Parallel Sysplex Availability Checklist

June 2020

Authors:
David Raften
Gene Sale

Contents

Table of Contents

Introduction

The increasingly competitive global economy is driving the need for higher and higher application availability. Both IBM and independent consultants have testified to the ability of Parallel Sysplex® to deliver near-continuous availability. As a result, many customers are turning to Parallel Sysplex to help them deliver the availability their businesses require. Continuous application availability for zSeries□ applications cannot be achieved without the use of Parallel Sysplex. However, Parallel Sysplex on its own cannot provide a continuous application availability environment. Continuous or near-continuous application availability can only be achieved by properly designing, implementing, and managing the Parallel Sysplex systems environment. This document provides a checklist of items for achieving near-continuous application availability and should be used as a guide when creating a high availability Parallel Sysplex.

Establishing availability objectives

Designing and implementing a continuous availability solution to meet your business objectives is not an easy task. It can involve considerable effort and expense - there is a direct correlation between the level of availability achieved, and the cost of providing that availability, with the costs increasing steeply as you approach 100% availability. Customers need to make intelligent and informed business decisions when designing or configuring z/OS® Parallel Sysplex clusters to attain high availability. When determining the "right level" of availability to configure for, customers should assess the business costs and risk associated with outages to mission-critical applications against the costs of attaining the desired level of availability. The following checklist will help you assess your businesses availability requirements:

- Perform a business impact analysis:
  - Identify all critical applications.
  - Determine the availability requirements for each application (hours of operation).
  - What are the application's recovery requirements (maximum recovery time).
  - Determine costs of an application outage (loss of revenue, customers, business).
  - Determine application dependencies (resources, other applications, and so on).
- Identify requirements to extend service hours:
  - Internet services or applications.
  - Time zone considerations.
  - 24-hour services.
  - Others.

The following checklist will help you assess your IT availability requirements:

- Define Service Level Objectives for applications:
  - Hours of Service.
  - Maximum number of outages that can be tolerated.
  - Application recovery times.
  - Minimum/maximum application response times.
  - Batch windows (turnaround times).
  - Capacity requirements.
- Identify components that need special focus or improvements:
  - Perform a Component Failure Impact Analysis study to identify and evaluate all the components involved in supporting a given application.
  - Perform an Outage Analysis:
    - ✓ Analyze the time lost through planned outages. This should include the time required to do backups, database reorgs, software installations, and so forth.
    - ✓ Analyze the time lost through unplanned outages. Root cause analysis should be done for each outage to identify:
      - The components whose failure caused the outage.
      - How did this defective component get into my system?
      - What process changes need to be made to prevent that from happening again?
      - How to detect the problem sooner?
      - How to react sooner?

- How to restore service faster?
  - How to reduce scope of impact?
- Identify maintenance requirements:
  - Minimum and maximum service times for hardware/software.
  - Concurrent maintenance or downtime requirements.

When you have completed this exercise, you should have a good understanding of the availability requirements of your various applications, and the cost of an outage for each one. This information will help you configure your applications and software and hardware to meet your availability requirements and quantify the cost constraints within which you must operate.

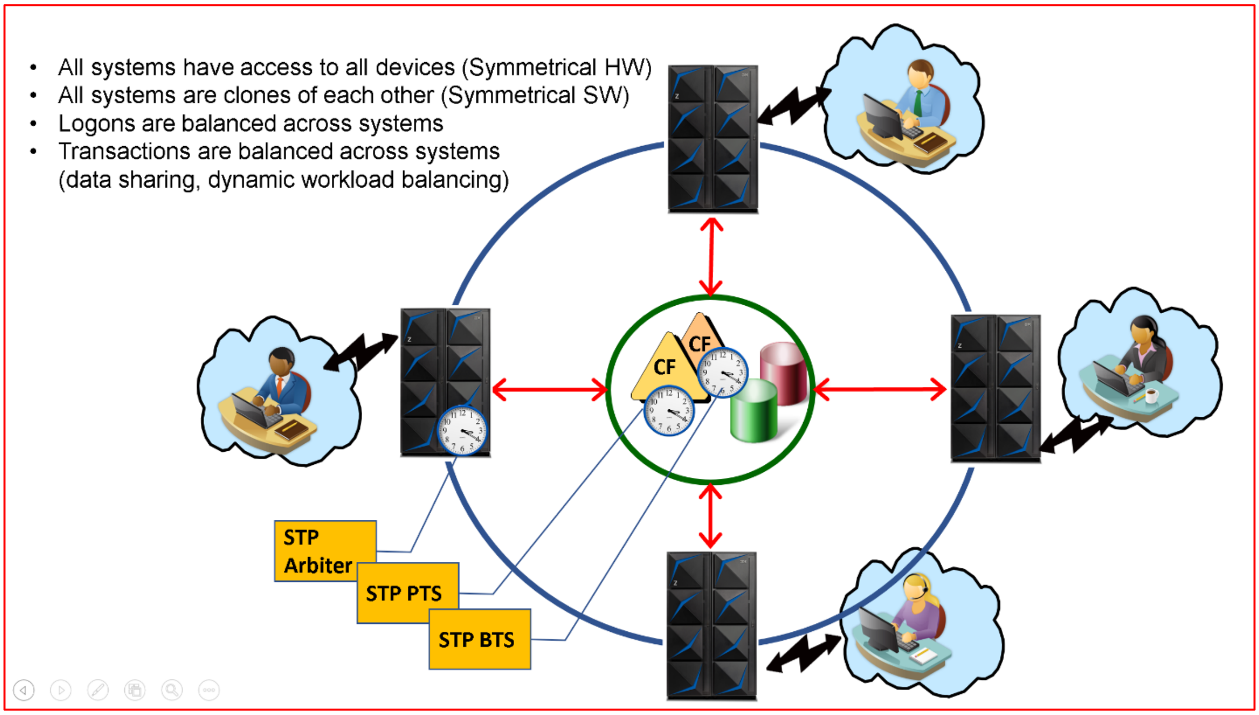
Designing a high availability Parallel Sysplex

Even in a well-designed system environment, the facilities used to provide applications to end users - hardware and software - *will* be unavailable at some time. The lack of availability may be caused by a change to increase capacity, add additional function, or to apply preventative maintenance, or it may be caused by a failure.

The overall objective of designing a Parallel Sysplex for high availability is to create an environment where the loss of any single component does not affect the availability of the application. This is achieved by designing a fault-tolerant systems environment where:

- Hardware and software failures are masked by redundant design.
- Hardware and software systems are configured symmetrically, thereby enabling workloads to run anywhere in the sysplex.
- Workloads and logons are balanced among systems by cloning applications and utilizing high availability features such as data sharing.
- Recovery events are automated so that when a failure does occur, end-user impact is minimized by fast restart mechanisms.

Example of a four-way Parallel Sysplex:



- All systems have access to all devices (Symmetrical HW)
- All systems are clones of each other (Symmetrical SW)
- Logons are balanced across systems
- Transactions are balanced across systems (data sharing, dynamic workload balancing)

The above figure shows a configuration of four z/OS images in a Parallel Sysplex, configured for availability with data sharing and workload balancing. In the event of a complete system failure, the absence of a single point of failure provides continued application availability and enables end-user logons and transaction workloads to be transferred to the surviving systems in the sysplex.

In the remainder of this document, we discuss how to configure your computing environment, hardware, and software for maximum availability. We start with the most general, the computing environment, and move on to the most specific, an individual software feature. Each section should be considered to be a prerequisite to the subsequent ones - there is little point defining duplex page data sets to protect yourself from DASD failures if your whole installation loses power on a weekly basis!

Maximizing computer environment availability

The high availability design point of Parallel Sysplex is based on the ability of a *collection of systems* to tolerate the failure of a single system or subsystem. Therefore, *global* single points of failure must be eliminated. For example, if the entire sysplex is being fed by a single power source and the power source fails, the entire sysplex will fail, resulting in poor availability.

One of the most basic requirements for continuity of service is that the computing hardware is operating in a safe and secure environment, within the environmental parameters specified by the manufacturers, and with a power supply that meets the requirements of that equipment.

The following checklists cover the areas of power, cooling, and computer room design.

Power

- Provide an uninterruptible or redundant power supply for critical components:
  - Use a local UPS system with motor generator and battery backup
  - Implement dual power feed distribution from separate electric utility power grids.
  - Ensure that there are no single points of failure in the redundant power configuration (that is, use separate circuit protectors, separate utility towers/poles, enter the building at different points, and so on).
  - In the event of a power interruption, switchover to alternate power source must be automatic and nondisruptive.
- Ensure that critical components have redundant power connections:
  - Select components that have dual power connections.
  - Connect redundant external power connections to separate AC power circuits to ensure that hardware functions are not affected by external power interruptions.
- Connect critical components to redundant power sources:
  - Central Electronic Complexes (CECs)
  - Coupling Facilities
  - FICON switching devices
  - Network Controllers
  - Peripheral and other devices (DASD, tape, monitors, etc.)
  - Environmental equipment (heating, cooling, humidity controls, monitors, lighting, etc.)
  - Dense Wavelength Division Multiplexers (DWDMs)
- Establish procedures to test failover to the redundant power source on a regular basis

Cooling

- Provide sufficient cooling capacity and redundancy to ensure that environmental conditions can be maintained in the event of an equipment failure:
  - Install redundant or multiple cooling units.
  - Provide for automatic monitoring of air temperature and humidity and ensure that switchover to backup systems is automatic if the environment thresholds are exceeded.
  - Establish procedures to specify actions to be taken in the event of an environmental failure.

Building/Computer room design

- The building and machine room that houses your systems should be secure:
  - Access to the building and/or computer room should be controlled and monitored.
  - All critical components and devices should be located within the secure area.
- Ensure that equipment is properly installed and labeled:
  - Computer equipment should be installed according to manufacturer's specifications.
  - All equipment should be properly labeled to provide for quick/easy identification.
  - Use a cable management system and ensure that cables are properly routed and labeled.
- Establish a business contingency plan to protect the business in the event of a disaster or in event of any prolonged outage. (The building represents a single point of failure):

  For near-continuous availability, establish a second site located several miles from the original site, and implement GDPS® to manage the sites. More information about GDPS can be obtained on the Web at: https://www.ibm.com/it-infrastructure/z/technologies/gdps .
  - If applications can sustain a prolonged outage without jeopardizing the business, provide a hot site or contract with a disaster recovery provider to provide business continuance services.
  - Disaster recovery is a complex and intricate topic in its own right, and increasing in importance as applications grow ever more complicated, and the demands for availability continue to increase. More information about disaster recovery planning and GDPS may be obtained from "*IBM GDPS Family: An Introduction to Concepts and Capabilities*" available in http://www.redbooks.ibm.com/abstracts/sg246374.html?Open .

Configuring hardware for availability

Having provided a secure and well-designed computing environment, the next step is to configure your hardware for maximum availability. This starts with selecting components that have high availability characteristics, and continues with configuring and connecting those components to provide the availability that you require.

Some hardware components are critical to the entire sysplex, whereas other hardware components may be utilized by a subset of systems, a single system, a subsystem, a particular application, and so on. Hardware failures can therefore have a varying degree of impact on system availability. Hardware components that support mission-critical workloads must be configured redundantly to avoid outages.

In this chapter, we discuss the availability considerations for processors, DASD, and other critical devices.

Availability features

When deciding which servers a high availability application should run on, you should review the following list of some of the availability features and try to select a server that has the features you require. Some of these features are standard, some are automatically enabled if present, and some are optional:

- Independent dual power feeds.
- N+1 power supply technology.
- N+1 cooling.
- Concurrent maintenance and repair.

- Concurrent LIC upgrades.
- Dynamic I/O reconfiguration management.
- Hot plugging of FICON® channels, CF links, OSA adapters.
- Nondisruptive replacement of I/O.
- Multiple Image Facility (MIF).
- High Performance FICON (zHPF)
- Console Integration Feature.
- Transparent CPU sparing (CP/SAP/ICF/zIIP).
  - SAP reassignment.
- Dynamic Coupling Facility dispatching.
- Dual cryptographic coprocessors.
- RAIM technology for memory
- Plan Ahead for nondisruptive upgrades:

Capacity Upgrade on Demand (the ability to non-disruptively bring additional CPs online).
Concurrent conditioning for hot plugging new I/O components (the pre-installation of supporting hardware to allow additional channels to be installed non-disruptively).
Capacity backup (the ability to quickly and non-disruptively bring additional CPs online on a temporary basis, to assist in recovery from a disaster).

- CICS® Subsystem Storage Protection.
- CICS Subspace Group Facility.
- Logical Partitions.
- Multiple processors.
- Enhanced LPAR dynamic storage reconfiguration.
- Automatic Reconfiguration Facility (ARF).
- Automatic I/O Interface Reset Facility.
- System Isolation Function via Sysplex Failure Manager (SFM).
- Auto-IPL
- HyperSwap®
- Capacity Provisioning Manager

Configuring for availability

Even with all the availability features on modern processors, there is still the possibility that a processor might fail. However, even if a box is 100% reliable, there will still be occasions when you will wish to remove it from the sysplex, possibly for an upgrade or to apply preventative service. Therefore, to ensure the continued availability of applications running on the affected processor, you must configure your processor resources redundantly. This section provides a checklist of recommendations to help you ensure your applications can continue to run across the outage of a given processor.

Servers

Servers are those processors that run z/OS images. We use the term *server* to differentiate these processors from the ones that only run Coupling Facility images.

- Configure a minimum of two CPCs or two LPARs on different CPCs to ensure continuous application availability in the event of a CPC failure or outage.
- Verify that sufficient CPC capacity (storage, MIPS, and channels) exists to support peak critical workloads.

- Ensure that each CPC has sufficient “extra capacity” to handle fail-over workloads in the event of a CPC or system outage:

Storage. And don’t forget the impact on virtual storage (especially CSA/ECSA) of taking over additional workload.

Processor cycles.

Channel bandwidth.

Connectivity to the required resources.

Plan to sustain workloads in the case where the largest z/OS system or largest CPC fails.

Consider implementing automatic shutdown of discretionary workloads as an alternative to duplicating the hardware resources used by your largest system.

Make sure there is sufficient available capacity to handle the spike in activity during recovery processing. This capacity must be available immediately and should not rely on some workloads being manually stopped or quiesced.

- Identify expendable workloads that can be displaced by production work in the event of a failure, and configure WLM to ensure that critical workloads get the service required according to your business requirements.
- For Logical Partitions, ensure that the following LPAR definitions are set:

Enable the Automatic I/O Interface Reset Facility if you are not using Automatic Reconfiguration Facility (ARF). If you *are* using ARF, you should not use the Automatic I/O Interface Reset Facility as it is incompatible with ARF. ARF and the Automatic I/O Interface Reset Facility are described in the System Overview manual for the relevant processor.

Ensure that storage and LPAR weight definitions are correct for your environment.

Do not configure an extremely low weight LPAR in the sysplex as it may impact performance. The low weight LPAR may not be able to keep up, which induces delays for peers. From an availability and/or fast recovery perspective, that low weight system may be the one driving the recovery. So you may not achieve your SLA’s because the recovery process intended to restore service is running too slowly.

- Give serious consideration to configuring two images from the sysplex on each server. Depending on your LPAR definitions, this may give you the ability to continue to utilize all the available MIPS on the processor even if one of the images has an outage (planned or otherwise).

Coupling Facilities

- Install at least two CFs to ensure that structures can be rebuilt in an alternate CF in the event of a CF failure. The inability to rebuild critical structures in the event of a CF failure could result in a sysplex-wide outage. Don’t forget to specify at least two CFs on every preference list in your CFRM policy, to ensure automatic rebuild can take place.
- If your applications genuinely require complete 24x7 availability, you should consider configuring a third CF. This is to provide fallback for the single remaining CF while one CF is down for maintenance. This implies that you either configure all of the CFs with enough links, MIPS, and storage to handle *all* the structures, or else setting up your CFRM preference lists in such a way that only critical structures will get rebuilt into the third CF if both of the other CFs are unavailable.

Ensure that sufficient “extra capacity” exists on each CF to allow for structure rebuild and movement of CF workloads in the event of CF failure or to support CF maintenance activities. Plan to sustain workloads in the case where the largest CF fails.

  - ✓ Storage. The Coupling Facilities should be configured with enough storage so that the total requirement of all allocated structures can be accommodated by the remaining CF if one CF experiences an outage.

- ✓ Processor cycles.
- ✓ CF link bandwidth and connectivity.

Depending on the CF exploiters in use, it may be possible to reduce the backup "extra CF capacity" needed to support REBUILDs. For example, the JES2 checkpoint structure can be forwarded to DASD instead of an alternate CF.

- Plan ahead to ensure you have the required CF Levels installed. Some functions such as Asynchronous Duplexing require a specific CF Level and/or z/OS level. As upgrading the CF Level requires an outage on the CF, you should ensure the new level is installed before you need it, and as part of an already-planned outage if possible. Plan to have at least two CFs at the required level, to allow you to move the structures if necessary.
- There is no formal coexistence policy for CFs. However, the hardware that the CFs run on need to be at (n, n-2) coexistence.
- Install redundant coupling links from all CFs to all CPCs:

Ensure that there are no single points of failure in the redundant link configuration. For example, ensure that redundant links are connected to separate driver cards so that a card failure does not result in complete loss of connectivity.

If using two providers, verify one provider is not leasing the same fiber as the other.

- Evaluate the use of Standalone Coupling Facilities versus Integrated Coupling Facilities (ICF). More information can be found in ***Coupling Facility Configuration Options*** at: https://www.ibm.com/downloads/cas/JZB2E38Q:
  Certain CF exploiters require failure isolation. Failure isolation means that CFs and the z/OS systems they are connected to reside on different CPCs. Ensure that failure isolation is maintained for CF exploiters that cannot tolerate concurrent CF and z/OS failures. Without this, the applications may be forced to do log based recovery which takes longer. Frequent commits may help reduce the time needed for log based recovery.

Consider System Managed CF Structure Duplexing to remove the single point of failure issue for structures requiring failure isolation.

- ✓ Asynchronous Duplexing should always be used for Lock Structures. As of 2020, this is only supported for Db2® lock structures.

More information on Duplexing can be found at: **System-Managed CF Structure Duplexing**
https://www.ibm.com/downloads/cas/YAZZADAB

Standalone Coupling Facilities (running on a single CPC with dedicated CPs, containing no z/OS images) have an advantage from a recovery standpoint.

The use of an ICF running in a CPC, where none of the z/OS images are in the same sysplex as the CF, is functionally equivalent to a Standalone CF. For example, placing a production CF in a CPC that contains only development images, none of which are in the same sysplex as the CF, would be equivalent to using a Standalone CF.

