Draft Document for Review April 4, 2024 3:27 pm SG24-8544-00

Redbooks
ibm.com/redbooks

IBM LinuxONE Resiliency

Ewerson Palacio

Brian Hugenbruch

Wilhelm Mild

Filipe Miranda

Livio Sousa

Anderson Augusto Silveira

Bill White

IBM

Redbooks

IBM Redbooks

IBM LinuxONE Resiliency

January 2024

SG24-8544-00

Note: Before using this information and the product it supports, read the information in “Notices” on page 9.

First Edition (January 2024)

This edition applies to IBM LinuxONE Emperor, LinuxONE Rockhopper 4 and IBM z16.

This document was created or updated on January 30, 2024.

Contents

Notices

This information was developed for products and services offered in the US. This material might be available from IBM in other languages. However, you may be required to own a copy of the product or product version in that language in order to access it.

IBM may not offer the products, services, or features discussed in this document in other countries. Consult your local IBM representative for information on the products and services currently available in your area. Any reference to an IBM product, program, or service is not intended to state or imply that only that IBM product, program, or service may be used. Any functionally equivalent product, program, or service that does not infringe any IBM intellectual property right may be used instead. However, it is the user's responsibility to evaluate and verify the operation of any non-IBM product, program, or service.

IBM may have patents or pending patent applications covering subject matter described in this document. The furnishing of this document does not grant you any license to these patents. You can send license inquiries, in writing, to:
IBM Director of Licensing, IBM Corporation, North Castle Drive, MD-NC119, Armonk, NY 10504-1785, US

INTERNATIONAL BUSINESS MACHINES CORPORATION PROVIDES THIS PUBLICATION "AS IS" WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESS OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF NON-INFRINGEMENT, MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE. Some jurisdictions do not allow disclaimer of express or implied warranties in certain transactions, therefore, this statement may not apply to you.

This information could include technical inaccuracies or typographical errors. Changes are periodically made to the information herein; these changes will be incorporated in new editions of the publication. IBM may make improvements and/or changes in the product(s) and/or the program(s) described in this publication at any time without notice.

Any references in this information to non-IBM websites are provided for convenience only and do not in any manner serve as an endorsement of those websites. The materials at those websites are not part of the materials for this IBM product and use of those websites is at your own risk.

IBM may use or distribute any of the information you provide in any way it believes appropriate without incurring any obligation to you.

The performance data and client examples cited are presented for illustrative purposes only. Actual performance results may vary depending on specific configurations and operating conditions.

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

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

This information contains examples of data and reports used in daily business operations. To illustrate them as completely as possible, the examples include the names of individuals, companies, brands, and products. All of these names are fictitious and any similarity to actual people or business enterprises is entirely coincidental.

COPYRIGHT LICENSE:

This information contains sample application programs in source language, which illustrate programming techniques on various operating platforms. You may copy, modify, and distribute these sample programs in any form without payment to IBM, for the purposes of developing, using, marketing or distributing application programs conforming to the application programming interface for the operating platform for which the sample programs are written. These examples have not been thoroughly tested under all conditions. IBM, therefore, cannot guarantee or imply reliability, serviceability, or function of these programs. The sample programs are provided "AS IS", without warranty of any kind. IBM shall not be liable for any damages arising out of your use of the sample programs.

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 https://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.

Db2®
DS8000®
FICON®
FlashCopy®
GDPS®
Guardium®
HyperSwap®
IBM®
IBM Cloud®
IBM Spectrum®
IBM Z®
IBM z16™
Interconnect®
OMEGAMON®
Parallel Sysplex®
PIN®
Redbooks®
Redbooks (logo) ®
Resilient®
Resource Link®
System z®
Tivoli®
WebSphere®
z Systems®
z/Architecture®
z/OS®
z/VM®
z13®
z15®
z16™
zEnterprise®
zSystems™

The following terms are trademarks of other companies:

Intel, Intel logo, Intel Inside logo, and Intel Centrino logo are trademarks or registered trademarks of Intel Corporation or its subsidiaries in the United States and other countries.

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.

Ansible, OpenShift, Red Hat, are trademarks or registered trademarks of Red Hat, Inc. or its subsidiaries in the United States and other countries.

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

Preface

This IBM® Redbooks® publication explores the concepts of resiliency as they relate to information technology systems. Focus will be given to a LinuxONE server as a foundation model; however, understanding why we care about uptime beyond vague notions that "time equals money" will help readers to understand the importance of availability for a business, institution, or governmental agency Authors.

This book was produced by a team of specialists from around the world working at IBM Redbooks, Poughkeepsie Center.

Ewerson Palacio is an IBM Redbooks Project Leader. He holds a Bachelors's degree in Math and Computer Science. Ewerson worked for IBM Brazil for over 40 years and retired in 2017 as an IBM Distinguished Engineer. Ewerson co-authored a large number of IBM Z® publications, created and presented at IBM Redbooks seminars around the globe.

Brian Hugenbruch CISSP, is a Senior Software Engineer with IBM Z and LinuxONE. Officially, he is the z/VM® Cryptography and Security Development lead. He has also served as the LinuxONE Resiliency Technical Lead since 2020; in this capacity, he drove the platform to its "Eight 9's" calculation. He combines these two sides of his work on the topic of Cyber Resiliency and Digital Forensics. He writes and speaks on all these topics in numerous forums and conferences, with the goal of making complicated topics as easy to understand as possible.

Wilhelm Mild is an IBM Executive IT Architect and Open Group Distinguished Architect which is passionate since more than 3 decades for new technologies and their adoption to enterprise solutions. He is architecting IBM LinuxONE and IBM zSystems™ complex solution Architectures featuring, Containerization and Red Hat OpenShift Container Platform, end-to-end security, Resiliency HA / DR and Network topologies for complex application landscape in worldwide customer engagements. He is a speaker on international customer events and education classes and is active in teaching hands-on labs and enjoys Nature in free time with hiking and traveling to funny natural places

Filipe Miranda is a principal technical specialist for Red Hat Synergy team/IBM. 15+ years of experience using Open Source technologies. Author of several IBM Redbooks and technical articles spread out on LinkedIn and IBM Developer, he is member of the zAcceleration team, where key skills on the toolbox includes, IBM CloudPaks, Red Hat OpenShift, DevOps and many other technologies part of the hybrid cloud portfolio.

Livio Sousa is an IT Specialist with over 20 years of experience with high-end platforms, which encompasses servers, storage, networking equipment and enterprise architecture integration. He is assigned to support hybrid cloud solutions. Throughout his career he had the opportunity to work with several different software platforms, such as z/OS®, Unix, Linux and Windows as well as different hardware processors architectures like the z/Architecture®, Power and the IA-32 architecture and its extensions.

Anderson Augusto Silveira is a Senior Technical Specialist for LinuxONE, Linux on Z and Red Hat OpenShift within IBM. With 12+ years of experience in Virtualization and Open Source technologies he has been participating in mission critical projects around the world, training teams of IBM Business Partners and Clients, and helping and supporting with adoption of new technologies and products.

Bill White i is a Project Leader and Senior Infrastructure Specialist at IBM Redbooks, Poughkeepsie Center.

Thanks to the following people for their contributions to this project:

Steven Cook
IBM US

Justin VanSlocum
IBM US

Now you can become a published author, too!

Here's an opportunity to spotlight your skills, grow your career, and become a published author—all at the same time! Join an IBM Redbooks residency project and help write a book in your area of expertise, while honing your experience using leading-edge technologies. Your efforts will help to increase product acceptance and customer satisfaction, as you expand your network of technical contacts and relationships. Residencies run from two to six weeks in length, and you can participate either in person or as a remote resident working from your home base.

Find out more about the residency program, browse the residency index, and apply online at:

ibm.com/redbooks/residencies.html

Comments welcome

Your comments are important to us!

We want our books to be as helpful as possible. Send us your comments about this book or other IBM Redbooks publications in one of the following ways:

- Use the online **Contact us** review Redbooks form found at:

  ibm.com/redbooks

- Send your comments in an email to:

  redbooks@us.ibm.com

- Mail your comments to:

  IBM Corporation, IBM Redbooks
  Dept. HYTD Mail Station P099
  2455 South Road
  Poughkeepsie, NY 12601-5400

Stay connected to IBM Redbooks

- Find us on LinkedIn:

  https://www.linkedin.com/groups/2130806

- Explore new Redbooks publications, residencies, and workshops with the IBM Redbooks weekly newsletter:

https://www.redbooks.ibm.com/subscribe

- Stay current on recent Redbooks publications with RSS Feeds:

  https://www.redbooks.ibm.com/rss.html

1

IT resiliency matters

IT resiliency is not just ensuring systems are up and running, but rather the entire IT infrastructure is consistently available and that services and data are always accessible. In addition, many organizations operate in highly regulated industries and must comply with strict IT resiliency standards, while service providers offer certain service levels pertaining to quality, availability, and responsiveness that they are obligated to meet.

In this chapter we explore the concepts of resiliency as they relate to IT systems. Focus is given to the IBM LinuxONE platform as a foundation model, however, discussions on why we care about uptime beyond vague notions that “time is money”, and the importance of high availability, business continuity, and disaster recovery is also provided.

This chapter includes the following topics:

1.1 The impact of downtime

There is no room for downtime in today's always-on digital world. Consumers of any service - from someone using an app on their phone, to application developers or business partners relying on infrastructure, expect services to be available. Services must be available without difficulty or disruption.

There is a cost when those expectations are not met, and that cost can be significant: legal consequences, lost revenue, damaged reputation, stalled productivity, or failure to meet regulatory and compliance standards, to name a few.

The impact of downtime can vary greatly depending on the business. Some might be able to survive an outage of a few hours, while for others, even consumers experiencing poor performance for several seconds can impact the bottom line. Not being able to deliver services as expected or at the agreed level can potentially turn into loss of business or hefty fines.

IT resiliency must protect your business from downtime. IT infrastructure solutions must work together to optimize availability, keep your systems running, detect problems in advance and recover your critical data.

1.1.1 Outages do happen

Sometimes bad things happen to good systems. Outages are the result of planned and unplanned events.

A *planned outage* is an anticipated and controlled disruption of a service or services. It is a scheduled event where consumers are notified in advance with a description and scope of the outage with a specific date, time, and duration. Reasons for a planned outage can include testing or implementing new hardware or software components, maintenance to apply fixes, or configuration changes that require a system reboot. This can involve a partial or a full site outage.

On the other hand, an *unplanned outage* is an unforeseen disruption to a service or services, when one or more hardware or software components of the service fails. Reasons for an unplanned outage can be caused by a natural disaster, cyberattacks, infrastructure failure, or human error[1].

The best approach to remedying any type of outage is to eliminate single points of failure by creating an environment that has redundant resources with an automated and seamless takeover process. Without such measures, you must first identify and fix the problem, and then manually restart the affected hardware and software components.

Even planned outages for businesses with 24x7 service offerings can come at a significant cost. Businesses need to be ready and able to recover as quickly and seamlessly as possible from any type of event to minimize cost.

Refer to The Real Costs of Planned And Unplanned Downtime, for the cost of downtime.

[1] The global average total cost of a data breach is $4.45 million per the Cost of a data breach 2023: https://www.ibm.com/reports/data-breach

1.2 IT resiliency is a journey

IBM defines IT resiliency as the ability to rapidly adapt and respond to any internal or external opportunity, demand, disruption, or threat, and continue business operations without significant impact. This includes continuous and near-continuous application availability, keeping your applications up and running throughout the far more common planned and unplanned outages, and disaster recovery, which concentrates on recovering from an unplanned event.

1.2.1 Reducing risk and impact on service delivery

Resiliency refers to a component's ability to deliver the intended outcome despite any type of failure or abnormal conditions. It is not just ensuring that the hardware is up and running. Instead, system resiliency applies to the entire stack (compute, operating system, workload, network, and storage layers). It should consistently be available to make certain that services and data are accessible. In addition, transactions ought to be completed successfully and on time, with good performance.

There can also be a requirement to adhere to regulatory guidelines and standard rules for the entire IT infrastructure that also impact system resiliency, examples include:

- NIST SP 800-160 and NIST SP 800-193
- PCI DSS 9,5,1
- US White House Executive Order 14028, “Improving the Nation's Cybersecurity”
- European Union Digital Operational Resilience Act (DORA)[2]
- ISO 27001(Information Security Management)

Because of its checklist-oriented nature, many businesses and organizations use the Payment Card Industry Data Security Standard (PCI DSS) as a starting point for building system resilience. They might then apply stronger standards when either needed for the client experience, or when required by law.

Regulations and standards that apply to various industries are frequently updated to reflect what is happening in the cyber resiliency aspect of their industries. For multinational businesses, laws, policies, and regulations might differ from one geographic location to the next, causing additional stress on the requirements for IT infrastructure resiliency.

In the end, the design of systems supporting critical IT services depends on the interaction between the criticality of the service and its business profile—regarding it as a journey toward reducing risk and impact on service delivery. Consequently, it becomes important to address downtime from a business viewpoint that is directed by service level agreements.

1.2.2 Service level agreements drive resiliency requirements

For any service provider, service level agreements (SLAs) will determine how their IT infrastructure will be designed, configured, and managed to provide the required level of resiliency. This is true whether you are discussing a service as wide-ranging as cloud-hosted platforms, or as small as an application shared within your IT department.

