Parallel Sysplex: Achieving the promise of high availability (part 2)

Session 17431 13 August 2015

Note: this material was not presented during the session

Mark A Brooks
mabrook@us.ibm.com
z/OS Sysplex Development
IBM Poughkeepsie, NY

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® (logo)
ibm.com®	MVS™	Service Request Manager®	z10™	AIX® BladeCenter®
CICS®	OS/390®	Sysplex Timer®	z/Architecture®	DataPower®
CICSPlex®	Parallel Sysplex®	System z®	zEnterprise™	DS4000®
DB2®	Processor Resource/Systems Manager™	System z9®	z/OS®	DS6000™
eServer™	PR/SM™	System z10®	z/VM®	DS8000®
ESCON®	RACF®	System/390®	z/VSE®	POWER7®
FICON®	Redbooks®	Tivoli®	zSeries®	ProtecTIER®
GDPS®	Resource Measurement Facility™	VTAM®	zEC12™	Rational®
HyperSwap®	RETAIN®	WebSphere®	Flash Express®	System Storage®
IMS™				System x®
IMS/ESA®	Geographically Dispersed Parallel Sysplex™			XIV®

The following are trademarks or registered trademarks of other companies.

Adobe, the Adobe logo, PostScript, and the PostScript logo are either registered trademarks or trademarks of Adobe Systems Incorporated in the United States, and/or other countries.
Cell Broadband Engine is a trademark of Sony Computer Entertainment, Inc. in the United States, other countries, or both and is used under license there from.
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.
Intel, Intel logo, Intel Inside, Intel Inside logo, Intel Centrino, Intel Centrino logo, Celeron, Intel Xeon, Intel SpeedStep, Itanium, and Pentium are trademarks or registered trademarks of Intel Corporation or its subsidiaries in the United States and other countries.
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.
ITIL is a registered trademark, and a registered community trademark of the Office of Government Commerce, and is registered in the U.S. Patent and Trademark Office.
IT Infrastructure Library is a registered trademark of the Central Computer and Telecommunications Agency, which is now part of the Office of Government Commerce.

* 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.

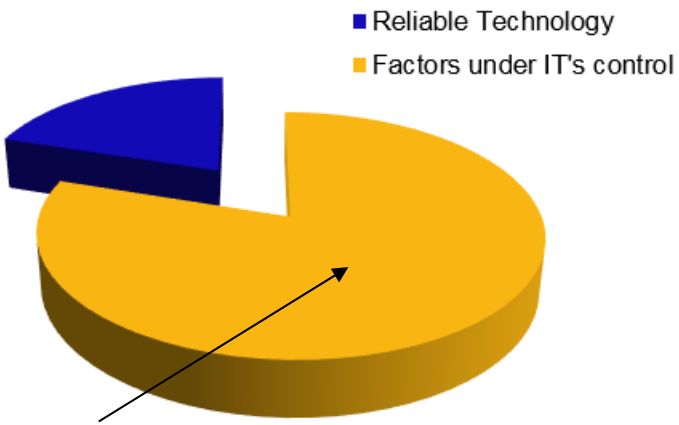
Information about non-IBM products is obtained from the manufacturers of those products or their published announcements. IBM has not tested those products and cannot confirm the performance, compatibility, or any other claims related to non-IBM products. Questions on the capabilities of non-IBM products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

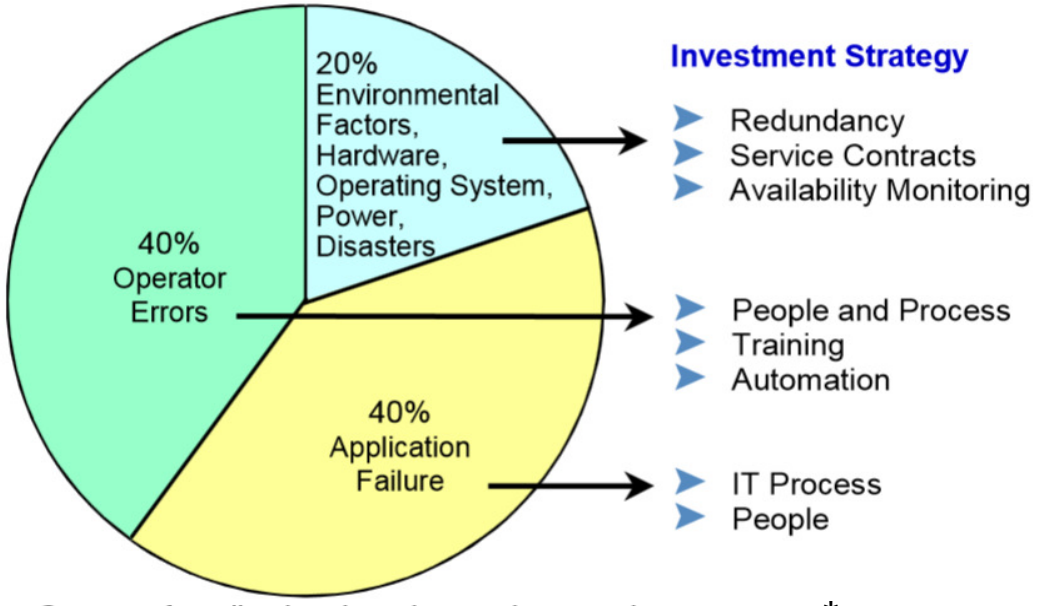
Trouble in paradise

- Companies face increasing pressure to deliver business services 24x7
- A properly configured parallel sysplex can deliver near continuous availability
- And yet, installations with highly redundant sysplex configurations fail to meet their availability objectives
  - Sometimes suffering significant detrimental business impact
- What's going wrong?

Unplanned Outages



HACoC Experience:
75-80% of unplanned service outages are the result of issues related to people, processes, and procedures



*Causes of application downtime and appropriate responses**

Availability of IT services requires more than simply installing reliable technology. You must also consider:

- How the Technology is implemented
- How the applications are designed, deployed and integrated with the underlying resilient infrastructure
- How the service is managed

** Source: Enterprise Guide to Gartner's High-Availability System Model for SAP, R-13-8504 dated 12/20/01 http://www.tarrani.net/mike/docs/HiAvailModel4SAP.pdf*

Our goal is to improve availability of business services

- "How can we mask failures so that critical business services appear to be highly available?"
  - Component failures will occur
  - When they do, we want to ensure that the business service does not experience an outage
  - Failing that, we want to restore service as soon as possible
- My focus is on how one would configure the sysplex to achieve maximal "high availability"
- I generally ignore some important business considerations:
  - What level of availability is required
  - What it will cost

Terminology

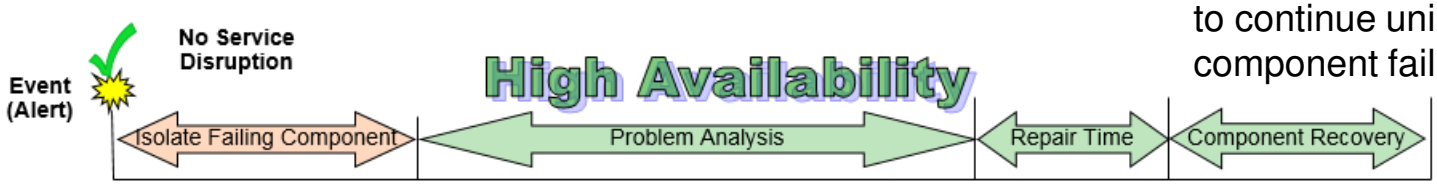
- **High Availability (HA) masks unplanned outages from end users**
  - The attribute of a system to provide service during defined periods, at acceptable or agreed upon levels. HA is achieved through:
    - Fault tolerance
    - Automated failure detection, recovery, bypass, and reconfiguration
    - Thorough testing and effective problem and change management

- **Continuous Operations (CO) masks planned outages from end users**
  - The attribute of a system to continuously operate. CO is achieved through:
    - Non-disruptive changes to hardware, software, and configuration
    - Software coexistence

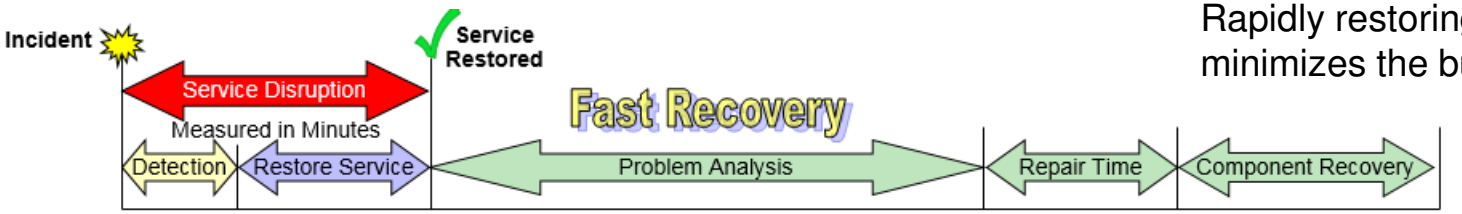
- **Continuous Availability (CA) masks both planned and unplanned outages from end users**
  - The attribute of a system to deliver non-disruptive service to the end user 24 hours a day, 7 days a week.

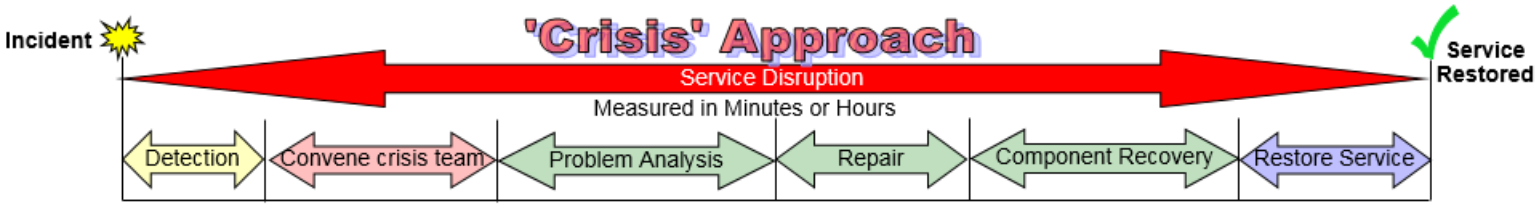
Possible approaches to dealing with failure

Ideally, a fault tolerant architecture and infrastructure allows service to continue uninterrupted despite component failure.



