High Availability Environments and Disaster Recovery
September 2004

Resiliency with Linux and zSeries

Table of Contents

Introduction

The implications of downtime can be considerable. Planned and unplanned system outages can negatively impact both customer loyalty and your business bottom line. In the end, you have to keep your system up and running with disaster recovery, repair, upgrade, redundancy and geographical dispersal programs.

In today's environment, customers are examining their options to cut costs and streamline their operations by simplifying their IT infrastructure. Simplification often includes the consolidation of multiple stand alone servers onto a single IBM @server® zSeries® server running Linux with z/VM®. While economy through consolidation was being aggressively sought, customers were not willing to compromise continuity of their business.

For years customers have struggled with the implications of not being able to respond in the event of a system outage. The simple desires to remain responsive and productive during an outage, and to maintain public confidence and customer loyalty has been expanded from within and from outside of the business. Pressures from within the business like corporate directives and audit requirements combined with external pressure from government regulation and insurance requirements have driven businesses to implement continuity strategies.

Given the combination of having to economize along with the challenge of being responsive during an outage, reliable solutions are most sought after in infrastructure simplification scenarios. zSeries servers running z/VM and Linux on zSeries can provide important autonomic capabilities and rich function to help provide customers with the tools to make their technology responsive during outages.

Linux for zSeries can inherit the hardware's reliability, but software faults can still cause outages. Thus a serious need exists to discuss and demonstrate how a highly resilient solution on Linux for zSeries running under z/VM can be implemented.

Before discussing the options available in a Linux on z/VM environment, it is important to provide some introduction to the subject of high availability and disaster recovery. A highly available system is a system that is designed to eliminate or minimize the loss of service due to either planned or unplanned outages. High availability doesn't equate to near continuous availability (that is, a system with nonstop service). You can achieve nearly continuous availability with, for example, the Parallel Sysplex® technology. The IBM @server zSeries operating system z/OS® exploits this technology to achieve nearly continuous availability. While z/VM and Linux cannot participate as active operating environments in a Sysplex, they can enjoy recovery benefits provided by a Parallel Sysplex operating in Geographically Dispersed mode.

Disaster recovery refers to how a system recovers from catastrophic site failures. Disaster recovery requires a replication of the entire site. The recovery time in case of a disaster is in the range of hours. The tasks necessary to recover from disaster differ from those needed to achieve a highly available system.

Resiliency: a key attribute of zSeries servers

It is important to note that disasters are not the only cause of downtime. After all, some companies must shut down their systems to make scheduled updates or perform maintenance. The built-in availability features on the IBM @server zSeries platform can help empower you to avoid both scheduled and unscheduled outages and aid in disaster recovery. How? By detecting potential problems at the earliest possible moment and taking the necessary actions to correct them, thereby helping to minimize the impact they might have on your applications.

The zSeries product line is designed to offer layer upon layer of *fault tolerance* and *error checking* features. If a failure occurs, the *built-in redundancy* on the zSeries platform is intended to shift the work over from failing components to ones that work to prevent the end-user service from being interrupted. The failed components may be removed and replaced while the processor is still active, so service may continue.

Geographically Dispersed Parallel Sysplex Technology

There is more to availability than just the server being up —the application and the data must be available as well. zSeries platforms' availability features include the hardware ,the operating system, application and database availability and the connection to disk. At the heart of zSeries platform availability is IBM Geographically Dispersed Parallel Sysplex™ (GDPS®) technology, which is positioned to provide a comprehensive business continuity solution for the z/OS platform. Based on geographical separation and automation, GDPS is a multi-site application availability solution designed to provide the capability to manage remote copy configuration and storage subsystem(s), automate Parallel Sysplex operational tasks and perform failure recovery from a single point of control. GDPS provides the resource sharing, workload balancing and near continuous availability benefits of a Parallel Sysplex environment. It can also significantly enhance the capability of an enterprise to recover from disasters and other failures and to manage planned exception conditions, helping businesses to achieve their own near continuous availability and disaster recovery goals.

Recent GDPS Enhancements

GDPS/PPRC has been enhanced to provide a new function called "GDPS/PPRC MULTIPLATFORM RESILIENCY FOR ZSERIES". This function is especially valuable for customers who share data and storage subsystems between z/OS and Linux on zSeries. For example, ERP applications like SAP utilizing an application server running on Linux on zSeries and an SAP database server running on z/OS. Since the solution utilizes a mult-tiered acrhitecture, there is a need to provide a coordinated Disaster Recovery solution for both the z/OS and zLinux based components of the solution. GDPS/PPRC can now provide that.

z/VM 5.1 provides a new HyperSwap function that enables the virtual device associated with one real disk to be swapped transparently to another disk. HyperSwap can be used to switch to secondary disk storage subsystems mirrored by Peer-to-Peer Remote Copy (PPRC). HyperSwap may also be helpful in data migration scenarios to allow applications to move to new disk volumes without requiring them to be quiesced.