Service-level agreements (SLAs) are used between a service provider and consumers to describe the service and define the level and quality that can be expected. It includes the duties and responsibilities of each party, the *metrics* for measuring service quality, and the

[2] Refer to The Digital Operational Resilience Act (DORA) - Regulation (EU) 2022/2554

course of action or penalties should the agreed upon service levels not be met. SLAs also specify terms and conditions, like:

- Effective start dates, end dates, and review dates.
- Key performance indicators that track performance over time, for example:
  - Service *ABC* must be able to run at least 10,000 transactions/second with 95% finishing under 0.1-seconds response time as measured by application records.
  - Service *ABC* must be available 24/7, except for a 1-hour planned outage, once per quarter. Warning of this planned outage must be given no later than 2 weeks before the outage. Availability level is based on the value that is indicated in the outage record and agreed to by all parties.
- Commitments that are related to the type of record that the SLA applies to. For example, problem tickets will specify target dates and times for response, resolution, delivery, availability, and other values. Work orders will specify target start, finish, and delivery times.
- Other criteria that may be measured include defect rates, technical quality, security controls, and compliance standards.

1.2.3 SLA criteria will dictate requirements for recovery

Resiliency requirements should reflect SLAs and their recovery objectives, which in turn will drive the technology needed to satisfy them. It should not be the technology driving the resiliency requirements.

Recovery events are typically measured by the targets that are defined for Recovery Point Objective (RPO) and Recovery Time Objective (RTO) targets.

Looking backward from an event, the RPO represents how much data is lost due to a failure. This value can vary from zero or near-zero with synchronous remote copy, to hours or days with recovery from tape. Looking forward from the event, the RTO is the goal for how much time can pass before service is restored. It includes bringing up the operating system on the recovery system, the workloads, management tools, enabling the network, and bringing the services online. This time span can be anywhere from minutes to days, depending on the recovery solution.

It seems obvious to say that you must choose the replication techniques that achieve the recovery objectives. However, business realities often complicate the choices, for example:

- An existing building is owned at a remote location so it must be used, even though it is too far away to support synchronous replication to get RPO=0.
- Tape-tape replication is chosen because it is cheaper, even though the amount of potential data loss might impact business results.
- The SLA, RPO, and RTO have been defined when the environment was different and are now outdated. Regular reviews and updates of existing SLAs, RPO, and RTO are needed to capture changes to the business environment.

Also, in case of problems with the recovery process there will be dependencies on the infrastructure support staff. Automation can help in greatly reducing the time that is needed to recover the environment by optimizing the recovery process.

1.3 IT resiliency goes beyond disaster recovery

IT resiliency is related to, but broader in scope, than disaster recovery. Disaster recovery concentrates solely on recovering from an unplanned event.

IT resiliency is the ability to ensure continuity of service and support for its consumers and to maintain its viability before, after, and during an event. This means that a business must make use of people, process, and technology:

- People who understand their roles-and-responsibilities during an outage or disruption. They must design the strategies and solutions to enable the continuity of business operations. Operators must practice *recovery procedures* to ensure a smooth recovery after an outage or disruption.
- Processes include crisis management and business continuity, but can be expanded to include problem, change-and-configuration management, and business controls.
- Technologies are often emphasized too much. However, they include backup, replication, mirroring, and failover solutions.

1.3.1 IT resiliency challenges

The challenges when achieving IT resiliency include:

- Evaluating, planning, and mitigating the business impact of various types of risks.
- Responding to regulatory pressures and compliance associated with business resiliency.
- Ensuring data is protected, available, and accessible as needed by the business.
- Achieving availability objectives by reducing frequency and duration of infrastructure applications and data outages.
- Recovering from and responding to disruptive events.

Several terms are used in for different levels or types of availability, they include high availability, continuous operations, continuous availability, business continuity, and disaster recovery. These are describe in the subsequent sections.

High availability

A high availability (HA) environment has a fault-tolerant, failure-resistant infrastructure that supports continuous workload processing. The goal is to provide service during defined periods, at acceptable or agreed upon levels, and mask *unplanned outages* from consumers. Many hardware and software components have built-in redundancy and recovery routines to mask possible events. The extent of this support greatly varies.

HA refers to systems and services that typically operate 24x7 and usually require redundancy and rapid failover capabilities. These capabilities minimize the risk of downtime due to failures in the IT infrastructure.

HA is distinguished from disaster recovery (DR) primarily through the duration and scope of outage that the technology is designed to mitigate.

Continuous operations

A continuous operations (CO) environment enables non-disruptive backups and system maintenance that is coupled with continuous availability of services. The goal of business continuity is to mitigate or mask the effect of *planned outages* from consumers. It employs non-disruptive hardware and software changes, non-disruptive configuration, and software coexistence. z/VM Live Guest Relocation, for example, enables CO because workloads can be moved from one system to another system at will.

Continuous availability

Continuous availability (CA) is a combination of CO and HA. It is designed to deliver non-disruptive service to consumers seven days a week, 24 hours a day. There is no impact from planned or unplanned outages. Red Hat Openshift Container Platform (OCP) is an example of a CA solution. It enables HA by removing single points of failure (SPOFs) within the hardware and its software environment. It enables CO by allowing “rolling system reboots”, non-disruptive updates and workload relocation to mask planned events.

Disaster recovery

Disaster recovery (DR) is protection against unplanned outages through a reliable, predictable recovery process. Examples include inability to access storage or a site disaster.

DR is the ability to respond to an interruption in services by implementing a disaster recovery plan to restore critical business functions. DR is more reactive than HA solutions. DR solutions typically enable recovery within 24 to 72 hours, while HA solutions enable recovery within one to two hours of an event. As with any event, the duration and scope of the event will dictate when use of the DR plan is necessary. A DR process might be an appropriate response, even with events that will last less than the 24-to-72-hour standard.

Business continuity

Business continuity is the capability of a business to withstand outages and to operate important services normally and without interruption in accordance with service-level agreements.

Business continuity includes disaster recovery (DR) and high availability (HA) and can be defined as the ability to withstand all outages (planned, unplanned, and disasters) and to provide continuous processing for all critical workloads. The goal is for the outage time to be less than 0.001% of total service time. A high availability environment typically includes more demanding recovery time objectives (seconds to minutes) and more demanding recovery point objectives (zero user disruption) than a disaster recovery scenario.

High availability solutions provide fully automated failover to a backup system so that users and applications can continue working without disruption. HA solutions must have the ability to provide an immediate recovery point. At the same time, they must provide a recovery time capability that is significantly better than the recovery time that you experience in a non-HA solution topology.

1.4 How to prepare for IT resiliency

It can seem like a daunting challenge to create an always-on service in the face of those challenges. And there is no one technology that solves all problems. IT resiliency is a choice. So, what should be considered when tackling this problem?

First, consider points of failure. Any component of your system which does not have a backup can lead either to reduced throughput, or reduced processing power, or a service outage.

Basic assumptions for building a resilient IT system:

- Build the approach, test the approach. Include the following:
  - Redundancy
  - Early detection
  - Automation and failover
  - Test, practice, improve — it is not a plan, if you have never tried it. It is just an idea.
- Resiliency best practices (generally speaking)
- How do these things address the concerns?
  - Financial — cost-benefit analysis of downtime versus technical investment
  - Regulatory — cost of failing to comply

To solve these challenges, invest in technologies that help reduce all downtime.

1.4.1 Balancing risk and costs of mitigation

Higher availability comes at a higher cost. The cost will vary depending on which services are affected. Calculating it can be done by looking at:

- Number of consumers affected by the outage
- Duration of the outage, from the consumers' perspective
- Average number of transactions executed per hour, per consumer
- Estimated revenue per transaction

While such calculations are rough estimates, they can provide an investment baseline on the risk of investing in resiliency technologies or trusting to luck or fate.

Such calculations are done using relative percentiles of availability, versus the cost of an outage. Having a 99% available system means that, in a given year, one might anticipate 3.65 days of downtime. If one measures the cost of downtime by the hour, this can still lead to an expensive cost, even if the outage is planned.

In comparison, a system with seven 9's of availability (99.99999% uptime) is, in a given year, unavailable for approximately 3.16 seconds, and the cost of this outage is significantly lower (see Table 1-1).

Table 1-1 Cost of outages in US$

	99%	99.9%	99.95%	99.99%	99.99999%
$150,000+ / hour (98% respondents)	$8,766,000	$876,600	$438,300	$87,660	$88
$300,000+ / hour (88% respondents)	$26,000,000	$2,640,000	$1,315,000	$262,300	$263
$1,000,000+ / hour (40% respondents)	$87,660,000	$8,766,000	$4,383,000	$876,600	$877

It is worth noting that outages are never precisely as short or as long as the table above, and these estimates are based on a mean-time between failures - how likely, in instances per year or per decade, they're likely to occur.

Balancing risk and costs of mitigation is essential to achieving business goals. By balancing risk of a type of failure and cost of a solution, you can determine how much redundancy and availability management you can design into your IT infrastructure. By understanding the real cost of service unavailability, you can calculate the costs that you can sustain to provide this availability. These factors also determine when you can do maintenance and upgrades to the IT infrastructure.

As illustrated in Figure 1-1, you must evaluate the risk and impact of the outage and decide whether the design needs to be changed to meet the cost implications of service outages. Or maybe these cost implications should be adjusted?

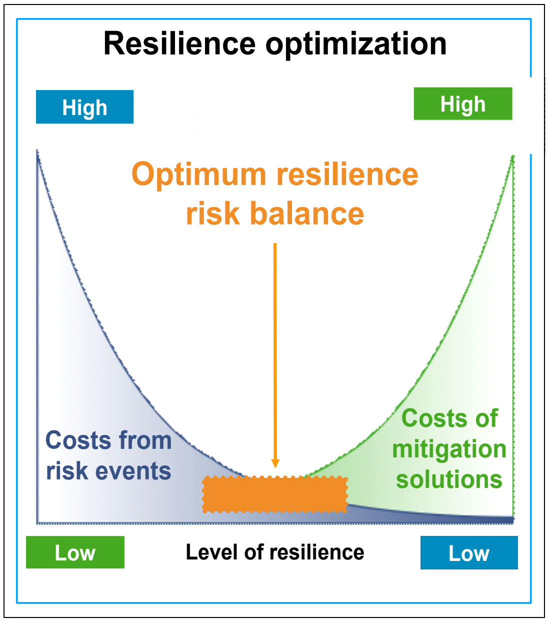

Figure 1-1 Resiliency Optimization

Under-investment can result in “spending money by not-spending money,” increasing risks to the business. Should an outage occur, you might be able to bear the cost of recovery at the time of the outage, but the cost of recovery can be greater than the cost of prevention. Over-investment can result in excess solution costs. By overspending on unnecessarily high levels of protection, you can incur expenses and risks associated with lost opportunity.

1.4.2 How does IBM LinuxONE help?

IBM LinuxONE has a sterling record of system uptime. This means that the relative availability of the underlying server hardware, through a host of technical means discussed in Chapter 2, “Technical View on LinuxONE Resiliency” on page 29, can give a boost to the uptime of applications just by virtue of the system on which it is running.

This is not to say that one server is enough, or that LinuxONE hardware solves all problems. Again, system resiliency is a lifestyle choice, and this means that every single point of failure—in storage, compute, networking, virtual infrastructure, physical site maintenance, and so forth—needs to be addressed as part of an overall resiliency and DR plan.

1.5 Deployment models for IBM LinuxONE resiliency

The goal of a deployment model is to provide a degree of horizontal scaling—if one copy of the workload fails, is there another to take its place.

Deployment models, for the purposes of this book, is a framework for planning and implementing resiliency in a IBM LinuxONE environment. The focus of such models is less about workload configuration and more about the level of resource needed in a particular part of your IT environment.

The IBM LinuxONE deployment models are discussed briefly in this section and described in Chapter 3, “Deployment Scenarios for Resiliency” on page 75 using different use case scenarios with architectural patterns.

An architectural pattern is a specific selection of hardware and software meant to illustrate in detail just how the various parts of an infrastructure can work together to anticipate unplanned outages, mitigate planned outages, or assist in times of disaster.

The goal of an architectural pattern is to provide vertical scaling for resiliency - in a given stack, how can the parts help offset one another? Architectural patterns are constructed of the components discussed in Chapter 2, “Technical View on LinuxONE Resiliency” on page 29, and an illustration of both a deployment model and architectural pattern is provided in Chapter 4, “Resiliency for Red Hat Openshift Container Platform” on page 95.

1.5.1 IBM LinuxONE deployment models

To assist with improving the resiliency of an IBM LinuxONE environment, we have developed a series of deployment models for resiliency. Those models map to different levels of resiliency based on common IBM LinuxONE configurations and specific business requirements. The models can guide you through assessment of your current resiliency investment and selection of new targets for workloads. The models can then help to identify practical steps to achieve availability goals for your IT infrastructure. In addition, the models suggest disaster recovery considerations between data centers as part of your assessment.

Table 1-2 provides an overview of the four IBM LinuxONE resiliency deployment models

Table 1-2 Deployment models for IBM LinuxONE resiliency

Model #	Model name	Risk/impact to business	Time to recovery	Typical RTO*
1	Reliable base	High	8+ hours	3+ day
2	Flexible site	Medium	2+ hours	8 hours
3	Multi-site resiliency	Low	Minutes	< 1 hour
4	Continuous availability	Minimum	Seconds	< 120 seconds
* Recovery Time Objective (RTO) is the target time for system restoration, while Recovery Point Objective (RPO) is the data loss that is acceptable. (RPO could be zero data loss.)				

Each of the four models supports different capabilities of the hardware and software technologies that are represented by the layers that are listed in Table 1-3 on page 24. The table also emphasizes that the layers relate to each of the four resiliency deployment models.

Table 1-3 Deployment Models Layers

Layer	Components examples of each layer
Infrastructure	Compute (CPUs, Memory, and Firmware), Storage, Network
Operating environments	Virtualization (IBM z/VM, KVM, IBM PR/SM, DPM), Linux, Container Platforms
Applications	Containerized Workloads, Middleware, ISV Applications, Databases
Management and Automation	► Hypervisors' Management ► Container Management ► Data Replication
All models in table 1-2 do have a relationship to aspects of all the layers.	