Not all failures will be masked. Rapidly restoring service minimizes the business impact.





When service restoration requires human intervention, outages tend to have unpredictable (long) duration. Risks significant business impact.

An Architecture for Achieving Highly Available Business Services

- Reliable - Minimize number of incidents
- Resilient - Mask failures that do occur
- Recoverable - Minimize duration of unmasked failures
- Resourceful - Be prepared when all else fails

Layered protection built on:

- The premise that failures will occur
- A robust, fault tolerant architecture
- Well implemented technology
- Excellent processes and procedures
- Skilled staff
- Well understood objectives

Reliable

How can we make the system more reliable?

Reliable: Minimize chance of failure

- Comprehensive testing
  - Remove defects prior to production
- Regular maintenance
  - Policy with rationale
  - Periodic review of effectiveness
- Effective change management
  - Facilitates successful changes
- Effective problem management
  - Iteratively improve system, processes, and skills

Testing
- System volume and stress;
- Production-like platform and data;
- Test scripts and Transaction Driver;
- Failure injection and recovery testing

Release Management
- Establish currency policies
- Deploy into production
- Review functional upgrades
- Package into testable releases

Change Management
- Enable growing quantity of changes;
- Measure Effectiveness by function and educate;
- Risk assessment and mitigation;
- Exception handling for emergencies and business needs;
- Accountability lies with developer;
- Readiness Checklists and implementation scripts;
- Focus on Quality of Change itself

Problem Management
- Quick analysis (Cause of incident) and timely correction;
- Track corrections and escalate exceptions;
- Reduce recurring incidents;
- Knowledge DB;
- DB structure supports reporting needs
- Problem Prevention - True root cause (Cause of defect);
- Pursuit of secondary contributors;
- Failure Pattern and Trend Analysis

Hardware eventually fails. Software eventually works.

Comprehensive Testing

Does it work?

- Unit test
- Function test
- System test
- Integration test

Can it survive?

Load test

Failure test

Recovery test

Safely change?

- Installation test
- Backout test

For HA, these questions must be answered before we get into production...

Configure suitable test environments

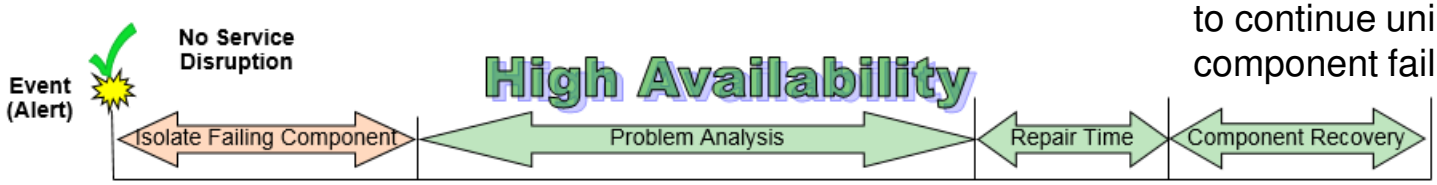
- To minimize the risk to the availability of business services, comprehensive testing must be done in an environment that is NOT production
- So you need a completely separate test environment that is as similar to the production environment as you can make it
  - Hardware, software, data, workload
- The greater the dissimilarity between production and test, the greater the risk
  - Cost concerns often lead to compromises
  - Don't ignore the risk, manage it
  - Excellent service management processes help ensure that you make the right tradeoffs

Resilient

How can we mask failures so that the business service appears to be highly available?

Mask failures to make business services appear available

Ideally, a fault tolerant architecture and infrastructure allows service to continue uninterrupted despite component failure.



We want to design and implement a robust infrastructure that allows us to mask component failures so that the business services are not disrupted.

Achieving “no disruption” also requires excellent service management processes and procedures.

Since the subject of this presentation is “configuration”, we'll focus on the technology. Note well, however, that technology alone is not enough. Only 20% of unplanned outages are attributable to technology. The other 80% is attributable to people, processes, and procedures.

Parallel Sysplex

- Sysplex is about eliminating “single points of failure” by providing redundancy for all sysplex components:
  - Servers/CECs
  - z/OS Systems
  - DASD Controllers
  - Coupling Facilities
  - Links
  - Middleware regions
  - Application regions
  - etc.
- In order to effectively mask failures in the sysplex:
  - Redundant components must be “clones” so that any survivor is capable of processing the same work as the failed instance
  - Work must be able to flow freely to the surviving instances
  - The resources needed to process work must be accessible from wherever the surviving instances happen to run
- With this infrastructure, we can minimize the chance that business services will be impacted by a component failure
  - However, potential for service disruption will never be zero

Parallel Sysplex ...

- In other words, work must be capable of running anywhere in the sysplex
  - On any active system
  - On any active middleware instance
  - On any active application region
- If the sysplex has redundant components which are not "clones" of one another …
  - Whether due to affinities, lack of workload routing capabilities, software licensing restrictions, capped processors, etc.
- Then that redundant infrastructure ***will be ineffective*** in providing high availability
  - Service disruptions will occur

Analyze work as it flows through your infrastructure

Resources used?
Single points of failure?
Affinities?

Network

SYS1

Applications

Subsystems

z/OS

CP | memory | network

Local Queue

Local Cache

CF

Shared Queue

Shared Cache

Locks

Shared storage
-data base
-system files
-catalogs

SYS2

Local Queue

Local Cache

Applications

Subsystems

z/OS

CP | memory | network

Local storage
-logs
-work files

Local storage
-logs
-work files

Redundant hardware infrastructure

- Hardware
  - Servers
  - Coupling facilities
  - Links and cables
  - Network (adapters, switches, routers)
- CF Structures
  - Duplex appropriately
  - Failure isolation
- Data
  - Use hyperswap to eliminate DASD as a single point of failure
  - Tape replication to eliminate tape as a single point of failure

How much hardware redundancy is needed ?

- We tend to think “two”. But what about ...
- Reliability
  - Does “two” actually provide sufficient reliability?
    - A parallel system of two components has an expected reliability of 96% if each component is 80% reliable.
  - As hardware ages, it tends to be less reliable
    - So expected reliability drops over time
- A component instance might be down due to failure or planned maintenance
  - Do the surviving components have enough capacity to support the workload?
  - Is the expected reliability of N-1 components sufficiently reliable?
    - Our previously 96% reliable parallel system now has an expected reliability of 80%. Is it sufficient?

Hardware Redundancy

- To eliminate single points of failure during planned outages and unplanned component failures, at least three component instances are needed
- Must ensure that the surviving N-k components have sufficient capacity to handle the workload
  - Excellent service management processes (capacity planning, availability management) will account for changes in workload
- Must ensure that the surviving N-k components provide the desired degree of expected reliability
  - May not be computable (lack of data, complexity, …)
  - Excellent service management processes will provide installation specific failure trend analysis
  - Component failure impact analysis could help assess degree of risk

k = number of simultaneous failed component instances to be tolerated

Hardware Redundancy

So we understand the key principles:

Provide sufficient redundancy to maintain reliability and capacity in the face of the expected worst case “k” simultaneous failed component instances.

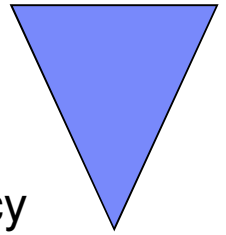
What are some specific sysplex configuration guidelines?

Address exceptions reported by z/OS Healthchecker

- XCF_CDS_MAXSYSTEM
- XCF_CDS_SEPARATION
- XCF_CDS_SPOF
- XCF_CF_ALLOCATION_PERMITTED
- XCF_CF_CONNECTIVITY
- XCF_CF_MEMORY_UTILIZATION
- XCF_CF_PROCESSORS
- XCF_CF_STR_AVAILABILITY
- XCF_CF_STR_DUPLEX
- XCF_CF_STR_EXCLLIST
- XCF_CF_STR_NONVOLATILE
- XCF_STR_POLICY_SIZE
- XCF_CF_STR_PREFLIST
- XCF_CF_SYSPLEX_CONNECTIVITY
- XCF_CFRM_MSGBASED
- XCF_CLEANUP_VALUE
- XCF_DEFAULT_MAXMSG
- XCF_FDI
- XCF_MAXMSG_NUMBUF_RATIO
- XCF_SFM_ACTIVE
- XCF_SFM_CFSTRHANGTIME
- XCF_SFM_CONNFAIL
- XCF_SFM_SSUMLIMIT
- XCF_SFM_SUM_ACTION
- XCF_SIG_CONNECTIVITY
- XCF_SIG_PATH_SEPARATION
- XCF_SIG_STR_SIZE
- XCF_SYSPLEX_CDS_CAPACITY
- XCF_SYSSTATDET_PARTITIONING
- XCF_TCLASS_CLASSLEN
- XCF_TCLASS_CONNECTIVITY
- XCF_TCLASS_HAS_UNDESIG

Health checker identifies single points of failure and other configuration issues that compromise HA.
It does not find them all. Still, a good start.

Coupling Facility Configuration



- Relative to coupling facilities, redundancy to a fair extent, permits resiliency
  - Have at least 2 Coupling Facilities defined in the CFRM policy and physically available.
  - Have at least two coupling links to / from each operating system to the coupling facility. Additional paths may be required with heavy workloads.
- External CFs are preferred to internal CFs.
  - “External” = CF in use by z/OS systems does not reside on CEC with any z/OS image using the CEC
  - Internal CF resides on a CEC with at least one z/OS image using it
  - Certain structures become unrecoverable if they reside in a non-failure-isolated CF.
- Use dedicated CPs on the CFs whenever possible.
- NonVolatile CFs are preferred.
- Provide enough space for all the structures and enough white space for structures on the other coupling facilities to rebuild into this coupling facility should there be a CF outage.

CF Structure considerations

- Follow best practices for CF structure and z/OS image placement on CECs, to ensure proper failure-isolation between them
  - Implement pseudo-standalone CF images (no co-resident z/OS images from the same sysplex) that therefore provide failure-isolation without the use of system-managed CF duplexing
  - Spread z/OS images from all sysplexes around to all CECs, except those CECs hosting pseudo-standalone CFs
- Exploit system managed duplexing for structures requiring:
  - Good sysplex recoverability
    - MQ shared queues, CICS structures, …
  - Failure-isolation but do not have it
    - DB2 lock and SCA structure recoverability
- Exploit user-managed duplexing
  - DB2 GBP cache structure recoverability

Effective service management ensures that failure isolation is maintained during maintenance windows.