- Carefully plan structure placement, sizing and recovery. See section **Sysplex Policie**s for additional considerations.
- Ensure that the Coupling Facility LPARs are properly defined:

Power save, rideout and volatility settings. Make sure that these reflect your actual configuration. For example, if you have an external UPS, set the mode to NONVOLATILE, and set the rideout value to reflect the capacity of the UPS.

Storage definitions.

CP resource definitions.

- Monitor the “Coupling Facility and Structure Activity” RMF™ report on a ongoing basis to ensure sufficient capacity and resources are allocated. Enable SMF recording for type 74 records and ensure that RMF Monitor III is active.
- Establish and document operational and recovery procedures:

Document procedures for:

- ✓ Moving structures from one Coupling Facility to another and restoring them back to their original CF. The REALLOCATE option of the SETXCF START command is very useful for this task.
- ✓ Changing CFRM policies and handling “policy change pending” situations.
- ✓ Shutting down and removing a Coupling Facility. The procedure for doing this is documented in the appendix Coupling Facility Maintenance and Shutdown in the z/OS Parallel Sysplex Systems Management manual.
- ✓ Adding a Coupling Facility to the sysplex.

Plan for structure recovery in the event of a CF failure.

When possible, automate operational and recovery procedures to prevent outages due to human or procedural errors.

- Define dedicated Coupling Facility CPs for production CFs. The use of shared engines can impact performance and introduces the risk that abnormal activity in the sharing sysplex could impact the production sysplex. If you are forced into sharing a CP between a production and non-production sysplex, enable Dynamic CF Dispatching (DCFD) with Thin Interrupts (DYNDISP=THIN) for the test sysplex, and give the production LP a sufficiently high weight to ensure it always get the required resource. For optimal performance, DCFD should not be enabled for a production CF.
- Define at least two CPs for the production Coupling Facilities for performance.

The considerations and performance effect of sharing CF engines is discussed in "CF Configuration Options” paper available on https://www.ibm.com/downloads/cas/JZB2E38Q.

DASD

In this section we first provide a list of the high availability features available on many modern DASD subsystems. After that, we provide a checklist of things to consider when configuring and connecting your DASD that will help you achieve the highest possible availability.

Availability features

When deciding which DASD a high availability application should be using, you should review the following list of some of the availability features and try to select a device type that has the features you require. The devices used for the system and sysplex volumes should have the highest availability characteristics, as there is no point having the application data available if the system or sysplex is down because of a DASD problem. Some of these features are standard, some are automatically enabled if present, and some are optional:

- Independent dual power feeds.
- N+1 power supply technology/hot swappable power supplies, fans, disks.
- N+1 cooling.
- Battery backup.
- Storage subsystem cache, to ensure required performance for critical system and Sysplex data sets.
- Nondisruptive maintenance.

- Concurrent LIC activation.
- Concurrent repair and replace actions.
- RAID architecture.
- Redundant microprocessors and data paths.
- Concurrent upgrade support (that is, ability to add disks while subsystem is online).
- Redundant shared memory.
- Spare disk drives.
- FlashCopy®/SnapShot/Concurrent copy.
- Dual copy/mirroring.
- Remote Copy (PPRC, P/DAS, XRC).

Configuring Disk for Availability

Even with all the availability features on modern DASD subsystems, there is still the possibility that a device, or a component in the path from the processor to the device, might fail. Also, there may be times when a device, or a component in the path to that device, may need to be taken offline. Therefore, to ensure the continued availability of applications that are using that device, you must configure and connect your DASD subsystem in a manner that provides the required level of redundancy. This section provides a checklist of recommendations to help you ensure that your applications can continue to run across the outage of any given device. More information can be found in **IBM Z Connectivity Handbook**
http://www.redbooks.ibm.com/redpieces/abstracts/sg245444.html?Open .

- Establish multiple/redundant channel paths to all critical devices to eliminate single points of failure. For redundant channel paths, ensure the following:

Configure paths through different FICON directors.

Be aware of the implications of using EMIF to share channels between LPs. The failure of a single channel can now impact a number of LPs instead of just one.

Ensure that I/O connections for redundant paths are not connected to the same I/O driver card or board at the physical device level.

The actual number of paths is dependent on device type and performance requirements. For Parallel Sysplex, a minimum of two paths to every device is recommended.

Use the recommendations in the Systems Assurance Product Review for the associated processors to distribute the channel paths across multiple channel cards, STIs, and SAPs. This will protect you from a failure in any one of these entities.

- Use the optional availability features of the DASD subsystem (dual copy, remote dual copy) to protect you from HDA failure.
- Maintain I/O symmetry across all systems within the Parallel Sysplex by designing your I/O configuration to appear as one I/O pool that is shared by all images. By maintaining symmetry in your I/O configuration, you will simplify operations and significantly reduce the amount of work required when adding new systems.

Establish naming conventions that are meaningful and flexible.

I/O device numbers must be unique; ensure the number is the same for the same device across all images.

If possible, use the same CHPID numbers across all servers.

Use HCD to define and maintain your I/O configuration. Maintain one logical IODEF for the entire sysplex. For availability reasons, you might decide to keep more than one physical copy, but their contents should all be the same.

- Attach all peripherals and devices to redundant power sources.
- Every time any maintenance activity is carried out on a DASD controller, ensure that the configuration is restored when the maintenance activity is complete (for example, all paths are online, DASD Fast Write and cache are enabled, and so on).
- If running in an LPAR environment, use MIF to reduce channel requirements.
- Use Dynamic I/O Configuration when adding or changing hardware to avoid planned outages.
- Use IBM Z High Performance FICON
- See the additional recommendations in **Sysplex Data sets** relating to data set placement on DASD volumes.

Other critical devices

In addition to processors and DASD, there are other devices that are vital to system availability. These include STP, FICON routers, and consoles. In this section, we include a list of recommendations about configuring and managing these devices for maximum availability.

Server Time Protocol and Precision Time Protocol

Server Time Protocol (STP)

STP is a message-based protocol, similar to the industry standard Network Time Protocol (NTP). STP allows a collection of IBM Z® servers to maintain time synchronization with each other using a time value known as Coordinated Server Time (CST). The network of servers is known as a Coordinated Timing Network (CTN). The mainframe's Hardware Management Console (HMC) plays a critical role with STP CTNs: The HMC can initialize Coordinated Server Time (CST) manually or initialize CST to an external time source. The HMC also sets the time zone, daylight savings time, and leap seconds offsets. It also performs the time adjustments when needed

In a multisystem Parallel Sysplex, any z/OS image that loses timing signals when STPMODE YES is specified in CLOCK(xx) will issue WTOR message IEA394A

> **Note:** In GDPS/PPRC configurations running GDPS the controlling system is allowed to remain running in an unsynchronized state for a limited amount of time to guarantee a consistent set of secondary PPRC volumes and fulfill its role as the driver of GDPS's recovery actions.

STP requires Feature Code 1021.   Amount of time a Stratum 2 or Stratum 3 server can remain synchronized without receiving messages from its clock source is approximately 10 seconds.

Arbiter: Provides additional means to determine if BTS should take over as the CTS under unplanned outages if "OLS" not received by BTS.

- RPQ 8P2263 required for distances greater than 100 km
- Since STP uses coupling links to transmit timing signals, it is often most convenient to use a Coupling Facility as the Preferred Time Server (PTS) as it already has connectivity to the z/OS systems. Timing-only signals may be required to other serves in the timing network such as

another stand-alone CF if they do not have CF link connectivity. STP can use the CF – CF links as required for CF duplexing.

- If there are three or more servers in the timing network, use an arbiter for improved availability if there is a loss of connectivity to the PTS.
- In a two site configuration, locate the Arbiter in the same site as the PTS. The BTS is at the other site.
- In a two site configuration with remote copy, the PTS and Arbiter should be with the primary disk.

Only with GDPS and FREEZE=STOP or SWAP, STOP should the Arbiter be with the secondary disk

- HMC Application level must be equal or higher than the level of Support Element consoles it manages
- All servers and CFs in the Coordinated Timing Network (CTN) are defined objects to any HMC in order to have the capability for CTN reconfigurations
- If a timing-only link already exists between two servers, HCD will not allow defining a CF connection between these servers. Even if not using it now, define a coupling link in the HCD. This will provide better positioning in case the link is needed for CF data in the future.
- Through the IBM z14® and the IBM z15™ with the BPA power option, the Internal Battery Feature (IBF) can optionally be installed. PDU power does not support IBF.   The main purpose of an IBF is to keep a IBM Z server running for several minutes, when a power outage to the data center occurs and there is no Uninterruptible Power Supply (UPS) available. In this time the workload of the server running on IBF can be moved or shut down in a controlled manner. Especially for Coupling Facilities this can prevent costly recovery of structures.

Install the Internal Battery Feature on one or more servers in the CTN. In order to plan for improved STP recovery when power has failed for a single server (PTS/CTS) or when there is a site power outage in a multi-site configuration. If an IBF is installed on an IBM Z server, STP has the capability of receiving notification that external power has failed and the IBF is engaged (CPC running on IBF power). When STP receives this notification from a server that has the role of the PTS/CTS, STP can automatically reassign the CTS role to the BTS, thus automating the recovery action and improving availability.

If the CTN in a single data center has three or more servers, we recommend assigning the Arbiter. The IBF does not provide any additional benefit for server power outages.

If the entire CTN is located in a single data center and only has two servers (PTS and BTS), then we recommend installing the IBF on both the PTS and the BTS. This should provide recovery capabilities when the server that is the CTS experiences a power failure.

- If the CTN spans two data centers, we recommend installing the IBF on the servers that will be assigned the roles of PTS, BTS, and Arbiter. This should provide recovery capabilities when the site where the CTS is located experiences a power failure.

Install the Internal Battery Feature on the server where the PTC/CTS and Arbiter are located. This allows time for the PTS to reconfigure the Backup Time Server (BTS) as CTS.

Recommend also installing the Internal Battery Feature on the BTS to provide protection when the BTS is the Primary Time Server and on Arbiter.

- The list of qualified DWDM vendor products can be found on Resource Link® at: https://www.ibm.com/servers/resourcelink/
  They are listed Under Hardware Products for Servers on the Library page
- Plan for redundant, diverse fiber routes between sites to help avoid a single point of failure of the fiber trunk. Ensure that the vendor qualification document is inspected carefully by a DWDM technical expert.

- When using NTP,

Configure at least one unique NTP server on both the PTS and the BTS. For redundancy, up to two separate NTP servers can be configured for both the Primary Time Server (PTS) and the Backup Time Server (BTS)
Use Pulse per second from the NT server to maintain an accuracy of 10 microseconds as measured at the PPS input of the IBM Z server. By comparison, if STP is configured to use an NTP server without pulse per second, it provides only a time accuracy of 100 milliseconds to the ETS device

- Enable Leap Seconds into the CTN definitions. Leap seconds are used to synchronize the Coordinated Universal Time (UTC) to the international atomic time (TAI). When a leap second is inserted into UTC (when a leap second is needed, it will generally be added at midnight UTC on either June 30 or December 31), UTC effectively jumps backward by one second. If leap seconds are not specified for the CTN, UTC will be one second ahead of the NTP server that STP is synchronizing to when the leap second event occurs. STP will automatically steer to the correct time, but this steering adjustment is made at the rate of seven hours per second to ensure no duplicate timestamps are seen. Customers who do not specify leap seconds will not meet the FINRA clock synchronization requirement during the seven-hour window required for steering to complete. This will be highlighted by message IEA032E issued by the z/OS accuracy monitor at every 60 minute interval until steering has corrected the time to within the ACCURACY threshold
- Plan for reassigning special roles if one of the special role servers (PTS, BTS, Arbiter) is unavailable.
- **SFM allows installation to code a policy to define the recovery actions to be automatically initiated following detection of a Parallel Sysplex failure.** Actions include fencing off the failed image that prevents access to shared resources, logical partition deactivation, or dynamic storage reconfiguration. **SFM to recognize that WTOR IEA394A issued.** WTOR message issued by all the z/OS images in the sysplex, the user is not time constrained to do timing network reconfiguration before replying to IEA0394A. Once WTOR on the first system image responded to with *"RETRY".* **z/OS system images will enter disabled-wait states should the user not be able to respond to the IEA394A WTOR message in the allotted time**. If the message is issued only on a subset of participating sysplex images, the SFM settings specified in the SFM Policy must be considered **WTOR allows time window to correct the problem and respond *"RETRY"* if problem corrected or *"ABORT"* if problem cannot be corrected**
- *"ABORT"* will load wait state 0A2-158
- In SYSx.PARMLIB(CLOCKxx), ACCURACY should be set to 20 for customers that need to meet the FINRA clock synchronization requirement. This will cause IEA032E to be issued when the steering adjustment exceeds 20 milliseconds. To help customers meet the FINRA clock synchronization audit requirement, a message is issued at a user specified interval, with a default of every 60 minutes, giving details of the current status from the PRT correction steering information block.
- MiFID II regulations require trading systems to time stamp key trading event records within 100 microseconds to UTC. This requires using STP with Pulse per Second, or PTP (see below)
- Establish and document procedures to handle time Standard / Daylight Savings time changes

IEEE 1588 Precision Time Protocol

IEEE 1588 Precision Time Protocol (PTP) can be used as an external time source for IBM Z Server Time Protocol (STP) for an IBM Z Coordinated Timing Network (CTN). PTP is designed to allow sub-nanosecond clock synchronization and security improvements over NTP.

The initial implementation will be for PTP connectivity via the IBM Z HMC/SE. At that time there will be no change to the use of STP CTNs for time coordination, other than the potential to use a PTP-based external time source. Future implementation is planned to include full connectivity of an external PTP time source directly to the IBM Z CPC, and re-introduction of the concept of a mixed CTN, with support for traditional STP and native PTP implementations. Beyond that, the goal is to enhance the role of IBM Z machines in a PTP environment that addresses the many governmental regulations and security concerns that clients are facing.

- Use PTP for applications that require tight synchronization, such as in some financial applications

Redbooks®

• Server Time Protocol Planning Guide	SG24-7280
• Server Time Protocol Implementation Guide	SG24-7281
• Server Time Protocol Recovery Guide	SG24-7380
• IBM Z Server Time Protocol Guide	SG24-8480

Consoles

Consoles are a critical resource in a Parallel Sysplex. The complete loss of consoles removes the operator’s ability to control the system. If there is a prolonged loss of consoles, the console address space buffers will eventually fill up causing loss of messages, and giving the appearance that the system has died.

Another reason that the correct console configuration is so important in a sysplex is that the way the systems manage consoles in a sysplex is fundamentally different from how they are managed in a multisystem environment that is not a sysplex. Many installations are not fully aware of the impact of these changes, and have not changed their console configuration accordingly as part of their migration to a sysplex environment.

Multiple Console Support (MCS) consoles are devices that are locally attached to a z/OS system and provide the basic communication between operators and z/OS. MCS consoles are attached to control devices that do **not** support systems network architecture or SNA protocols.

SNA Multiple Console Support (SMCS) consoles are devices that do not have to be locally attached to a z/OS system and provide the basic communication between operators and z/OS. SMCS consoles use z/OS Communications Server to provide communication between operators and z/OS, instead of direct I/O to the console device.

Extended Multiple Console Support (EMCS) consoles are devices (other than MCS or SMCS consoles) from which operators or programs can enter commands and receive messages. Defining EMCS consoles as part of the console configuration allows the system programmer to extend the number of consoles beyond the MCS console limit, which is 99 for each z/OS system in a sysplex.

If automation is used to remotely IPL z/OS, the HMCS console (the Integrated 3270 Console that is located on the Hardware Management Console) must be disconnected from the z/OS partition that is to be IPLed before the IPL is initiated. Failure to do so will prevent automation from receiving messages that are issued during the IPL and prevent automation from taking any appropriate action.

In this section we include a list of recommendations for both hardware consoles and operating system consoles. While these are different resources, they are both critical, and would normally be managed by the same group:

- Hardware consoles:

Ensure that critical hardware consoles (HMCs, ETR, ESCD, etc.) are configured redundantly and/or have standby consoles.
Place hardware consoles on private LANs.
Implement regular backups and establish recovery procedures.

- z/OS consoles are critical components of a basic and Parallel Sysplex. Ensure that they are properly defined and managed:

Ensure that there are a minimum of two MCS or SMCS consoles configured in the sysplex and that there are no single points of failure related to these consoles.
Consoles should be attached to different CPCs, have separate physical screens, separate power supplies, etc.

- For systems that do not have MCS consoles physically attached, ensure that there is a console available to these systems by:

Defining SMCS consoles to the Communications Server (VTAM) to allow an SMCS console to become active on that system.
Ensuring that systems with attached MCS consoles are the first systems IPLed into the sysplex, and the last system to leave the sysplex.

  - ✓ Update your Startup and Shutdown IPL procedures to ensure that systems that have consoles attached are never all (intentionally) deactivated at the same time, to protect against potential “no console” conditions

Providing the ability to physically switch consoles between systems to make an MCS console available when needed.

- Maintain symmetric console device numbers between systems to simplify maintenance and recovery, and to facilitate console switching between systems that do not have dedicated consoles.
- Configure the hardware “system console”. This will allow the system to be operated from the Hardware Management Console (HMC) when necessary. The system console function can be used just like any other console. The performance shortcomings that affected earlier implementations have been removed. .

If only SMCS consoles are used, the system console plays an important role in availability - NIP messages and SYNCHDEST messages will be displayed on the system console. If the SecureWay Communications Server fails and all SMCS consoles become unavailable, the system console becomes the “*console of last resort*” and must be used to reestablish system availability.

- In a sysplex environment, up to 250 MCS, SMCS, HMCS, and subsystem consoles can be defined per z/OS® image with a limit of 99 active MCS, SMCS, HMCS, and subsystem consoles per z/OS image.

Verify that software products (Automation, TSO, NetView®, etc.) using console services are exploiting Extended MCS console support and not '*subsystem*' consoles to avoid exceeding the 99 MCS console limit.

Review console configurations and where possible consolidate consoles that perform duplicate functions to eliminate unnecessary consoles. With the ability to route messages from all systems to a single console, the consolidation of messages reduces the number of consoles required to manage multiple systems.

- Assign a unique name to all MCS and subsystem consoles. (SMCS consoles always require a name).

If consoles are not named, repeated system IPLs may result in exceeding the "99" console limit. Should this happen, console slots already in use may be freed either by using the IEARELCN (OW05419) in SYS1.SAMPLIB, or by performing a sysplex-wide IPL (where all systems are partitioned from the sysplex before any system is reIPLed back into the sysplex).

- Ensure proper setup of SYSx.PARMLIB(CONSOLxx) parameters:

When two or more systems in a sysplex require a CONSOLxx member, you can do one of the following actions:

- ✓ Code a separate CONSOLxx member for each system in the sysplex (the least efficient method).
- ✓ Code a single CONSOLxx member for all systems in your sysplex. Specify parameters with sysplex scope to be used by all systems in the sysplex. Consider using system symbols to represent unique values in the member.