All deployment models in the Table 1-2 on page 23 do have a relationship with the aspects of all layers shown above in Table 1-3.

In the relationship of the layers to the four resiliency deployment models, the question is this: *What are the essential features of the layers for each of the four models?* This will be discussed in Chapter 2, "Technical View on LinuxONE Resiliency" on page 29.

Here are some example issues regarding layers in relation to the four models:

- Model 1 to Model 2: In the transition from resiliency model 1 to 2, the hardware layer of typically grows from a single system to a multi-system environment.
- Reference architectures: Various IBM solutions can be a part of your implementation of a model. Each solution has a 'reference architecture,' which can include hardware components and software.

For descriptions of the capabilities of each layer, see Chapter 2, "Technical View on LinuxONE Resiliency" on page 29.

The selection of a deployment model should be based on a combination of risk and cost. If a smaller business, for example, only has one data center, then this business is constrained by the limits of physical site construction as to the number of threats that can be defeated.

Deployment model 1: Reliable base

This deployment model is designed as the "starter" for resiliency, with a focus on vertical scaling. It contains one physical site, one IBM LinuxONE system.

It would be best never to have failures. You can get closer to this by starting with quality base technology together with regular maintenance. Application code that you write on top of that base should have extensive problem detection and near-instantaneous correction. You take these steps with IBM LinuxONE hardware and the operating system.

By design, the IBM LinuxONE platform is a highly resilient system. Its architecture has built-in self-detection, error correction, and redundancy. The IBM LinuxONE reduces single points of failure, to deliver the best reliability of any enterprise system in the industry. Transparent processor sparing, and dynamic memory sparing enable concurrent maintenance and seamless scaling without downtime. Workloads that fail can be restarted in place.

If the operating system or logical partition (LPAR)[3] is down for a planned or unplanned outage, another instance of the workload (in another LPAR) can absorb the work. An automatic recovery process can reduce the workload impact by avoiding the delays associated with a manual intervention.

Resiliency for a single IBM LinuxONE platform can be enhanced by using a clustering technology with data sharing across two or more Linux images. Although this does not provide for additional hardware redundancy, it can keep workloads available and accessible across software upgrades by allowing rolling restarts.

For more information on this model, see “ 3.1, “Reliable base” on page 76.

Deployment model 2: Flexible site

This deployment model expands upon the reliable base model by adding redundancy for compute, storage, and network within a single site. This allows for greater control of outages.

Despite the best base technology, failures can still occur. You should anticipate failures with the right technology implementation. The Flexible site model with the ability to restart workloads quickly on a backup component, can significantly reduce the impact of an outage.

With the installation of a second IBM LinuxONE system in the IT infrastructure, your component resource sharing begins the road to full system redundancy. Additionally, single points of failure for data can be mitigated by setting up Peer-to-Peer Remote Copy (PPRC), “Metro” (synchronous) data mirror if the systems are installed separated by a metro distance, (up to 300 km[4]), and “Global” (asynchronous) data mirroring for longer distances. IBM Copy Services Manager (CSM) can enable fast data replication with little or no data loss. For more information see 2.8.8, “Infrastructure placement” on page 71

For more information on this deployment model, see 3.2, “Flexible site” on page 77

Deployment model 3: Multi-site resiliency

This deployment model sometimes met without having gone through deployment model 2, introduces one IBM LinuxONE system and associated storage into a second data center, removing the limitation of physical site recovery from single points of failure. This step is necessary on the journey to higher availability and disaster recovery.

Even the best fault tolerant designs can fail. When failure happens and service is impacted, be prepared to restore service quickly. One method to reduce the impact of an outage is to avoid outages whenever possible. Anticipate failures with the right technology implementation.

Most hardware and software components in the IBM LinuxONE system can be cloned to a second IBM LinuxONE system with dynamic workload balancing spreading the work. For planned or unplanned events on one LPAR, the workload flows to another LPAR seamlessly, without affecting availability.

By adding GDPS® capabilities to the disk replication, you can automate management of system and site recovery actions. In failover situations, automation can reduce the business impact by minutes or even hours.

For more information on this deployment model, see 3.3, “Multi-site resiliency” on page 82.

[3] A logical partition (LPAR) is a subset of the IBM LinuxONE hardware that is defined to support an operating system. An LPAR contains resources (such as processing units, memory, and input/output interfaces) and operates as an independent system. Multiple LPARs can exist in an IBM LinuxONE platform. See 2.8.3, “Automation with Linux HA components” on page 63 for more information.

[4] IBM Storage DS8870 supports up to 300km (186 mi) with Metro Mirror - Delays in response times are proportional to the distance between volumes.

With multiple systems, multiple storage units, automation software, at multiple data centers, every possible risk can be addressed before it happens. The distance at which physical sites are set also becomes a defining factor, as two data centers located at a metro proximity may fall subject to the same disasters.

Each step of the journey does involve time and money in order to procure, install, configure, and test each of the components involved. And testing is a requirement - investing in all of the above, without validating that it works despite the pressures of on-going business, could lead to unwanted surprises in the event of an actual disaster.

The last of the four IBM Z resiliency deployment models is as follows:
Fault Tolerant within a data center plus GDPS Continuous Availability

When you have Fault Tolerance within the Primary Data Center, plus GDPS xDR, Metro and Global Mirror, plus Continuous Availability for key workloads, Disaster Recovery looks like this architecturally:

- GDPS Global Mirror consists of two sites, which are separated by virtually unlimited distances.
- The sites run the same applications with the same data sources, to provide cross-site workload balancing, continuous availability, and Disaster Recovery (DR).
- As a result, workloads to fail over to another site for planned or unplanned workload outages within seconds.
- GDPS allows recovery time in 1 - 2 minutes with 3 - 5 seconds of data loss, also with full end-to-end automation.

You can combine Metro-Mirror and Global-Mirror solutions into 3- and 4-site solutions to provide IBM HyperSwap® for very rapid recovery for disk outages. You also enable zero data loss across long distances, without affecting user response time.

Deployment model 4: Continuous Availability (CA)

The linuxONE continuous availability deployment model requires the implementation of strategies and technologies that ensure uninterrupted and reliable operations on IBM's LinuxONE systems.

GDPS for Continuous Availability:

GDPS is the technology used to provide continuous availability and disaster recovery for IBM Z, including linuxONE. It orchestrates workload routing and facilitates failover in the event of disruptions, ensuring uninterrupted operations.

In order to implement LinuxONE Continuous Availability, z/OS[5] partitions running on an IBM Z in a GDPS environment are required. GDPS xDR function extends the near-continuous availability and disaster recovery capabilities provided by GDPS Metro to other platforms, or operating systems, running on LinuxONE servers.

GDPS Metro Multiplatform Resiliency for IBM Z (xDR)

GDPS Metro provides a function called “GDPS Metro Multiplatform Resiliency for IBM Z,” also referred to as cross-platform disaster recovery, or xDR. This function is especially valuable for customers who share data and storage subsystems between z/OS and Linux z/VM guests on IBM Z. For example, an application server running on Linux on IBM Z and a database server running on z/OS.

[5] LinuxONE does not support/run z/OS.

With a multi-tiered architecture, there is a need to provide a coordinated near Continuous Availability/Disaster Recovery solution for both z/OS and Linux. GDPS Metro can provide this capability when Linux is running as a z/VM guest or native on LinuxONE or IBM Z. Using the HyperSwap function so that the virtual device associated with one real disk can be swapped transparently to another disk, HyperSwap can be used to switch to secondary disk storage subsystems mirrored Metro Mirror. If there is a hard failure of a storage device, GDPS coordinates the HyperSwap with z/OS for continuous availability spanning the multi-tiered application.

For site failures, GDPS invokes the Freeze function for data consistency and rapid application restart, without the need for data recovery. HyperSwap can also be helpful in data migration scenarios to allow applications to migrate to new disk volumes without requiring them to be quiesced.

When using ECKD formatted disk, GDPS Metro can provide the reconfiguration capabilities for the Linux servers and data in the same manner as for z/OS systems and data. To support planned and unplanned outages these functions have been extended to KVM on LinuxONE and IBM Z with GDPS V4.1. GDPS provides the recovery actions listed below:

- Re-IPL in place of failing operating system images
- z/VM Live Guest Relocation management
- Manage z/VM LPARs and z/VM guests, including Linux on Z
- Heartbeat checking of Linux guests
- Disk error detection
- Data consistency with freeze functions across z/OS and Linux
- Site takeover/failover of a complete production site
- Single point of control to manage disk mirroring configurations
- Coordinated recovery for planned and unplanned events

Additional support is available for Linux running as a guest under z/VM. This includes:

- Re-IPL in place of failing operating system images
- Ordered Linux node or cluster start-up and shut-down
- Coordinated planned and unplanned HyperSwap of disk subsystems, transparent to the operating system images and applications using the disks
- Transparent disk maintenance and failure recovery with HyperSwap across z/OS and Linux applications

The Continuous Availability Deployment model discussed in 3.4, “Continuous Availability” on page 89 is based on a configuration running z/OS images on IBM Z, z/VM and KVM and Linux on Z LPARS running on LinuxONE. See Figure 3-10 on page 90.

2

Technical View on LinuxONE Resiliency

This chapter explores the IBM LinuxONE capabilities designed to achieve the highest platform resiliency and availability.

It describes the various aspects of a virtualized environment, including the IBM LinuxONE processor Reliability, Availability and Serviceability (RAS) characteristics, Processor Resource System Manager (PR/SM) and Dynamic Partition Manager (DPM) operation modes and Virtualization and Hypervisors such as z/VM and KVM. Resiliency automation for the infrastructure is also covered. It also covers the LinuxONE Input/Output (I/O) capabilities with FICON® and FCP SCSI, as well as Networking with the use of internal network capabilities and network cards. This chapter includes the following topics:

- 2.1, “Layers of Resiliency based on IBM LinuxONE”, on page 30
- 2.2, “Infrastructure”, on page 30
- 2.3, “Operating environment”, on page 39
- 2.4, “LinuxONE Virtual Storage”, on page 49
- 2.5, “LinuxONE Virtual Network”, 51
- 2.6, “Applications”, on page 55
- 2.7, “Management”, on page 59
- 2.8, “Automation”, on page 62
- 2.9, “Building the Related Deployment Model”, on page 73

2.1 Layers of Resiliency based on IBM LinuxONE

To build an environment for highest resiliency on IBM LinuxONE machines, we have to consider all the layers that are involved and could impact the resiliency of a running environment as shown in Figure 2-1.

We also need to consider which layers could build the most reliable environment and best availability of the application services, no matter of planned or unplanned outages of components on any of the layers that are part of the environment. The picture below illustrates these layers and has options on the right side which could contribute to build an environment with the highest resiliency in IT industry with IBM LinuxONE.

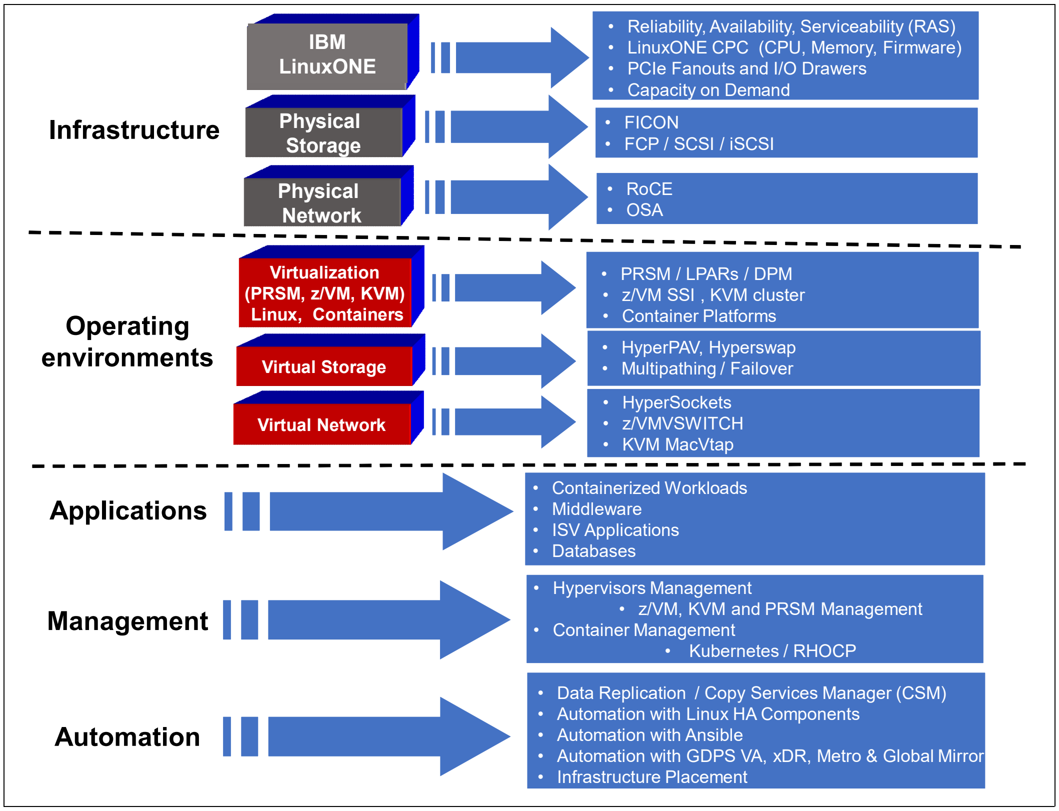

Figure 2-1 LinuxONE Resiliency Layers

2.2 Infrastructure

This section describes the infrastructure layer, which includes the IBM Linux ONE built-in hardware resiliency components and its RAS platform enhancements. It also covers the IBM LinuxONE CPC components and the infrastructure layer's key capabilities, including compute, physical storage and physical network.