CFRM - SMDUPLEX

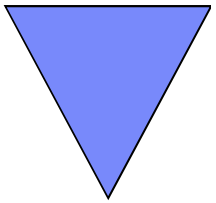
- System managed duplexing allows applications to transparently recover from failures automatically
  - Structure failure, CF failure, loss of connectivity to CF
  - Critical for applications which do not support user rebuilds
- But not without cost
  - Service times for duplexed requests are longer than simplex
  - Need links between CFs (and a pair of CFs)
- Setup Required
  - Format CFRM CDS
    - ITEM NAME(MSGBASED) NUMBER(1)
  - CFRM policy updates for relevant structures
    - DUPLEX(ALLOWED) – manual control of when to start duplexing
    - DUPLEX(ENABLED) – system seeks to maintain duplexing when feasible

Recommendation: Consider leveraging SMDUPLEX processing.

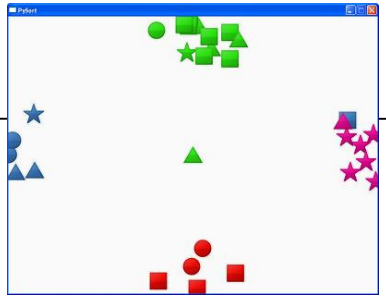
User Managed Duplexing

- With system managed duplexing, the system maintains both instances of the structure
- With user managed duplexing, the exploiter determines which requests to duplex
- Refer to application recommendations for best practices
  - For example, best practice for DB2 Group Buffer Pools is to exploit user managed duplexing

Sizing CF Structures



- IBM recommends using the CFSizer website or SIZER batch utility whenever the CFCC level is upgraded or there is a significant change in the workload using the structures
- CFSizer
  - http://www.ibm.com/systems/support/z/cfsizer/
- SIZER batch utility
  - http://www.ibm.com/systems/support/z/cfsizer/altsize.html
- IBM suggests that the INITSIZE to SIZE ratio not exceed 1:2



Alter Processing

- Coupling Facility structures can be altered to meet the needs of exploiters.
  - Applications can issue IXLALTERs to change the entry to element ratio and the size of the structure.
  - Operators can use the SETXCF command to alter the size of the structure
  - ALLOWAUTOALT(YES) allows z/OS to initiate alters of the structure to align with in-use counts when the FULLTHRESHOLD is surpassed.
- When a structure is being altered and it is at its maximum size and near full, alter processing may cause undue burden on the coupling facility.

Eliminate SPOFs on DASD data

- Implement synchronous DASD mirroring either locally (same site) or to a remote site within synchronous replication distance, via PPRC or Metro Mirror, to provide effective redundancy for data on storage subsystems
- Enable that redundancy to be used for sysplex HA by implementing Hyperswap failover capability
  - Basic Hyperswap (TPC-R) or GDPS-based hyperswap management
  - Provide HA for storage subsystem failures
- Consider out-of-region replication of data to a long-distance remote site (XRC or Global Mirror) in addition to synchronous replication
  - Preserve data redundancy even after a Hyperswap
  - Evaluate GDPS 2- and 3-site DR alternatives

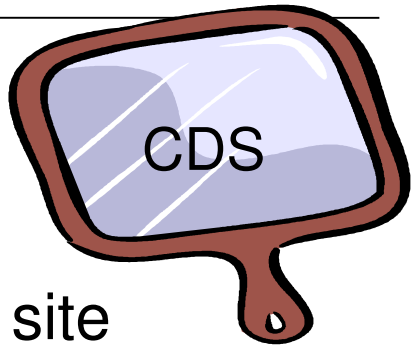
General Recommendations for all Couple Datasets

- Always run with a primary and alternate CDS
- Place CDSs on different volumes
- Place CDSs on different physical devices whenever possible
- Be prepared to deal with loss of a CDS by having:
  - A third one pre-formatted for use
  - Operational procedures or system automation to add it as an alternate, thus restoring redundancy
  - Mechanism to notify system programmer so that problem can be resolved and sysplex restored to “normal” configuration

Synchronous Mirroring of Couple Datasets

- Avoid synchronous mirroring of CDS
  - Risk of I/O delay or long busy conditions, which can:
    - Degrade timely access to the CDS by users
    - Lead to permanent I/O error and CDS being removed from service
- Especially Sysplex CDS and CFRM CDS
  - Removing both primary and alternate from service results in a **sysplex-wide outage**
- Possible Exception: LOGR CDS
  - If log-stream data is being mirrored to DR site, and data in the log stream is to be used at DR site, LOGR CDS must be mirrored as well
  - Need time consistent copies of all relevant data if the log stream is to be usable
    - Off-load data sets, staging data sets, LOGR CDS, MVS Catalogs

Using Copies of Couple Datasets at DR Site is Risky



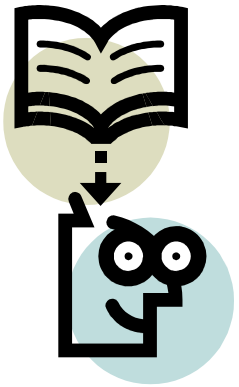
- Customer Goal: Simplify DR configuration and minimize time to recovery by copying CDS to DR site
  - Allows all volumes on device to be copied or asynchronously mirrored without need for manual exclusion of volumes containing CDS
  - Need not run format utility to create CDS at DR site
  - Need not run policy utility to create policies in the CDS at DR site
- Problem: Requires great care to avoid sysplex outages
- Risks:
  - Sysplex outages
  - Data Integrity issues
  - CDS at primary site unexpectedly removed from service
  - 0A3 wait-state when GRS cannot allocate ISGLOCK structure
  - Residual data for inaccessible structures at other site
  - One, some, or all CF's ripped away from active sysplex

Recommendations if Using Copies of Couple Datasets at DR Site

- All CFs used by the DR site should be defined in the CFRM policy used by the primary site
- Do not allow DR site to gain access to CF's in primary site
- Do not allow DR site to gain access to DASD in primary site
- When configuration changes, be sure you maintain these conditions
  - Needs to be part of your change management procedures
- Reference: Hot Topics February 2011 Issue 24 p.69 "*Mirror, mirror, on the wall, should couple dataset be mirrored at all?*"
  - http://publibfp.dhe.ibm.com/epubs/pdf/eoz2n1c0.pdf

CRITICALPAGING

- Problem: Loss of system(s) during hyperswap (or other dasd swap) which were expected to survive
- Cause: Page fault in critical code path while DASD freeze/swap is in progress
- Solution: CRITICALPAGING Function
  - "Hardens" storage of critical address spaces
  - Reduces potential for page faults in address spaces that participate in the critical path:
    - RASP (RSM), GRS, CONSOLE, XCFAS, address spaces associated with Basic HyperSwap in base (HSIB), Basic HyperSwap API (HSIBAPI), and GDPS HyperSwap Communication Task (often jobname GEOXCFST)



CRITICALPAGING

- Real storage assessment needed prior to enabling CRITICALPAGING to ensure application performance is not impeded
  - If the system "never" pages perhaps no real storage needs to be added
  - If the system pages often, to maintain current performance, a simple guideline
    - PLPA+EPLPA and CSA+ECSA
- References:
  - WSC Flash
    www.ibm.com/support/techdocs/atsmastr.nsf/WebIndex/FLASH10733
  - White Paper
    www.ibm.com/support/techdocs/atsmastr.nsf/WebIndex/WP101800

Recommendation: Perform real storage assessment and enable CRITICALPAGING function in DASD swap environments

Alternative to CRITICALPAGING

- Flash Memory for Paging
  - zEC12 or z13 with Flash Express
  - z/OS V1R13 and up

- Page fault during critical code path can be resolved from flash memory even though DASD freeze/swap is in progress

Redundant software infrastructure

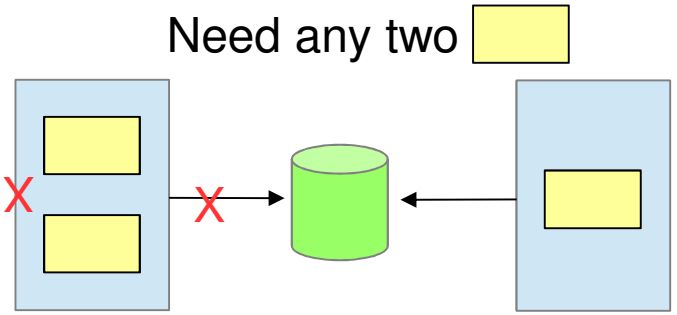
- Multiplicity of software instances
  - z/OS images
  - Subsystems (DB2, CICS, IMS, ...)
  - Applications

Spread across systems

Possibly within a single system image too

How much software redundancy is needed ?

- Just as for hardware, we must:
  - Have at least three software instances to eliminate the single point of failure that would occur during planned outages or unplanned failures
  - Ensure that the surviving N-k instances have capacity to process the workload
- Must also ensure that the software instances are sufficiently failure isolated from one another
  - Would not want all instances to reside on one CEC
  - A given component failure must never impact more than the tolerable number of "k" software instances
    - Failure of CEC or z/OS system image
    - Loss of I/O capacity (as an example)



Software redundancy

- You maximize your chance of achieving your desired availability when you have functional software instances on every system in the sysplex
  - Anything less tends to increase the risk of a service outage
  - But there are pressures that motivate installations to "subset":
    - Software license charges
    - Capping
    - Affinities
    - Application issues

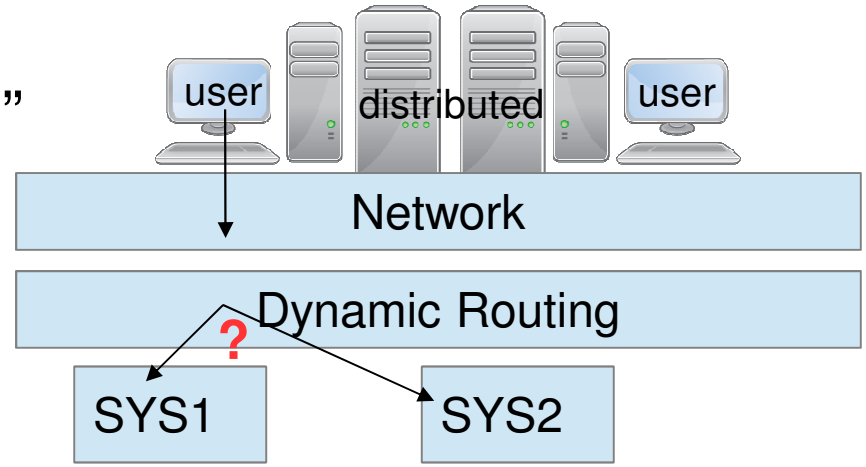
Recognize the risk.
Takes steps to manage it.