GDPS/PPRC provides the reconfiguration capabilities for the Linux on zSeries servers and data in the same manner as for z/OS systems and data. In order to support planned and unplanned outages, GDPS provides the following recovery actions:

- *Re-IPL in place of failing operating system images*
- *Site takeover/failover of a complete production site*
- *Coordinated planned and unplanned HyperSwap of disk subsystems, transparent to the operating system images and applications using the disks*

Linux on zSeries environments running with z/VM that are running within zSeries servers participating in a GDPS environment may be recovered at a remote location as well as the z/OS based environments operating in the same GDPS environment.

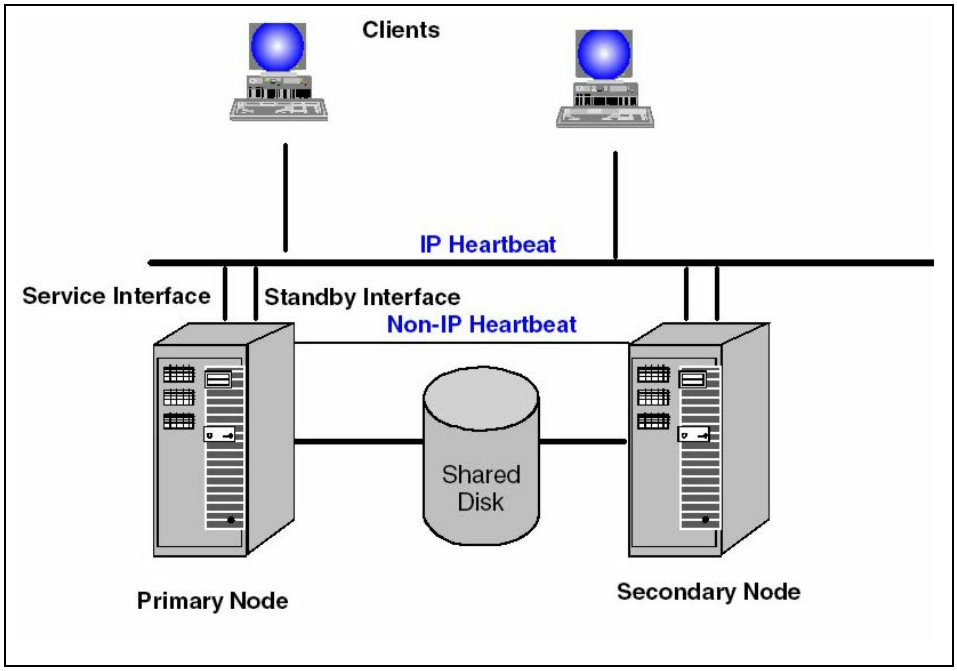
Highly Available Data

Data high availability means that data survive a system failure and are available to the system that takes over the failed system. Data high availability may not be equivalent to transaction safety. “Transaction safety” means that the application, with or without additional middleware products, is designed to provide the data integrity (for example, with two-phase commit) even if a failure occurs in the middle of a transaction. For the purposes of this discussion, data high availability means that the data is highly available once it is stored onto a disk.

Data high availability can be achieved by sharing the data across the systems. In the zSeries environment you can configure different channels to I/O subsystems, thus the devices are available to different Linux guest systems.

High Availability Example with z/VM and Linux on zSeries

To illustrate the abilities of a Linux-based solution running with z/VM on a zSeries server, let's consider a two-node high availability cluster to illustrate the basic technologies that make up the solution.



A typical high availability environment consists of:

- *Two or more Linux for zSeries systems in one z/VM, or in two Logical Partitions (LPARs) on the same zSeries processor complex, or on different zSeries servers*
  - *A set of shared disks for data availability, or a network file system*
  - *At least two network connections*
  - *A heartbeat and/or monitoring tool on each node*

Figure 1 summarizes a typical HA system. Each node has two network interfaces (service and standby) and in addition, a non-IP heartbeat connection. The active node is called the primary or service node; the other is the secondary or backup node.