2.2.1 IBM LinuxONE - built-in Hardware Resiliency

The IBM LinuxONE system has its heritage in decades of extensive research and development by IBM into hardware solutions to support mission-critical applications for the most diverse industries and delivers exceptional overall resiliency.

Resiliency is accomplished with built-in redundancy, simultaneous replacement, repair, and upgrade functions for the Central processor complex (CPC), processing units, memory and I/O drawers, as well as I/O components for storage and networking. IBM LinuxONE resiliency also extends the capability to install firmware updates (known as LIC, Licensed Internal Code) non-disruptively.
In most cases, LinuxONE capacity upgrades can be concurrent, without a system restart. See , "Capacity on Demand - (Temporary Upgrades)" on page 36.

Resilience capabilities vary according to equipment type & model. Figure shows available IBM LinuxONE 4 models.

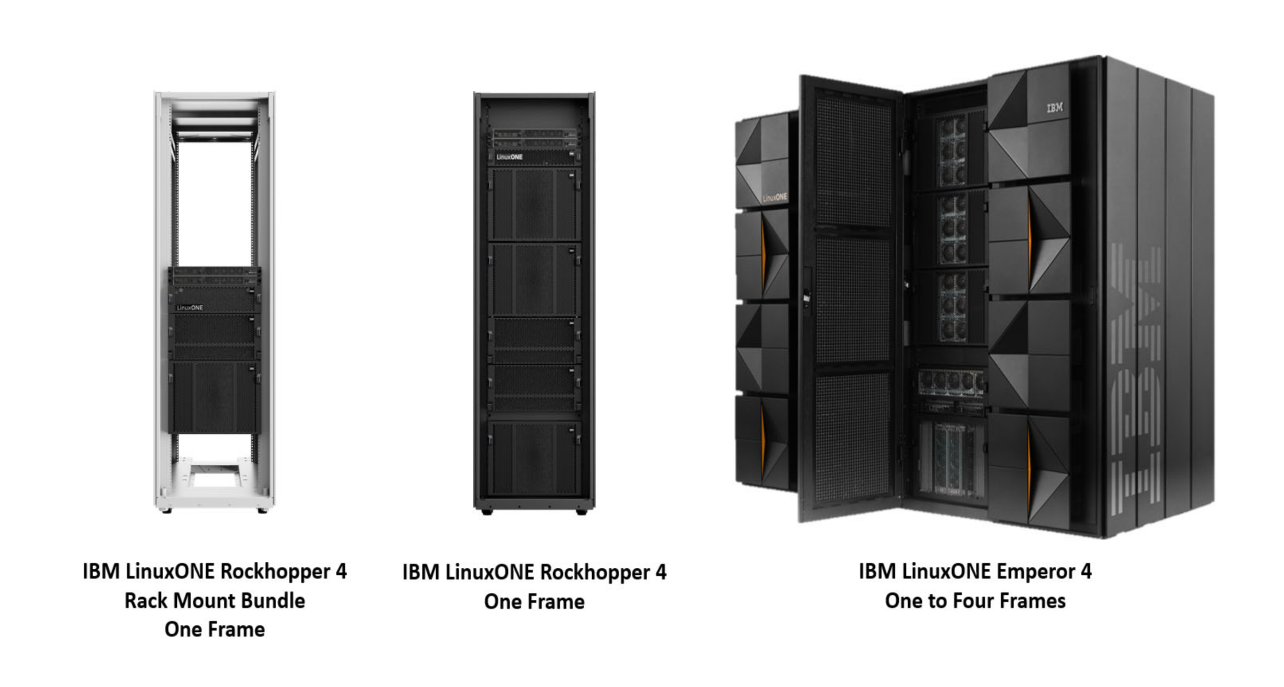

Figure 2-2   IBM LinuxONE 4 available models

LinuxONE Reliability, Availability and Serviceability (RAS)

One of the most basic requirements for continuity of service is that the components (hardware and software) in the IT infrastructure are operating in a resilient environment. This means, if you want to attain the highest levels of availability for your business-critical applications and data, you must start with sound fundamentals. These fundamentals include ensuring that critical IT components have available backup (redundant) capacity, redundant power sources, and redundant connections across critical paths to storage, networks, and other systems, as well as multiple instances of software (operating systems, middleware, and applications).

Note: Redundancy, by itself, does not necessarily provide higher availability. It is essential to design and implement your IT infrastructure by using technologies such as system automation and specific features. These technologies can take advantage of the redundancy and respond to failures with minimal impact on application availability.

From a redundancy and resiliency perspective, the IBM LinuxONE platform design (hardware and software) includes RAS principles that are driven by a set of high-level program objectives that move toward continuous reliable operation (CRO) at the system level. The key objectives of IBM LinuxONE are to ensure data integrity, computational integrity, reduce or eliminate planned and unplanned outages, and reduce the number of repair actions.

The RAS strategy is to manage change by learning from previous generations of IBM Z and LinuxONE and investing in new RAS functions to eliminate or minimize all sources of outages. The RAS strategy employs a building-block approach that is designed to meet stringent requirements for achieving CRO. These are the RAS building blocks:

- Error prevention
- Error detection
- Recovery
- Problem determination
- Service structure
- Change management
- Measurement
- Analysis

Infrastructure Layer

Enhancements to IBM LinuxONE current RAS designs are implemented in the next IBM LinuxONE platform through the introduction of new technology, structure, and requirements. Continuous improvements in RAS are associated with new features and functions to ensure that the IBM LinuxONE platforms deliver exceptional resiliency.
IBM LinuxONE RAS is accomplished with concurrent replace, repair, and upgrade functions for processing units, memory, CPC and I/O drawers, as well as I/O features for storage, network, and clustering connectivity.

The IBM LinuxONE hardware and firmware are a physical implementation of the z/Architecture. The key capabilities that are featured in the infrastructure layer include:

- Compute (CPU, Memory and Firmware).
- Storage
- Network

Compute

The IBM LinuxONE hardware platform can have one or more frames based on the model. The LinuxONE Rockhopper 4 models will always be just one frame whereas the LinuxONE Emperor 4 models can be configured with one to four frames. Even the model with just one frame will have many redundant components and features, because resiliency is a design point of these machines. The frames contain the following components:

- CPC drawers which contain processing units (PUs) also known as cores or General Purpose Processors (GPU), memory, and connectivity to I/O drawers.
- I/O drawers with special cores for I/O features
- Special purpose features, such as on-chip crypto feature and on-chip AI accelerator
- Cooling units for either air or water cooling
- Power supplies
- Oscillator cards for system clock

The LinuxONE model will depend on the number of processing units (PUs), the amount of memory, and how much I/O bandwidth you require to run your workloads. The PUs, memory, and I/O features have built in resiliency and the power, cooling, and system clocking are redundant components in each system.

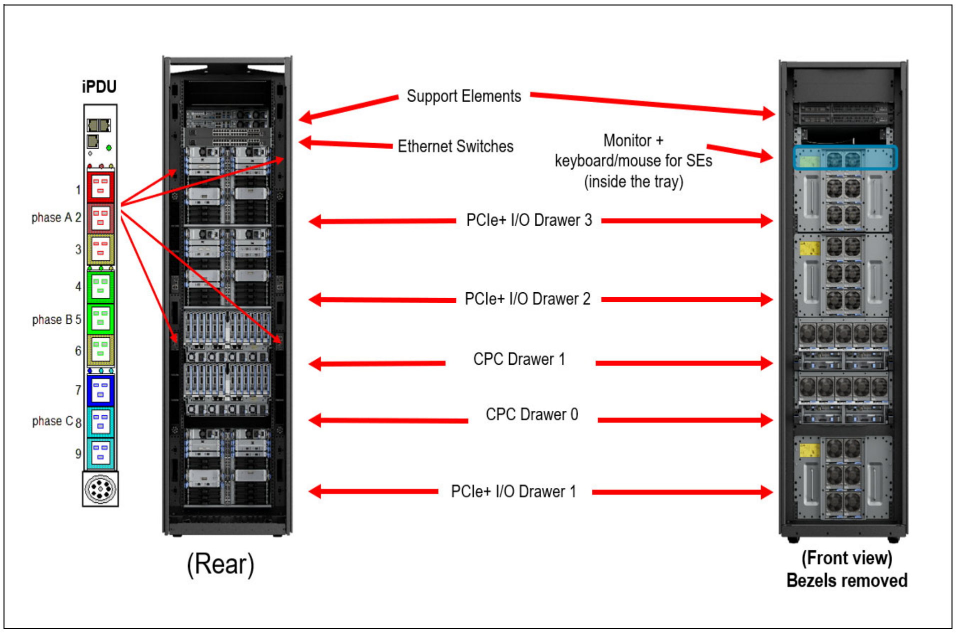

Figure 2-3 BM z16™ LinuxONE Rockhopper 4 under the covers (front and rear views)

LinuxONE CPCs

For highest Resiliency the IBM LinuxONE CPC contains almost all components installed redundant, to avoid any type of outage or it uses the redundancy for self healing.

Each CPC drawer in a LinuxONE hosts the processing units, memory, and I/O interconnects. The CPC drawer design aims to reduce, or in some cases even eliminate, planned, and unplanned outages. The design does so by offering concurrent repair, replace, and upgrade functions for the CPC drawer. Figure 2-3 shows a front and back views of a LinuxONE Rockhopper 4 under the covers.

The process through which a CPC drawer can take over for a failed CPC drawer is called Enhanced (CPC) Drawer Availability (EDA). EDA allows a single CPC drawer in a multi-drawer
configuration to be removed and reinstalled concurrently for an upgrade or a repair.

Processing Units (PUs)

All cores in the LinuxONE are physically the same but can be differentiated by their characteristics. The cores can be characterized in advance or dynamically. Certain core characterizations are well suited to specific types of tasks or as accelerators. The core characterization types in LinuxONE are as follows:

– Integrated Facility for Linux (IFL) core/processor is used for Linux and for running the IBM z/VM hypervisor or KVM (Kernel Virtualization) in support of Linux. z/VM or KVMs often used to host multiple hundreds of Linux virtual machines (known as guests).

- System assist processor (SAP) is used for offload and to manage I/O operations Several SAPs are standard with the platform. More SAPs can be configured if increased I/O processing capacity is needed.
- Integrated Firmware Processor (IFP) - Two cores dedicated to supporting native PCIe features (for example, RoCE Express, zEDC Express, zHyperLink Express, and Coupling Express), and other firmware functions.
- Central processor (CP) - In LinuxONE, a CP is permitted for exclusive use of Virtual Appliance, which is a fully integrated software solution to provide Continuous Availability / Disaster Recovery protection and fail-over automation of workloads on LinuxONE. Refer to "Virtual Appliance (VA)" on page 58.

In the unlikely event of a permanent core failure, each core can be individually replaced by one of the available spares. Core sparing is transparent to the operating system and applications. The resiliency capabilities for the PUs include:

- Transparent core sparing.
- Concurrent processor drawer repair/add, including Processor/Cache Chips, Memory and other internal components.
- Transparent SAP Sparing.

System clocking

LinuxONE has two oscillator cards (OSCs) for system clocking purposes: one primary and one secondary. If the primary OSC fails, the secondary detects the failure, takes over transparently, and continues to provide the clock signal to the system.

Power

The resiliency capabilities for power include transparent fail-over and concurrent repair of all power parts and redundant AC inputs. The power supplies for LinuxONE are also based on the N+1 design. The additional power supply can maintain operations and avoid an unplanned outage of the system.

Cooling

LinuxONE can provide N+1 cooling function for the radiator-based, air cooled model, which suits the needs of typical business computing. The N+1 (redundant) cooling function for the fluid-cooled model suits the needs of enterprise computing. The resiliency capabilities for cooling include transparent fail-over and concurrent repair of cooling pumps, blowers, fans, and so on. The single frame models do not have radiators. The cooling is accomplished by forced air using redundant fans.

System control structure

The system control structure includes redundant sideband control access to all units in the platform, and redundant network switches. The Support Elements (SEs) are connected to support processors in the CPC drawer, I/O drawers, power supplies, and cooling units, where Hardware Management Appliances (HMA) run.

The HMA and Support Element (SE) are appliances that provide hardware management for IBM LinuxONE servers. Hardware platform management covers a complex set of configuration, operation, monitoring, and service management tasks, and other services that are essential to the operations of the platform.

The SEs are stand-alone 1U rack-mounted servers and closed systems that run a set of LinuxONE platform management applications. When tasks are performed at the HMA, the commands are routed to the primary SE of the platform. The primary SE then issues those commands to the LinuxONE.

Two rack-mounted SEs (one is the primary and the other is the alternate) are implemented to manage the LinuxONE platform. SEs include N+1 redundant power supplies. Information is mirrored once per day between the primary and the alternative SEs. The Remote Support Facility (RSF) provides communication with the IBM Support network for hardware problem reporting and service.

Memory

LinuxONE platforms implements Redundant Array of Independent Memory (RAIM), which detects and recovers from failures of dynamic random access memory (DRAM), sockets, memory channels, or Dual Inline Memory Module (DIMM). LinuxONE memory includes these resiliency capabilities:

- DIMM-level failure protection based on RAIM technology.
- Memory channel and bus protection based on CRC and RAIM technology.
- Concurrent memory repair/add through the concurrent drawer repair process.
- Concurrent memory upgrades.