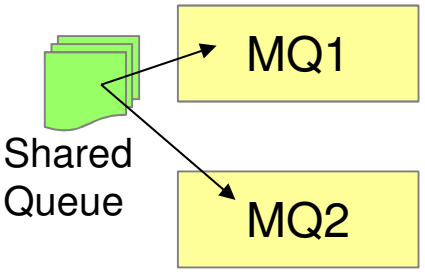
- Must ensure that the surviving N-k instances have sufficient capacity to handle the workload
  - Excellent service management processes (capacity planning, availability management) will account for changes in workload

Sufficient reliability ?

Dynamic workload routing

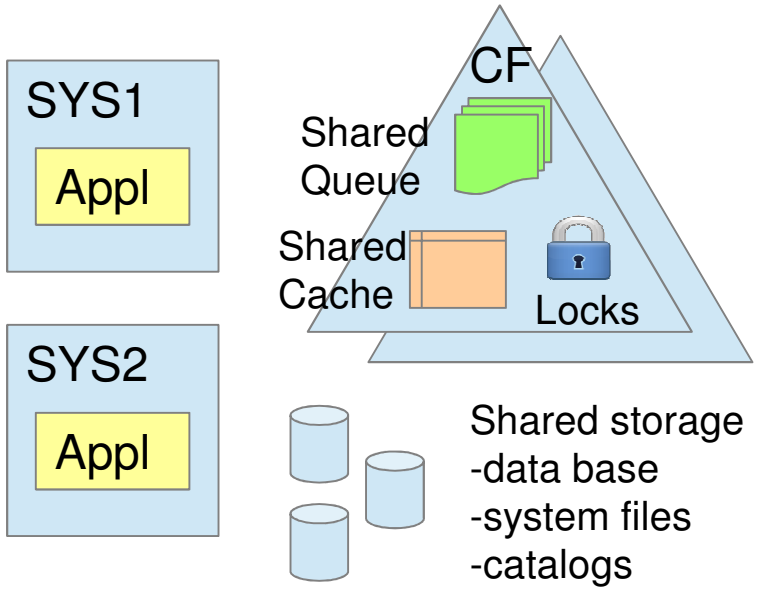
- Redundancy is useless if we can't exploit the redundant instances
- Work must flow to the “good guys”
  - Those that survive a failure
  - Those with capacity
- Must configure system to enable dynamic routing of work
  - VIPA and Dynamic VIPA
  - Sysplex Distributor
  - VTAM Generic Resources
  - CICSPLEX
  - MQ Shared Queues
  - IMS Shared Queues
  - Dynamic workload management (WLM)





Data sharing

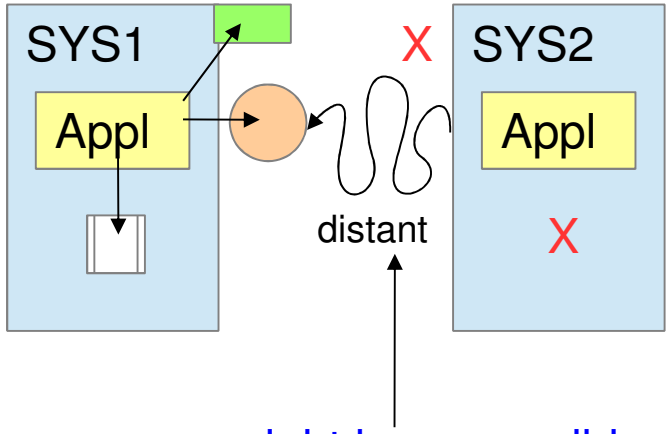
- Dynamic routing is useless if the application does not have access to the data it needs to perform its function
- Data must be accessible from wherever the application runs
- Must configure system to enable data sharing
  - Shared disk
  - Coupling Facility



But more generally ...

Cloned systems

- Dynamic routing is useless if the application does not have access to the resources needed to perform its function
- Whatever resources the application needs must be available from wherever the application runs
- Whether hardware
  - Features, encryption, printer, tape, network, ...
- Or software
  - Subsystem, applications,...
  - With needed features, levels, …



The resource might be accessible, but remotely. Can the application provide service with acceptable performance and capacity with the additional latency?

Sysplex Cloning Recommendations

- Implement one or more forms of sysplex dynamic workload routing/balancing so that any work can be routed to any active region that can process it
  - Using Sysplex Distributor, MQ shared queues, IMS shared queues, …
  - Eliminate application-level affinities which get in the way of dynamic workload routing
- Get rid of static routing affinities in current SNA/TCPIP network routing infrastructure; replace with one or more of the above dynamic routing mechanisms
- Eliminate partitioned data and “data affinities” to specific systems or regions by making all critical data into sysplex shared data – in DB2, IMS DB, or VSAM RLS data sharing

About those applications

- Applications are often a significant inhibitor to achieving availability goals
- z/OS and IBM middleware were designed and implemented with the goal of allowing applications to inherit the availability characteristics of running in a sysplex without having to be rewritten
- But applications can be written in ways that make it impossible for them to enjoy these benefits ...

Applications can inhibit availability

- Affinities
  - Logical resources
  - Physical resources
  - Local data
- Local time stamps
- "global" counters
- Hot spots
- Lack of parallelism
  - Assumes locality of execution
  - Assumes FIFO execution
- Lack of fault tolerance

Subsystem specific exploitation issues
  CICS, DB2, IMS, Websphere...

Legacy architecture

Lack of coexistence and versioning

Poor implementation

Inadequate design
  Granularity
  Serialization
  Commit points
  Understanding

Affinities

- Eliminate workload affinities so that any work can run on any of the multiple supporting subsystem instances
- Affinities are static bindings where work can only run in one system or in one subsystem instance
- Even the most resilient sysplex infrastructure is ineffective if workload affinities exist and the subsystem instance hosting the affinity is down
  - In effect, you still have a single point of failure even though there are multiple subsystem instances

Application governance

- Architect applications for HA
  - Cloning
  - Sharing
  - "Rolling IPL"
  - Coexistence
  - Versioning
- Enforce standards
  - Ensure compliance
  - Verify through test

Availability Management
- Proactive, high level strategic;
- Cross-function;
- Awareness and communication;
- Process and Architecture Governance;
- Improvement initiatives;
- Value of Availability

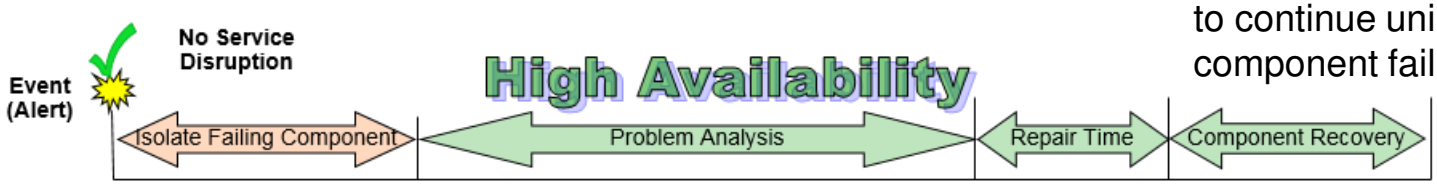
Recovery Development
- Understand potential failures;
- Develop Problem Determination (PD) tools;
- Develop Recovery procedures;
- Automate recovery;
- Test and practice recovery (fire drills)

Testing
- System volume and stress;
- Production-like platform and data;
- Test scripts and Transaction Driver;
- Failure injection and recovery testing

Problem Management
- Quick analysis (Cause of incident) and timely correction;
- Track corrections and escalate exceptions;
- Reduce recurring incidents;
- Knowledge DB;
- Problem Prevention - True root cause (Cause of defect);
- Pursuit of secondary contributors;
- Failure Pattern and Trend Analysis

Resilient: component failure does not disrupt business services

Ideally, a fault tolerant architecture and infrastructure allows service to continue uninterrupted despite component failure.



- Architect a technology solution
- Design in fault tolerance at every layer
- Build with components that support HA
  - Systems
  - Storage
  - Applications
  - Network

Design, Build, Maintain the Technology for HA

- Ensure redundant capacity
- Provide capability to remove any component at any time without service impact.
- Provide automation to quickly restore services.

High Availability requires excellence across all areas that support the business services: Technology, Processes, People.

- Assign Organizational functions
  - Proactive culture
  - Service Planning (linking development with delivery)
  - Availability Management
  - Enterprise Architecture (Systems and Applications)
  - The right skills at all support levels
- Define Effective Processes
  - Architecture and standards
  - Developing HA solutions
  - Testing and validation
  - Managing the technology
  - Implementing updates
  - Fast service restoration
  - Correcting errors
  - Proactive prevention

Design, Build, Manage the Service for HA

- Provide information for sound management decisions
  - CMDB and component status
  - Services catalog and business impact
  - Change history
  - Cost of down time and down time history
- Prepare ability to remove any component at any time without service impact.

Design, Build, and Maintain the Technology for HA

- Redundant Hardware components
  - Processors, CFs, Links, IO, Storage
- Redundant Software components
  - Including middleware and application components
- No single points of failure
  - Find and eliminate workload SPOFs/affinities
- Dynamic routing capabilities for all workload
  - No static routing affinities
- Data sharing for all critical data
  - No data-related affinities

Simplification/cloning of the environment – symmetry and "anything runs anywhere"

Sufficient failover capacity for recovery

"White space" and/or automated CoD
Processors, memory, I/O
z/OS capacity and CF/link capacity

Health Checks with follow-up

Build sysplex infrastructure to mask component failures so that business services continue to function without interruption.