A non-IP connection is usually used to check the health of the nodes. It is also possible to use the normal network connection for the heartbeat, but an additional non-IP connection is preferred because it does not rely on the IP stack, so a failure of the network adapter can be identified. Since the system can differentiate between a node and a network failure, more sophisticated error handling routines are possible and the negotiation between the two nodes

in case of a network failure can still happen. This is especially important for the proper release of shared resources (such as disks) before the takeover occurs.

The business-critical data reside on a shared disk. The zSeries has no internal disk, thus an external shared disk subsystem is always available.

If the active node fails, the heartbeat tool detects the failure and performs the IP, data, and application takeover. Both servers have a real IP address for each adapter. The application connects to the server with a so-called virtual IP address. The virtual IP address is an alias address defined on top of the real network interface. Both real IP addresses are visible from outside and are incorporated in the address resolution protocol (ARP) to provide the IP-to-MAC mapping. During the IP takeover the virtual IP address is transferred to the second node and an ARP message (gratuitous ARP) is sent to inform the other computers in the network of the new active node. The heartbeat tool has to provide scripts to handle the necessary actions such as start/release and stop the resources.

The situation where the two nodes each falsely believe the other to be dead is called a partitioned cluster. If both have shared write access to the disks, data integrity can be damaged. One way to help provide data integrity is called STONITH (Shoot The Other Node In The Head). STONITH achieves high data integrity by stopping one node, and therefore only one node has access to the data.

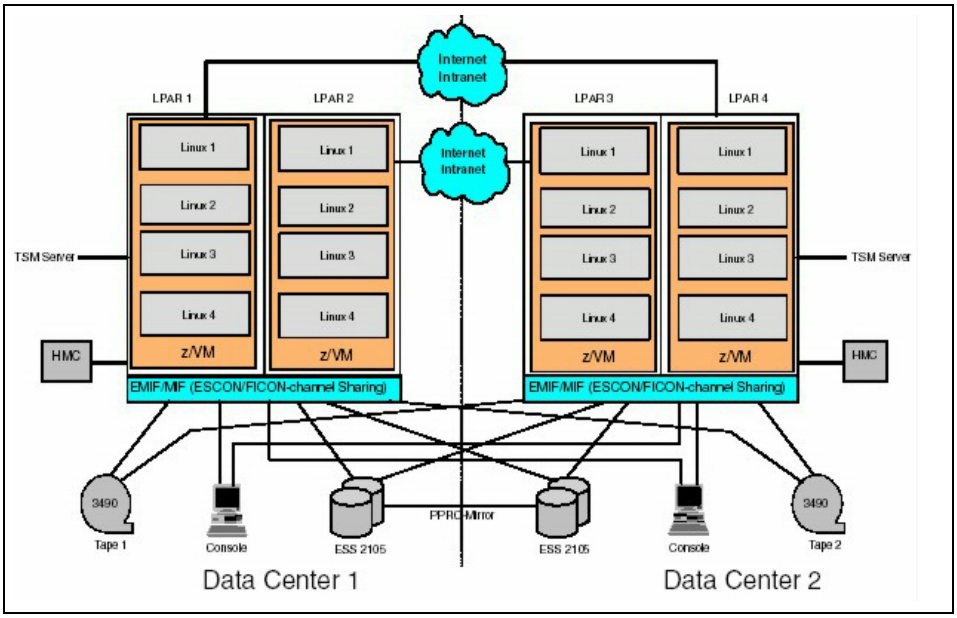
A Disaster Recovery Example with z/VM and Linux on zSeries

A common technique for disaster recovery is the "dual site concept," where two data centers reside in different locations. The entire hardware configuration is redundant, and the two systems are connected to each other.

Figure 2 shows a typical dual site environment with duplicated hardware (CPUs, DASD, and tapes). In this example two Logical Partitions (LPARs) with z/VM and Linux guest systems are shown on each site. Data mirroring is necessary to accomplish data recovery in case of an I/O subsystem failure. To enable a site takeover to take place, the sites must be connected to each other with separate connections.

The IBM ESS has implemented the Point-to-Point Remote Copy (PPRC) feature of the zSeries disk controller hardware. PPRC allows a remote disk unit to hold a synchronized copy of the data stored on a production disk subsystem. The synchronization of the data takes place at the time of the I/O operation; thus the remote copy is identical to the production data.