IBM LinuxONE firmware

The IBM LinuxONE firmware provides the flexibility to update dynamically the configuration. You can perform the following tasks dynamically:

- Add a logical partition (LPAR)
- Add a logical channel subsystem (LCSS)
- Add a subchannel set
- Add a logical core to an LPAR
- Add a cryptographic coprocessor
- Remove a cryptographic coprocessor
- Enable I/O connections
- Swap processor types
- Add memory
- Add a physical core

To help minimize planned outages, the following tasks are also possible:

- Concurrent driver upgrades
- Concurrent and flexible customer-initiated upgrades
- Concurrent firmware (patches) updates
- Dynamic PU and SAP reassignment

PCIe fanout

The PCIe fanout in the CPC drawer provides the redundant paths for data between memory and the I/O drawers, which houses the I/O features. The PCIe fanout is hot-pluggable. If a PCIe fanout fails, a redundant I/O interconnect allows a PCIe fanout to be concurrently repaired without loss of access to its associated I/O domains within the I/O drawer.

PCIe I/O drawer

The I/O drawer supports multiple features that are organized in four hardware domains per I/O drawer. Each domain is driven through an IBM Interconnect® PCIe switch adapter. The two PCIe switch cards provide a backup path for each other through the passive connection in the I/O drawer backplane. During a PCIe fanout or cable failure, all I/O features in the two domains can be driven through a single PCIe switch card.

I/O features are supported in any combination and can be concurrently added and removed. The resiliency capabilities for I/O include:

- Multiple channel path support
- Concurrent repair/add of all features in an I/O drawer
- Concurrent repair/add of I/O drawer
- Concurrent upgrade of any I/O feature type
- Domain fail-over based on Redundant I/O interconnect
- Dynamic activation of I/O configuration changes

Capacity on Demand - (Temporary Upgrades)

Businesses must handle unpredictable market opportunities, customer needs, and external pressure without missing a beat or interrupting existing processes. This means your IT infrastructure must support changing business objectives. You should have access to the resources you need, when you need them.

This is the basic principle underlying the Capacity on Demand offerings for IBM Z and LinuxONE. The Capacity on Demand offerings allow you to get the resources you need, when you need them.

The Capacity on Demand offerings provide permanent and temporary upgrades by activating one or more LICCC records. These upgrades occur without disruption to the operation of the server. Depending on the type of upgrade, you can order upgrades yourself using the Customer Initiated Upgrade (CIU) application on Resource Link® or you can call your IBM sales representative to order the upgrades.

Various dynamic “capacity on demand” capabilities for resiliency and capacity change on demand are available for IBM LinuxONE:

- **Non-disruptive Capacity Backup (CBU)**
  - Capacity Backup (CBU) records allow you to replace model capacity or specialty engines to a backup server in the event of an unforeseen loss of server capacity because of an emergency.
- **On-Off Capacity on Demand (OOCoD)**
  - On/Off Capacity on Demand (On/Off CoD) records allow you to temporarily add model capacity or specialty engines due to seasonal activities, period-end requirements, peaks in workload, or application testing.
- **IBM Z Flexible Capacity for Cyber Resiliency**

- Flexible Capacity for Cyber Resiliency records allow you to shift production capacity between participating IBM z16™ servers at different sites.This offering is available for IBM Z beginning with IBM z16 servers and LinuxONE 4.

2.2.2 Physical storage attachment capabilities

FICON / FCP SCSI

The Storage attachment to an IBM LinuxONE system is realized using a separate dedicated I/O subsystem using separate special cores dedicated for I/O.
The I/O attachments for the Storage servers can use FICON cards. FICON cards can be configured to use (FICON Channel) FC protocol for traditional DASDs disks in Extended Count Key Data (ECKD)[1] format, or (Fiber Channel Protocol) FCP protocol for Fibre Channel SCSI Storage devices.

For Resiliency, these cards can be coupled and are acting as a unit that can run the entire workload if one card would malfunction.
IBM LinuxONE systems have dedicated processors (SAPs) to handle I/O which are independent from the application cores. They do not require any capacity planning and are not considered in Licensing models for the software stack.

ECKD over FICON

Following are some of the I/O processing characteristics when using ECKD over FICON:

- LinuxONE firmware handles multiple paths to a disk
- I/O requests are serialized and queued in the Linux guest per disk
- Disk blocks sizes are 4 KB
- High availability is handled by the infrastructure
- DASDs models 3390-3 (2.2 GB), 3390-9 (6.6 GB), 3390-27 (19,8 GB) and 3390-54
- (39,6 GB) have predefined sizes. DASD model 3390-A, known as Extended Address Volumes (EAVs), have a flexible size and support up to 1 TB capacity per disk

Linux Fibre Channel Protocol - FCP

Linux zFCP device driver adds support for Fibre Channel Protocol (FCP)-attached SCSI devices to LinuxONE. FCP is an open, standards-based alternative and supplement to existing FICON connections.

Following are important I/O characteristics of this type of configuration:

- Several I/Os can be issued for a LUN at the same time (asynchronous I/O).
- Disk blocks are 512 bytes
- No ECKD emulation overhead of fixed block devices
- I/O queues occur in the FCP card or in the storage server
- No disk size restrictions
- High availability and load balancing is provided by Linux multipathing, type fail-over, multibus, or z/VM hypervisor exploiting the EDEVICE feature
- Explores N-Port ID Virtualization (NPIV)[2]

[1] ECKD-Extended Count Key Data and CKD-Count Key Data are used interchangeably in this chapter.

[2] N-Port ID virtualization (NPIV) is a Fibre Channel (FC) standard that makes it possible to create multiple virtual ports on a single physical node port (N-Port), with each virtual port appearing as a unique entity to the FC network.

For Resiliency purposes, the storage servers can be used via a SAN Volume Controller (SVC), which can build a Storage Stretched Cluster with high consistency and resiliency based on the storage controller itself. The SVC can handle storage fail-overs between different storage servers.

2.2.3 Physical network attachment capabilities

IBM RoCE Express

Network cards called RoCE (**R**emote Direct Memory Access **o**ver **C**onverged **E**thernet) cards, are similar to OSA cards, can share their ports as well, but can run different network protocols, beside the traditional ETH protocol.

RoCE Express cards use a communication mechanism called Shared Memory Communication (SMC) that enables two SMC-capable peers to communicate using memory buffer allocated by each peer for the partner's use. There are two types of Shared Memory Communications:

- Shared Memory Communications over Remote Direct Memory Access (RDMA) or SMC-R:
  - SMC-R is a network capability takes a new leap, strengthening performance for sharing data and reducing data transmission network overhead. SMC-R provides application transparent exploitation of this new RoCE feature reducing the network overhead and latency of data transfers, effectively offering the benefits of optimized network performance between two LinuxONE processors.
- Shared Memory Communications-Direct Memory Access, or SMC-D:
  - SMC-D enables direct access to shared virtual memory providing a highly optimized network interconnect for IBM LinuxONE intra-CPC communications using Internal Shared Memory. Refer to “Internal Shared Memory (ISM)” topic below.

Open Systems Adapter (OSA)

The Open Systems Adapter is actually a network controller that you can install in a LinuxONE I/O Cage. The OSA-Express features integrate network interface hardware and support many networking transport protocols. Every OSA-Express feature provides capabilities in the following areas:

- Standard Ethernet support
- Operating modes
- Connectivity options (bandwidth and data throughput)
- Reliability, availability, and serviceability (RAS)

Effectively, the OSA integrates the control unit and device into the same hardware. It does so by placing it on a single card that directly connects to the central processor complex I/O bus.

Standard Ethernet Support

The following Ethernet standards are applicable for OSA-Express features (standard transmission schemes):

- IEEE 802.3 and IEEE 802.1
- Ethernet V2.0
- Data Integrity Extensions (DIX) V2

OSA Operating Modes

With the IBM LinuxONE platform, the integration of a channel path with network ports makes the OSA a unique channel or channel path identifier (CHPID) type, which is recognized by the hardware I/O configuration on a port-by-port basis. The following CHPID types are described in further detail in the *IBM Z Connectivity Handbook*, SG24-5444 subsequent sections:

- Queued direct input/output (QDIO) (OSD)
- Non-queued direct input/output (OSE)[3]

Open System Adapter Support Facility (OSA/SF)

OSA/SF is a host-based tool that is used to customize and manage OSA-Express features:

- OSA/SF is not required for the OSA feature that is configured for the QDIO mode or the default IP pass-through non-QDIO mode. However, it can be used for problem determination purposes.
- OSA/SF is a required facility when the OSA feature is being configured for shared non-QDIO mode and where SNA definitions are involved.
- With the IBM LinuxONE platform and OSA/SF, the Hardware Management Console (HMC) is enhanced to provide configuration, validation, activation, and display support for the OSA-Express7S, OSA-Express7S 1.2, and OSA-Express6S features:
  - OSA/SF on the HMC is required for OSA-Express7S 1.2, OSA-Express7S, and OSA-Express6S features.
  - One OSA/SF application can communicate with all OSA features in an IBM Z platform.
  - OSA/SF communicates with an OSA feature through a device (type OSD) that is defined by using Hardware Configuration Definition (HCD) or the Input/Output Configuration Program (IOCP).

2.3 Operating Environments

This section introduces the IBM LinuxONE platform virtualization capabilities such as Processor Resources Systems Manager (PR/SM) and Dynamic Partition Manager (DPM). It also discusses virtualization implementations with z/VM and KVM hypervisors and container platforms.

2.3.1 Virtualization

IBM LinuxONE is a completely virtualized environment which understands ‘bare metal’ as the virtualization mode with Processor Resource/System Manager (PR/SM) entity. This level is already a virtualization layer that is implemented in the hardware and is highly efficient for resiliency and fail-over capabilities between Logical Partitions (LPARs) and the resources attached to it.

An LPAR is a subset of the processor hardware that is defined to support an operating system. An LPAR contains resources (processors, memory, and input/output devices) and operates as an independent system. Multiple logical partitions can exist within Linux ONE mainframe hardware system.

[3] IBM LinuxONE 4 is planned to be the last Server to support OSE networking channels. (This refers to: IBM support for the System Network Architecture (SNA) protocol being transported natively out of the server by using OSA-Express 100BASE-T adapters).

Partitioning control specifications are partly contained in the Input/Output Configuration Data Set (IOCDS) and are partly contained in a system profile. The IOCDS and profile both reside in the Support Element (SE). An IOCDS contains information to define the I/O configuration to the processor channel subsystem and is created by a program called I/O Configuration Program (IOCP).

A second layer of virtualization is in each LPAR – and can be realized by implementing an hypervisor in the LPAR which enables the virtualization of all hardware resources, and thousands of Linux guests which can be run at the same time in a single LinuxONE system.

The two hypervisors that can be installed in a LinuxONE LPAR are z/VM, and KVM.

With z/VM you can have IBM’s premier hypervisor matured over decades and exploiting very granular resource sharing and resource shifting between the virtualized Linux virtual guests controlled by z/VM.

With KVM you can enable on LinuxONE an hypervisor developed by the Open Source Community, which is adapted by our premier Linux distribution partners, Red Hat, SUSE and Canonical.

Platform virtualization

Platform Virtualization is a principal strength of the IBM LinuxONE. It is embedded in the architecture and built into the hardware, firmware, and operating systems. For decades, the IBM LinuxONE platforms have been designed based on the concept of partitioning resources (such as CPU, memory, storage, and network resources). So, each set of features can be used independently with its own operating environment.

Every LinuxONE platform is highly virtualized, with the goal of maximizing utilization of computing resources, while lowering the total number of resources and cost needed to run critical workloads and solutions.

Virtualization can help secure and isolate application workloads and data within virtual servers and storage devices for easier replication and restoration. This added resiliency can provide you with greater flexibility to maintain a highly available infrastructure, while performing planned maintenance, and to configure low-cost disaster-recovery solutions.

2.3.2 IBM Processor Resource Systems Manager (PR/SM)

The hardware level virtualization capability in the IBM LinuxONE, PR/SM, is a Type-1 hypervisor that runs directly on bare metal, allowing you to create multiple logical partitions called LPARs. PR/SM is a highly reliable, proven, and secure, firmware-encapsulated virtualization technology that allows multiple operating systems to run on the same physical platform. Each operating system runs in its own logical partition (LPAR).

IBM PR/SM is a technology used in IBM LinuxONE to provide logical partitioning and resource management capabilities. PR/SM allows a LinuxONE system to be divided into multiple logical partitions, each running its own operating system instance and set of applications with highest isolation.

PR/SM logically partitions the platform across the various LPARs to share resources, such as processor units and I/O (for networks and storage), allowing for a high degree of virtualization and highest isolation at the same time.

The main goal of PR/SM is to maximize the utilization of system resources and improve overall system performance. It allows up to 85 logical partitions to run concurrently on a single physical LinuxONE Emperor and up to 40 on a single LinuxONE Rockhopper, enabling organizations to consolidate workloads and at the same time isolate them for Resiliency and therefore reduce hardware costs.

Here are some key features and capabilities of IBM PR/SM:

- **Logical Partitioning**: PR/SM divides a physical LinuxONE into logical partitions, also known as LPARs. Each LPAR functions as an independent virtual LinuxONE, with its own processors, memory, and I/O devices. This allows different operating systems to run concurrently on the same hardware.
- **Resource Management**: PR/SM manages the allocation of system resources, such as processors, memory, and I/O channels, to each logical partition based on predefined rules and policies. It ensures that each partition gets the necessary resources to meet its workload requirements while preventing resource contention and optimizing overall system performance.
- **Dynamic Resource Allocation:** PR/SM supports dynamic resource allocation, allowing system administrators to adjust the allocation of processors and memory to individual partitions without disrupting the operation of other partitions. This flexibility enables organizations to respond to changing workload demands and optimize resource utilization on-the-fly.
- **High Availability and fail-over:** PR/SM provides built-in resiliency features for high availability and disaster recovery. It supports the use of redundant components, such as processors and I/O devices, to ensure system availability in the event of hardware failures. Additionally, PR/SM permits automation to support workload movement or restart on different physical machines for planned maintenance or in case of failures, minimizing downtime.
- **Workload isolation:** Each logical partition in PR/SM is isolated from others, providing a secure and controlled environment for running different workloads. This isolation has Evaluation Assurance Level (EAL) 5+ Common Criteria certification, prevents disruptions caused by one partition from affecting others, improving system stability and security.
- **LPAR weights:** In an environment with shared capacity it is important to make use of LPAR weight capabilities to avoid capacity stealing between LPARs and a definition of capacity guarantees for certain workloads in the LPARs. More details see LPAR weights here: https://www.youtube.com/watch?v=VVviT4N4BiE 2.8.3, “Automation with Linux HA components” on page 63

Overall, IBM PR/SM is a powerful virtualization technology that enables efficient resource management and workload consolidation on IBM LinuxONE systems. It helps organizations maximize the utilization of their LinuxONE hardware and achieve better performance, flexibility, and reliability in their computing environments.