CONSOLxx parameters with sysplex scope are valid only for the first system that enters a sysplex. Because these parameters are ignored by systems that later join a sysplex, you do not need to set them up to specify unique values for different systems in a multisystem environment. For a complete list of parameters in CONSOLxx that have sysplex scope, refer to z/OS MVS Planning: Operations.

If different systems require unique values on parameters that do not have sysplex scope, you can use system symbols to represent those unique values in a shared CONSOLxx member. When each system processes CONSOLxx, it replaces the system symbols with the values it has defined to them.

A named console can be defined on multiple systems, but can only be active on one system at one time.

The system uses default values for the CONSOLxx statements INIT, HARDCOPY, and DEFAULT if you do not code them. If the default values for these statements are acceptable to your installation, do not code them for the systems in your multisystem environment.

If multiple CONSOLxx members are used, ensure parameters are consistent across the sysplex members. For example, if one system has RLIM set to 999, then all systems should have RLIM set to 999.

Specify DEVNUM(SYSCONS) to specify the system consoles attributes when it must be used, for example, if when other consoles have failed. (You don't need the console card for it to be used, but it makes it faster and easier)

Use of MSCOPE=(*ALL) will generate a lot of console messages and system-to-system console traffic. Use MSCOPE=(*) or MSCOPE with a specific subset of systems as much as possible. Use MSCOPE=(*ALL) only on a few, or select group of consoles that really require a sysplex-wide scope. The objective should be to avoid having all the messages being routed to all of the systems in the sysplex.

Use ALTGRPs instead of "ALTERNATE" (SMCS consoles only support ALTGRP).

- Utilize alternate console groups to ensure that MCS and SMCS consoles have backups.

The sysplex only has one "MASTER" (cond=m) console. The sysplex master console is assigned when the first system is IPLed into the sysplex, or if no MCS consoles are available then the first SMCS console with master authority is activated. It is therefore possible for the sysplex master console to change depending on which system is IPLed first. Using ALTGRPs will minimize the risk of the sysplex entering into a "*no master console*" (IEE141A) condition.

Ensure that all MCS and SMCS consoles have backups on other systems in the sysplex.

- Define the SYNCHDEST group to provide predictability in the rare event of a synchronous (emergency, disabled) message. SMCS consoles will be ignored if specified in the SYNCHDEST group.
- Use the following functions for operational efficiency:

Implement message suppression (MPFLST) on all systems. You should suppress from display on the consoles all nonessential message traffic. Suppression rates of 95%, and higher, of message traffic is typical on well managed systems.

Specifying MPFLSTxx members, the MMSLSTxx member, and the PFKTABxx member within the CONSOLxx member makes it easier to maintain these related members of parmlib rather than setting them after IPL with the SET operator command.

Minimize the route codes that are defined to a console to only those that are needed by the console's function. The objective is to avoid having all route codes defined to all the consoles defined to a system.

The Command Prefix Facility (CPF) allows you to enter a subsystem command on any system and the command automatically routed to the system in the sysplex that supports the command without any need for the operator to know what system that is.

Use CMDSYS to automatically route commands to a target system other than the system to which the console is attached. This reduces the need to have consoles configured on all the systems in the sysplex.

SYNCHDEST messages for synchronous WTO/Rs (also know as disabled WTO/Rs) cannot be sent to other systems in the sysplex. You should configure your hardware system console to receive these messages if you do not have local consoles attached to this system.

The ROUTE command can be used to direct commands to other systems in the sysplex when necessary, avoiding the need to have consoles defined on each system in the sysplex.

When possible, Extended MCS consoles should be defined a DOM(NONE). If Extended MCS console has no need to see DOMs performance is improved if it does not receive them.

The Action Message Retention Facility (AMRF) saves action messages across the sysplex in order to retrieve them later via a 'D R' command. This list of messages is scanned for every DOM issued in the system and can impact console performance if allowed to grow too large. Verify that you are retaining only messages you need to retain. This can be controlled by the MPFLSTxx. Is suggested that you do not retain any "E" type (eventual) action messages.

- Review Ensure that the CONSOLE address space has the highest dispatching priority (FF), or, if in WLM goal mode, make sure that the CONSOLE address space is assigned to the default service class of SYSTEM.
- Verify automation procedures to determine if modifications are required. Automated message activities may need to be modified to determine the originating system that issued the message before taking action.
- Put automation in place and establish procedures to handle WTO/WTL buffer shortages.

Automate the following messages:

IEA405E WTO BUFFER SHORTAGE - 80% FULL

IEA404A SEVERE WTO BUFFER SHORTAGE - 100% FULL

IEA230E WTOR BUFFER SHORTAGE. 80% FULL

IEA231A SEVERE WTOR BUFFER SHORTAGE. 100% FULL

IEA652A WTO STORAGE EXHAUSTED - WTOs WILL BE DISCARDED

IEA653I JOBNAME= jobname ASID= asid HAS REACHED nn% OF THE WTO BUFFER LIMIT

IEA654A WTO BUFFER SHORTAGE ON SYSTEM sysname1 IS AFFECTING MESSAGES ON SYSTEM sysname2

IEA099A JOBNAME= jobname ASID= asid HAS REACHED THE WTO BUFFER LIMIT

- The following are possible causes of WTO buffer shortages and additional areas that should be considered for automation:

The buffer limit is too low to handle the message traffic.

A console might not be ready because:

- ✓ A multiple-line WTO message is hung.
- ✓ An intervention required condition exists.
- ✓ The console has been powered off.
- ✓ The path to the device is not fully functional; for example, an I/O interface is disabled

The console is not in roll mode.

The console has a slow roll time.

The console is in roll deletable (RD) mode, but the screen is filled with action messages.

- An application is in a loop issuing messages with or without a needed reply.
- There is a WTO/R buffer shortage on another system in the sysplex.
- JES2 might have a problem with continuous writing of the syslog, causing the hardcopy function to hang.
- Create documentation for operations:

Mapping of MCS, SMCS and system consoles.

Document console operations and recovery procedures, including:

- ✓ The use of command prefixes, route commands, etc.
- ✓ Recovering from console failures and performing console switches.
- ✓ Switching console modes and recovering from buffer shortages.

The use of the VARY CN(*), ACTIVATE command to enable the system console when needed for recovery.

Maximizing software availability

At this point, we have discussed the things you should do to ensure the computing environment is supportive of continuous processing, and we have discussed how to configure your hardware to remove as many single points of failure as possible. The next area to address is software. In this chapter, we start off by listing the considerations for providing maximum system availability. This is to ensure that each individual system has the highest possible availability, and also to ensure that there is nothing in the software configuration that could negatively impact the availability of the sysplex.

After that, we address the features of z/OS that provide support for workload balancing. Finally, we discuss individual products, such as IMS and Db2 and list the availability considerations for each of those products.

Configure software for high availability

One of the first things to consider is the placement of the critical sysplex data sets. All critical sysplex data sets should have an alternate data set that is used in case of a problem with the primary data set. Therefore, you have to make sure that there are no single points of failure that could affect both data sets. You also have to be careful to place the data sets so that they achieve the required performance.

Sysplex data sets

The sysplex data sets are classified as follows:

1. Sysplex couple data set - This is the data set that is used by XCF to manage the sysplex.

2. Functional couple data sets - These data sets are associated with a particular function such as Coupling Facility Resource Management (CFRM), Workload Manager (WLM), and so on.

The sysplex couple data set and the CFRM couple data set are *critical* to the sysplex. Any system that loses access to these data sets is placed into a non-restartable WAIT state. Therefore, the following is recommended for the sysplex and all functional couple data sets:

- Define primary, alternate and spare couple data sets with the same attributes.
- Allocate the primary, alternate and spare couple data sets on high availability DASD.
- The primary, alternate and spare couple data sets should be placed on different DASD subsystems:
- Establish redundant I/O paths through different ESCON® directors and storage directors, ensuring that there are no single points of failure in any of the primary, alternate or spare data set configurations.
- Allocate data sets as a single extent on a single volume.
- Do not over-specify parameters when formatting CDSes. If you specify values that are far larger than necessary, this can unnecessarily delay sysplex initiation.
- Allocate CFRM, SFM, ARM, WLM and LOGR in separate data sets.
- Allocate data sets on storage subsystems with caching and fast write enabled.
- Do NOT mirror the Couple Data Sets using Metro Mirror. Instead, have the alternate CDSs be at the remote site.

Also, consider implementing automation to periodically check to ensure that the cache remains enabled.

- Do not use DFSMSdss™ or other utilities that issue a reserve against volumes containing couple data sets.
- Avoid placing CDSes on volumes with high use data sets such as page or dump data sets. If possible, allocate CDSes on isolated volumes to avoid I/O delays and reserve activity.
- Do not allocate the primary sysplex couple data set on the same volume as either the primary CFRM or the primary LOGR couple data sets. The recommended mix is as follows:

VOL001	**VOL002**	**VOL003**
Sysplex – Pri	Sysplex – Alt	Sysplex - Spare
CFRM – Alt	CFRM – Pri	CFRM - Spare
SFM – Pri	SFM – Alt	SFM - Spare
ARM – Pri	ARM – Alt	ARM - Spare
WLM – Pri	WLM – Alt	WLM - Spare
LOGR – Alt	LOGR – Pri	LOGR - Spare
SHCDS – Pri	SHCDS – Alt	SHCDS – Spare

- Put automation in place to issue the SETXCF command to add the spare couple data set as the alternate CDS in the event that an alternate becomes unavailable (and notify the system programmer of the recovery event). After activating the spare data set, recover the failed CDS quickly so it can become a spare. Remember to update the COUPLExx member to reflect the new data set names for the primary and alternate couple data sets. If using GDPS, let GDPS handle this automation.
- Use the XISOLATE Tool to ensure data sets remain allocated for high availability available at ftp://ftp.software.**ibm.com**/s390/mvs/tools/

Other important data sets

There are also other critical data sets that are not unique to the sysplex environment, but are still critical to system, and possibly sysplex, availability. Like the sysplex data sets, these data sets should also be placed on high availability DASD. When a component supports multiple copies of a data set (like the JES2 checkpoint), it is recommended that a primary, secondary and spare data set be defined. Also ensure that:

- There are no single points of failure related to the primary/secondary/spare data sets and the paths that serve them.
- Automation be put in place to automatically:

Switch to the secondary data set when the primary fails.
Switch the spare data set into use when a primary or secondary data set fails.
Notify the system programmer when a recovery action is invoked.
Use XISOLATE to ensure data sets remain allocated for high availability

The following list is not exhaustive. You must evaluate all your data sets to determine which ones are critical to your business and ensure that these data sets are configured and managed for high availability:

- Master and user catalogs
- SYSRES
- Common and PLPA page data sets
- Local page data sets. Ensure all the page data sets are sized to handle the loss of a data set, unexpected levels of system activity, the data spaces used by the System Logger, and a number of system dumps. Also make sure there is adequate redundancy in the connection of these devices to the systems.
- CICS Journals, GCD, LCD, RSD
- Db2 logs, bootstrap, catalogs, directories
- RACF® database
- JES2 Checkpoint data sets
- JES2 Spool data sets
- If you are using VSAM/RLS, place the primary and alternate SHCDS control data sets on separate volumes.
- SMS CDSes (ACDS, SCDS, COMMDS)
- HSM CDSes (MCDS, BCDS, OCDS, and journal)
- IMS™ RECON, WADS, and OLDS data sets
- IMS and Db2 databases
- Application data
- Data sets containing vendor products and data.

Sysres and Master Catalog sharing

Evaluate the pros and cons of sharing the sysres data sets and master catalogs:

- Consider the use of multiple shared data sets to establish data set failure domains. For example, in an 8-system sysplex, if you share 2 sysres's (4 systems on each) and a sysres fails, it will impact 4 systems instead of 8. You need to balance the single point of failure risk against the risk introduced by having to synchronize many different versions of what is essentially the same data.

- To ease the burden of managing multiple parmlibs, consider sharing parmlibs among systems in the sysplex by defining common Parmlib members that utilize symbolic substitution for unique system parameters.
- The recent Redbook, *Parallel Sysplex - Managing Software for Availability*, SG24-5451, discusses the benefits and drawbacks of sharing these system files.

Consoles

It is vitally important that your consoles are configured for maximum availability. Recommendations about how to achieve this are provided in the “Consoles” section earlier. We add a pointer here because the configuration of consoles is a software issue as well as a hardware one.

Naming conventions

To facilitate the implementation of automation, and to reduce the incidence of human error, you should develop meaningful naming and numbering conventions for:

- Sysplexes
- Processors
- LPARs
- Systems
- Subsystems
- Catalogs
- Data sets
- Parmlib and proclib member suffixes
- Consoles
- CF structures
- SMP zones
- Job and proc names
- I/O paths and devices
- Define names such that technical staff and management can easily identify the sysplex, system, device, and so on.
- If possible, configure CHPIDs and device addresses on multiple systems using the same numbering scheme.
- When deriving naming conventions for these entities, bear in mind the use of system symbols. These can make the implementation of naming conventions easier, simplify system operations, and reduce the cost of managing a multisystem sysplex. This is discussed in more detail in the Redbook *Parallel Sysplex Managing Software for Availability*, SG24-5451.

Software symmetry

As explained earlier, it is important to maintain a symmetrical hardware configuration to enable applications to run anywhere in the sysplex. The symmetrical hardware configuration provides any-to-any pathways to hardware resources. Likewise, software must be configured symmetrically in order to exploit symmetrically configured hardware.

You should also configure subsystems symmetrically, even though many subsystems allow for asymmetric implementations. For example, JES2 allows multiple JES2 MASes (JESplexes) to be configured within a sysplex. It also supports a JESplex that contains systems both inside and outside the sysplex. To the extent possible, you should configure subsystem “plexes” symmetrically by matching the subsystem plex to the sysplex. Configuring subsystem plexes symmetrically will allow access to the same subsystem "logical image" on each system. This enables work units that use these subsystems to run anywhere in the sysplex, providing both improved availability and load balancing. Furthermore, the symmetrical software configuration will simplify operations, recovery and systems management. The major subsystems that should be configured symmetrically are as follows:

- GRSplex (must equal the Parallel Sysplex if GRS Star is used)
- JESplex
- RACFplex
- Batchplex
- SMSplex
- HSMplex
- RMFplex
- IBM Workload Scheduler (Batch)
- WLMplex
- TCPplex
- VTAMplex

Database subsystems, transaction managers, and applications must also be configured symmetrically to enable datasharing. They are covered in later sections.

Sysplex Policies

The sysplex policies are central to the way your sysplex performs, and to how it reacts to error situations. To ensure the optimum performance and availability within the sysplex, it is important that the policies are carefully planned and managed.

Two new web-based tools are available to help you define your CF structures, and to verify the size of the structures. The CFSizer can be used to quickly check the recommended size for a structure, given a minimum amount of sizing information. And the Parallel Sysplex Configuration Assistant provides an easy to use way of creating Parmlib members, CDS allocation JCL, and policy statements for the non-data sharing CF structures. Both tools are accessible via the Parallel Sysplex home page at: https://www.ibm.com/it-infrastructure/z/technologies/parallel-sysplex . The CFRM policy governs Coupling Facility usage. Ensure that the policy is properly defined:

- Ensure that structures are properly sized. Use the INITSIZE and SIZE parameters in the CFRM policy to control structure sizes. Monitor structure allocations and adjust storage allocation if necessary.
- Whenever a structure is rebuilt, the new structure will be allocated using the INITSIZE value from the CFRM policy, so remember to always update the policy any time you increase a structure size dynamically with the SETXCF START, ALTER command.
- Specify CF structure placement preference by using the PREFLIST and EXCLLIST parameters of CFRM. Use the PREFLIST to isolate related structures from each other.
- Understand the structure "allocation rules" and what it means to "transition" to a new CFRM policy.
- Understand the characteristics of the structures that are placed in CFs and ensure that:

Automatic recovery mechanisms are put in place to recover the structure in the event of a CF failure. For example, ensure that the PREFLIST contains multiple CFs to enable automatic structure rebuild in an alternate CF. If necessary, specify REBUILDPERCENT to enable the initiation of the structure rebuild

process. Specify REBUILDPERCENT(1) to ensure that rebuild is initiated in event of *any* loss of connectivity between a CPC and CF.
High activity structures are placed in different CFs, thereby balancing CF workload activity.
Evaluate use of FULLTHRESHOLD structure monitoring.
Use AutoAlter to automatically tune structure size. It is recommended for the following structures only: XCF signaling, IRLM Lock, Db2 SCA, Db2 GBP, DFSMS ECS, DFSMS™ RLS, WLM Multi-sys Enclave

Ensure that LOGR, WLM, SFM, and ARM policies are properly defined to implement the priorities and actions that have been agreed upon for the sysplex.

Maintain multiple policy definitions for planned and unplanned events. For example:

- When a system is removed from the sysplex, the production workloads presumably shift to other systems in the sysplex. When this occurs, you may want to change the system weights that are defined in the SFM policy. Predefined SFM policies can be switched automatically using automation.
- Predefine a set of CFRM policies that can be used when shutting down CFs for maintenance.

Sysplex policies are defined to govern important functions within the sysplex. The policies and the functions they govern must be proactively managed to ensure that they reflect changes in your systems.

Health Checker for z/OS and Sysplex

Download the IBM Health Checker for z/OS and Sysplex available at:
.software.**ibm.com**/webapp/download/search.jsp?go=y&rs=hchk
The IBM Health Checker for z/OS and Sysplex is a tool that checks the current, active z/OS and sysplex settings and definitions for an image and compares their values to those either suggested by IBM or defined by you, as your criteria. The objective of the IBM Health Checker for z/OS and Sysplex is to identify potential problems before they impact your availability, or in worst cases, cause outages.

This tool is updated periodically with additional function and runs on all z/OS releases.

Sysplex Failure Management (SFM)

- Specify CONFAIL(YES) to automate recovery from system-to-system connectivity failures.