Design, Build, and Manage Service for HA

- Proactive
  - Understand and plan
  - Justify and build
  - Analyze and fix
- Application architecture
  - Fault tolerant
  - Sysplex enabled
- Effective processes
  - Closed loops
  - Spawn improvements
- Failure injection/testing
  - Practice, verify, improve

Service Level Management
- Business Requirements;
- Common IT Service Level Objectives Understood;
- Manage Expectations;
- Service Planning;
- Measuring service level achievements;
- Customer Satisfaction

Availability Management
- Proactive, high level strategic;
- Cross-function;
- Awareness and communication;
- Process and Architecture Governance;
- Improvement initiatives;
- Value of Availability

Event Management
- Define and govern Monitoring Strategy
- Define Monitoring Architecture;
- Implement monitoring tools;
- Correlate component and service events
- Automate responses
- Continuously tune thresholds

Configuration Management
- Component Location;
- Connectivity and linkages;
- Contacts;
- Business Impact;
- Index to recovery procedures;
- Horizontally and vertically integrated DB

Problem Management
- Quick analysis (Cause of incident) and timely correction;
- Track corrections and escalate exceptions;
- Reduce recurring incidents;
- Knowledge DB;
- Problem Prevention - True root cause (Cause of defect);
- Pursuit of secondary contributors;
- Failure Pattern and Trend Analysis

Testing
- System volume and stress;
- Production-like platform and data;
- Test scripts and Transaction Driver;
- Failure injection and recovery testing

Recovery Development
- Understand potential failures;
- Develop Problem Determination (PD) tools;
- Develop Recovery procedures;
- Automate recovery;
- Test and practice recovery (fire drills)

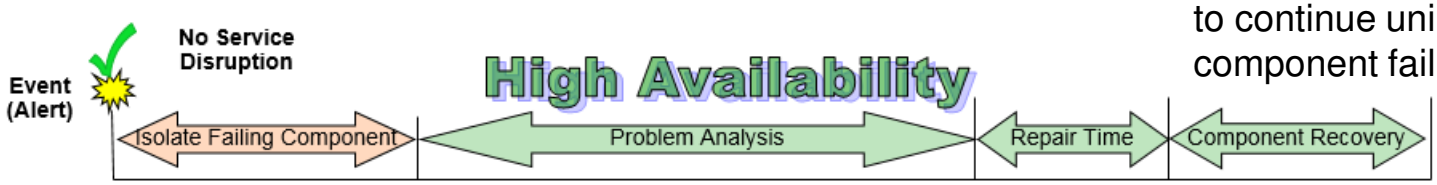
Manage the delivery of business services in a way that proactively supports and enables the continued masking of failures.

Recoverable

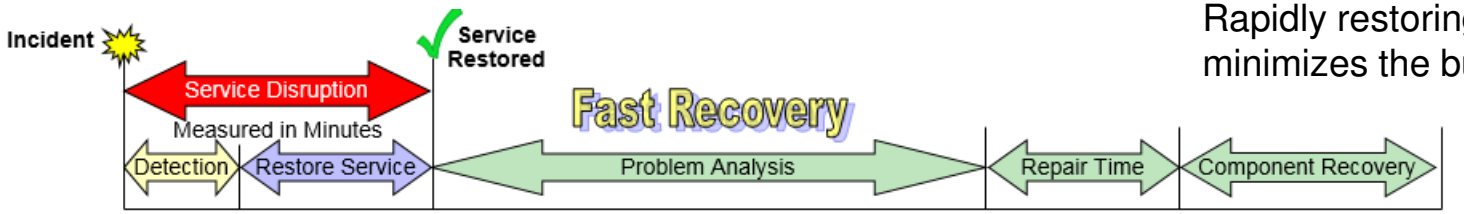
We failed to mask the failure.
How can we restore service as fast as possible?

Fast recovery minimizes duration of disruption to business services

Ideally, a fault tolerant architecture and infrastructure allows service to continue uninterrupted despite component failure.



Not all failures will be masked. Rapidly restoring service minimizes the business impact.



- The key objective for fast recovery is to restore normal operation of the business service as quickly as possible
- Diagnosis and repair are secondary issues to be addressed later
- People often need to change their mindset

Depending on your SLA's, fast recovery might achieve "no disruption"

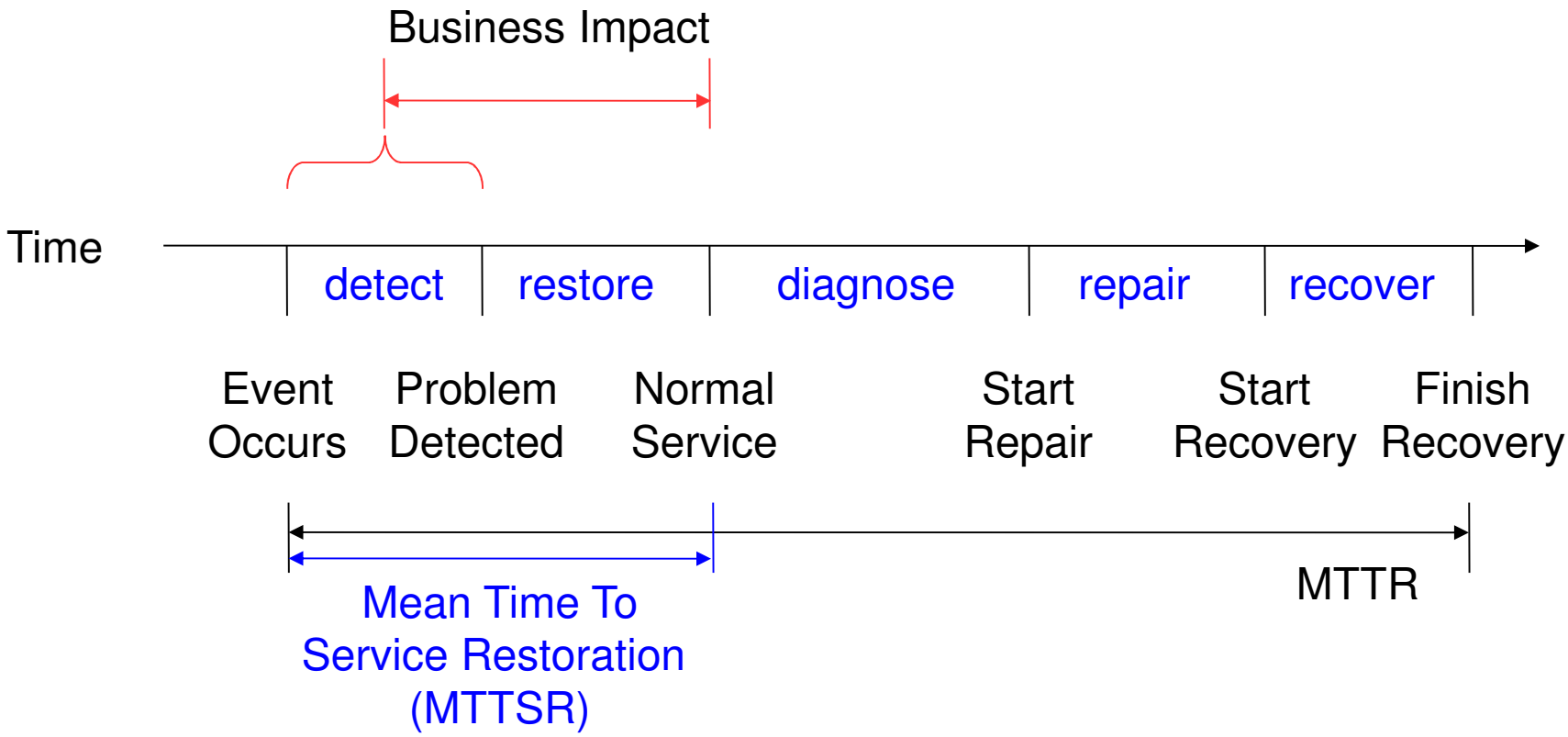
Applicability of fast recovery

- When “resiliency” fails
  - Must be prepared for the possibility that our resilient infrastructure fails to mask the problem
  - Some possible reasons:
    - Failed to consider the problem
    - Defects
    - More failures than expected
- When “resiliency” is not achievable
  - Unable to resolve affinities, application or infrastructure issues

Fast recovery principles

- Fast recovery will never be fast enough if manual intervention is required to restore service
- The system must be configured to automatically
  - Detect the problem
  - Resolve it
- Fast recovery requires investment in technology, people, and process
  - System understanding
  - Automated event monitoring
  - Automated recovery actions
  - Test and verification
  - Root cause analysis
  - Remediation

Use fast recovery to minimize service outage



The number one priority is to restore normal business operation ASAP. Everything else is secondary.

Approaching the problem

- What can fail?
- What is the impact?
  - Which services
  - Scope
- How to detect failure?
  - Component
  - Business service
- Recovery architecture
  - Recovery model
  - Scope of recovery

Primary focus must be on how quickly the business service is restored.

How to accomplish recovery
- Business service
- Component

Measure
- How fast did we detect
- How fast did we recover
- How long was business service impacted

How can we do better?
- Mask the failure
- Faster detection
- Faster recovery

Fast recovery considerations

- Infrastructure concerns
  - Largely same considerations as for “resiliency”
  - Need place that can access all the necessary resources
- Restoration model
  - Failover
    - Reroute work to an alternate provider
    - Task, space, system, sysplex, site, DR
  - Hot standby
    - How will work get here?
    - Application issues? Reconnect?
  - Fast restart
    - Scope: task, space, subsystem, system, sysplex, DR

You may need to resolve application issues regardless of the model.

Governance !

Fast recovery considerations ...

- Where to restore service
  - Same system? Some other system? Sysplex? DR site?
- Prior to restoration
  - Might need recovery action prior to restoring service
  - Quiesce, release locks, shutdown, failover, CBU, …
- After restoration
  - Might need to come back
    - Performance, capacity, simplicity, ...

Fast recovery is a critical business service

- You analyzed your infrastructure to identify and resolve single points of failure that would impact your critical business services
- You must similarly resolve issues that would impact your ability to accomplish fast recovery
  - The set of resources and services needed to perform fast recovery are likely not the same as those required for the business service
- **If fast recovery is the key to minimizing the duration of a service outage, then the fast recovery "service" must be highly available**

Why might fast component recovery be needed?

- When component is required by business service. Perhaps:
  - Component was single point of failure
  - Scope of failure greater than expected
- To restore normal operation. Perhaps:
  - Single point of failure until failed component is restored to service
  - Less capacity to deal with spikes and variability of workload
  - Redundant components have longer service times
  - Want to resume operation with “normal” provider of the service
    - Operational simplicity
    - Performance?
    - Capacity?