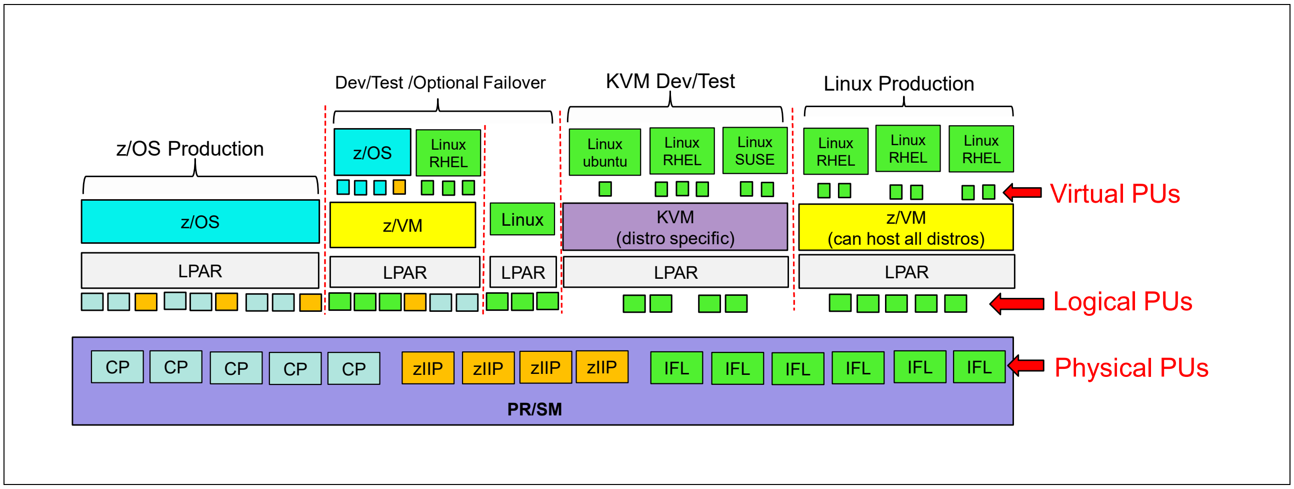

Figure 2-4 Exploiting PR/SM virtualization capabilities.

Figure 2-4 shows the various options for exploiting the IBM PR/SM virtualization capabilities such as z/VM, KVM as well ad native z/OS and native Linux distributions. IBM Linux ONE supports only z/VM, KVM and native Linux LPARs.

Resiliency in workloads necessitates the utilization of one or more LPARs, along with various modes of software-level control to ensure availability. It is essential to predetermine the desired availability levels for each workload to understand the corresponding requirements.

To provide application fail-over capabilities for multi-LPAR application workloads, it is recommended to implement high availability clustering solutions such as Pacemaker, Red Hat High Availability Add-on for Red Hat Enterprise Linux, or SUSE High Availability. These solutions facilitate automatic fail-over and workload recovery in the event of hardware or software failures.

Regular testing of LPAR application workloads' resiliency through simulation and controlled failure scenarios is vital. Conducting disaster recovery drills, fail-over testing, and load testing will validate the effectiveness of resilience measures and identify areas for improvement.

Developing a comprehensive disaster recovery plan tailored to specific needs is of utmost importance. This plan should encompass backup and replication strategies, data replication, fail-over procedures, and data integrity validation, all aimed at expediting recovery processes and minimizing downtime.

LinuxONE can be configured as either PR/SM or Dynamic Partition Manager (DPM) mode.

2.3.3 Dynamic Partition Manager (DPM)

DPM is a configuration manager that is designed for setting up and managing Linux servers that run on a LinuxONE system. On a DPM-enabled system, the runtime environment for your LinuxONE is called a partition. A partition is also the runtime environment for a hypervisor and its guest operating system images.

On a LinuxONE system, a partition is a virtual representation of all of the physical hardware resources of that system, which include processors, memory, and input/output (I/O) adapters. On LinuxONE systems, as on other platforms, an adapter is a physical device that connects the system to other computers or devices. In contrast to other platforms, adapters on a

LinuxONE system can be shared between partitions, which reduces the amount of adapters that might be required to handle a specific workload.

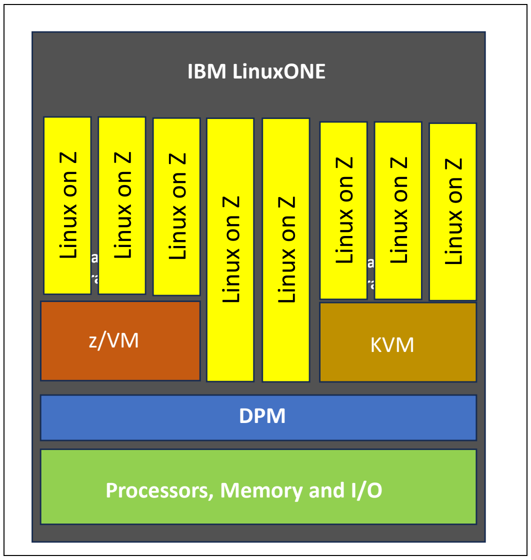

Figure 2-5 Dynamic Partition Manager Diagram

DPM is a feature that is enabled by the systems' initialization process and is mutually exclusive with PR/SM. Once enabled, the DPM mode will be ready when the system is powered on.

With DPM you will use the HMC to configure the running environment for your LinuxONE server. DPM automatically discovers and display the system resources that are available for you to use, and indicates how your selections might affect other servers and applications that are already defined or running on the same system.

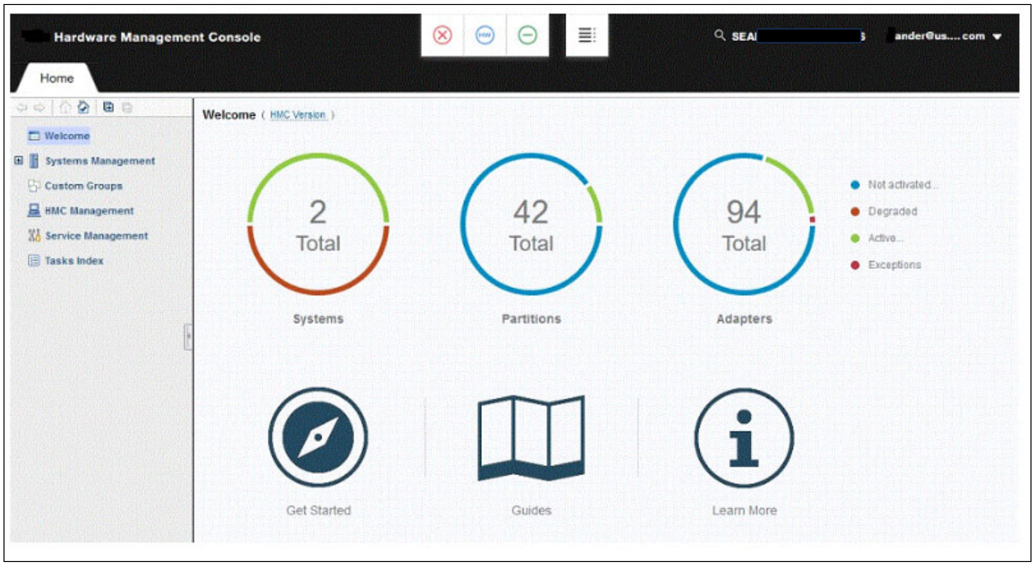

Figure 2-6 HMC "Welcome" page in DPM mode

After your system is up and running in DPM mode, you can use DPM to:

– Modify system resources without disrupting running workloads.

- Monitor sources of system failure incidents and conditions or events that might lead to workload degradation.
- Create alarms so that you can be notified of specific events, conditions, and changes to the state of system resources.
- Update individual partition resources to adjust capacity, redundancy, availability, or isolation.

For additional DPM mode information, refer to *Dynamic Partition Manager (DPM) Guide,* SB10-7176.

2.3.4 Highest workload isolation in LPAR

The LPARs in an IBM LinuxONE system have highest isolation, certified as Evaluation Assurance Level 5 (EAL5) – which is the highest isolation in IT industry, which can be seen as isolated as physical servers.
Therefore, workloads in LPARs have 2 unique resiliency characteristics on IBM LinuxONE, the resources can be shared across LPARs and at the same time reach the highest workload isolation and resiliency in a LPAR.

Two LPARs can be used as separate entities with shared resources of computing and if one LPAR is not functioning, the other LPAR can run the entire workload due to the capacity shift to it. This is built-in resiliency per hardware design.

As shown in Figure 2-5 on page 43, DPM is not fully covering all features of PR/SM, but is enhancing the user experience for an IBM LinuxONE especially in a cloud based environment where Software defined Hardware is important and the Hardware Resiliency is controlled and managed with external tools.

2.3.5 The z/VM hypervisor on IBM LinuxONE

IBM z/VM is a highly secure and scalable virtualization technology. It is a hypervisor that runs on LinuxONE systems, allowing multiple virtual machines to run on a single or multiple logical partitions on a physical machine. Each virtual machine can run its own operating system, such as Linux. With z/VM, organizations can consolidate workloads and improve resource utilization and resiliency, while maintaining high levels of security and performance.

z/VM is a Type-2 hypervisor, that allows sharing the LinuxONE platform’s physical resources, such as disk, memory, network adapters, and cores (called Integrated Facility for Linux - IFLs). These resources are managed by the z/VM hypervisor, which runs in an LPAR, and manages the virtual machines (VMs) that run under the control of the z/VM hypervisor. Typically, the z/VM hypervisor is used to run Linux virtual servers, but it can also be nested that means a second layer virtualization is possible. This enables granular testing capabilities and verification for resiliency.

With z/VM the workload can scale efficiently horizontally, by enabling more virtual servers in the same z/VM, and vertically, which means that applications can scale without modification by just dedicating them dynamically more resources via the z/VM hypervisor.

You can combine the horizontal and vertical scalability and reach the best virtualization capability in an IT environment. That is unique to the IBM LinuxONE environment.

z/VM can host Linux guests from all distribution partners in the same LPAR, because its EAL4+ certified isolation for Virtual server guarantees the isolation of each server in a safe, secure, and scalable environment.

Within z/VM, the network topology can be simplified using z/VM VSWITCH technology, a virtual software defined network that can run in Layer 2 or Layer 3 connecting to one or multiple OSA network cards for resiliency.

z/VM VSWITCH connected to multiple OSA cards can implement port aggregation, which enables a resilient fail-over and enlarged network bandwidth at the same time.

The direct connection to a LinuxONE inter-network topology implemented with Hipersockets across LPARs is another unique characteristic which uses the resiliency of the hardware features built in LinuxONE and the z/VM hypervisor.

z/VM Resiliency

IBM z/VM delivers resiliency for the virtual machines and workloads in several levels:

- Linux multipathing of ECKD and FBA emulated devices (as EDEVs or EDEVICEs).
  - For ECKD devices it is called HyperPAV
- Capacity of attaching NPIV FCP devices directly to the guests for Linux multipathing
- Network redundancy with 2 or more Open Systems Adapters (OSAs) through z/VM Virtual Switch (VSWITCH)
  - z/VM VSWITCH supports various configurations including Layer 2 and 3 networking, Active/Backup, and IEEE 802.3ad Link Aggregation, Virtual Edge Port Aggregator (VEPA) IEEE 802.1Qbg, VLAN aware and unaware, among others. Other than OSA Express, RoCE Express does not support z/VM VSWITCH technology to provide path redundancy
- z/VM Single System Image (SSI) and Live Guest Relocation:
  - SSI is a concept in z/VM that enables up to eight z/VM instances to function as a single system. With SSI, these instances are interconnected and collectively manage a shared set of resources, such as storage and network infrastructure. The goal is to provide a cohesive and transparent environment for running VMs across the interconnected z/VM instances. See Figure 2-7 on page 46
  - Live Guest Relocation is a feature of z/VM SSI that allows for the movement of running VMs from one z/VM instance to another while the VMs continue to operate without interruption. These features provide flexibility and high availability by enabling workload balancing, disaster recovery, and maintenance operations without affecting the availability of applications running in the VMs.

For effective workload balancing a prioritization between virtual guests in the LPAR is helpful. Therefore, it can be guaranteed that certain VMs have always enough capacity and get as much as is available if possible. This technology of VM SHARE can be defined on the VM level. For Details see:
https://www.ibm.com/docs/en/zvm/7.3?topic=resources-set-share-command

Another concept for effective workload definition is CPU pooling with z/VM, that uses a pool of CPU capacity within an LPAR and distributes it on a VM base to the workloads. This is used for effective licensing within an LPAR. See: Using CPU Pools.

z/VM Single System Image (SSI)

A z/VM single system image (SSI) cluster is a multisystem environment in which the z/VM systems can be managed as a single resource pool and guests can be moved from one system to another while they are running. SSI is included as part of the z/VM base.

