is causing signalling sympathy sickness. Message IXC640E is issued by the culprit system to alert the operator and indicate the intended resolution. Message IXC440E is issued by the impacted system. XCF issues ABEND 00C rsn 020F000C to cut a logrec entry and verify internal XCF control structures.

If the SFM Policy MEMSTALLTIME specification allows the culprit system to **take action**, then roughly 30 seconds before doing so XCF issues ABEND 00C rsn 020F000D to cut a logrec entry, verify internal XCF control structures, and initiate an SDUMP. When the MEMSTALLTIME interval expires, message IXC615I indicates which member was selected for termination. Most members are terminated with ABEND 00C rsn 160. For Global Resource Serialization (SYSGRS), Consoles (SYSMCS), and XCF itself (SYSXCF), the system is removed from the sysplex, resulting in wait-state 0A2 rsn 160.



This chart summarizes how XCF deals with stalled members that are impeding inter-system signal traffic.

XCF recognizes a "stall" condition if a signal exit routine is unresponsive for 30 seconds, or if a signal remains queued for the signal exit for more than 30 seconds. Since there are various "normal" conditions that can interfere with the member's ability to process signals (dumps for example), XCF does not immediately raise an alert. If one happens to issue DISPLAY XCF, GROUP commands however, the stall conditions would appear.

If the stall persists for an additional 3 minutes (roughly), XCF issues message IXC431I to the hardcopy log to document the problem. Some 2 minutes later, assuming the condition persists, XCF issues message IXC430E to alert the operator to a potential problem. At this point one might have the operator issue DISPLAY XCF, GROUP commands to determine more information about the stalled member. One might then use this information to guide further diagnosis. Note that message IXC431I may be issued periodically to provide updated information about the stall condition.

The stall condition may or may not impact XCF's ability to perform signal transfer. For sure, XCF resources are tied up. If in fact, XCF signal buffers are being consumed, the stalled member may prevent XCF from being able to receive signals via the inbound signalling paths. If so, signal transfers may stall. Eventually the signalling path might be restarted for a "stalled I/O" condition. The path restart process will re-establish the signal path. However, XCF will recognize that the stalled I/O condition was the result of a stalled member on the target system consuming XCF signal buffers. If the sending system confirms that there has been an impact, namely, some user could not get a signal sent because all the eligible paths are stalled, XCF declares there to be a signalling sympathy sickness condition. The sending system issues message IXC440E (not shown on the diagram). The target system issues message IXC631I to the hardcopy log to indicate that the stalled member is causing sympathy sickness, and message IXC640E to the console to alert the operator as to what will be done about the problem.

After issuing the IXC640E message, XCF waits MEMSTALLTIME seconds. If MEMSTALLTIME(NO) is in effect, XCF takes no action. If the member is still stalled at approximately MEMSTALLTIME minus 30 seconds, XCF issues abend 00C reason 020F000D and takes a dump for diagnosis. If at the end of the MEMSTALLTIME interval, the member is still stalled and causing signalling sympathy sickness, XCF terminates the member per the TERMLEVEL specification from the member's IXCJOIN invocation. XCF issues message IXC615I just before it initiates said termination.

z/OS 1.12 Critical Members



- A system may appear to be healthy with respect to XCF system status monitoring
 - Updating status in sysplex CDS and sending signals
- But is the system actually performing useful work?
- There may be critical functions that are non-operational
- Which in effect makes the system unusable, and perhaps induces sympathy sickness elsewhere in the sysplex
- Action should be taken to restore the system to normal operation

42



z/OS 1.12 extends XCF System Status monitoring to incorporate status information about critical components (such as GRS). Currently, a system is deemed unresponsive if it stops sending XCF signals and stops updating its status in the sysplex Couple Data Set (CDS). However, these indications of activity do not necessarily imply that a system is able to accomplish useful work. Indeed, an apparently active system could in effect be causing sympathy sickness because critical components are unable to accomplish their intended function. The goal of the “critical member” support is to resolve the sympathy sickness by expeditiously partitioning a sick system out of the sysplex whenever any critical XCF member on that system is deemed unresponsive. Though still not a perfect indicator of whether a system is performing useful work, the discovery of unresponsive critical components should provide an incremental improvement that helps the sysplex better identify (and remove) unresponsive systems.

z/OS 1.12 Critical Members ...