When using GDPS, specify CONFAIL(NO)

- Assign an appropriate system weight value to prioritize recovery actions.
- Specify REBUILDPERCENT(1) for structures in your CFRM policy, to automatically initiate rebuild in case of any loss of connectivity to the structure. Use caution, as improper REBUILDPERCENT and system weight values may cause structure rebuild to be suppressed. Refer to z/*OS MVS*™ *Setting Up a Sysplex*, GC28-1779, for more information on SFM. There is also a white paper about SFM, entitled *Improve Your Availability with Sysplex Failure Management* in the availability section of the Parallel Sysplex home page, and a WSC Flash, W98025B, that discusses the use of SFM.
- Use the ISOLATETIME parameter as is default.
- Enable “system status detection”, XCF’s ability to use BCPii to determine whether a system is up or down.
- For systems running discretionary workloads, set up SFM to automatically partition the system out of the sysplex when:

You get a status update missing condition. (Isolation)
You get a connectivity failure. (Connfail)

Automatic Restart Manager (ARM)

- Implement ARM on all images where ARM-managed subsystems will run.
- Ensure that an active ARM Policy exists for the sysplex.
- Verify that selected elements are capable of exploiting ARM, and products are at the correct levels.
- For cross-system automatic restarts, ensure that those systems have access to the required databases, files, logs, program libraries, and so on that are required to execute the restarted workload.
- Define restart processes for all restartable elements and enable cross-system restarts in the sysplex.

System Logger (LOGR)

The z/OS System Logger is used by numerous z/OS features (APPC/MVS, OPERLOG, LOGREC, RRS) and also by other products such as CICS and IMS. The performance and availability of the System Logger (LOGR) therefore have a critical impact on the availability of applications using these features or products. The following recommendations should help optimize the availability and performance of LOGR:

- In GDPS environment, use LOGRY and LOGRZ on the K-Systems.
- If possible, isolate the Primary LOGR Couple Data Set from *all* other CDSes. If this is not possible, then at least isolate it from the Primary Sysplex CDS. This is done because of the high I/O rate to this data set when many logstreams are in use.
- Don't oversize the LOGR structures. The SMF Type 88 records provide information to help evaluate the utilization of the LOGR structures.
- Place the offload data sets on the fastest available DASD.
- The offload and staging data sets must be allocated with SHAREOPTIONS(3 3).
- Ensure the LOGR couple data set is formatted at the current level.
- Ensure the PTF for APAR OW29849 is applied.
- Try to have at least two active logstreams per CF structure, connected to more than one system, to allow peer recovery in case of failure.
- When another logstream is connected to a LOGR structure, the existing logstreams will be resized. This could potentially cause short-term logstream-full conditions and impact exploiters. Therefore, try to start all connectors to a given LOGR structure at the same time.
- If using CF structures, try to avoid use of staging data sets. As the number of IXGWRITEs increases, staging data sets will become a bottleneck long before the CF is maxed out.

Consider System Managed CF Structure Duplexing as a better performing alternative to Staging data sets.

Avoid volatile CFs and ensure that CFs and the z/OSs they are connected to are *not* in the same failure domain. If the CF is volatile, or is in the same failure domain as a connected z/OS, LOGR will automatically use staging data sets to ensure the integrity of the log data. If you are using ICFs, try to place the structure in a CF that is in a different CPC to the z/OS images that are using that structure.

Define ACS routines and SMS classes to handle staging data sets, even if you don't intend to use them, to cater for loss of failure independence in Logger structures.

Staging data sets should be large enough to hold the maximum amount of data created by the busiest connector.

Use two different data classes for the staging and offload data sets. The staging data sets should have a CI size of 4 KB, and the offload data sets should have a CI size of 24 KB (assuming APAR OW29849 is installed).
Make sure you have sufficient auxiliary storage to back the Logger data spaces.

Configuring individual components

More information on each of these can be found in Redbook: Getting Started with IBM Z Resiliency
http://www.redbooks.ibm.com/abstracts/sg248446.html?Open

XCF Signaling

Every system in the sysplex must be able to communicate with every other system in the sysplex. A loss of intersystem signaling connectivity may have an impact on the entire sysplex.
Ensure that there is redundancy built into the XCF signaling configuration:

- Configure at least two hardware paths between systems, using:

Signaling structures.
FICON CTCs.
Or a mix of structures and CTCs.

- Ensure that there are no single points of failure related to the hardware paths:

If you are using exclusively signaling structures for cross system communication, define the signaling structures in separate CFs, using the exclude list to enforce this. XCF will still be able to place all the signaling structures in the same CF if that is the only CF available, so this does not cause any exposure.
Configure ESCON CTCs channel paths through different ESCON directors and on different channel cards in the processors.

- Define PATHIN and PATHOUT statements such that signaling structures and CTCs are configured bi-directionally:

Establish a sound naming convention to easily identify links.
Initially assign all PATHOUT signaling paths to the DEFAULT transport class. Review the RMF “XCF Activity” report to determine if tuning is required.

  - ✓ If PATHOUT buffer tuning is required, define additional transport classes and segregate messages based on message size using GROUP(UNDESIG).
  - ✓ Monitor inbound and outbound buffer availability. If buffer unavailable conditions occur, MAXMSG may need to be adjusted. If you specify an overly large MAXMSG value XCF will only use what it actually requires, but if you do have delays, really large MAXMSG values can cause large queue effects and lots of fixed real below the bar in use.
  - ✓ Monitor XCF Path statistics and ensure that signaling paths are performing efficiently.
  - ✓ Use the SETXCF MODIFY command to implement any changes you wish to make, making sure to remember to reflect the changes back in the COUPLExx member.

Maintain a map of intersystem signaling connectivity for Operations.

- CTCs can be configured either as unidirectional (one PATHOUT or one PATHIN per physical CTC), or bi-directional (one or more PATHOUTs and PATHINS on a physical CTC).
- For the majority of customers bi-directional CTCs will perform sufficiently. For those customers with very high XCF activity (requiring 4 or more XCF paths per CPC connection), considering using unidirectional CTCs.

Refer to WSC Flash 10011: Parallel Sysplex Performance: XCF Performance Considerations
www.**ibm.com**/support/techdocs/atsmastr.nsf/PubAllNum/Flash10011

XCF Controls

- Ensure the proper setup of COUPLExx parameters:
  Decrease CLEANUP to 15 seconds or less.
  Use the default values for OPNOTIFY and INTERVAL. INTERVAL default is based on the SPINTIME parameter specified in EXSPATxx.
  Ensure that COUPLExx is updated whenever dynamic changes are made via the SETXCF command.
- Ensure EXSPATxx member is coded properly:
  Unless there is a valid reason to change, leave SPINTIME values default. If values are coded, ensure proper adjustments are made to corresponding COUPLExx (OPNOTIFY & INTERVAL) parameters.
  Specify **SPINRCVY TERM,ACR**

GRS

- Implement a GRS Star configuration.
- Ensure proper placement and sizing of the GRS lock structure. Under-sizing the lock structure can negatively impact performance.
- Ensure that RNLs are properly defined.
- Use the GRS Monitor to track ENQ activity to aid in defining RNLs.
- Convert as many RESERVEs as possible to global enqueues.
- If there is no resource sharing outside the sysplex, convert ALL RESERVEs.

JES2

- Run with two checkpoint data sets and use either the DUAL or DUPLEX feature of JES2 to maintain current checkpoint data across the data sets.
- Define backup checkpoint data sets and use automatic checkpoint reconfiguration (OPVERIFY=NO) to forward the checkpoint data set in the event of a checkpoint failure.
- System Managed Rebuild supports the JES2 checkpoint structure for planned configuration changes. No JES2 definition or parameter changes are required to enable this support. For unplanned outages, the checkpoint reconfiguration dialog must still be used.
- Use the JES2 checkpoint reconfiguration dialog to move a checkpoint data set. Do not use DFSMSdss™ to move checkpoints. Similarly, do not use DFSMSdss to move a spool volume.
- Define a sufficient number of spool data sets and specify FENCE=YES to limit the impact of a spool data set failure.

The FENCE parameter allows the customer to specify the number of volumes a job will be fenced to. Specifying multiple volumes should mitigate any performance issues associated with fencing.
Specify **FENCE=(ACTIVE=YES,VOLUMES=n)** where n is a number greater than 1

- Spool and checkpoint data sets should be placed on high availability devices and should not be SMS-managed. Avoid placing other data sets on volumes containing checkpoint and spool data sets.

- When placing checkpoint data in a Coupling Facility, place CKPT1 in the CF and place CKPT2 on DASD.
- Define the JESplex to be the same as the sysplex.
- Spread JES2-managed devices across multiple systems in the MAS to limit the impact of a JES2 or system failure. Define all devices on all members to allow them to be switched in the case of an extended system loss.
- Specify MASDEF(RESTART=YES) or install fix for OW32320 to automatically free the JES2 checkpoint lock in the event a JES member fails, thereby holding the lock.
- Enable automatic requeuing of jobs in the JES2 MAS via the MASDEF(RESTART,AUTOEMEM) parameter.
- Know your JES2 exits and what they do. Ensure that the exits are compatible with new releases of JES2 before migrating to the new release.
- Code $ESTAE in JES2 user exit routines.
- Specify REGION=0 in the JES2 proc and monitor JES2 storage usage.
- Implement automation that initializes the JES2 environment during system startup.
- Implement automation that initializes the JES2 environment (start of NJE links, and so on) after a JES2 hot start. It should not be necessary to IPL to restart JES2.
- JES2 resources should be monitored (HASP050) and automation/procedures put in place to handle shortages. Do not set WARN=00 or WARN greater than 90 for any resources.
- Define more resources than you think you will need and monitor usage periodically.
- Monitor JES2 devices to ensure that they remain ready (especially NJE, RJE and unattended devices).
- Define a backup JES2 PROC and JES2 initialization deck to be used in case the production versions have a problem. Start the backup PROC by specifying S backup,JOBNAME=JES2,....
- Test the JES2 PROC after all updates and after any changes to PROCLIB data sets. This can be done by simply issuing a S JES2 (even if JES2 is up). This second JES2 will not complete initialization, but it will verify that the PROC is error-free.
- Define an alternate IEFSSNxx Parmlib member or COMMNDxx member as needed to allow MVS to start without automatically starting JES2.
- Do not place a $S command in the JES2 initialization deck.
- Ensure that automated replies to JES2 message can be turned off for an IPL.
- Avoid initialization statements that use * as part of the range.
- Define maximum PURGE, SPIN, and CNVT PCEs.
- Consider using products such as IBM Workload Scheduler to improve batch handling and improve turnaround.
- Use WLM Scheduling Environments to ensure that jobs will run on whichever system a particular required resource is available on.
- Use the z/OS JES2 Health Monitor to help to quickly identify and issue alerts for problem areas

z/OS waits
Local lock waits
Long hold of JES2 checkpoint
Non-dispatchable
Busy - No z/OS waits
Not running - Not waking up from main WAIT
Paging waits
Loops
Long PCE dispatches

Displays percentage of total JES2 CPU used by each specific PCE type
Monitors JOB and BERT lock PCE waits
Displays what resources PCEs are WAITing for, and for how long

WLM

- Intelligent Resource Director (IRD)
- Define a WLM service definition and workload goals to meet your business objectives.

Identify business/application requirements and establish service level objectives.
Make your “loved ones” the most important work in the system (importance 1 or 2).
Establish Velocity and Response Time Goals that reflect your true requirements.

- Establish meaningful naming conventions for the different constructs used in the WLM policy. This will allow for easier identification of elements such as Service Classes, Report Classes, Classification Groups, or workloads. An example naming convention might be:

Type	Example	Description
Workload	BAT_WKL	Workloads have a suffix of ‘_WKL’. Workload names are not addresses by operator commands and so the use of special characters like the ‘_’ is not disruptive. By using a suffix of ‘_WKL’ we can quickly differentiate workloads from service classes.
Service Class	BATREG	Try and use as descriptive a name as possible. Watch sort orders
Resource Groups	LIMIT_RG	We use a suffix of ‘_RG’ for resource groups. LIMIT_RG Resource Groups
Report Classes	RJES2 RVTAM RTCPIP RICSP	We prefix all report classes with an ‘R’ to indicate report class. This would require no service class to have the need to start with an R. A suffix of ‘_R’ would also work, but report classes may need to be specified in RMF control cards.
Transaction Name Group	STC_TNG	Again we use a suffix to indicate the construct is a named group
Transaction Class Group	JES_TCG	Another suffix and we try and make the prefix meaningful, JES or STC, or IWEB groups

- Establish a WLM policy management process for backing up, modifying, activating, and restoring your WLM service definition.

Before modifying your WLM service definition, always save a copy of the current version as backup.

- For the first time activation of WLM:

Use the IBM Migration Aid Tool to create your initial policy. Cheryl Watson’s “Quick Start” Policy is also a good starting point for a initial policy. Consider blending output of Migration Aid Tool with “Quick Start Policy” recommendations to create your first policy.

- Set up report groups using the SRVCLASS parameter and activate your policy in compatibility mode for several days.
- Use the RMF Post Processor reports to review the actual response times or velocities for the service classes defined. If needed, the goals defined in the service policy should be adjusted before switching to goal mode.
- Pick a period of time during off-peak hours, and switch to goal mode. (You can dynamically switch between goal mode and compatibility modes using the Modify WLM command).

- Continue to monitor performance. When you're satisfied that service objectives are being met, switch another system to goal mode and monitor performance. Continue this process until all systems in the sysplex are converted to goal mode.

Service Classes

- Keep the number of service classes to a minimum (25 - 30), especially high importance periods. This helps ensure WLM responsiveness in managing all service class periods to their goals during periods of high CPU contention. There are a couple reasons why it helps to limit the number of service classes:

WLM Responsiveness - at each 10 second policy interval, WLM selects the service class in the most need of help and takes one policy action on it's behalf. The more service classes that need assistance, the more 10 seconds cycles needed for WLM to take the needed policy actions on behalf of these service classes.

Better WLM Decisions - as part of its' decision-making, WLM makes projections for how work will behave when resource changes are made. The accuracy of these projections depends on the quality of history data gathered for a service class. The more similar work that is grouped together in a single class, the more statistically valid the history data, and the better WLM decisions will be.

- Ensure that WLM has good information available for decision making. For Response Time based service classes ensure there are enough completion's, and for Velocity Goals ensure there is enough ready work, to provide sufficient information for WLM.
- Limit the number of started task service classes to 2-3 service classes, plus SYSSTC and SYSTEM.

Ensure that "system" work: such as VTAM, JES, IRLM, RACF, TSS, etc.; are assigned to the SYSSTC service class. If classified otherwise SRM will try to manage these workloads, delaying resource adjustments to other service classes

Identify other critical system started tasks, and if CPU requirements are reasonable and there is sufficient storage, assign them to SYSSTC.

Monitor the SYSSTC service class's CPU usage, if high and/or online systems are experiencing delay, adjust SYSSTC by moving some of the heavier CPU users to a different service class.

Let all z/OS system tasks that default to SYSTEM go to System. ***Do not second guess***. z/OS-provided address spaces, such as CONSOLE, GRS, DUMPSRV, and so on, should be allowed to default to the SYSTEM service class.

For the remainder of the started tasks, create as few service classes as possible, and for Long running STCs, use Velocity Goals. For Example:

1. CPU intensive and important (High importance, High velocity goal)
2. All other started tasks (High importance, Low velocity goal)

- Consider establishing "special situation" service classes to avoid having to modify, install, and activate a new policy to handle special circumstances. For instance:

When implementing Goal Mode for the first time, consider creating several "migration" service classes (e.g. SRVL10L for velocity 10 / importance 2, SRVL10H for velocity 10 / importance 1, SRVL20L... and so forth). These service classes can be very helpful in moving work around on the fly while "tuning" your policy, and will not be as intrusive as activating a new policy. These service classes should be removed at a later date.

Evaluate the need for a service class, with a sufficiently high velocity, which can be used to push "Emergency" batch jobs through the system during periods of heavy CPU utilization.

Create a special, high velocity, TSO service class for emergencies.

To handle runaway jobs and/or transactions, consider creating a "sleeper" service class. Associate with a resource group and specify a maximum service unit capacity of 1.

Modify any operator documentation and/or procedures with the processes used by your installation for runaway tasks.

- Ensure that you have classification rules for all subsystem types.

If there are no classification rules for a subsystem type, the default is SYSOTHER with a discretionary goal. Aim to never have work run in SYSOTHER
Ensure enclave transactions are properly classified in the WLM service definition to prevent them from being managed as discretionary work.
The most common error by far is not classifying Distributed Db2 Transactions (running under the DDF subsystem type) and having them end up managed as discretionary work.

Response Time / Velocity Goals.

- Use Response Time Goals rather than Velocity Goals wherever possible. Response Time Goals have several advantages over Velocity Goals, such as:

Velocity Goals must be reevaluated whenever hardware is upgraded to reflect the difference in velocities due to differences in processor speed or number of CPs. Whereas Response Time Goals, based on End-User requirements, would not necessarily change across an upgrade.
CICS workloads, using Velocity Goals, will get storage protection only after a page delay problem is detected by WLM. CICS workloads, defined with Response Time Goals, is assumed to be interactive and WLM proactively protects its' working set, even before page delay issues are detected.
For storage constrained systems, use of Response Time goals could provide an important advantage for interactive workloads.

- Define a high velocity goal for CICS, Db2, and IMS regions to ensure fast initialization, with the transactions defined with Response Time Goals.
- For a sysplex with mixed machine types, Velocity Goals for the same work can achieve different velocities on processors of different speeds and/or different numbers of CPs. some helpful hints for selecting Velocity Goals in this environment are:

Higher Velocity Goals are more sensitive to processor differences, so select a goal that is attainable on the smaller processor(s).
Do not try to be too precise in selecting a number, small differences in velocity goals have little noticeable effect. For example, velocities that represent "slow", "medium", and "Fast", might be 10%, 40%, and 70% respectively.

- Always review your Velocity Goals after a processor change.
- Consider using low importance levels and/or resource groups for work with the potential to dominate system resources
- Consider having classification by system name for Velocity Goals.
- Move "well-behaved" production applications which are consistently small consumers of CPU cycles into SYSSTC. Consider the impact to your environment.

Other Considerations

- Try to avoid using Resource Groups.
- If resource groups will be used, it may be advantageous to change the CPU and SRB user coefficients to 1. If they will not be used, you should probably continue using the current user coefficients.
- Do not mix diverse work types in the same service class with CICS or IMS transactions
- Avoid setting to many aggressive goals that can not be met, otherwise WLM will spend a lot of time managing and not getting other work done (e.g. High Velocities - over 80)

- Use batch initiator management to spread batch work across the sysplex.
- Reevaluate velocity goals when you turn on I/O priority queuing or batch initiator management. These two functions change the velocity calculation. The projected velocity values (“migration velocities”) are reported in RMF to help you decide on new goals before you migrate to these functions.
- Ensure that SMF recording for type 99 records is turned off. These records are produced in large volumes and are only required for WLM problem determination. Also, WLM buffers the last 15 minutes of SMF type 99 records, and these will be contained in a dump if required for problem determination. If you wish to process summary information, this is kept in the SMF type 99, subtype 6 records which are produced in much smaller quantities.
- Use the VTAM Generic Resource function with CICS, IMS, TSO, and APPC to distribute sessions across a sysplex. However, if you have a very large number of logons in a very short amount of time (1 to 5 minutes), you may find that the use of VTAM GR with WLM produces an imbalance of session allocations across systems in the sysplex. If this applies to your installation, consider using the ISTEXCGR exit to override the session allocation criteria.
- Use the WLM/DNS support to spread TCP/IP work across the sysplex.
- CPU Critical and Storage Critical support is available for systems that are CPU and/or Storage constrained to guarantee that critical work does not become a resource donor to lower importance work.
- Take into consideration your installation testing and change control processes
- Ensure that operators are properly trained on the use of WLM and that all revenant operator documentation and/or procedures are updated

RACF

- Share the same RACF database among all systems in the sysplex**:**
- Turn on the “sysplex communication” bit in the DSNT (ICHRDSNT) and IPL all systems (one at a time) to enable RACF sysplex communication *before* you enable RACF sysplex data sharing. When you are ready to do RACF sysplex data sharing, turn on the “datasharing” bits in the DSNT.
- Ensure that the DSNT is compatible on all systems in the sysplex.
- Ensure that the RACF database is not shared with systems outside the sysplex.
- Ensure that the database range table is identical on all systems.
- Ensure that the class descriptor table is compatible on all sharing systems.