A z/VM SSI cluster consists of up to eight z/VM systems in an Inter-System Facility for Communications (ISFC) collection. ISFC is a function of z/VM Control Program (CP) that provides communication services between transaction programs on interconnected z/VM systems. A group of interconnected domains consisting of z/VM systems that use ISFC to communicate with each other is known as an ISFC collection.

Each z/VM system is a member of the SSI cluster. Figure 2-7 shows the basic structure of a cluster with four members. The cluster is self-managed by CP using ISFC messages that flow across channel-to-channel devices between the members. All members can access shared DASD volumes, the same Ethernet LAN segments, and the same storage area networks (SANs).

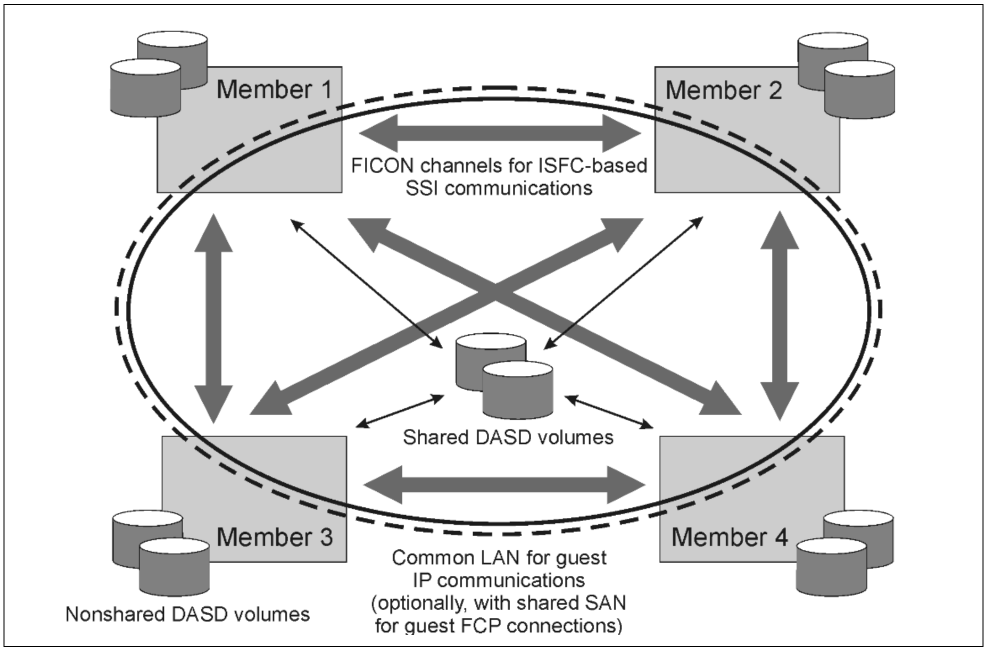

Figure 2-7 A four member z/VM SSI cluster example[4]

Single System Image (SSI) enhances the z/VM systems management, communications, disk management, device mapping, virtual machine definition management, installation, and service functions to enable multiple z/VM systems to share and coordinate resources within a Single System Image structure. This combination of enhanced functions provides the foundation that enables Live Guest Relocations (LGR), which is the ability for a Linux guest to be moved from one z/VM system to another within the SSI cluster.

z/VM Live Guest Relocation (LGR)

A z/VM SSI cluster provides a virtual server mobility function called Live Guest Relocation.

A running virtual server (guest virtual machine) can be relocated from one member to another. Relocating virtual servers can be useful for load balancing and for moving workload off of a physical server or member system that requires maintenance. After maintenance is applied to a member, guests can be relocated back to that member, thereby allowing you to

[4] A z/VM SSI cluster can have up to eight members.

maintain z/VM and to keep your Linux virtual servers highly available.When a relocation is initiated, the guest continues to run on the source member until the destination environment is completely prepared. At that point the guest is briefly quiesced on the source member and then resumed on the destination member. Refer to: Planning for a Single System Image and Live Guest Relocations.

2.3.6 The KVM hypervisor on IBM LinuxONE

KVM is an Hypervisor which is developed by the Open Source community and is then adapted and integrated in every Linux distribution for IBM LinuxONE.
KVM represents an hypervisor that provides server virtualization for Linux workloads that run on the IBM LinuxONE platform, leveraging the existing skill set used in other hardware architectures.

It is also a Type-2 hypervisor that runs in an LPAR and enables sharing CPUs (IFLs), memory, and I/O resources through platform virtualization.
KVM allows the creation and execution of multiple Virtual Machines (VMs) on a Linux host, providing full hardware virtualization capabilities. It leverages the hardware virtualization extensions available on IBM LinuxONE systems for efficient and secure virtualization.
It can coexist with z/VM environments and is optimized for scalability, performance, security, and resiliency, and provides standard Linux and KVM interfaces for simplified virtualization and operational control.
The KVM hypervisor is integrated in the major supported Linux distributions:

– Red Hat Enterprise Linux Server
– SUSE Linux Enterprise Server
– Canonical Ubuntu

KVM on LinuxONE relies on hardware, LPAR and Linux host redundancy and resiliency as a starting point. As such it implements features of high availability, redundancy, and resiliency at hypervisor level, such as:

– Multipathing of virtual disks with Linux Device Mapper-Multipathing (DM-MP).
– Network redundancy based on Linux bonding and teaming. Active-backup, round-robin, and IEEE 802.3ad Link Aggregation are supported.
– Virtual Machine Live Migration, which enables you to move a running VM from one physical host to another without disrupting its operation.
– Live Migration: The process of transferring a running VM from the source host to the destination host while it continues to run without any noticeable downtime. It allows you to achieve load balancing, perform hardware maintenance, or migrate VMs between different physical hosts for various reasons, such as a workload balancing or a planned maintenance.
– Because RoCE Express does not provide a promiscuous mode, you cannot use Open VSWITCH in a KVM host to provide path redundancy for its guests, instead MacVtap can be used for internal communication of KVM guests.

KVM Cluster

KVM Cluster is a virtualization environment that leverages the KVM hypervisor to create and manage Virtual Machines on Linux Systems. KVM Clusters connect multiple nodes to work collectively and manage virtual resources. KVM Clusters implement High Availability and

Continuous Operations by distributing Virtual Machines across nodes. In case of a node failure the Virtual Machine can migrate to operational nodes minimizing downtimes.

2.3.7 Container Platforms

IBM LinuxONE offers robust support for various container technologies and platforms, enabling the utilization of cloud-native applications, including Linux-based workloads, microservices, hybrid cloud deployments and their associated ecosystems. The configuration options available include:

Linux Applications

IBM LinuxONE facilitates the containerization of Linux-based applications through technologies such as Open Container Initiative (OCI) compliant containers. Containers provide a lightweight and isolated runtime environment, ensuring consistency and portability across different environments. Leveraging LinuxONE's powerful hardware resources and scalability, it becomes an optimal choice for running large-scale containerized workloads.

Microservices Architecture

IBM LinuxONE is highly suitable for deploying microservices-based architectures using containers. By breaking down complex applications into smaller, decoupled services, developers can independently develop, deploy, and scale these services. Containers enable efficient resource utilization and rapid deployment, making LinuxONE an ideal platform for managing microservices workloads.

Cloud-Native Applications

IBM LinuxONE fully supports the development and deployment patterns of cloud-native applications. These applications are designed to harness the benefits of containerization, scalability, and elasticity. Leveraging technologies such as Kubernetes, IBM LinuxONE enables automated container orchestration, scaling, and load balancing, providing a reliable and high-performance foundation for running cloud-native applications.

Hybrid Cloud Deployments

IIBM LinuxONE facilitates the deployment of containerized workloads in hybrid cloud environments. Utilizing solutions like Red Hat OpenShift Container Platform, which supports multiple hardware architectures including IBM LinuxONE, enables the management and deployment of containers across both on-premises LinuxONE systems and public cloud platforms. This approach allows users to leverage containerization advantages while maintaining a consistent and secure environment throughout their hybrid cloud infrastructure.

Container Platforms

IBM LinuxONE supports a wide range of container platforms that provide capabilities for managing containerized applications. An example is the Red Hat OpenShift Container Platform (RHOCP), which is available for various hardware architectures, including IBM LinuxONE. RHOCP, built upon Kubernetes, offers a comprehensive environment for developing, deploying, and managing containerized applications. It includes features like automated scaling, load balancing, and container lifecycle management.

2.4 LinuxONE Virtual Storage

This section covers the virtual storage capabilities exploited in the LinuxONE Platform. Topics discussed include Parallel Access Volumes PAV, HyperPAV, HyperSwap capabilities, and Multipathing.

The Parallel Access Volume, or PAV, facility allows a controller to offer multiple device numbers that resolve to the same DASD, which allows I/O to the same DASD to happen concurrently.

With this concept, the device addressed by the operating system to perform the I/O operation is the “Base” device. When the Base device is busy working and another I/O operation to the same Base address is started, it can be executed by one of the “Alias” devices, if PAV or HyperPAV is used.

If there is no aliasing of disk devices then only one I/O transfer can be in progress to a device at a time, regardless of the actual capability of the storage server to handle concurrent access to devices. Parallel access volume exists for Linux on System z® in the form of PAV and HyperPAV. Compared to PAV, HyperPAV is much easier to administer and provides greater flexibility. PAV and HyperPAV are optional features that are available on the DS8000® Storage Subsystems series

Parallel Access Volume (PAV)

PAV uses *“base”* and *“aliases”* devices. Base devices are the target device to carry the I/O operation. With PAV, if a base device is busy, and an alias is available, it will be used to perform the I/O. Refer to Figure 2-8

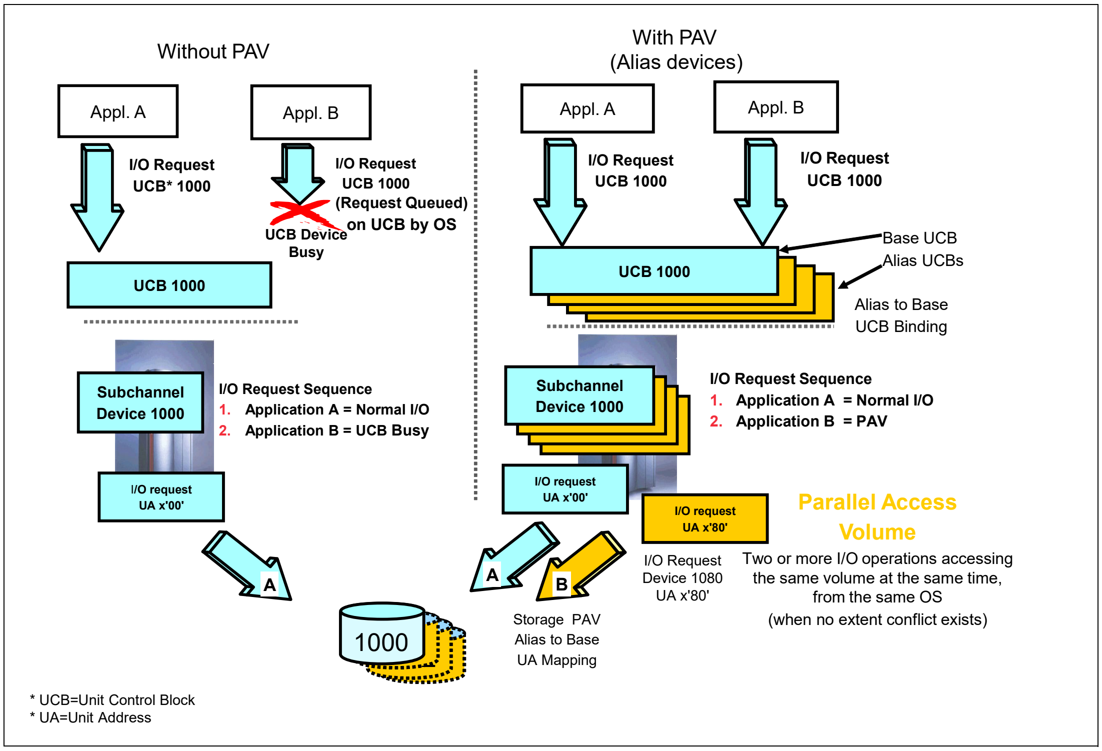

Figure 2-8 Parallel Access Volume (PAV) exploitation example

HyperPAV

The PAV capability is extended dynamically with IBM Hyper-Parallel Access Volume (HyperPAV). HyperPAV allows multiple I/O operations to a DASD through a pool of base and alias subchannels (devices) that are shared within a logical subsystem (LSS). HyperPAV eliminates the need for users to map volumes to aliases and takes care of the aliases and I/Os automatically. See Figure 2-9 on page 51.

ECKD with HyperPAV

The Base and HyperPAV devices (alias) are defined on the storage server and on the IOCDS on LinuxONE, where the DASD devices are defined with unit addresses. A (Logical Control Unit) LCU is a logical set of DASD devices on a DS800 series disk storage unit and it can have up to 256 possible DASD device addresses from 00 to FF. After the base devices are defined any remaining device numbers in the LCU can be used as an alias by the system to access any of the base addresses in the same LCU.

Using ECKD devices with HyperPAV, Figure 2-11 on page 54, can tremendously improve performance and increase availability. Following are some characteristics of ECKD with HyperPAV:

- The DASD driver sees the real disk and all alias.
- Load balancing with HyperPAV is done in the DASD driver.
- It uses less processor cycles than Linux multipathing.

For resiliency, storage server with ECKD disks will typically be mirrored via the storage server function of Metro Mirror for synchronous replication and metro distances, or with Global Mirroring if the distances are larger. The swapping between the mirrored discs needs to be

triggered and managed by external tooling like, GDPS.
For more information about HyperPAV support, refer to: LinuxONE and Linux on Z HyperPAV.