- Member Impairment
 - A member is **confirmed** to be impaired when its status exit indicates “status missing”
 - A member is **deemed** to be impaired if it is stalled with no signs of activity
- XCF now surfaces impairment for all members

43



An XCF group member may tell XCF that it is impaired via the member status exit routine. XCF shares this information with peer members that have a group exit routine, but prior to z/OS 1.12, did not take any overt action to surface the condition or mitigate it. With z/OS 1.12, XCF will (1) surface the fact that a member is impaired, and (2) take action to mitigate the problem if the member is identified as being “critical”. Surfacing the condition should make it easier to identify situations where an application may not be operating normally.

A monitored member is “confirmed impaired” if it indicates to XCF via its status exit that it is “status missing”. The member is “deemed impaired” if the member’s XCF exits appear to be stalled with no signs of activity.

Message IXC633I is issued to indicate that a member is impaired. It will be interesting to see how frequently these impairment conditions occur. Since they were never externalized in the past, no one really knows.

z/OS 1.12 Critical Members ...



- A Critical Member is a member of an XCF group that identifies itself as “critical” when joining its group
- If critical member is impaired long enough, XCF will eventually terminate the member
 - Per member’s specification: task, space, or system
 - MEMSTALLTIME determines “long enough”
- GRS is a “system critical member”

44




z/OS 1.12 extends XCF Member Status monitoring to take some form of action when a critical XCF group member appears to be non-operational. In particular, XCF will terminate the critical member if the impaired state persists long enough. Termination of the critical member should relieve the sympathy sickness condition and allow the application to resume normal operation. Alternatively, such termination may also make it possible for more timely restart of the application (or other appropriate recovery action) that can then lead to full recovery. The application determines whether it is “critical” and if so, the means by which it should be terminated. Said termination could entail termination of the member’s task, address space, or system. If the system is to be terminated, the member is presumed to be “system critical”. GRS is “system critical”.

If a critical member remains continuously impaired for as long as the system failure detection interval (FDI), XCF inspects the Sysplex Failure Manager (SFM) specification for the MEMSTALLTIME parameter. For MEMSTALLTIME(NO), XCF delays termination of the member for FDI seconds, or two minutes, whichever is longer. If MEMSTALLTIME(nnn) is specified, XCF delays termination for the indicated number of seconds.


This function is intended to help reduce the incidence of sysplex-wide problems that can result from unresponsive critical components. GRS exploits these XCF critical member functions in both ring and star modes. GRS monitors its key tasks and notifies XCF if it detects that GRS is impaired.

z/OS 1.12 Critical Members ...



- Key Messages
 - IXC633I “member is impaired”
 - IXC634I “member no longer impaired”
 - **IXC635E “system has impaired members”**
 - IXC636I “impaired member impacting function”