Enabling fast recovery

- Staff needs deep understanding of system
  - What is normal
  - What can fail
  - Failure impact
  - Manifestation
  - Appropriate recovery actions
- Must automate event monitoring and recovery actions to minimize duration of business service outage
  - Need to react at system speed, not human speed
  - Embody staff's deep understanding of system into the automation
- Excellent service management processes are critical
  - Especially root cause analysis and remediation
  - Ability to test and verify recovery procedures

Fast recovery: Minimize duration of service outage

- Fast recovery requires
  - Comprehensive monitoring
  - Automated detection
  - Automated recovery
- Knowledge
  - What can fail?
  - What will it impact?
- Testing
  - Failure injection
  - Validation of recovery
- Robust technology
  - Some place to go
  - Application enablement

Service Level Management
- Business Requirements;
- Common IT Service Level Objectives Understood;
- Manage Expectations;
- Service Planning;
- Measuring service level achievements;
- Customer Satisfaction

Availability Management
- Proactive, high level strategic;
- Cross-function;
- Awareness and communication;
- Process and Architecture Governance;
- Improvement initiatives;
- Value of Availability

Event Management
- Define and govern Monitoring Strategy
- Define Monitoring Architecture;
- Implement monitoring tools;
- Correlate component and service events
- Automate responses
- Continuously tune thresholds

Configuration Management
- Component Location;
- Connectivity and linkages;
- Contacts;
- Business Impact;
- Index to recovery procedures;
- Horizontally and vertically integrated DB

Problem Management
- Quick analysis (Cause of incident) and timely correction;
- Track corrections and escalate exceptions;
- Reduce recurring incidents;
- Knowledge DB;
- Problem Prevention - True root cause (Cause of defect);
- Pursuit of secondary contributors;
- Failure Pattern and Trend Analysis

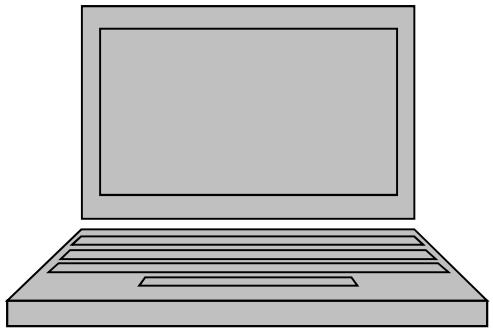
Testing
- System volume and stress;
- Production-like platform and data;
- Test scripts and Transaction Driver;
- Failure injection and recovery testing

Recovery Development
- Understand potential failures;
- Develop Problem Determination (PD) tools;
- Develop Recovery procedures;
- Automate recovery;
- Test and practice recovery (fire drills)

SYNCHDEST

- Ensure CONSOLxx contains SYNCHDEST(groupname)
- Groupname is the name of a console group declared in CNGRPxx
- SYNCHDEST will try to present the SYNCH WTOR or SYNCH WTO to the physically attached MCS consoles on the system where the message was issued
- SYNCHDEST increases the likelihood of a system programmer noticing the message sooner during the hours when the systems are actively monitored
- The SYNCH message will go to the first declared console and stay there for 125 seconds then roll to the next console in the list. Eventually, the SYNCH message will go to the system console and stay there until replied to.
- Check the priority box and respond to SYNCH WTOR
- SYNCH WTOs will stay for 10 seconds and then system processing will resume.
- Refer to WSC FLASH10761 on synchronous WTOR processing
  - www.ibm.com/support/docview.wss?uid=tss1flash10761&aid=1

Last Resort



- System Console
  - Ensure access to the system console
  - z/OS 1.11 and above V CN(*),ACTIVATE not required to enter commands
- If MCS, SMCS and EMCS consoles are not responding, access the system via the System Console on the HMC
- Strongly consider enabling accessing of the HMC over the web
- Consider leveraging the Console Actions -> Monitor System Events to monitor for SYNC WTORs or SYNC WTOs, especially for times when system programmers are not actively monitoring the system

Configuring for fast recovery

- Enable sysplex to automatically remove unresponsive systems from the sysplex
  - z/OS can automatically detect and recover
  - Generally the fastest and most reliable way to deal with problem
  - Minimizes duration of sympathy sickness impact
- Take advantage of features that enable recovery processing to run faster

Configuring for fast recovery

Unresponsive systems

- SFM with BCPii
- SFM Policy
  - ISOLATETIME
  - SFM CONNFAIL
  - SFM SSUMLIMIT

Unresponsive applications

- SFM Policy
  - MEMSTALLTIME
  - CFSTRHANGTIME

Faster recovery

CFRM MSGBASED

Serial Rebuild

RECPRTY

ARM

Auto IPL

Sympathy Sickness

- Sick systems don't play well with others
  - They don't respond when spoken to
  - They don't share their toys
- Hangs occur because others are:
  - Waiting for a response
  - Waiting to get an ENQ, latch, lock
- What can make a system sick?
  - Being dead
  - Loops (spin, SRB)
  - Low weighted LPAR
  - Loss of a coupling facility
- If a "sick" system does not recovery swiftly, or is not removed from the sysplex swiftly, other systems in the sysplex may be adversely impacted
  - ***In many cases, a long period of sympathy sickness has a greater negative impact on the sysplex than does the termination of an XCF group member, address space, structure connector, or even a system***

Root Causes of Sysplex Performance Problems

- **Real storage**
  - Change in workload
  - Defect
  - Poor configuration
- **CF slow downs**
  - Service time degradation
  - Not meeting expectations
  - More requests going ASYNC
  - New configuration
  - Increase workload
- **CPU**
  - Not enough CPU to handle workload
  - Capping LPARs
  - All LPARs running hot at the same time and there are vertical lows

Virtual storage shortages

SRBs Looping

Contention for global resources

Spin loops

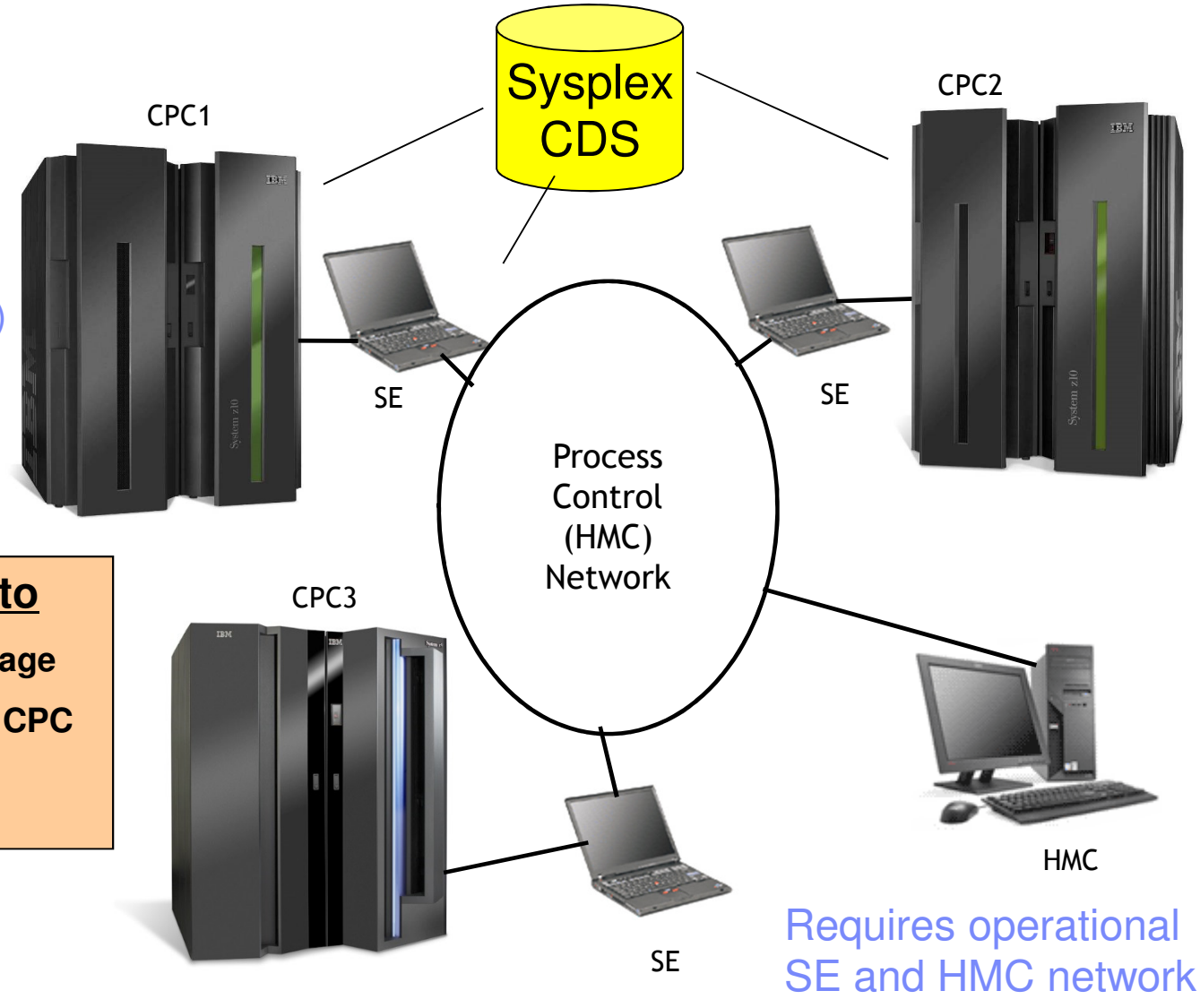
IO delays

CDS access delays
Logger offload hangs

Try to have your event monitors detect automatically.

SFM With BCPii

z/OS Images
(not VM guests)



XCF uses BCPii to

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

Requires operational SE and HMC network

BCPii, Why wait the FDI+ if the system is truly dead?