In case of a failure of the production disk subsystem, the backup disk subsystem can be brought online and the processing resumes at the point of failure. For a more detailed discussion on this topic see the redbook IBM TotalStorage Solutions for Disaster Recovery SG24-6547. PPRC may also be used to allow a site takeover by issuing PPRC commands. PPRC commands are not directly supported from the Linux shell, but users can implement ESS PPRC on z/VM systems. The control and management functions of PPRC are done with the device support program ICKDSF of z/VM, or with the IBM Total Storage® Enterprise Storage Server® Web interface, which can also be used in a Linux-only environment. However, ICKDSF contains only a subset of the commands and functions available in a z/OS environment. You must establish procedures to control recovery in the event of a disaster—for example, procedures to respond to failure scenarios.



Automated site takeover requires a thorough understanding of the whole environment, as well as the use of some advanced procedures. Implementing such procedures is no easy task, and is beyond the scope of this paper.

Linux on zSeries with z/VM with z/OS - Disaster Recovery

The design principles of a disaster recovery solution can be adapted to implement a z/VM high availability solution. Thus, the question arises whether the zSeries disaster recovery techniques can be used for a z/VM environment in conjunction with z/OS. Key technology for implementing a disaster recovery solution with zSeries and z/OS is the Geographically Dispersed Parallel Sysplex™ (GDPS®).

GDPS provides switching capability from one site to another site, for planned and unplanned outages. The latest version of GDPS is designed to manage zSeries images that execute externally to the Parallel Sysplex environment. This support offers only a restart of the LPAR in which the z/VM environment runs the PPRC management.

In a mixed z/VM and z/OS environment, failover by means of GDPS can be implemented. In this disaster recovery configuration the guest images on both sites, the primary and the secondary, access the data from their local storage subsystems, which are kept in sync by PPRC. This helps in case of a hardware or site failure, to allow the secondary site to take over the data and services. This makes it necessary that even in a disaster case where the primary DASD is not involved, a site switch of these volumes has to be performed.

In most cases the Linux guests in the disaster recovery case will take over all network resources (IP address, host names, routing table, and so forth) depending on the installed network infrastructure. In this situation the network configuration can be stored on minidisks that are copied from the primary to the secondary site using PPRC. If a different network setup is required for the recovery site, you have to create separate configuration files for each location, which are placed and maintained at both sites. Therefore, the Linux guests should be defined with the same device addresses for the primary and the secondary site.

Currently there is no interface between the GDPS and z/VM or the Linux guest systems under z/VM. But GDPS-controlled z/OS systems, which are running on the same Central Processor Complex (CPC) and/or are attached to the same storage subsystems as the z/VM with the Linux guests, can pass the failure conditions of the devices to the GDPS.

In case of a failure of a critical resource, like a storage subsystem, the failover procedure for the z/VM with the Linux guests is expected to be triggered and following actions are performed:

1. Reset the z/VM LPAR.
2. Break the PPRC DASD pairs.
3. Load the z/VM system on the secondary site, with automatic startup of all Linux guest systems.

These very critical actions are designed to be performed automatically, without any operator interaction.

Since GDPS is not aware of the operational state of the z/VM and Linux logical partition, restart and recovery scenarios should be put in place. For in-depth information on these topics see IBM Redbook “IBM zSeries and S/390 HIGH AVAILABILITY FOR z/VM and Linux”

Summary

The expanding scope of security threats and resultant continuing uncertainty serve to remind us just how critical it is for businesses to be prepared for disasters. Much work has been done by public and private interests to analyze the lessons learned and try to identity sound practices that can strengthen the resilience of the business infrastructure. The core lessons learned – a need for geographically dispersed facilities and resources, having an up-to-date business continuity plan and a highly automated disaster recovery solution – are not lost on IBM. The zSeries platform, along with related IBM products and service offerings, puts you in a position to take advantage of new technologies and evolve along with the demands of your business. And it lets you do so without having to sacrifice the flexibility you so desperately need in these uncertain and rapidly changing times. Choose IBM @server zSeries and rest easy with the knowledge that it is designed to help you protect your business.

For more information please see:

For more information on all IBM @server zSeries products, please visit the Web at **ibm.com**/servers/zSeries

For more information about high availability options for Linux and z/VM:

High Availability for z/VM and Linux for zSeries available at the following URL: www.redbooks.**ibm.com**/redpapers/pdfs/redp0221.pdf

For more information about business continuity for Linux and zSeries see: *Business Continuity Considerations and the IBM @server zSeries* - GM13-0256

IBM Corporation
Marketing Communications, Server Group
Route 100
Somers, NY 10589
U.S.A.

Produced in the United States of America
09/04