45

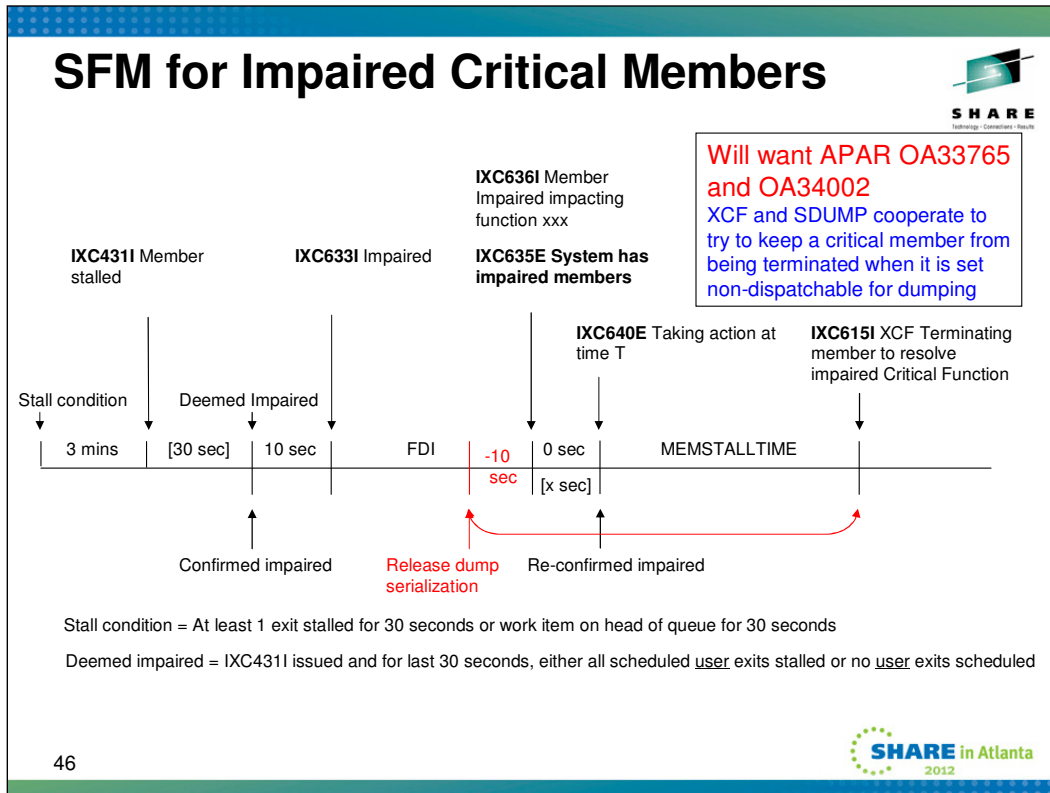


This slide summarizes the essence of the key messages related to impaired member processing.

Installations may choose to develop automation and/or operational procedures to deal with impaired member conditions. The Sysplex Failure Management (SFM) policy MEMSTALLTIME specification should be specified accordingly. For example, if an installation wants operators to investigate and resolve such problems, one will likely specify a longer MEMSTALLTIME value to allow time for such actions to occur. MEMSTALLTIME in effect serves as a back stop to allow the system to take automatic action to alleviate the problem if the operations personnel are unable to resolve the problem in a timely manner. Of course one should recognize that the higher the MEMSTALLTIME value the longer the potential sympathy sickness impact on other systems in the sysplex will persist.

Message IXC635E is likely the key message that would be used to trigger the relevant automation and/or operational procedures.

The sysplex partitioning messages (IXC101I “system being removed”, IXC105I “system has been removed”, and IXC220W “XCF wait-stated system”) have new inserts to indicate that the system was removed as the result of terminating an impaired member. The XCF wait-state code 0A2 has a new reason code (x194) to indicate this condition as well.



This chart summarizes how XCF deals with member impairment conditions. It combines both “stalled member” monitoring and “member impairment” monitoring as both monitors need to interact in an appropriate fashion. For clarity, messages IXC430E, IXC440E, IXC631I, IXC632I, and IXC640E as they relate to signalling sympathy sickness conditions are omitted.

A member can be “impaired” either because XCF “deems” it to be so, or because the member “confirms” itself to be impaired.

Prior to z/OS 1.12, XCF monitoring detected “stall conditions” wherein one or more exit routines (group or signal) have not made progress, or work items pending for these exits have not been processed. With z/OS 1.12, the monitor is extended to consider the member status exit routine as well. If the stall condition persists long enough (approximately 3 minutes), XCF issues message IXC431I to document the “stalled member”. Once the stall condition is externalized, the monitor looks to see if the member should be deemed impaired. To qualify, XCF must find no signs of “user activity” for a 30 second period. Thus, each time a new user exit is scheduled and/or each time XCF observes an exit make progress, XCF restarts the “no user activity” timer. If the 30 second timer expires, the member is **deemed to be impaired**.