- BCPii allows XCF to query the state of other systems via authorized interfaces through the support element and HMC network
- Benefits:
  - XCF can detect and/or reset failed systems
  - Works in scenarios where fencing cannot work
    - CEC checkstop or powered down (via SE, not EPO)
    - Image reset, deactivated, or re-IPLed
    - No CF
  - Eliminates the need for manual intervention
  - Prevent human error that may lead to data corruption problems
  - **Reduction in sympathy sickness time**
- Requirements
  - z10 GA2, z196, or z114 with appropriate MCL's; BC12, EC12
  - Pair of systems at z/OS 1.11 or later
  - BCPii configured, installed available
  - XCF has security authorization to access BCPii FACILITY class resources
  - **New version of sysplex CDS**

Recommendation: Set this up. IT IS A CRITICAL COMPONENT OF RESILIENCY

Sysplex Failure Management

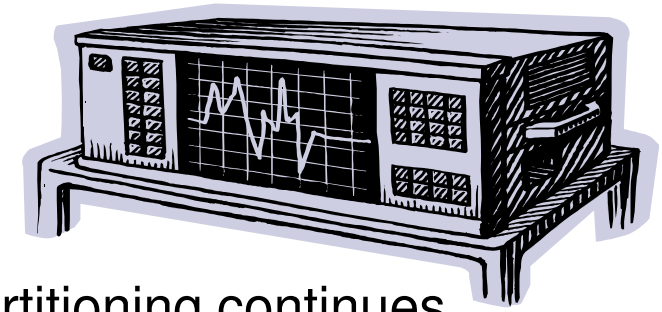
- A Sysplex Failure Management (SFM) policy that implements best practices is a critical component of a resilient sysplex
- A good SFM policy enables automatic, timely, corrective action to be taken when applications or systems appear to be causing sympathy sickness
- SFM is your backstop that protects your sysplex when your operators and/or your automation are inattentive, unable, or incapable of resolving the problem
  - Every SFM parameter was created in response to actual incidents
  - You have full control over how quickly SFM reacts
  - It is vitally important to have the backstop in place

Sysplex Failure Management

- Define an SFM policy to help meet your availability and recovery objectives
  - Applications or systems are not permitted to linger in an extremely sick state such that they adversely impact other systems in the sysplex
  - Applications or systems are not terminated prematurely
  - SFM settings may also vary depending on if there are operators continuously monitoring systems or if operators must be paged
- A suitable SFM policy is but a component of a resilient sysplex. You must still:
  - Ensure no hardware or software single points of failure
  - Have sufficient redundancy to allow for recovery
  - Sysplex enable workloads
  - Workload balancing

System Not Updating Status, System Not Sending Signals

- ISOLATETIME(x)
  - X seconds after the FDI exceeded fencing is initiated by all systems
    - Fencing commands sent via the coupling facility to target system
    - I/O is isolated
      - No new I/O is initiated
      - Any ongoing I/O is terminated



- After fencing completes successfully, sysplex partitioning continues
  - Other systems in the sysplex clean up for system that was removed
  - Shared resources are released
- If fencing fails IXC102A is issued
  - Operator must reset the image and respond down to IXC102A

Recommendation: ISOLATETIME(0)

(As of z/OS 1.11, the default if not otherwise specified in SFM policy)

Health

System Not Sending Signals, System Updating Status

- System delays, performance issues, device/CF issues
  - Stalled I/O restarts, no buffer conditions, response times
  - SFM has nothing for these issues
  - Manual intervention to diagnose and repair
- Loss of signal connectivity
  - CONNFAIL(YES)
    - SFM determines sets of systems that do have full signal connectivity
    - Selects a set with largest combined system weights
    - Systems in that set survive, others are removed
    - To ensure CONNFAIL makes the best decisions for the sysplex, ensure the weights assigned to each z/OS system adequately reflect the relative importance of the system
  - CONNFAIL(NO)
    - Operator prompted with IXC409D to determine which system to terminate

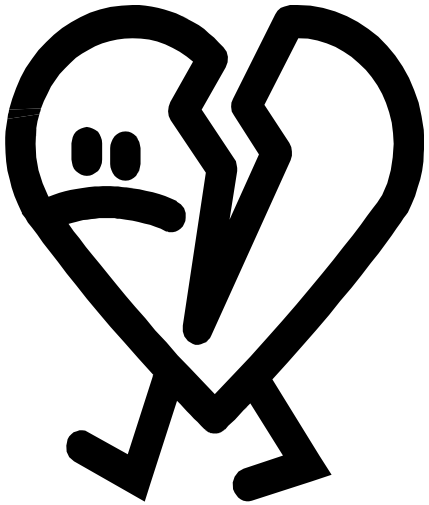
Recommendation: CONNFAIL(YES)

Exception: CONNFAIL(NO) for GDPS environment

Health

System Sending Signals, System Not Updating Status

- SSUMLIMIT(x)
  - Indicates the length of time a system can remain in the state of not updating the heartbeat and sending signals, aka, the amount of time a system will remain in a “semi-sick” state.
  - Once the SSUMLIMIT has been reached the specified action will be initiated against the system
    - ISOLATETIME(0)



Recommendation: SSUMLIMIT(900)

Health

MEMSTALLTIME

- Enable XCF to automatically take action when XCF signals are backing up to the point of adversely impacting other systems in the sysplex
- XCF action: terminate the stalled member that is consuming the highest quantity of buffer space related to the problem

Recommendation: MEMSTALLTIME(600-900)

CFSTRHANGTIME

- Enable XES to automatically take action if a connector does not respond to a structure event in a timely fashion
- XES corrective actions:
  - Stop rebuild
  - Force user to disconnect
  - Terminate connector task, address space or system
  - RAS: ABEND026 dumps collected

Recommendation: CFSTRHANGTIME(900-1200)

Health

CFRM - MSGBASED

- Minimize serialized writes to the CFRM CDS by enabling one system to be the manager to coordinate structure recovery / rebuild protocols
  - Reduces duration of structure rebuild and duplex failover
  - With z/OS 2.1, “serial rebuild” further reduces time

- Enable MSGBASED
  - Format CFRM CDS
    - ITEM NAME(MSGBASED) NUMBER(1)
  - SETXCF START,MSGBASED - switch occurs when there are no events outstanding for a structure
    - Events - connect, disconnect, rebuild events .. the reasons XES reaches out to connectors to a structure.

Recommendation: Leverage MSGBASED processing.

Serial Rebuild – influencing priority order

- **You have some input as to what order structures will be selected for processing during LossConn Recovery**
  - **Presumably this would relate to the order in which you want your business applications to be restored to service**
- **Optional RECPRTY(value) specification in CFRM Policy**
  - **Decimal value in the range 1 to 4, default is 3**
  - **Low numbers imply rebuild sooner, high numbers later**
  - **Takes effect immediately when policy activated**
- **Order is determined by:**
  - **RECPRTY specification**
  - **"Distance" from completion**
  - **Lock structures**
  - **Other structures**

ISGLOCK should have RECPRTY(1)

Automatic Restart Manager (ARM)

- Use ARM policy (or automation) to quickly restart failed elements "in place" when they fail on a running system
- Use ARM policy or automation to quickly restart failed elements "cross systems" when a system fails.
  - Extremely important for subsystems
  - Process the logs and release the retained locks (DB2)

Exception: Only use “in place” for GDPS environment

Provide Automation for Critical Sysplex Functions

- Implement GDPS automation for:
  - General sysplex automation (respond to or alert operations for critical messages and prompts)
  - Management of DASD replication sessions and hyperswap coordination
  - Disaster recover failover coordination/orchestration
  - Planned site switch failover/fallback and DR testing
  - Etc.
- Implement reliable restart recovery via automation (for middleware products that require such restart to free up sysplex resources via log recovery performed at restart time)
  - DB2, IMS, VSAM RLS restart processing
  - Restart in place on same system, and cross-system restart on other system
  - Via system automation or ARM (Automatic Restart Manager)
  - HA for the shared data cannot be maintained in failure scenarios without reliable, rapid restart for these middleware components

Comprehensive testing

Does it work?	**Can it survive?**	**Safely change?**
▪ Unit test	Load test	▪ Installation test
▪ Function test	Failure test	▪ Backout test
▪ System test	Recovery test	
▪ Integration test		

For fast recovery, must verify ability to detect failure and restore the service.

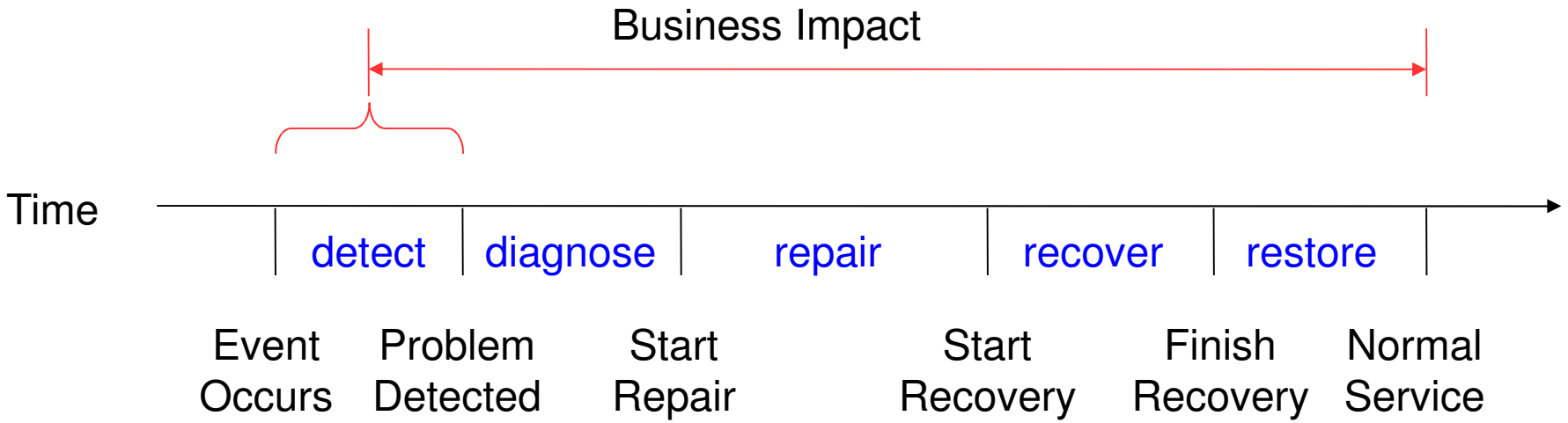
Configure suitable test environments

- To test fast recovery, must be able to inject failures into the environment
  - Obviously, such testing must be done in an environment that is NOT production
- Want to verify all aspects of fast recovery
  - Detection of failure
  - Ability to accomplish recovery procedure
  - Confirm business service is operational
  - Whether time objectives were met
- Environmental considerations
  - Component failure functionality
  - Production like workload
  - Stress

Resourceful

We could not mask the failure. Fast recovery failed.
We have an outage. Now what?