Db2

There are specific parameters which the user specifies during installation/migration on the Db2 ISPF panels. Those values are converted to keyword values contained in your customized SDSNSAMP(DSNTIJUZ). When you submit the DSNTIJUZ job, it assembles the parameters into the DSNZPARM control block. This section identifies the values by their DSNZPARM keywords, with most Db2 systems staff are familiar, rather than by their installation panel name and field names. Please refer to the Installation Guide for your version of Db2 and to the appendix "Directory of subsystem Parameters" for a cross referenced list of the parameter, macro, panel id, page reference, and for Db2 for z/OS, whether the parameter can be updated "online".

- Implement Db2 datasharing**:**

Start by reviewing the Db2 Data Sharing Planning and Administration Guide for your version. This is an excellent reference and “how to” source for planning your data sharing project.

Establish and document naming conventions early on to avoid confusion, eliminate operator error, simplify problem determination, and facilitate changes to the data sharing group.
Review CPU capacity requirements to ensure there is sufficient CPU capacity available to handle the additional resource overhead associated with Data Sharing:
Data Sharing overhead can vary greatly from one system to another, depending on such factors as the degree of inter-Db2 read/write interest, configuration of coupling facilities, CF links, structure sizes, etc..
Review all ISV software with vendors and verify that products are at the proper release and maintenance levels to support Data Sharing and any new Db2 features that you plan to exploit.
Ensure your monitor supports a single view or activity for all members of group. Db2 Performance Expert is one product that can view either single members or all members from workstation.
Use the tool, **Health Monitor** and **Health Center**, that alert you to potential system-health issues before they become real problems that affect performance and availability.

- Coupling Facility considerations:

Review Coupling Facility Guidelines as described in the beginning of this document.
Ensure that Coupling Facilities have sufficient resources (CPU cycles, storage, and link capacity) and that CF Structures are properly sized:

> Use the Coupling Facility Structure Sizer Tool to assist you with sizing issues. The tool is available on the Parallel Sysplex web site at: https://www.ibm.com/support/pages/cfsizer

For high availability, it is recommended that at least one (1) standalone coupling facility be present in the configuration
Ensure that AUTOREC(YES), the default, is specified for each GBP. This enables automatic GBP recovery in case of a CF failure.
If you use an ICF for Db2 structures:
Implement GBP duplexing. The use of GBP duplexing virtually eliminates the GBPs as a single point of failure

- ✓ If a CPC containing both an ICF and a z/OS image from the same Parallel Sysplex fails, there is a double failure. If the lock or SCA structures are in the ICF, all members come down and group restart is necessary. If there are simplexed GBPs in the failed ICF, then recovery cannot proceed until the failed member is restarted. For duplexed GBPs, recovery is not needed since structures are designed to automatically use the secondary copy within a few moments after the Db2 members on that mare are marked FAILED.
- ✓ Make sure the Db2 SCA and IRLM lock structures are allocated in a failure-isolated CF or (System Managed) Duplexed. When the Lock structure is duplexed, each lock requested has a CPU cost of 3x-4x that of a simplexed structure.

- For initial startup, over allocate structures and monitor for a period of time, then adjust sizes downward or upward as necessary.

AutoAlter may be specified on the Structure statement of the CFRM policy to assist you to tune your GBPs. This makes the GBPs as self tuning as possible. XES can increase or decrease the size (gradually). It can increase the directory to data ratio (but never decrease). Auto Alter is designed to handle gradual changes in activity, not sudden fluctuations.
For Group Buffer Pools (GBP), specify the INITSIZE and SIZE parameters, allowing some room for growth, to avoid the disruption of having to modify, install, and activate a new CFRM policy and issue a rebuild to increase the size. From the CF Activity Report Summary, investigate any directory reclaims and increase the number of directory entries when there are "Dir Rec XI".

- ✓ An excellent Db2 command to use is -DIS GBPOOL (*) TYPE(GCONN) GDETAIL. Issue it from any member connected to all GBPs and it will show only those GBPs that are connected (thus reducing output greatly). GDETAIL shows statistics issued since the last time the command was issued from the same member. The first time you issue it you receive statistics since the allocation of the structures, a good way to identify if you have any of the "bad" non-zero values for:
- ✓ Cross Invalidations due to directory reclaims or Writes Failed due to Lack of Storage

The former affects performance, while the latter affects both performance and availability. It is possible for pages to be placed on LPL if the storage is severe.

- The SCA structure can impact availability if under allocated. Allocate at least 32MB to the SCA initially. It contains exception states for objects such as those that are in a Read-Only status, as well as for Copy / Recovery / Reorg / Check Pending.
- Start with a 32MB Lock Structure, monitor false contention, if high, increase size to the next power of 2.

High is > 2% contention and 1% false contention as indicated in the RMF Coupling Facility Activity Report for the Lock Structure (Req. Total, Req. Deferred, False Cont) or the Db2PM Statistics report

- Set REBUILDPERCENT to 1% for all structures for which rebuild is desired when there is a loss of CF connectivity.
- Spread structures across more than one coupling facility in a manner that balances CF CPU and Link utilization as evenly as possible.
- Db2 supports dual logging. This is similar to JES2's use of dual checkpoints, where one backs up the other:

Implement Db2 Dual logging support .

Each set of logs (active and archive) should reside on different volumes/control units/paths. Copy1 should be on a separate “box” from Copy2.

Exclude the Db2 Active Logs and BSDS’s from HSM migration activities .

Define a sufficient number of Db2 active logs. Optimize the following:

- ✓ Input and output log buffer sizes.
- ✓ Write thresholds.
- ✓ Archive log frequency.
- ✓ Log update rate.

Consider using the Archive log option, TSTAMP=YES for ease in identifying archive logs if a problem exists.

- Optimize Db2 Recovery and Restart times: Evaluate media used for Db2 archive logs and ensure that log access time is optimized. Consider archiving the primary Db2 log files to DASD to facilitate a quick recovery. The second archive copy can be written to tape and used for disaster recovery.

Establish a SMS pool with enough primary space to maintain a sufficient number of logs on disk to support a recovery up to 24 hours back.

Establish automated procedures (use HSM if available) to monitor space within the pool to ensure that sufficient space exists at all times to create new archive logs. In the event that the pool becomes full and Db2 is unable to dump the active logs, Db2 will stop until the situation is corrected and an active log is available again to write to!

Automate and monitor the following messages:

DSNJ110E - LAST COPYn ACTIVE LOG DATA SET IS nnn PERCENT FULL

DSNJ111E - OUT OF SPACE IN ACTIVE LOG DATA SETS

DSNJ10%I - csect name LOG WRITE ERROR DSNAME=..., LOGRBA=..., ERROR STATUS=cccc

- There should be enough tape drives to be able to support each Db2 in the data sharing group if they each have to mount tapes to go back to Archive logs or to HRECALL migrated DASD archives. This value is specified as DSNZPARM MAXRTU, though it can be changed by an operator -SET LOG command. A related value is specified as DSNZPARM DEALLCT with zero, so that a tape drive will be released as soon as it is not used. and its data set becomes available to other Db2 members that may request it.
- In the event of a Db2 or System failure, restart any failed member (non-quiesced members) to minimize restart time and free retained locks so that its data can be accesses by other Db2 members of the group.

Consider using the Automatic Restart Manager (ARM) to restart the Db2 subsystem in place when a Db2 member failures, or on another z/OS image in the event of a z/OS and/or system failure.
Consider using "Restart Light" to quickly resolve retained locks with minimal disruption to other systems. Restart Light applies only for z/OS image failures (for a Db2 failure, the member restarts in place on the same system, so there is no need to use Restart Light)

- Ensure that the DSNZPARM value for the RETLWAIT parameter is set correctly for your installation. It is expressed as a multiplier of the resource timeout value (IRLMRWT) that Db2 will wait before returning a "Resource Not Available" condition to the application. Good values are 1 or 2 for use when you have automation perform restarts in place of Db2. Otherwise, specify 0.
- For heavy data sharing with large GBPs, try to dribble castout write activity between GBP checkpoints. CLASST (1-5%) and GBPOOLT (10-25%) are quite common for large GBPs. Monitor aggressively monitor for GBP write failures (0) through the -DIS GBPOOL command.
- A long-running unit of work may elongate Db2 recovery and restart time. Therefore, incidents of "long-running units of work" should be minimized.
- The warning message (DSNR035I) for long running unit of recovery (UR) was based on the number of checkpoint cycles to complete before Db2 issues a warning message for an uncommitted unit of recovery (DSNZPARM URCHKTH). But the number of checkpoints depends on several factors which may not include the long running job. The warning mechanism is additionally based on the number of log records written by an uncommitted unit of recovery, specified by DSNZPARM URLGWTH. Message DSNJ031I is issued when the threshold is reached. The number of updates is cumulative for a UR and the message is repeated every time the threshold is reached.

Ensure that policies for application checkpoint, commit frequencies and restart guidelines are established and being followed.
Even read only transactions must commit, or they can cause utilities such as an online reorg to fail. Reorg must drain even readers during the switch phase, which is extremely short but requires exclusive access. Commits reduces the total number of locks held by the unit of work, reduce "false contention", and avoid lock escallation.

- Consider putting automation and operational procedures in place to identify and cancel long-running units of work.
- Evaluate system checkpoint frequency for impact on Db2 restart times.
- Create partitioned table spaces to ensure granular utility domains. A restrictive state on one partition is unlikely to affect the availability of the others (limit is up to 4096)

Backup and Reorg Processes

- Minimize the disruption of ongoing work by utilizing the ESS FlashCopy features to obtain point-in-time copies of data for use in Disaster Recovery.

- Optimize the use of Db2 COPYs and REORG to reduce planned and unplanned outage time. Consider SHRLEVEL CHANGE options for both. Updaters are allowed while COPY executes, while they are allowed almost all the time during online Reorg.
- Utilize Db2's Online Reorg capability. Use the REORG parameters to drastically improve reorg times: SORTDATA and SORTKEYS. For Db2v7: DRAIN_WAIT, RETRY, RETRY_DELAY, and Fastswitch.

Optimize Db2 shutdowns:

- At Db2 Shutdown, z/OS performs processing for all Db2 data sets opened since Db2 startup. You can reduce shutdown time by setting the z/OS parameter DDCONS(NO). It can take many minutes to do SMF processing. The last reference data for the VSAM files is not updated if you choose this option.
- Stopping DDF earlier in the process can help purge distributed threads, particularly inactive ones.
- You can use -STOP Db2 CASTOUT(NO) to cause Db2 to shutdown quickly. It causes the GBPs to remain allocated with changed pages not yet written out to DASD. It is intended to be used in situations where you plan to immediately restart Db2; for example, if you are applying service.

Minimize Db2 Lock Contention

- Tune to keep Db2 Lock Contention to less than 2%, and keep False Contention to less than half the total of the total data sharing lock contention (1%):

Leave the IRLMPROC parameter MAXUSRS at the default 7, unless you will have more than 7 members in the data sharing group

If false contention is high, increase size of the lock structure to the next power of 2.

- Ensure that IRLM is running at a higher priority than the IMS, CICS, and Db2 address spaces, but not higher than the XCFAS address space. IRLM SRB time should always be less than Db2 SRB time.
- Reevaluate Lock Escalation parameters. Lock escalation occurs on a table basis after a certain number of lower level locks (page or row) have been granted and the tablespace itself becomes locked. This option (LOCKSIZE=ANY) is not recommended as it can greatly affect concurrency, even while it minimizes locking.
- Carefully select the IRLM startup parameters (PC, DEADLOK, and MAXCSA).

PC=YES is the recommended option for all data sharing customers, due to the frequency of lock requests to XES that must occur in PC mode.

MAXCSA is not applicable when PC=YES

DEADLOK - with faster z/Series processors the value should be in the range of less than 1 to 1 (the default is 5,1 that indicates deadlock interval checking will occur ever 5 seconds). A value of (.5,1) would be a better value than (5,1). The second parameter refers to the ratio of global detection to local detection and is always 1.

- Review DSNZPARM LOGAPSTG (Fast Log Apply) and make sure it is not zero (which disables it). FLA improves drastically (orders of magnitude) LPL/GRECP recovery time, and Recover utility time. Values can be 10-100 MB and reserves storage in the DBM1 address space. Please read the Db2 V6 Performance Topics Red Book, SG24-5351, 5.1 and 5.2 for detailed information. FLA is used during restart no matter what you choose.
- Review DSNZPARM CHKFREQ, PCLOSEN, PCLOSET and CLOSE=YES parameters

CHKFREQ is used to specify checkpoints in terms either of time or number of updates. If CHKFREQ is <=60, it denotes a steady number of minutes between checkpoints. If it is >60, it refers to number of

updates. For a data sharing environment where a member is infrequently updated, the strong recommendation is for time-based checkpoint. This is necessary in order for the GBP Checkpoint LRSN to move forward.
PCLOSEN is frequently disabled by specifying a high value such as 32000. That leaves PCLOSET as the number of minutes following an the last update before a pageset is converted to read only via pseudo close logic.
CLOSE=YES "Respect Close NO attribute during Pseudo Close." Behavior indicates that after a PCLOSET interval a pseudo-closed data set will be physically closed. While this option is beneficial from the standpoint of eliminating GBP-dependency (and overhead of writing to the GBP in the case of the last updating member), when GBP-dependency is re-established (by the first update to that data set), it causes physical open of the data set. Most users prefer this option as it eliminates data sharing overhead as soon as possible.
CLOSE=NO on the Create or Alter Tablespace DDL statement indicates that the data set is not to be physically closed following pseudo close of the data set. While it does not decrease the GBP overhead, it also does not involve physical close and subsequent physical open of the data set on the first access following the close. The situation this option avoids is a lot of open/close activity.

Application Design/Coding Techniques

Commit Work as Soon as Is Practical: To avoid unnecessary lock contentions, issue a COMMIT statement as soon as possible after reaching a point of consistency, even in read-only applications. To prevent unsuccessful SQL statements (such as PREPARE) from holding locks, issue a ROLLBACK statement after a failure. Statements issued through SPUFI can be committed immediately by the SPUFI auto commit feature.

Taking commit points frequently in a long running unit of recovery (UR) has the following benefits:

- Reduces lock contention
- Improves the effectiveness of lock avoidance, especially in a data sharing environment
- Reduces the elapsed time for Db2 system restart following a system failure
- Reduces the elapsed time for a unit of recovery to rollback following an application failure or an explicit rollback request by the application
- Provides more opportunity for utilities, such as online REORD, to break in

Set the DSNZPARM URCHKTH to help you identify those applications that are not committing frequently. The setting is a multiplier of Db2 checkpoint frequency and should conform to your installation standards for applications taking commit points. As long as one update was performed, it can identify even READERS who have not taken commits.

Even though an application may be conforming to the commit frequency standards of the installation under normal operational conditions, variations can occur based on system workload fluctuations. For example, a low-priority application may issue a commit frequently on a system that is lightly loaded.. However, under a heavy system load, the use of the CPU by the application may be pre-empted, and, as a result, the application may violate the rule set by URCHKTH. For this reason, add logic to your application to commit based on time elapsed since last commit, and not solely based on the amount of SQL processing performed. In addition, take frequent commit points in a long running unit of work that is read-only to reduce lock contention and to provide opportunities for utilities, such as online REORG, to break in (during the switch phase).

You can identify long running units of work in Db2 for z/OS with the DSNZPARM URLGWTH parameter of installation panel DSNTIPL. The value indicates the number of updates performed by this unit of work without a commit. The message is issued every time the threshold is hit and is cumulative for a given unit of work. If you set your default at 5000 records, you receive DSNJ031I the first time at 5000, the second time at 10000 etc.

Retry an Application After Deadlock or Timeout: Include logic in a batch program so that it retries an operation after a deadlock or timeout. That could help you recover from the situation without assistance from operations personnel. Field SQLERRD(3) in the SQLCA returns a reason code that indicates whether a deadlock or timeout occurred.

Close Cursors: If you define a cursor using the WITH HOLD option, the locks it needs can be held past a commit point. Use the CLOSE CURSOR statement as soon as possible in your program, to release those locks and free the resources they hold.

- Design for Lock Avoidance:
- Avoid using Table Space Locks where possible
- Lock avoidance, the ability to avoid taking read locks, is based on the continual advancement of the Global Commit Lock Sequence Number (the oldest Begin UR for all active URs for all members of the data sharing group). When it cannot advance (at commit time), more locks must be taken in the entire group. This condition is hard to diagnose without the long running warning messages described previously. Despite minimal updating, a rogue job increased locking by 4X recently in actual experience (5000 locks to 20,0000 locks)
- Review applications to determine if BIND options ISOLATION(CS) and CURRENTDATA(NO) can be used to reduce the number and duration of locks.
- Typically, ISOLATION(CS) lets Db2 release acquired locks as soon as possible. CURRENTDATA(NO) lets Db2 avoid acquiring locks as often as possible. After that, in order of decreasing preference for concurrency, use these bind options:

ISOLATION(CS) with CURRENTDATA (YES), when data you have accessed must not be changed before your next FETCH operation.

ISOLATION(RS), when rows you have accessed must not be changed before your application commits or rolls back. However, you do not care if other application processes insert additional rows.

ISOLATION(RR), when rows you have accessed must not be changed before your application commits or rolls back. New rows cannot be inserted into the answer set.

- Use ISOLATION(UR) cautiously: UR isolation acquires almost no locks. It is fast and causes little contention, but it reads uncommitted data. Do not use it unless you are sure that your applications and end users can accept the logical inconsistencies that can occur.

Consider defining tables with LOCKPART YES. This tells Db2 to track Inter-Db2 R/W Interest at the partition level instead of the table level.