A member is “**confirmed impaired**” if its status exit routine reports that the member is “status missing”. If a member confirms itself to be impaired, the state of the exit routines is (mostly) irrelevant. The member is responsible for determining its own status correctly. If the member indicates that it is impaired, then XCF assumes it must be so.

When the member is deemed impaired or confirmed impaired, XCF continues to observe the member. If the impairment condition persists for 10 observations (seconds), XCF issues message IXC633I to the hardcopy log to indicate that the member is impaired. If the impairment condition persists for the system failure detection interval (FDI), XCF issues message IXC636I to the hardcopy log to document the impaired member and the function that it is being impacted. This message also indicates whether the member is “critical” and thus subject to being terminated. XCF also issues message IXC635E to the console to alert the operator. At this point one might have the operator issue DISPLAY XCF, GROUP commands to determine more information about the impaired member. One might then use this information to guide further diagnosis.

After issuing IXC636I for the member, XCF determines whether the impaired member is critical, and if so, whether the member is “deemed impaired” or “confirmed impaired”. If “confirmed impaired”, XCF redrives the status exit routine to have the member confirm its status one more time. If the critical member confirms “status missing”, or if the member is “deemed impaired”, XCF issues IXC640E to the console to indicate that impaired members are impacting the sysplex. The message indicates when XCF (SFM) intends to take action to alleviate the situation.

After issuing the IXC640E message, XCF waits MEMSTALLTIME seconds. If MEMSTALLTIME(NO) is in effect, XCF waits FDI seconds or 120 seconds, whichever is greater. For brevity, we just say XCF waits MEMSTALLTIME seconds. If the member is still impaired at approximately MEMSTALLTIME minus 30 seconds, XCF issues abend 00C reason 020F000D and takes a dump for diagnosis. If at the end of the MEMSTALLTIME interval, the member is still impaired, XCF issues message IXC615I and terminates the member per the TERMLEVEL specification from the member’s IXCJOIN invocation.

With APAR OA33765 installed (and SDUMP APAR OA34002), approximately 10 seconds before issuing IXC636I and every second thereafter until the member resumes or is terminated, XCF inspects the ASSB of the critical member to determine whether its space has been set non-dispatchable for dumping. If so, XCF updates the ASSB to tell SDUMP to make the space dispatchable.

Unresponsive Structure Connectors



- Connectors to CF structures need to participate in various processes and respond to relevant events
- XES monitors the connectors to ensure that they are responding in a timely fashion
- If not, XES issues messages (IXL040E, IXL041E) to report the unresponsive connector
- Users of the structure may hang until the offending connector responds or is terminated

47



Applications that connect to a coupling facility structure are expected to participate in sysplex wide event protocols. In general, the Coupling Facility Resource Management (CFRM) subcomponent of the XES component of z/OS will present an event to the connector's event exit routine. For some events, the participating connector is expected to perform some application specific processing, and then provide a confirmation to indicate that it has finished processing the event. Failure to provide the expected confirmation means the process related to that event is hung (rebuild processing for example). Thus the peer connectors, and the applications that depend on the connector's function, may experience sympathy sickness. Thus failure to respond in a timely manner can lead to a sysplex wide hang.

The XES monitor that detects missing confirmations was introduced in OS/390 V1R8 (HBB6608). If an event needs confirmation, the system establishes a monitor to watch for the expected response and to report cases where it is not received in a timely manner. After 2 minutes without a response, XES issues message IXL040E or message IXL041E to identify the unresponsive connector, the associated structure, the event, and the affected process.