Be prepared to minimize duration of business impact through manual intervention



Ensure that staff has necessary skills, training, and resources:

- Experience dealing with failures
- System knowledge
- Documented procedures
- Rehearsed procedures
- Failure testing

Consider ways for the system to help the staff resolve the issue quickly:

- Comprehensive monitoring
- Automated alerts
- Automated recovery procedures
- Diagnostic tools

Be prepared

- Test environment
- Practice
- Analyze, proactively move towards “fast recovery”
- Automated alerts
- Automated procedures
- Diagnostic tools

Root Causes of Sysplex Performance Problems

- **Real storage**
  - Change in workload
  - Defect
  - Poor configuration
- **CF slow downs**
  - Service time degradation
  - Not meeting expectations
  - More requests going ASYNC
  - New configuration
  - Increase workload
- **CPU**
  - Not enough CPU to handle workload
  - Capping LPARs
  - All LPARs running hot at the same time and there are vertical lows

Virtual storage shortages

SRBs Looping

Contention for global resources

Spin loops

IO delays

CDS access delays
Logger offload hangs

The Back Up Plan – Notify System Programmer

- If the resiliency options are not implemented then operators must be engaged to assess the situation and determine which actions to take

- Call for help as quickly as possible

The Back Up Plan – Notify System Programmer

Message	System Programmer Action
IXC102A	Reset system and respond DOWN immediately.
IXC402D	Reset system and respond DOWN immediately.
IXC409D	Assess status of systems Respond with the name of the system to be removed
IXC426D	System is sending signals but not updating its heartbeat. Investigate swiftly and react before sysplex sympathy sickness ensues. Respond with the system to take down if unable to resolve immediately.
IXC631I IXC633I IXC635E IXC636I IXC640E	Investigate stalled members, pursue recovery options which include termination of stalled members.

Recommendation: Leverage resiliency options, ISOLATETIME, SSUMLIMIT, CONNFAIL and MEMSTALLTIME.

The Back Up Plan - Notify System Programmer

Message	Suggested Action
IXL040E IXL041E	Determine why connector has not responded. Consider terminating the connector. If the hang exceed 2 minutes ABEND026 RSN08118001 dump will be taken. Open a PMR to the application failing to respond.

Recommendation: Leverage CFSTRHANGTIME

Notify System Programmer

Message	Suggested Action
IXC518I	XCF not using CF xyz *Normal when a CF is being removed from a sysplex Action: D XCF,CF and D CF to determine which CFs are physically and logically available, recover as needed
IXC101I IXC105I	Partition has started for a system Partition has completed for a system *Normal when a system has been varied out of a sysplex Action: Collect a standalone dump if the system was removed unexpectedly by SFM

Notify System Programmer

Message	Suggested Action
IXL008I	Path to CF invalidated (links miscabled) Action: D CF to determine if corrective action for the CF paths needs to be taken.
IXL044I	IFCCs for a coupling facility were detected. Action(s): Consider collecting a nondisruptive dump of the CF while the problem is occurring. Also consider collecting dumps on all systems in the sysplex. Contact the IBM Hardware Support Center. SLIP SET,ACTION=SVCD,MSGID=IXL044I, JOBLIST=(XCFAS),DSPNAME=('XCFAS'.*), SDATA=(ALLNUC,CSA,PSA,LPA,LSQA,NUC,RGN,SQA,SUM,SWA,TRT,XESDATA,COUPLE), REMOTE=(DSPNAME,SDATA,JOBLIST),END

Notify System Programmer

Message	Suggested Action
IXL045E	XES SRBs encountering delays. Action(s): Determine if the system is overburdened and resolve the bottleneck. Consider taking a dump while the condition is occurring and contact the IBM Software Support Center (compid 5752SCIXL). DUMP COMM=(IXL045E) JOBNAME=(XCFAS,impacted_job),DSPNAME=('XCFAS'.*), SDATA=(ALLNUC,CSA,PSA,LPA,LSQA,NUC,RGN,SQA,SUM,SWA,TRT,XESDATA,COUPLE), REMOTE=(SYSLIST=*('XCFAS','impacted_job'),DSPNAME,SDATA),END Slip to capture dump upon recreate: SLIP SET,ACTION=SVCD,MSGID=IXL045E, JOBLIST=(XCFAS),DSPNAME=('XCFAS'.*), SDATA=(ALLNUC,CSA,PSA,LPA,LSQA,NUC,RGN,SQA,SUM,SWA,TRT,XESDATA,COUPLE), REMOTE=(DSPNAME,SDATA,JOBLIST),END
IXL158I	Path to CF not operational Action: Verify the desired configuration for that path, take action as needed. Consider collecting a nondisruptive dump of the CF while the problem is occurring. Also consider collecting dumps on all systems in the sysplex. Contact the IBM Hardware Support Center. SLIP SET,ACTION=SVCD,MSGID=IXL158I, JOBLIST=(XCFAS),DSPNAME=('XCFAS'.*), SDATA=(ALLNUC,CSA,PSA,LPA,LSQA,NUC,RGN,SQA,SUM,SWA,TRT,XESDATA,COUPLE), REMOTE=(DSPNAME,SDATA,JOBLIST),END

Summary

Summary of configuring for high availability

- Reliable
  - Test environment
- Resilient
  - Sufficient redundancy
  - Cloning
  - Workload routing
  - Application architecture

Recoverable

- SFM with BCPii
- SFM policy
- CFRM MSGBASED
- Automation

Resourceful

- Test environment
- Automated alerts

A Resilient Sysplex

- Redundant Hardware components
  - Processors, CFs, Links, IO, Storage
- Redundant Software components
  - Including middleware and application components
- No single points of failure
  - Find and eliminate workload SPOFs/affinities
- Dynamic routing capabilities for all workload
  - No static routing affinities
- Data sharing for all critical data
  - No data-related affinities
- Automated Recovery / Restart
  - Aggressive sysplex automation and alerting

Production-like Volume Testing

Separate from production sysplexes

Good operational processes and procedures, especially around sysplex problem determination and recovery

Take advantage of latest tools and technology

Simplification/cloning of the environment – symmetry and “anything runs anywhere”

Sufficient failover capacity for recovery

“White space” and/or automated CoD

Processors, memory, I/O

z/OS capacity and CF/link capacity

Maintenance strategy

Stay up to date, avoid defect rediscoveries

Health Checks with follow-up

Appendix

Check list for HA sysplex

Sysplex HA Recommendations

1. Eliminate workload affinities so that any work can run on any of the multiple supporting subsystem instances
   - Affinities are static bindings where work can only run in one system or in one subsystem instance
   - Even the most resilient sysplex infrastructure is ineffective if workload affinities exist, and the subsystem instance hosting the affinity is down
     - SPOFs still exist even though there are multiple subsystem instances
2. Eliminate workload routing affinities that prevent workload from being routed to any subsystem instance
   - WLM workload routing/balancing will be ineffective unless work can freely be routed to any of the supporting subsystem instances
   - Make use of shared queues and sysplex-enabled routing support such as Sysplex Distributor and MQ or IMS shared queues
3. Eliminate partitioning of data using DB2, IMS DB, or VSAM RLS data sharing, so that any work can run locally on any data management subsystem instance, and access the required data

Sysplex HA Recommendations ...

4. Ensure that all critical sysplex messages are monitored by automation, and either acted upon directly or “alerted” to operations by automation
   - Sysplex hangs and “sympathy sickness” can result when these critical messages are not acted upon quickly and correctly
5. Ensure that operational procedures are in place and understood to take quick action when critical sysplex messages are alerted to operations
   - Use of modern problem determination technology: zAware, RTD, PFA, Fault Analyzer, etc.
   - Documented or automated/scripted recovery procedures
6. Modify recovery operations procedures to attempt recovery at the most granular level possible – attempt hot/warm starts, subsystem recycles, “group restarts” and other granular actions before resorting to system IPL (or multi-system/sysplex-wide IPL)
   - Minimize the scope of the impact of a problem when it occurs
   - The more automated the recovery, the better

Sysplex HA Recommendations ...

7. Implement automated same-system and cross-system restart procedures for DB2/IRLM and other critical subsystems, using either ARM or system automation
   - Sysplex recovery for some data sharing subsystems is **restart-based**, with sysplex-wide retained locks not released until the subsystem restarts, often causing sympathy sickness until resolved
8. Implement Metro Mirror synchronous replication and Hyperswap for all critical DASD volumes
   - Not so much as a disaster recovery mechanism (though it is used for that too), but for HA to avoid a SPOF on storage volumes
9. Implement BCPii services, including XCF use of BCPii for the SFM System Status Detection (SSD) partitioning protocol
   - Improved / more automatic removal of unresponsive systems from the sysplex

For more information

- Forbes: Reputational Impact of IT Risk
  - http://www-01.ibm.com/common/ssi/cgi-bin/ssialias?infotype=SA&subtype=
- Aligning IT with Strategic Business Goals
  - http://www.ibm.com/midmarket/it/it/att/pdf/it_IT_business_continuity_BCRS
- Redbooks
  - 2004 Achieving the Highest Levels of Parallel Sysplex Availability (SG24-6061)
  - 2004 Parallel Sysplex Application Considerations (SG24-6523)
  - 2010 System z Mean Time to Recovery Best Practices (SG24-7816)
  - 2011 System z Parallel Sysplex Best Practices

References

- Mission: Available white paper – (trying to get this restored to TechDocs)
  www.ibm.com/support/techdocs/atsmastr.nsf/WebIndex/WP101966
- System z parallel sysplex best practices -
  http://www.redbooks.ibm.com/redbooks/pdfs/sg247817.pdf
- z/OS parallel sysplex configuration overview -
  http://www.redbooks.ibm.com/redbooks/pdfs/sg246485.pdf
- Achieving the highest levels of parallel sysplex availability -
  http://www.redbooks.ibm.com/redbooks/pdfs/sg246061.pdf
- Achieving the highest levels of parallel sysplex availability for DB2 -
  http://www.redbooks.ibm.com/redpapers/pdfs/redp3960.pdf
- Parallel sysplex application considerations -
  http://www.redbooks.ibm.com/redbooks/pdfs/sg246523.pdf

References

- Communications Server HA - http://www.redbooks.ibm.com/redbooks/pdfs/sg247898.pdf
- z/OS automatic restart manager - http://www.redbooks.ibm.com/redpapers/pdfs/redp0173.pdf
- GDPS concepts and capabilities - http://www.redbooks.ibm.com/redbooks/pdfs/sg246374.pdf

Customer References

- Exploiting parallel sysplex: A Real Customer Perspective - http://www.redbooks.ibm.com/redbooks/pdfs/sg247108.pdf
- Fidelity reference architecture for system z - http://www.redbooks.ibm.com/redbooks/pdfs/sg247507.pdf