- Bind Plans with ACQUIRE(USE): That choice is best for concurrency. Packages are always bound with ACQUIRE(USE), by default. Design for thread reuse and BIND packages with options ACQUIRE(USE) RELEASE(DEALLOCATE) to reduce global locking contention for the 20% highest volume of transactions and all batch jobs.

Monitor size of EDMPOOL, as use of RELEASE(DEALLOCATE) will drive up use of EDMPOOL storage. If packages are seldom used, binding with Release(Deallocate) wastes storage in the EDMPOOL.

- Beware of ROW Level Locking, this feature will increase Data Sharing locking overhead.

Access Data in a Consistent Order: When different applications access the same data, try to make them do so in the same sequence. For example, make both access rows 1,2,3,5 in that order. In that case, the first application to access the data delays the second, but the two applications cannot deadlock. For the same reason, try to make different applications access the same tables in the same order.

Other Considerations:

- Set up and test operational procedures for planned and unplanned outages. Recovery procedures need to be developed and tested for all types of scenarios - for example, CF failure, CF link failure, DASD failure involving Db2 data, recovery from an image copy, incremental image copy, and so on.
- Operational and recovery procedures should be well documented and easy to access.
- Ensure that each Db2 member is sufficiently sized to handle additional workload in the event another member of the data sharing group becomes unavailable.
- Carefully select the IRLM startup parameters (PC, DEADLOK, MAXCSA, and so on).
- Db2 for z/OS data sharing users can experience significantly improved Delete Name performance when pages are purged from the Group Buffer Pool (GBP) of data sets that are no longer shared. Db2 works together with the Name Class Queue support of the CFCC > 8 to reduce spikes in CF utilization caused by this long running command. The Db2 GBP directory is organized into DBID, PSID, and partition number queues that help to greatly reduce the CPU cost to delete the entries, sometimes by a factor of 10 with faster IBM Z servers and CFs.
- If you have a large number of Db2 physical data sets, consider splitting across multiple z/OS ICF Catalogs. This reduces ICF catalog contention when opening tables and limit your exposure in the event of a ICF catalog or DASD failure.

Consider creating a ICF User Catalog for each Db2 application group to improve resiliency. Allocate across different volumes, subsystem, and so on.

- There is a single copy of the Db2 Catalog/Directory that all members of the data sharing group share this catalog. It constitutes a ***single point of failure***.

Know and understand this limitation, have procedures in place to provide a quick recovery of the catalog if necessary.
At a minimum, take full daily backups of the Db2 Catalog/Directory. Consider taking backups on a more frequent basis to facilitate a quick recovery in the event of a Catalog failure or corruption (e.g. Prior to batch window, after batch window has completed, etc..)
Consider using Remote Copy technology to keep an backup “active” copy of the catalog. With PPRC at the same or remote site, the catalog/directory is not a single point of failure.

- Review your disaster recovery process and update and/or revise as necessary

Use the ARCHIVE LOG SCOPE(GROUP) parameter to achieve a point of consistency for each active member for DR purposes if you are using traditional DR scenario described in the Db2 Administration Guide under "Remote Site Recovery from a disaster at the local site".
You also need a separate CF and CFRM policy for the recovery site.
ALL active Db2 members must be started at the recovery site (there is no such thing as being in data sharing locally, but not for DR)
If you have disk from which "instant" copies can be made, please refer to a DR scenario that illustrates the procedure
Faster + Easier = Improved Db2 Disaster Recovery (FlashCopy)
ibm.com/support/techdocs/atsmastr.nsf/PubAllNum/WP100227

- The essential points of this procedure are that you have the same Recovery Point Objective as the traditional DR (about 24 hours) and you are not relying on more infrastructure at the recovery site than is assumed for the traditional Db2 DR scenario (no mirrored DASD to a second site). You also have the ability to perform an "instant" DASD copy, using FlashCopy or a similar technology of another DASD vendor.
- You issue -Set Log Suspend on each member. This activity writes the log buffers and stops all logging, allowing NO updates - but NO "quiesce" point or syncpoint occurs or is necessary
- You begin the "instant" copy or FlashCopy for all volumes in your data sharing group: catalog/directory, user data, all members' logs and BSDSs. When the "instant" part of the copy is complete ("logical relationship established") - a few minutes...
- You issue -Set Log Resume on each member and all activity resumes.
- You dump the FlashCopy (or other instant copy) to tape and take them to the recovery site.
- Your DR "recovery" consists of

Restoring the dumped volumes to DASD,
Forcing all the CF structures,
Starting the Db2 members normally (that will bring Db2 to the last complete UR before the log was suspended);
Getting all shared objects out of GRECP
Starting the workload - No complex procedures, fast recovery time.

- Restart "Light" is enhanced to remain up for the commit coordinator, such as IMS or CICS, to be restarted and to resolve the indoubt URs. DDF is restarted if it is allowed. No new work is allowed, and after all indoubt URs have been resolved, Db2 self-terminates.
- Child L-lock propagation is no longer based on the parent L-lock, but instead on cached (held) state of pageset P-lock. This results in less volatility for child locks. Parent L-locks are no longer need to be held in retained state after Db2 failure. In other words, a pageset IX L-lock no longer held as a retained X lock and provides an important availability benefit in data sharing IRLM now maps IX L-locks to XES S. Db2 grants IX parent L-locks locally when only IS or IX L-locks held on the object. Parent L-locks still sent to Lock Structure - but never contention for common conditions. The default BIND option RELEASE(COMMIT) becomes more attractive with this improvement. Choosing this option after you are in New Function Mode on Db2 for z/OS V8 requires a group-wide outage as the locking algorithm cannot be changed in XES for some members and not others without loss of data integrity. While this activity is optional, it is strongly ecommended for performance improvement in reducing global lock contention.
- LPL Recovery initiated automatically for many exception events (but not if detected during restart or if there is a DASD I/O failure).
- CF batching. Db2 now can read/write multiple pages (batches) to the CF with a single command and can read multiple pages for CASTOUT processing.

CICS and CICSPlex SM (CP/SM)

- Implement cloning of CICS regions:

Implement Socket Owning Regions (SORs) and TORs on multiple systems.

- ✓ Balance end-user logons using VTAM GR, eNetwork Dispatcher, DNS, and so on.
- ✓ For LU6.2 sessions, implement VTAM Persistent Sessions or VTAM MNPS for faster logon reconnect times.
- ✓ Ensure that SORs and TORs have access to all or most AORs.

Implement AORs on multiple systems, using the configuration in the diagram below as an example of the “ideal” CICS configuration:

- ✓ Eliminate CICS affinities and clone your CICS applications.

- Implement datasharing for cloned AORs, using:

IMS/DB
Db2
VSAM/RLS
Temporary Storage Queues, Coupling Facility Data Tables, Global ENQ/DEQ support, and Named Counter Server support.

- Share the DFHCSD data set among all clones and among the region “types” which comprise an application set of regions (TOR/AOR/QOR).
- Use one JCL procedure, started with the S procname,JOBNAME=stcname command to start the cloned regions. Use symbolic overrides to create unique instances.
- Implement dynamic routing for transactions, DPLs, and STARTS using CP/SM.
- Use CP/SM single system image and single point of control to:

Enable grouping of regions.
Reduce complexity and burden on operations.
Ease problem diagnosis for system programmers and operators.

- Use CP/SM system availability monitoring to enable early problem detection.
- Set logical application scopes using CP/SM Business Application Services to reduce complexity and avoid operator errors.
- Isolate the Maintenance Point CMAS from managed CICS regions. This will allow recovery of the CMAS on any available LPAR, as well as insulating the CICS regions from software level changes during maintenance and upgrade cycles.
- Use CP/SM real time analysis to monitor critical resources and raise alerts:

Set thresholds to detect problems early.
Use CP/SM automation facilities to re-enable monitoring across threshold boundaries.

- Configure CMASs in an any-to-any configuration.
- Start up CAS and CMAS during MVS initialization.
- Utilize storage protection and transaction isolation.

If it is not viable to run with storage protection and transaction isolation in production, consider running with these protection mechanisms turned on during stress testing to discover and correct coding errors.

- Use ARM to quickly restart failed CICS regions in the event of a region or system failure.
- Consider using autoinstall without cataloging to improve CICS restart times.
- If you are using RLS, split the SMSVSAM “Cache Set” between two different CFs for availability (and performance).

WebSphere MQ

Implement a WebSphere® MQ Queue Sharing Group

- Ensure pre-requisite hardware and software is available. A queue sharing group also stores information in a Db2 data sharing group.
- Start with WebSphere MQ Concepts and Planning Guide (GC34-6051) which outlines the tasks and contains pointers to the relevant sections of other WebSphere MQ books.
- Review the WebSphere MQ Capacity Planning SupportPac, MP1D for information about likely performance and Coupling Facility resource requirements: **ibm.com**/software/ts/mqseries/txppacs/mp16.html
- Review “WebSphere MQ in a z/OS Parallel Sysplex Environment, SG24-6864-00” from

- **ibm.com/**redbooks which discusses setting up and using MQ in a sysplex environment to improve throughput and availability of applications.

Subsystem Configuration

- Affinities: Removing the affinities between a queue manager and a particular z/OS image allows a queue manager to be restarted on a different z/OS image in the event of an image failure:

All page sets, logs, bootstrap data sets, code libraries, and queue manager configuration data sets must be defined on shared volumes.
The subsystem definition must have sysplex scope, a unique name within the sysplex, and be defined on each LPAR in the SYSPLEX.
The level of “early code” installed on every z/OS image at IPL time must be at the same level.
TCP virtual IP addresses (VIPA) must be available on each TCP stack in the sysplex, and you must configure WebSphere MQ TCP listeners and inbound connections to use VIPAs rather than default host names.

- Logging and Recovery

Implement Dual Logging and Dual BSDS support for each subsystem.
Ensure pageset and CFSTRUCT backups are taken frequently: the critical factor in media recovery scenarios is the amount and location of log data which must be reapplied to fuzzy pageset and structure backups to bring them 'up to date'. SupportPac MP16
ibm.com/software/ts/mqseries/txppacs/mp16.html contains an analysis of restart times and tuning information.
Consider adding an additional queue managers to the queue sharing group to provide additional logging 'bandwidth', if required, for CFSTRUCT backups.

- Coupling Facility Resources

• A queue sharing group uses Coupling Facility list structures to hold data. These structures are known in MQ as an administrative structure or an application structure.
Application structures are defined to MQSeries as CFSTRUCT objects.
An administrative structure is used for communication between queue managers in the Queue Sharing Group, and for management of units of work which involve shared queues. IBM recommends a minimum structure size of 10 MB for the administrative structure.
Note that the administrative structure constitutes a single point of failure for the queue sharing group and should be considered as a candidate for System Managed Structure Duplexing.
One or more application structures hold the messages which are resident on shared queues. The size of structure required for these will be application dependent.
MQSeries manages the backup and recovery of persistent messages held on a application structures through the use of simple commands.
Automate the backup of CFSTRUCTs

Application Design

- • WebSphere MQ Queue Sharing Group in a Parallel Sysplex environment REDP3636 from **ibm.com**/redbooks discusses considerations for designing applications to take advantage of an MQ Queue Sharing Group.

IMS

To ensure the highest availability for IMS applications, the applications should be cloned to run in several IMS subsystems, and data sharing implemented to allow each subsystem direct read/write access to the data. The following items will help you maximize your IMS application availability:

- Balance logons by exploiting VTAM Generic Resources for IMS and APPC/MVS.
- Implement IMS shared message queue support for transaction balancing.
- Implement transaction balancing for fast path messages using shared Expedited Message Handling (EMH).
- Implement FDBR to release locks quickly after IMS, MVS, and system failures.
- Use ARM for IMS Control Regions, IRLM, and CQS address spaces. Do not use ARM for FDBR address spaces since its use with FDBR disables the use of ARM for the IMS Control Region FDBR tracks.
- Convert batch programs to BMPs. BMPs have better availability characteristics, such as dynamic backouts after all abends.
- Ensure that all batch and BMP update programs take checkpoints, and that they do so at regular intervals.
- Use the LOCKMAX option to catch runaway BMPs and batch programs and to identify insufficient checkpoint frequencies.
- Minimize CI/CA splits and data set extensions by:

Ensuring sufficient data set space allocation.
Performing regular database reorgs.

- Carefully select PSB processing options (PROCOPT) for high volume transactions and large volume batch jobs.
- Carefully select the IRLM startup parameters (PC, DEADLOK, MAXCSA, and so on).
- Ensure that IRLM has a higher dispatching priority than the IMS, CICS, and Db2 address spaces, but not higher than the XCFAS address space.
- Use duplexed structures for VSO data sharing.
- Use System Managed duplexing for the SMQ and EMHQ structures
- Use System Managed duplexing for the IRLM structure If not, then ensure that the IRLM lock structure is in a failure-isolated CF.
- Specify overflow structures for shared queues and shared EMH queues.
- Use the IMSGROUP parameter for IMS control regions to simplify the execution of BMPs across multiple control regions.
- Use the same name for all IMS IRLMs. This simplifies the execution of IMS batch jobs across multiple MVS systems and the emergency restart of IMS systems on different MVS systems.
- Specify SCOPE=NODISCON for the IRLMs. This is especially important in systems where IMS batch jobs participate in data sharing.
- If you use IMS Connect, implement multiple instances of it. Each instance should have the capability to send messages to each IMS system. Implement your messsage exit routines to be sensitive to the availability of each IMS and route messages only to active IMS systems.

Communications Server for z/OS

SNA

- Implement VTAM Generic Resources to enable generic logons:

Ensure proper Coupling Facility structure size and placement.
Enable subsystem exploitation of generic resources.

- Exploit APPN□ and HPR.
- Define two APPN network nodes.
- Use ARM to quickly restart VTAM in the event of a VTAM failure.
- Use XCF for data transportation between systems.
- Exploit VTAM and application cloning via system symbols for:

Easier systems management.

Dynamic application definitions.

- Configure all systems within a Parallel Sysplex with the same NETID.
- VTAM Generic Resources requires that all systems in the sysplex be part of the same network.
- For LU 6.2 sessions, use Persistent Sessions (PS) or Multi-node Persistent Sessions (MNPS) to reduce post failure network restart time. IMS Version 7 introduces a new facility called Rapid Network Recovery that exploits this feature. For 3270-type sessions, the user can often be recovered more quickly by using VTAM Generic Resources and letting the user log on to another instance of the application, rather than having to wait for the instance the user was previously logged on to to be restarted.
- Monitor the following Web page for a list of important or recommended service that should be applied to Communication Server for z/OS: **ibm.com/**software/network/commserver/support/fixes/csos390.html
- Implement a Network Management product, such as NetView, to monitor the status of all network components, assist in problem diagnosis, and automatically respond to network failures.
- Provide multiple paths from each host to each network controller, to ensure connectivity can be maintained across the failure of any connection.

TCP/IP

- Implement Dynamic XCF to ensure automatic IP connectivity among all the TCP/IPs in the sysplex.
- Exploit Virtual IP Addressing (VIPA) for critical IP applications and enable physical interfaces to be backed up in case of a failure. Use dynamic VIPA takeover in case of a failure. Use the z/OS Automatic Restart Manager (ARM) for application server restart, as well as in-place restart for TCP/IP itself.
- Implement Dynamic VIPAs on an application basis to ensure fast server availability after a failure, as seen by clients:

Use Automated Takeover for multiple homogeneous servers (such as TSO or Web Server), each of which can satisfy the same client requests.

Use Application-Defined Dynamic VIPAs for unique restartable or movable server applications.

- Use Dynamic IP to balance TCP/IP clients over multiple stacks.
- Implement a WLM-supported Domain Name Server (DNS) for workload balancing of traditional business applications.
- Use multiple Telnet (TN3270E) servers with WLM/DNS.
- For Web Server connections, implement a WLM-guided Network Dispatcher to enable load balancing. The Cisco MultiNode Load Balancing component is another example of how WLM can be used to intelligently distribute IP network requests across the systems in a Parallel Sysplex.
- Use application cloning for the CS for z/OS TN3270E server so that there can be freedom of movement without corresponding VTAM definition coordination.
- Use system symbolics to simplify the task of TCP/IP configuration file maintenance.

- As mentioned in the SNA section, ensure that each host has multiple network interfaces. This, along, with Routing Daemons such as OMPROUTE, will help in situations where one of the interfaces had a failure.
- The TN3270E server supports multiple ports. You can define a "well known" backup port so that if the main telnet port (normally port 23) fails, the backup can be used.
- Use DHCP for dynamic assignment of client IP addresses. This makes system administration easier.
- The use of SNMPV3 for network management should be deployed.

Network

In just about every installation, the vast majority of the application users are remote to the computer center. Therefore the availability of the resources used to connect the users to the systems is critical. This section contains a checklist of actions for both SNA and TCP networks, to ensure they are configured for maximum availability:

- Configure multiple gateways, each having sufficient extra capacity to take over the other's workload in the event of an outage. Each gateway should be processing real work and have automated failover.
- Use a Communications Management Configuration (CMC) configuration:

Originally, CMCs were used to own the NCPs and attached devices. However, in an APPN environment, the CMC can be extended to being the DLUServer, the Central Directory Server (CDServer), and also possibly the APPN Border Nodes.

Isolate the primary CMC on its own CPC or LPAR image to prevent application-caused outages from affecting the network.

Configure a backup CMC on a failure-independent CPC image.

Use XCF for communications between the CMC and z/OS images.

Configure CMCs as APPN Network Nodes or Interchange Nodes. Interchange Nodes should be used if the CMCs still attach to NCPs and/or other VTAMs using subarea connections.

Configure remaining nodes as End Nodes or Migration Data Hosts.

Note: We strongly recommend placing the CMC(s) *inside* the sysplex. However, if the CMC *must* be outside the Parallel Sysplex, at least one system (and preferably two) within the sysplex must be configured as a Network Node.

- Implement High Performance Routing (HPR) to provide for nondisruptive path switching between nodes:

HPR/IP is the preferred method for SNA applications (non TN3270/E) to communicate over an IP backbone.

At a minimum, implement APPN/HPR between the gateway and the Parallel Sysplex.

If possible, move APPN/HPR out to the end user sessions to establish full end-to-end, nondisruptive recovery from failures.

Define backup routes to enable switchover in the event of a network or system failure.

If carrying SNA traffic over an IP backbone network, use the APPN Enterprise Extender to maintain high availability network characteristics, including the use of APPN/HPR end-to-end support.

If using TN3270E to carry SNA applications over an IP backbone, place the TN3270E servers on the host. The host will generally have better availability than a router, and you have the entire TCP/IP WAN between the client and the server, which means that full TCP/IP re-route will occur on any outage.