Operators often fail to react to these messages, thus allowing sympathy sickness to persist. Worse still, operators often react by terminating the wrong connector. Thus there is a need for the system to be able to automatically take corrective actions.

z/OS 1.12 CFSTRHANGTIME



- CFSTRHANGTIME
 - An SFM Policy specification to indicate how long the system should allow a structure hang condition to persist before taking corrective action(s) to remedy the situation
 - CFSTRHANGTIME(NO) is the default
- Corrective actions may include:
 - Stopping rebuild
 - Forcing the user to disconnect (signal structures only)
 - Terminating the connector task, address space, or system

48



In z/OS 1.12, the existing XCF/XES CF structure hang detect support is extended by providing a new CFSTRHANGTIME SFM Policy option that allows you to specify how long CF structure connectors may have outstanding responses. When the time is exceeded, SFM drives corrective actions to try to resolve the hang condition. This helps you avoid sysplex-wide problems that can result from a CF structure that is waiting for timely responses from CF structure connectors.

The interval specified with the CFSTRHANGTIME keyword begins after a hang is recognized, approximately 2 minutes after the delivery of the event that requires a response. CFSTRHANGTIME(0) means that the system is to take action immediately upon recognizing that the response is overdue. The initial suggestion, documented by way of the new XCF_SFM_CFSTRHANGTIME health check, was to set CFSTRHANGTIME at 5 minutes. This was soon changed to 15 minutes instead. This allows time for the installation to evaluate the situation and decide whether to take manual action, and possibly allow the hang to clear spontaneously, while still preventing the hang from persisting so long as to cause sysplex-wide problems. For some connectors, the time needed to process an event can vary due to a variety of application specific factors. Thus some installations may need to adjust the CFSTRHANGTIME value to accommodate their particular workload.

Initial action is taken at expiration of CFSTRHANGTIME interval. If hang persists, escalates to more aggressive actions. The escalation hierarchy begins with the least disruptive actions and progresses to more disruptive actions. Actions include: stop rebuild, stop signaling path (XCF signaling only), force disconnect (XCF signaling only), terminate connector task, terminate connector address space, partition connector system. Each system acts against its own connectors (no system will take action against any other system). No attempt to evaluate causal relationships between multiple events is made.

z/OS 1.12 CFSTRHANGTIME ...



Messages

```
IXL049E HANG RESOLUTION ACTION FOR CONNECTOR NAME: conname
TO STRUCTURE strname, JOBNAME: jobname, ASID: asid:
actiontext
```

```
IXL050I CONNECTOR NAME: conname TO STRUCTURE strname,
JOBNAME: jobname, ASID: asid
HAS NOT PROVIDED A REQUIRED RESPONSE AFTER noresponsetime
SECONDS.
TERMINATING termtarget TO RELIEVE THE HANG.
```

49



If no SFM policy is active or the policy specifies CFSTRHANGTIME(NO), no hang relief action is taken. If a policy is started, stopped, or changed, the monitor will re-evaluate the required response and possibly reissue IXL049E.

IXL049E may indicate that:

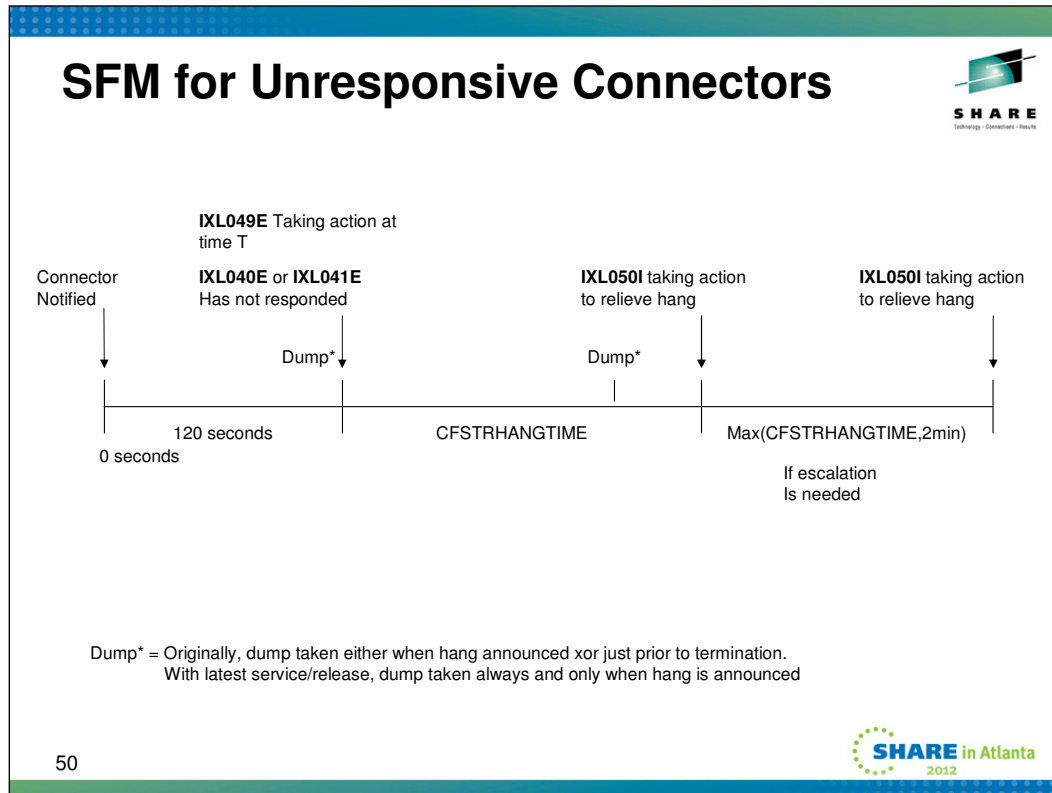
- (1) The system will not take action because (a) there is no SFM policy, (b) the SFM policy requires manual intervention, or (c) the system has tried everything it knows how to do, or
- (2) The system is taking action now (either because the policy specified CFSTRHANGTIME(0) or it was changed in a way that makes the action past due), or
- (3) The system will take action at the time specified in the message.

For IXL049E, *actiontext* is one of:

```
SFM POLICY NOT ACTIVE, MANUAL INTERVENTION REQUIRED.
SFM POLICY REQUIRES MANUAL INTERVENTION.
SYSTEM IS TAKING ACTION.
SYSTEM WILL TAKE ACTION AT termdate termtime
SYSTEM ACTION UNSUCCESSFUL, MANUAL INTERVENTION REQUIRED
```

For IXL050I *termtarget* is one of:

```
REBUILD
SIGNAL PATHS (ATTEMPT 1)
SIGNAL PATHS (ATTEMPT 2)
SIGNAL PATHS (ATTEMPT 3)
CONNECTION
CONNECTOR TASK
CONNECTOR SPACE (WITH RECOVERY)
CONNECTOR SPACE (NO RECOVERY)
CONNECTOR SYSTEM
```



This chart depicts how XES deals with stalled connectors that are not providing responses in a timely manner.

The connector is notified of an event. If connector does not provide the expected response in roughly two minutes, message IXL040E or IXL041E is issued to indicate that the member has not provided the expected response. Message IXL049E is issued to indicate the means by which the problem is to be resolved, should it persist. If the SFM policy is not active, or the policy specifies (or defaults to) CFSTRHANGTIME(NO), then manual intervention is required (as in the past). If CFSTRHANGTIME is specified to allow automatic action, message IXL049E indicates when that action will occur.

If the CFSTRHANGTIME interval expires and the connector has still not responded, XES issues message IXL050I to indicate the action that is being taken to attempt to relieve the hang condition. In most cases, it is expected that this first action will resolve the problem. However, XES continues to monitor the situation. If the problem persists as long as the CFSTRHANGTIME interval, or two minutes (whichever is longer), XES will escalate to a more “powerful” action. Message IXL050I is issued to indicate what action is being taken. This cycle repeats until the problem is resolved or no further actions apply.

SFM Suggestions



If using GDPS, use
their recommendations

- **Enable SFM with BCPii**
- **SFM Policy Specifications**
 - ISOLATETIME(0) -All releases
 - SSUMLIMIT(900) -z/OS 1.9
 - MEMSTALLTIME(300) -z/OS 1.8 and 1.12
 - CFSTRHANGTIME(900) -z/OS 1.12 (worth watching)
 - CONNFAL(YES) -All releases, YES is default
- **COUPLExx Specifications**
 - INTERVAL (omit for default) -spin FDI z/OS 1.11
 - OPNOTIFY -All releases, your call

51



This slide summarizes the various SFM related parameters and offers some suggested values (best practice). Note that in a GDPS environment, GDPS may have recommendations that differ from what I suggest here. If using GDPS, follow their recommendations.

- ISOLATETIME is used to remove unresponsive systems from the sysplex. Since the system has already been unresponsive for the entire FDI, there is no need to delay taking action. Hence, use ISOLATETIME(0) to act immediately.
- SSUMLIMIT is used to remove systems that are sending signals but not updating status in the sysplex CDS. The system is still alive, but will eventually have issues if it cannot access the sysplex CDS. 15 minutes allows enough time for the DASD issues (or whatever) to be resolved. Empirical evidence suggests that systems can often survive this long without suffering significant sympathy sickness issues.
- MEMSTALLTIME is used to terminate unresponsive XCF group members that are either causing signalling sympathy sickness, or who are unable to perform functions that are critical to the operation of the system. 5 minutes allows time for your automation or operations staff to try to investigate and resolve the problem. If you have neither, a smaller value might be appropriate. A longer value elongates a sympathy sickness condition whose impact is likely to be more severe with the passage of time.
- CFSTRHANGTIME is used to alleviate sympathy sickness that arises when a connector to a CF structure becomes unresponsive. Some connectors may have significant quantity of work to do in response to some events. 15 minutes is likely fine for most, but might be too short for some configurations and workloads. Regardless, CFSTRHANGTIME should be specified to provide protection. If you have concerns you might specify a higher value, but be sure to put automation or operational procedures in place to react to the IXL040E and IXL041E messages as soon as they occur.
- CONNFAL is used to remove systems that lose signalling connectivity. Specify system WEIGHT as appropriate.
- INTERVAL is used when determining whether a system (or critical member) has become unresponsive. Remove the specification from your COUPLExx parmlib member to allow the system to use the spin FDI by default.
- OPNOTIFY is used when determining whether the operator should be told about certain sorts of issues

Summary



- Failing to deal with an unresponsive system in a timely manner can cause sympathy sickness
- Appropriate configuration of the Sysplex Failure Management (SFM) policy allows the systems in the sysplex to automatically take corrective action to resolve sympathy sickness problems when your manual procedures and automation fail to resolve them in a timely manner
- **Enable SFM with BCPii**

52



Thank you for your attention.

Enabling SFM to exploit BCPii services to detect when a system has failed is a huge step forward. I urge you to get this set up as soon as possible.

Other Sources of Information



- *MVS Setting Up a Sysplex (SA22-7625)*
- *MVS System Commands (SA22-7627)*
- *MVS System Messages IXC-IZP (SA22-7640)*
- *MVS Initialization and Tuning Reference (SA22-7592)*
- *MVS Programming: Callable Services for High Level Languages (SA22-7613)*

- *Redbook: System z Parallel Sysplex Best Practices (SG24-7817)*

53



The MVS publications are available at <http://www.ibm.com/systems/z/os/zos/bkserv/>
Redbooks are available at www.redbooks.ibm.com

Setting Up a Sysplex is the source for most of the material discussed in this presentation, in particular SFM policies and parameters.

System Commands documents the SETXCF and DISPLAY XCF operator commands that were mentioned.

System Messages documents the various messages that were mentioned.

Initialization and Tuning documents parameters specified in various parmlib members, including COUPLExx for XCF couple data sets and policies, as well as EXSPATxx for excessive spin conditions that determine the "spin FDI".

Callable Services documents BCPii Setup and Installation as well as the BCPii APIs

The Redbook "*System z Parallel Sysplex Best Practices*" is available at <http://www.redbooks.ibm.com/redbooks/pdfs/sg247817.pdf>

This book is intended to provide one succinct place to get guidance for how to set up a sysplex to achieve high levels of resilience. It includes a discussion of SFM policy settings and much, much more.