Use of automation

When installations first started implementing automated operations, the intent in many cases was to use automation to reduce the number of operators required to manage the systems. As a result, the degree to which systems are automated varies widely from installation to installation.

However, for any installation facing increasingly demanding availability requirements, automation is no longer a "nice to have" - it is an absolute necessity. This is because of the following reasons:

- t is not possible to read and comprehend every message that is issued, even on a mid-sized z/OS system. Some action needs to be taken to reduce the volume of messages. And the larger the system, the more aggressive that reduction needs to be.
- The complexity and diversity of software and applications in most installations makes it difficult, if not impossible, for any operator to know the correct response for every possible message and error without referring to a message or procedures manual, by which time the problem may have escalated to the point that it cannot be resolved.

  Also, as requirements for continuous availability drive the amount of redundant components in a system, it is important that special attention is paid to ensuring that these components remain available. Because they are redundant, it may not be immediately obvious that one of them has become unavailable. Therefore, automation should be put in place to specifically monitor these redundant components.
- Modern systems are so fast that a problem can very quickly build up, impacting other components, to the point where the whole system starts locking up. Immediate reaction to any error is critical, and this cannot be achieved without automation.
- Because of the level of interaction between systems in a sysplex, it is no longer possible to effectively manage a sysplex without a sysplex-wide view of everything that is happening. While it is possible to route the messages from all systems to a single console, the resulting volume of messages would make it impossible for any human to understand and react to all the messages.
- To minimize outage time, the shutdown and startup of the system must be automated. Automation is also required to make sure all required subsystems and resources are activated after an IPL, and in the correct sequence.

In this chapter, we address each of this issues and also discuss the topic of ensuring the ongoing effectiveness of your automation.

Automation tools

One of the first steps in the implementation of an automation project is to decide *what* you want to automate, and *which* is the most appropriate tool to do the automation. Generally speaking, the recommendation is to automate as close to the source of the message as possible. This improves performance and reduces the number of components involved in the automation process. There are many automation tools available; some are standalone automation products, others are "built-in" functions. In this section we list some of the automation features and automation products available from IBM:

- Built-in automation functions such as:
- z/OS's Sysplex Failure Manager (SFM).
- z/OS's Automatic Restart Manager (ARM).
- z/OS's Message Processing Facility (MPF).

- z/OS’s Command Prefix Facility (CPF).
- z/OS’s Initialization Command Processing (COMMNDxx).
- Individual component functions such as:

Automatic MCS console switchover.
JES2 automatic command processing facility.
JES2 automatic checkpoint lock release.
JES2 automatic checkpoint forwarding.
Db2 automatic recovery (AUTOREC) and Db2 GBP failover if GBP duplexing is used.
Automatic structure rebuild (most CF exploiters).
MVS automatic job cancellation:

  - ✓ Job execution time exceeded.
  - ✓ Job wait time exceeded.

JES2 timed event.
and so on ...

- Other products to consider for easing system management tasks:

System Automation for z/OS.
Hardware Configuration Manager (HCM).
RMF.
IBM Workload Scheduler for tracking of batch jobs and, optionally, started tasks.

Message volume reduction

Once you have decided what you want to automate and which tools you are going to use for that automation, the next step is to reduce the volume of messages presented to the operators, and provide them with a mechanism to easily and quickly determine if everything is operating correctly. You can accomplish these goals in the following ways:

- Suppress as many messages as possible.
- Set up consoles to receive local message traffic using the MSCOPE=(*) parameter.
- Automate messages that result in repetitive operator actions.
- Automate recovery actions for failure events.
- Design an effective and manageable “exceptions console” for Operations:

Provide a topographical view of the sysplex.
Sysplex-wide view on the first “summary” screen.
Graphical representation of major system components.
“Drill down” capability (sysplex > system > subsystem > application).

- Issue alerts when operator intervention is required.
- Color code alerts to indicate criticality (Red, Yellow, Green).

An example of a well-designed status indicator for critical problems would be something like the following:

- A critical event turns on a summary “red light”.
- The operator clicks on the red indicator to drill down for detailed information about the problem. If the condition causing the red light is resolved, the red light should be turned off. If the condition causing the red light is unresolved, the red light should be turned off but the border of the light should remain red as a reminder that an unresolved critical condition exists. New critical alerts should cause the light to turn completely red again.

Managing complexity

An average z/OS system could easily have 100 software products installed, and twice that number of applications. To maximize the availability of the system and the applications, it is important that all messages and errors are addressed correctly. The best way to achieve this is by the operations and technical support or application support groups working together to agree on the most appropriate response for each message. The specialist for each product should work with the automation specialist to decide if the appropriate response is to suppress the message, automatically respond to it, or display it to the operators so they can respond manually.

- For each product in your software inventory, set up automation and message suppression routines for all the messages.
- Every time a new release or a significant amount of service is installed, check to make sure all the messages are addressed and that the responses are still appropriate.
- Run the message analysis program (which can be downloaded from the Web site: www.**ibm.com**/servers/eserver/zseries/software/sa/sadownload.html) after the installation of service or a new release to make sure there are no new messages that you were not aware of, or have not automated.
- As a rule, application programs should not send messages to the z/OS console. Abnormal conditions should be addressed by the program ending with a meaningful return code - this can then be handled by the scheduling package. Purely informational messages, if they are really needed, should be routed somewhere other than the console.

The other issue to address is monitoring of the redundant system components. You have to not only make sure that all the redundant components are available, but also that the operators and system programmers are aware if a failure in some component means that you no longer have redundancy in that area.

The following are examples of redundantly configured components that should be actively monitored to ensure that redundancy is maintained:

- Redundant data sets (Sysplex Couple Data Sets, JES2 checkpoint data sets, …)
- Server Time Protocol
- Coupling Facilities and structures (CF links, restoring CF structures to their original location after a planned or unplanned CF outage.)
- CPUs.
- FICON channels
- Channel paths.
- Ensure that hardware consoles (HMC, ETR, ESCD) remain available.
- Ensure that the hardware “Call Home” feature remains enabled.

Automation performance

It is important that the automation tools are able to respond to messages and errors in a timely manner. Even the worst-performing automation package should be faster than an operator, however, it is still important that the automation package has good performance, so that it can address problems before they can escalate, and also so that the automation itself does not place too large a load on the system. The following recommendations will help you achieve this objective:

- Make sure the automation package has very high priority, so that it can respond to loops or high-CPU conditions, rather than being locked out by them.
- Suppress as many messages at source as possible. If a product gives you any control over which messages are issued, do not issue a message unless you actually need the information it contains.
- Use MPF to suppress any messages that you do not need to display or automate.
- Within your automation routines, handle the most frequently occurring messages first, to avoid the overhead of scanning through many automation routines before hitting these messages that get issued very frequently.
- After these high volume messages, place the routines for any errors that could potentially cause an availability impact.

Sysplex-wide automation

While we earlier stated that messages should be automated as close to the source as possible, there are some messages that require a knowledge of what is happening on other systems before they can be responded to correctly. For such messages, your automation package should have the ability of forwarding them to a focal point system that has information about what is happening on all systems. The focal point system can then decide on the most appropriate response and route that response to each of the systems. The automation package should also provide the ability to automatically failover to a backup focal point, in case the focal point or the system it is running on fails.

Another reason for having a focal point is to give the operators a single, integrated view of the status of all systems. Having just a single place to monitor should improve their response time to any error situations and also help them see if a problem is just affecting a single system or if it is having a multisystem effect.

Startup and shutdown

To achieve high availability, system and subsystem startup, shutdown, and restart times must be minimized through the use of automation.

- Enabling AUTOIPL when not in a GDPS configuration should be the first thing done.
- Use System Recovery Boost starting on the IBM z15 to speed up shutdown and IPL.
- Fully automate system shutdown sequences:

Perform normal shutdown of applications and subsystems.

Perform shutdowns in parallel if possible.

Use SFM to automate system shutdown after the V XCF,sysname,OFFLINE command is issued. Verify BCPii and SSD are enabled for this.

Minimize or eliminate operator intervention after the shutdown command is issued.

- Fully automate system startup sequences:

Start up mission-critical functions (subsystems) concurrently.

Delay automatic startup on non-critical functions (monitors and other tools) until mission-critical subsystems are up and running.

Minimize or eliminate operator intervention during system/subsystem startup.

Effective systems management processes

In one analysis IBM did of multi-system outages, nearly half of the outages were caused by human error. This problem can be partly addressed by effective and comprehensive automated operations, but effective systems management practices can help avoid having the error situations arise in the first place.

In this chapter we discuss some of the system management processes, and how each relates to availability in a Parallel Sysplex.

Availability Management

Design and manage systems for high availability*:*

- Assign an *Availability Manager* who has a dedicated focus on availability and is responsible for ensuring that the organization's availability objectives are met.
- Assign an *Availability Architect* who is responsible for designing a high availability systems environment and who acts as the technical leader for high availability initiatives.
- Ensure that systems and applications are designed for high availability.
- Ensure that the system management processes are effective. Analyze all outages to see if they could have been avoided by better processes, or better adherence to existing processes.
- Perform a business impact analysis for each application:

Identify the availability requirements for each application.

Regularly review (at least once a year) to ensure that your documented availability requirements still meet the business' requirements.

Don't forget dependencies between applications. A very high availability application may actually depend on a smaller application that, by itself, does not have very high availability requirements.

- Establish meaningful availability measurements (usually at the system, subsystem, network and most importantly, at the end-user level).
- Track, analyze and report availability statistics.
- Create organizational "awareness" of availability objectives and results. Tying bonus programs into availability objectives focuses the mind very effectively!
- Perform a Component Failure Impact Analysis (CFIA) to identify availability exposures**:**

Determine any potential causes of application unavailability.

Identify complexity and lack of redundancy in elements in the transaction processing path.

Identify Single Points of Failure (SPOFs).

- Conduct detailed outage analysis for all outages:

Identify opportunities to reduce recovery time.

Identify system design, skills, process issues.

- Plan and manage availability improvements.
- Availability management is an iterative process. Iterate continuously.

Change Management

Change management is basically all about ensuring that errors are not introduced into the production environment - and giving you the information you need to back them out when they do! One of the key tools in weeding out all errors at an early stage is an effective test environment. The following list provides considerations specifically aimed at testing in a high availability Parallel Sysplex environment:

- Ensure that an adequate test environment exists to stress test changes. A completely separate testplex is optimal for achieving high availability. A testplex is advantageous because:

It provides a environment where tests and experiments can be conducted with no possible impact to production systems.
It provides the capability to stress test changes and identify errors before changes are introduced into the production environment.
It enables operators and system programmers to experiment with configuration changes and operational procedures. In this sense, the testplex is also used as an education plex (sometimes referred to as an “edplex”).

Can test and production systems coexist in the same sysplex? *Yes!* However, there are limitations on the type of testing and “experiments” that can be performed when test and production coexist. Technical staff must use caution to ensure that production systems are not impacted by test system activities. This is especially the case when functions are enabled in both test and production that would result in common usage of resources (for example, the IGWLOCK00 structure, CFs in general, and the CDSes). Also, a sysplex shared between test and production does not provide a test environment for functions that are sysplex-wide in scope (e.g., GRS STAR, CFRM POLICIES, WLM POLICIES). Installations must decide whether a separate testplex is necessary based on:

- Your availability requirements.
- How much cost can you bear.
- System programmer and operator skills.
- The effectiveness of existing test configurations and methodologies.
- The frequency and risk of current and projected system changes.

Regardless of whether you have a separate sysplex for testing or if you use the production one (more likely the development plex, in most cases), there are things to consider to maximize the
value of the testing you perform:

- Use TPNS and other simulators to create representative system loads.
- Stress testing and testing of changes that pose a risk to the production systems environment should take place in an isolated test environment:

Either create a “testplex“ which is fully separate from the production sysplex, or schedule a time when an outage of the production sysplex would be acceptable.
Create a separate pool of DASD for the test environment, ensuring that there is no production data on the test volumes or catalogs.
Ensure that other resources used by the test environment will not impact the production systems. For example, define separate Coupling Facilities for test structures.

- Implement a staged roll-out of changes:

Stress test changes in a separate testplex.
Migrate changes from the test environment to a quality assurance/development (QA) environment.
Migrate changes from the QA environment into production.

- Implement effective organizational processes to ensure that changes are:

Coordinated across the enterprise.
Reviewed by appropriate technical and management personnel.
Assessed and risk minimized (consider “freezing” changes during critical processing periods).
Approved and tracked.
Keep procedures, configuration diagrams and other documentation up to date.

Problem Management

- Assign a *Problem Manager* who is responsible for overseeing the execution of the problem management process.
- Establish a tiered problem management support system. For example:
- Level 1:

Help desk

- Level 2:

System operations
Network operations

- Level 3:

System programming staff

- Level 4:

Crisis team

- Establish a problem management escalation policy based on the severity and impact of problems.
- Ensure that help desk support and problem data is easily accessible to system users and technical staff.
- Create a single common database for problem records.
- Utilize effective problem management tools to record problem data and search for known problems.
- The problem management team must:

Provide an accurate and comprehensive description of reported problems.
Accurately record:

- ✓ Outage time.
- ✓ End-user impact.
- ✓ Recovery time.
- ✓ Response times from support staff and vendors.

Assess problem severity and impact.
Identify, circumvent, and resolve problems.
Identify and resolve secondary problems.
Participate in post-incident reviews.

Recovery Management

Anticipate failures and plan recovery actions:

- Ensure that effective back-out plans exist in the event changes result in disruptions.
- Establish recovery escalation policies based on impact scope (IPL instead of spending excess time in problem determination).
- Develop, document, and maintain effective recovery procedures for:

Environmental facilities.
System hardware.
Operating systems, subsystems and other software products.
Network.
Applications.
Data.

- Develop, document, and maintain emergency shutdown and startup procedures.

- Ensure that support staff are adequately trained to handle exceptions.
- Perform recovery “fire drills” frequently.
- Evaluate the effectiveness of recovery procedures and make improvements as needed.
- Automate recovery actions as much as possible.

Crisis Management

When crisis strikes, is there frenzied chaos, or is there an orderly approach to handling a crisis?

- Develop crisis management protocols:
- Establish a crisis team consisting of key technical staff and a Crisis Manager.
- The Crisis Manager must:

Be empowered to make decisions.
Be the focal point for all crisis activities.
Coordinate the crisis and communicate progress to management.

- Establish an effective communication mechanism to quickly assemble the crisis team.
- Ensure that the crisis team has access to systems and other important resources.
- Establish an effective communication mechanism for the crisis team to use when handling a crisis (usually a telephone conference line).
- Develop crisis protocols that enable an orderly approach toward handling a crisis.

Establish a system health check topology.
Crisis team “roll calls” to assess system status.
Establish a “recovery action” escalation policy.

- Ensure that support organizations understand the crisis policies and expectations.
- Practice crisis management techniques and drill staff regularly.
- Create a crisis timeline and document activities during a crisis.

Conduct a post mortem review.
Tune the crisis management techniques to improve effectiveness.

Proactive Maintenance

Outage analysis has shown that about 15% of outages affecting multiple systems/subsystems could have been avoided by better preventive maintenance practices. That is, the cause of the outage was fixed by a PTF that had been available for six months or more.

- Have a well-defined process to install preventive maintenance on a regular basis.
- To have the least impact on Parallel Sysplex availability, it is recommended that maintenance be installed and activated on one system at a time. This is known as ***rolling IPLs***.
- Review HIPER APARs weekly and installing the applicable PTFs on the test system, in preparation for production, as needed.

HIPER maintenance should be rolled out to production systems on a regular basis, such as monthly.
Severity 1 HIPERs and those HIPERs also marked pervasive should be strongly considered.

- Install maintenance based upon the Recommended Service Upgrade (RSU) level. More information about RSU, and ordering and installing the redefined recommended service is available at: **ibm.com**/servers/eserver/zseries/zos/servicetst

Proactive systems management

Anticipate system requirements. Don’t wait for the system to tell you that it has run out of resources!

- Monitor resources and put automatic mechanisms in place to handle resource shortages

XCF groups and members (D XCF,C).
Data set usage.
z/OS storage utilization (CSA utilization, paging rates).
Processor utilization.
I/O path performance.
Device performance.

- XCF signaling configuration.

Buffer utilization.
Path performance.

- Coupling Facility utilization.

Contention rates.
Response times.
Link performance.

- Capture diagnostic data (dumps, logrecs) and ensure that problems are properly resolved.
- Establish effective data backup and restore processes for critical databases, and subsystem and system data.
- Validate your current software maintenance practices. One outage analysis study in IBM showed that a large number of the outages would have been avoided had the systems in question been at more current service levels.
- Establish proactive maintenance programs for:

□ System support infrastructure (facility).
□ System hardware and software products.

✓ Use SMP/E HOLDDATA to ensure that HIPER fixes are evaluated and applied on a regular basis.

- Ensure that generations of software running in the sysplex are within the product specifications (for example, N, N-2).
- Create and maintain comprehensive and effective system documentation. Ensure that all technical staff has access to documentation needed to manage the IT environment:

Operational procedures.

✓ System operation and recovery procedures.
✓ Enterprise processes.

Configurations diagrams (hardware and software).
Mapping of system and application interdependencies.
Product documentation.

Staff technical skills

Your technical staff designs, implements, maintains, and operates the Parallel Sysplex systems environment. Without them, achieving high availability is not possible. Strong technical skills take a long time to develop. To protect themselves from a skills exposure, IT organizations must:

- Ensure that backup technical skills exist. (Eliminate “people” single points of failure).
- Ensure that the skill level of technical staff is maintained by regular hands-on exercises.
- Be cautious when transferring people to/from technical positions. (Technically skilled individuals are not “plug & play”).

In a general sense, system programmers, operators and support staff must be well trained in all of the components that comprise the Parallel Sysplex platform including:

- System support facilities:

Power distribution systems.
Environmental systems.

- Hardware components:
  - CPUs.
  - I/O and peripheral devices.
- Software components:
  - Operating system and subsystem software.
  - Vendor products.

Applications.

- Network components:
  - Routing and distribution networks.
  - Related software.

Technical staff must be able to operate, configure, and manage the Parallel Sysplex hardware environment. To manage systems for high availability, technical staff must be able, at a minimum, be able to perform the following tasks:

- Activate and deactivate hardware devices.
- Operate devices.
- Perform customization tasks.
- Perform reconfiguration tasks.
- Analyze hardware performance and perform tuning tasks.
- Determine hardware status.
- Recover from hardware failures.

Access and understand hardware logs.

- Understand hardware behavior and method of operation of all devices, including:

System Time Protocol (STP)
Coupling Facilities
z/OS Servers
FICON / SAN network
DASD
Network controllers/gateway devices
Tape
Hardware consoles (HMC)
Operator consoles
Device interconnections (cables, switches, patch panels, LANs)

The following lists some of the tasks that are expected of operators and system programmers. Depending on the enterprise, some of these tasks may be the responsibility of operators, system programmers, or both. Operators and/or systems programmers must be able to:

- Customize systems and subsystems.
- Add a system to the sysplex.
- Remove systems from the sysplex.
- Reinitialize the sysplex.
- Manage sysplex policies.
- Manage sysplex data sets.
- Start and stop subsystems.

- Start and stop database processing.
- Move workloads within the current system and to other systems.
- Perform database reorgs, backups, and image copies.
- Perform database forward and backward recovery.
- Manage z/OS systems:

Adjust address space dispatching priorities.
Adjust LPAR weights.
Dynamically change the I/O configuration.
Use appropriate commands and understand which are single system/subsystem and which are "plex-wide".
Know "tricks of the trade" such as:

- ✓ Using Multi-System DAE to reduce duplicate SVC dumps.
- ✓ Using PA1 to recall commands (CONSOLxx -> CONSOLE(RBUF)).
- ✓ Use of the ROUTE command.
- ✓ Defining and using system groups (IEEGSYS).
- ✓ Automatic routing of commands using CMDSYS.
- ✓ Use of wildcards.

Use tools such as ISPF, SDSF, TSO, and so on.
Manage system workloads.

- Manage system and subsystem logs.
- Use the hardware console to:

Specify load parameters.
IPL systems.
View system activity, operating system messages, and system status.
Communicate with the operating system (Use V CN(*),activate).
Define system "groups" (including a separate group for standalone dumps).

- Perform problem determination and recovery:

Determine if the problem is hardware or software.
Identify the failing hardware or software component.
Recover from failures and from problems encountered during recovery

- Recover from system failures including:

Enqueue deadlocks, reserve lockouts.
Device start pending conditions.
Console hangs, dead console conditions.
Resource shortages.
Address space hangs.
System enabled WAITs.
System disabled WAITs.
System or workunit enabled/disabled loops.
Data set corruption.
Problem previously encountered (learn from past outages!).

- Recover from subsystem failures.
- Manage the datasharing environment.
- Recover from failures in the datasharing environment.
- Monitor IRLM including recovery for:

Lock contention.
Retained locks.

Lost locks.

- Manage Coupling Facilities including:

A thorough understanding of Coupling Facility exploiters.

- ✓ Structure characteristics.
- ✓ Recovery characteristics.
- ✓ Policy change characteristics/policy “in transition”.

Displaying CF information:

- ✓ Displaying policy information.
- ✓ Defining and switching CFRM policies.

Displaying CF structures.
Rebuilding and duplexing/unduplexing structures.
Handling failed-persistent connectors.
Handling persistent structures.
Initiating structure rebuilds.
Altering the size and placement of structures.
Shutting down a Coupling Facility.
Reinitializing a Coupling Facility.

- Recover from CF failures:

CF outage.
Rebuild hangs.
Structures in transition.
Loss of connectivity.
Restore everything to its original state after the failure has been rectified.

- Recover from connectivity failures:

XCF signaling connectivity.
Device (DASD, tape, etc...) connectivity.

- Collect documentation using various utilities and tools:

Standalone dump.
SVC (Console) dump.
Coupling Facility dump.
GTF trace.
Component trace.

- Utilize system and subsystem utilities.
- Select and install maintenance.
- Create, understand, maintain, and utilize hardware system diagrams.
- Create, understand, maintain, and utilize operational and recovery procedures.
- Create, understand, maintain, and utilize a mapping of software systems, subsystems and workload interdependencies.
- Understand how the systems are automated:

Create automation scripts.
Operate the automation platform.
Recover from automation failures.
Maintain system operations during an automation outage.

- Follow escalation policies, and know when and how to:

Perform emergency hardware and software shutdowns.
Contact the next level of support (including vendors).
Contact management.

Implement disaster recovery protocols.

Obtaining skills

System programmer and operator skills can be obtained from a variety of sources. IBM has developed a roadmap for Parallel Sysplex operator and system programmer training and certification.

Education programs provide a good skills base for operators and system programmers. However, to maintain and enhance these basic skills, operators and system programmers should have access to as much "hands on" sysplex training as possible. The following steps are recommended for obtaining, enhancing and maintaining optimal operator and system programmer skill levels:

- Attend courses and obtain certifications recommended by the following programs:
- Parallel Sysplex Operator Certification.
- Parallel Sysplex Data Sharing Operator Certification.
- Parallel Sysplex System Programmer Certification.

For information on training and certification programs, visit the IBM certification Web site:
ibm.com/certify/index.shtml

To enable as much "hands on" training as possible, IBM recommends that a test sysplex (testplex) be created from a cloned version of the production environment. To the extent possible, the testplex should be completely separate from the production sysplex. The testplex should include a separate set of hardware resources and cloned operating system, subsystem, and application software. For training purposes, the testplex need only be a fraction of the size of the production sysplex in terms of:

- Number of systems (LPARs).
- LPAR resources (CPs, storage, weights).
- DASD and I/O resources.

The testplex should contain at least two systems (preferably three). The following summarizes the steps required to provide you with the infrastructure required to provided the required hands-on training capability.

- Create a testplex:

Separate the production and test environments as much as possible.
Create a testplex of z/OS and CF images using LPARs or VM guests. For additional information on Parallel Sysplex testing under VM, see the Web site: vm.**ibm.com**/os390/

- Define production interdependencies and testing limitations (for example, if the test LPARs reside on the same CPC as production images, you cannot run "loss of power" test scenarios by powering off the CPC - in this case, LPAR deactivation can be used as an alternative).
- Develop test and recovery scenarios based on:

The Parallel Sysplex Test and Recovery Planning manual, GG66-3270.
z/OS Parallel Sysplex Recovery as documented in the Parallel Sysplex Test Report:
www.**ibm.com**/servers/eserver/zseries/zos/integtst/library.html
For application and subsystem configuration, develop failure scenarios for components that are in the application transaction "pathway".
Course materials provided by training programs.

Here are some examples of test and recovery scenarios:

- Simulate a system failure:
- Start applications and simulate a workload (using TPNS or other simulator).
- Perform a hardware STOP on the HMC to place the system into manual state.
- Observe ALERTS/messages and respond to the failure based on installation procedures.
- Develop variations of this test scenario including:

Running the scenario with and without the use of ARM.

Running the scenario with different SFM policies (use PROMPT, ISOLATETIME, and so on).

Demonstrate manual and automatic workload switchover to other systems.

Practice system commands such as D XCF,S,ALL and D R,L.

Practice IPL, subsystem restart, and standalone dump procedures.

Practice moving workloads back to the restored system.

- Simulate a Coupling Facility failure:

Populate the CF with structures and activate workloads.

From the HMC, deactivate the CF LPAR.

Observe ALERTS/messages and respond to the failure based on installation procedures.

Develop variations of this test scenario including:

Modifying CFRM policies to disable structure REBUILD capability. When the failure occurs, observe the symptoms of the “lost structure”, then recover the function either by starting a new policy that allows REBUILD to proceed, or recover the function through some other means (move the JES2 checkpoint to DASD, start XCF signaling paths through CTCs, etc.).

Disconnect (or VARY OFFLINE,FORCE) CF links to simulate loss of CF connectivity and experiment with CFRM’s REBUILDPERCENT function.

- Simulate a Db2 subsystem failure:

Start the Db2 subsystem and workload.

Cancel the Db2 master address space or the IRLM address space.

Observe ALERTS/messages and respond to the failure based on installation procedures.

Develop variations of this test scenario including:

✓ Running the scenario with and without the use of ARM.
✓ Demonstrate manual and automatic workload switchover to other systems.
✓ Practice manual subsystem restart.
✓ Practice moving workloads back to the restored system.

Use HCD to dynamically change the I/O configuration.

Simulate a CPC failure by deactivating (powering off) the CPC or deactivating the LPAR.

Simulate a console hang by disconnecting the device cable:

✓ Use the ROUTE command to verify system and device status.
✓ Use the HMC system console to communicate with the system.
✓ Perform other tests appropriate to your environment.

- Develop scenarios based on day-to-day operations including:

Use of system and subsystem commands.

Customization and configuration changes.

System and subsystem startup and shutdown.

- Combine operations and recovery scenarios into skills development program:

Run periodic recovery “fire drills” for operators.

Train new operators.

- Keep the testplex consoles in the operations area for operators to use for experiments such as trying new commands.

- Encourage system programmers to use the testplex to perform customization tasks, dynamic changes, develop operational scenarios, and other experiments.

Develop high availability applications

Poor design and implementation of applications can defeat the best high availability system infrastructure. It is therefore important to ensure that applications are not the "weak link" in system availability.

- Functionally isolate or modularize applications to ease application design, development and management tasks:

Enable efficient use of storage by application "mainline" modules.

- Design applications for high availability:

Reduce application complexity and interdependencies.

Reduce or eliminate system and transactional affinities to enable applications to run cloned across multiple systems.

Develop coding standards which:

- ✓ Prevent the introduction of affinities into the application workload.
- ✓ Avoid requirements for exclusive use of resources.

Implement functions that provide high application availability:

- ✓ IMS, Db2, and VSAM/RLS datasharing.
- ✓ Avoid holding locks for long periods by doing frequent checkpoints/commits.
- ✓ Eliminate database locking deadlocks.
- ✓ Avoid designs that may result in database "hot spots".

IMS:

- Reduce the use of ROLL call.
- Avoid the use of large Scratch Pad Areas.
- Use MODE=SNGL or Express PCB to avoid tying up DB blocks.
- Reduce the number of DL/I calls to reduce locking contention rates.

Db2:

- Use type 2 indexes instead of type 1.
- Use page level locking and reduce the number of rows per page rather than using row level locking.
- Use UR isolation level.
- Use CS versus RR isolation level.
- Bind with CURRENTDATA(NO).
- Bind with RELEASE(DEALLOCATE) instead of RELEASE(COMMIT).
- Issue frequent checkpoints to ensure that recovery time is short.
- Ensure that applications can be restarted at the last declared checkpoint.
- Allow for the coexistence of batch and interactive workloads.

Applications must be error-free:

- Ensure that application changes are rigorously reviewed and tested.
- Applications must support at least N, N+1 compatibility to enable the roll-out of code changes within the datasharing group.

Summary

The purpose of this checklist is to make you *THINK AVAILABILITY* when you are:

- Selecting a computer site and supporting infrastructure.
- Deciding on hardware and software design (Host, Network, Clients).
- Selecting computer equipment and software products.
- Configuring hardware and software.
- Designing IT operations infrastructure and management processes.

Before system changes are made, technical staff should think “How will this change affect system and sysplex availability?” Before operators use a system command or operate a console, they should think “How will this change affect system availability?”

Technical staff and management should have an awareness of the IT enterprise availability goals. This *availability conscientiousness*, in conjunction with managing systems towards the availability goals, will result in improved system availability. Use this checklist as a guide. Depending on your installation and availability goals, additional product research may be necessary when designing and configuring systems for high availability. With appropriate technical skills, and an implicit awareness of availability requirements, designing, implementing and managing systems for high availability should be a natural evolution. There is *no plug and play solution* to achieving high availability. To achieve a high availability systems environment, organizations must *focus on availability* and invest in designing, configuring and managing systems for high availability.

Reference information

Listed below are some sources of information that should be helpful when designing a high availability Parallel Sysplex platform.

Redbooks

• Getting Started with IBM Z Resiliency	SG24-8446
• z/OS MVS Parallel Sysplex Configuration Volume 1: Overview,	SG24-2075
• z/OS MVS Parallel Sysplex Configuration Volume 2: Cookbook,	SG24-2076
• z/OS MVS Parallel Sysplex Configuration Volume 3: Connectivity,	SG24-2077
• Getting the Most Out of a Parallel Sysplex,	SG24-2073
• Continuous Availability S/390 Technology Guide,	SG24-2086
• Continuous Availability - Systems Design Guide,	SG24-2085
• System/390 MVS Parallel Sysplex Continuous Availability SE Guide,	SG24-4503
• System/390 MVS Parallel Sysplex Continuous Availability Presentation Guide,	SG24-4502
• Parallel Sysplex Test and Recovery Planning (Orange Book),	GG66-3270
• Parallel Sysplex Operational Scenarios,	SG24-2079
• Parallel Sysplex Managing Software for Availability,	SG24-5451
• z/OS Parallel Sysplex Application Considerations Presentation Guide,	SG24-4743
• Parallel Sysplex Automation Guidelines,	SG24-5441
• Batch Processing in a Parallel Sysplex,	SG24-5329
• TCP/IP in a Sysplex,	SG24-5235
• SNA in a Sysplex,	SG24-2113
• Parallel Sysplex Coupling Facility Online Monitor: Installation User's Guide,	SG24-5153
• IBM Z Server Time Protocol Guide	SG24-8480

Server Time Protocol Implementation Guide	SG24-7281
Server Time Protocol Planning Guide	SG24-7280
Server Time Protocol Recovery Guide	SG24-7380

Product Documentation

z/OS MVS Setting Up a Sysplex,	GC28-1779
z/OS Parallel Sysplex Systems Management,	GC28-1861
z/OS MVS System Commands,	GC28-1781
System Overview, S/390 9672 Generation 6,	GA22-1030

Web sites

https://www.ibm.com/services/business-continuity	(IBM Business Continuity Services)
www.redbooks.ibm.com/	(Redbooks)
https://www.ibm.com/it-infrastructure/z	IBM Z Home page
www.ibm.com/it-infrastructure/z/zos	(z/OS Home page)
www.ibm.com/it-infrastructure/z/technologies/parallel-sysplex	(Parallel Sysplex home page)
https://www.ibm.com/it-infrastructure/z/zvm	(z/VM Home Page)
www.ibm.com/it-infrastructure/z/zos-workload-management	(WLM Web site)
www.ibm.com/support/techdocs/atsmastr.nsf/Web/Techdocs	(Tech Docs)
www.ibm.com/support/pages/node/664965	(z/OS Consolidated Service Test)
http://public.dhe.ibm.com/systems/z/servicetest/zOS_Preventive_Maintenance_Strategy.pdf/	(z/OS Preventive Maintenance Strategy to Maintain System Availability)
www.ibm.com/it-infrastructure/z/software	(IBM Z software products)
www.ibm.com/it-infrastructure/z/cics	(CICS on IBM Z Home Page)
www.ibm.com/it-infrastructure/z/ims/	(IMS Home Page)
www.ibm.com/analytics/db2/zos	(Db2 for z/OS Home Page)
www.ibm.com/certify/	(Certification Programs from IBM)
www.ibm.com/training/M425350C34234U21	Digital Badges
Ftp://ftp.software.**ibm.com**/s390/mvs/tools/	(XISOLATE Tool/Documentation)

WSC Flashes

http://www.ibm.com/support/techdocs/atsmastr.nsf/Web/Flashes

- W98029: Parallel Sysplex Configuration Planning for Availability
- W9715A: GRS STAR APAR/PTF and lock structure sizing recommendations
- 10011: XCF Performance Considerations

FLASH10002	MVS/ESA Parallel Sysplex Performance - LPAR Performance Considerations for Parallel Sysplex Environments
FLASH10009	Using a Coupling Facility for the JES2 Checkpoint
FLASH10022	z/OS Parallel Sysplex: Performance Impacts of Using Shared ICF CPs
FLASH10786	Where is My Coupling Facility?
FLASH10159	z/OS Performance: Heuristic Algorithm for Managing CF Request Conversion
FLASH10819	Couple Datasets: Best Practices and Avoiding Disasters
FLASH10576	CLOCKxx member considerations when in Local Timing Mode
FLASH10786	Where is My Coupling Facility?

- FLASH10631 Server Time Protocol (STP) - when the operating system doesn't support it!
- FLASH10887 Time of Day (TOD) clock Accuracy Monitor

IBM Service Offerings

For information on the offerings available from IBM, contact your IBM representative or see:

- **https://www.ibm.com/services**

Acknowledgments

We would like to thank the following people for all the help, advice, and contributions that made this document possible:

- Jay Aiken IBM Raleigh
- Riaz Ahmad IBM Washington Systems Center
- Kim Bailey IBM Raleigh
- Norton Carey IBM Raleigh
- Mary Crisman IBM Poughkeepsie
- Mac Devine IBM Raleigh
- Johnathan Harter IBM Raleigh
- Frank Kyne IBM ITSO Poughkeepsie
- Rich Lewis IBM Dallas
- Mary Petras IBM Santa Teresa
- Rachel Pickering IBM UK
- David Raften IBM Poughkeepsie
- Dale Reidy IBM Poughkeepsie
- David Surman IBM Poughkeepsie
- Juha Vainikainen IBM Finland
- Tom Wasik IBM Poughkeepsie
- Gail Whistance IBM Poughkeepsie
- David Yackel IBM Poughkeepsie
- Jeff Josten IBM Santa Teresa

Special Notices

This document is intended to provide customers with a checklist of items that can affect application availability in a Parallel Sysplex. The list is not all-inclusive, and there are other items that may be unique to each installation that have an impact on availability. However, this list was compiled based on IBM experience in investigating the cause of outages. The use of this checklist should help customers create an environment supportive of continuous application availability.

References in this publication to IBM products, programs or services do not imply that IBM intends to make these available in all countries in which IBM operates. Any reference to an IBM product, program, or service is not intended to state or imply that only IBM's product, program, or service may be used. Any functionally equivalent program that does not infringe any of IBM's intellectual property rights may be used instead of the IBM product, program or service.

Information in this book was developed in conjunction with use of the equipment specified, and is limited in application to those specific hardware and software products and levels.

Trademarks

IBM, the IBM logo, and ibm.com are trademarks or registered trademarks of International Business Machines Corporation, registered in many jurisdictions worldwide. Other product and service names might be trademarks of IBM or other companies. A current list of IBM trademarks is available on the web at “Copyright and trademark information” at http://www.ibm.com/legal/copytrade.shtml

The following terms are trademarks or registered trademarks of International Business Machines Corporation, and might also be trademarks or registered trademarks in other countries.

CICS®
CICSPlex®
DFSMS
DFSMSdss®
DS8000®
FICON®
FlashCopy®
GDPS®
HyperSwap®
IBM®
IBM Z®
IBM z15™
InfoSphere®
Interconnect®
MVS
NetView
OMEGAMON®
Parallel Sysplex®
RACF®
Redbooks®
RMF
Resource Link®
System Storage™
System z®
Tivoli®
VTAM®
WebSphere®
z/Architecture®
z/OS®
z/VM®
z/VSE®
z14
z15™

The following terms are trademarks of other companies:

ITIL is a Registered Trade Mark of AXELOS Limited.

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

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

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

Red Hat, are trademarks or registered trademarks of Red Hat, Inc. 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.

Other company, product, or service names may be trademarks or service marks of others.