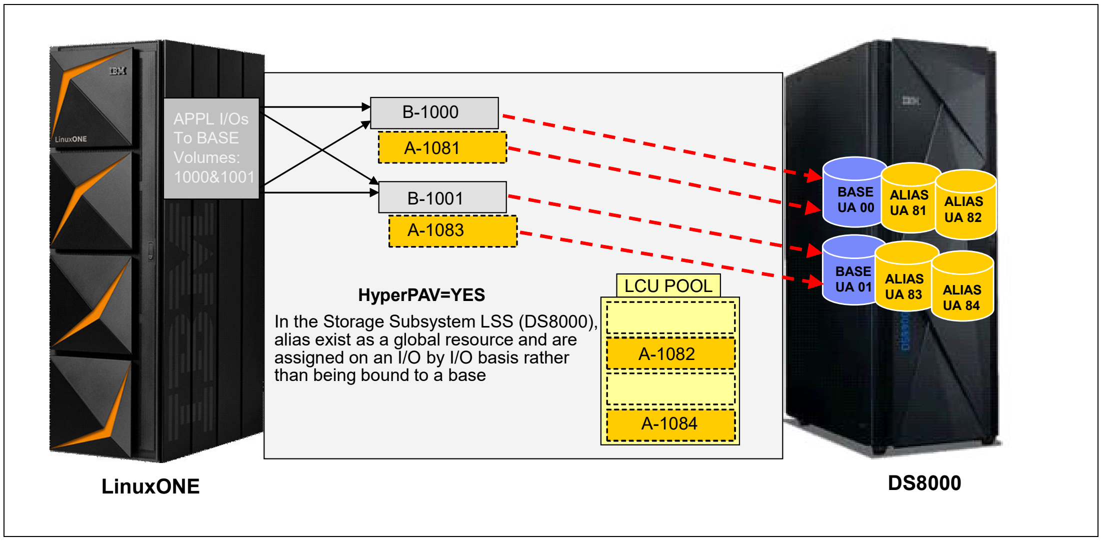
APPL I/Os To BASE Volumes: 1000&1001
B-1000
A-1081
B-1001
A-1083
HyperPAV=YES
In the Storage Subsystem LSS (DS8000), alias exist as a global resource and are assigned on an I/O by I/O basis rather than being bound to a base
LCU POOL
A-1082
A-1084
BASE UA 00
ALIAS UA 81
ALIAS UA 82
BASE UA 01
ALIAS UA 83
ALIAS UA 84
LinuxONE
DS8000

Figure 2-9 HyperPAV exploitation example

2.5 LinuxONE Virtual Network

The IBM LinuxONE system is a fully virtualized platform that can support many LPARs and thousands of Linux system images at once. Therefore, network connectivity covers the connections between the platform and external networks as well as the intra-system connectivity between the Linux system images or Virtual machines hosted on a system.
Each LPAR can have one or more network interfaces assigned to it. The network topology is dependent on the isolation requirements and the characteristics of the workloads, but sharing and isolation are concepts that are highly used in an IBM LinuxONE environment.
The network cards can have multiple ports that represent an individual network interface. These ports can be assigned to one or multiple LPARs and therefore the isolation of the LPAR and the sharing of a port are unique characteristics of IBM LinuxONE and contribute to the high resiliency of the environments hosted there.

An IBM LinuxONE system, can make use of multiple network cards and different network topologies.

Using system internal network capabilities

With IBM LinuxONE, there is a network topology possible which is for internal networks between LPARs, using microcode only and does not require any network card.

2.5.1 IBM Hipersockets

IBM HiperSockets is an integrated function of LinuxONE that supplies attachments to virtual local area networks within the same system. HiperSockets provides connectivity across multiple LPARs on the same LinuxONE system by performing memory-to-memory data transfers through microcode in the LinuxONE hardware, in a secure way. The HiperSockets function eliminates the use of I/O subsystem operations and the use of an external network connections.

For more information about the available OSA Express and RoCE Express features, see: *IBM Z Connectivity Handbook*, SG24-5444:
Depending on the hypervisor you can use virtual switches in combination with the network capabilities of IBM LinuxONE mentioned above.

The Figure Figure 2-10 on page 52 shows examples of the most used network capabilities with OSA, RoCE, and Hipersockets network interfaces. There are three Linux instances; two of them run as z/VM guests in one LPAR and a third Linux instance runs in another LPAR. Within z/VM, Linux instances can be connected through a guest LAN or VSWITCH. Within and between LPARs, you can connect Linux instances through HiperSockets. OSA-Express cards running in either non-QDIO mode or in QDIO mode can connect the LinuxONE to an external network.

> **Note:** For information about QDIO and Non-QDIO modes, please refer to *IBM Z Connectivity Handbook*, SG24-5444.

For resiliency it is recommended to use multiple network interfaces and even for different workloads. It can be beneficial to use multiple networks with different characteristics to support the application requirements regarding network bandwidth and resiliency.

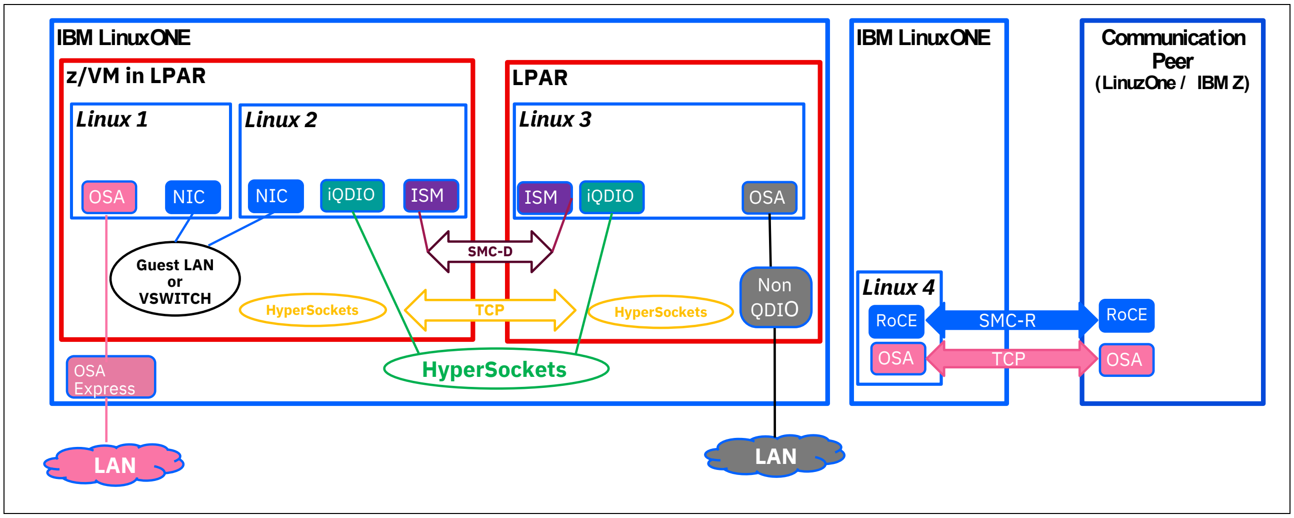

Figure 2-10 LinuxONE Network Capabilities

2.5.2 KVM MacVTap

MacVTap is a Linux kernel device driver that enables Network bridge-like networking and simplifies virtualized bridged networking setup. MacVTap driver is used to attach the guest's NIC directly to a specified physical interface of the host machine. It makes both, the guest and the host, to appear directly on the network switch the host is connected to. MacVTap also improves throughput and latencies to external systems.

A MacVTap endpoint is a character device that follows the **`tun/tap`** ioctl interface and can be used directly by kvm/qemu and other hypervisors that support the tun/tap interface.

MacVTap can be configured in any of three different modes which determine how the MacVTtap device communicates with the lower device in the KVM host. The three possible modes are VEPA, Bridge and Private mode. See "Configuring a MacVTap interface" for additional setup information.

VEPA

Virtual Ethernet Port Aggregator (VEPA) is the default mode. Data flows from one endpoint down through the source device in the KVM host out to the external switch. If the switch supports hairpin mode, the data is sent back to the source device in the KVM host and from there sent to the destination endpoint.

Bridge

Connects all endpoints directly to each other. Two endpoints that are both in bridge mode can exchange frames directly, without the round trip through the external bridge. This is the most useful mode for setups with classic switches, and when inter-guest communication is performance critical.

Private

Private mode behaves like a VEPA mode endpoint in the absence of a hairpin aware switch. Even when the switch is in hairpin mode, a private endpoint can never communicate to any other endpoint on the same lowerdev.

2.5.3 z/VM VSWITCH

z/VM Virtual Switch (VSWITCH) is a bridge-like network that emulates switch technology inside the hypervisor layer. The virtual switch is designed to help eliminate the need for virtual machines acting as routers. Virtual routers consume valuable processor cycles to process incoming and outgoing packets, requiring additional copying of the data being transported. The VSWITCH helps alleviate this problem by moving the data directly between the real network adapter and the target or originating guest data buffers.

VSWITCH is a z/VM CP system-owned switch (a virtual switch) to which virtual machines can connect. Each switch is identified by a *switchname.* A z/VM user can create the appropriate QDIO network interface card (NIC) and connect it to this switch with the NICDEF directory statement. Alternatively, a z/VM user can create a virtual network adapter (with the CP DEFINE NIC command) and connect it to this LAN (with the COUPLE command). See z/VM: CP Commands and Utilities Reference for more information on these commands.

The z/VM Virtual Switch can be configured in *Bridge* or *VEPA* mode, depending upon local requirements for network separation; this allows for either the performance boost of the "hairpin turn" or separation out to the physical switch. Additionally, the Virtual Switch supports Link Aggregation — the combination of multiple OSA ports into a single logical pipe. This allows for greater throughput and network load-balancing at the Layer 2 level when communicating with a physical switch. This technology extends to multiple systems in a z/VM Single System Image through Inter-VSWITCH Link Aggregation Groups.

For more information about virtual networking options in z/VM, see z/VM: Connectivity.

2.5.4 HyperSwap

HyperSwap is a storage high availability solution for IBM Storage Subsystems, such as DS8000, that enables switching logical units between two Storage Subsystems to ensure continuous operation. IBM HyperSwap is a high availability feature that provides single site or dual-site, active-active access to a volume. This function ensures continuous data availability in case of hardware failure, power failure, connectivity issues, or other unplanned outages. It is designed to offer a robust disaster recovery solution with minimal Recovery Point Objective (RPO) and Recovery Time Objective (RTO).

Additional details about HyperSwap function can be found in section 3.3.8, “HyperSwap Function” on page 88.

2.5.5 Multipathing

Multipath I/O provides failover and might improve performance. You can configure multiple physical I/O paths between server nodes and storage arrays into a single multipath device.

Multipathing thus aggregates the physical I/O paths, creating a new device that consists of the aggregated paths.

Multipathing provides I/O failover and path load sharing for multipathed block devices. In Linux, multipathing is implemented with multi-path tools that provide a user-space daemon for monitoring and an interface to the device mapper. The device-mapper provides a container for configurations, and maps block devices to each other.

A single SCSI device (or a single zFCP[5] unit) constitutes one physical path to the storage. The multipath user-space configuration tool scans *sysfs* for SCSI devices and then groups the paths into multipath devices. This mechanism that automatically puts each detected SCSI device underneath the correct multipath device is called *coalescing*.

Use a multipath setup to access SCSI storage in an Fiber Channel Storage Area Network (FC SAN). The multipath device automatically switches to an alternate path in case of an interruption on the storage system controllers or due to maintenance on one path.

The multipath daemon has default configuration entries for most storage systems, and thus you only need to do basic configuration for these systems.

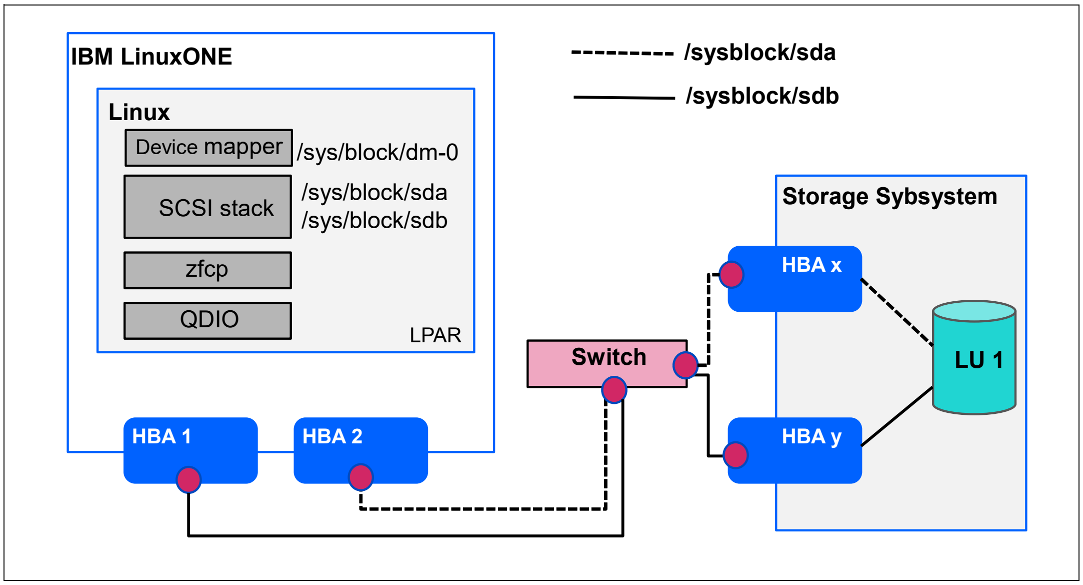

Figure 2-11 Example of Multipathing Implementation with Linux.

[5] zFCP device driver is a low-level driver or host-bus adapter driver that supplements the Linux SCSI stack. See: https://www.ibm.com/docs/en/linux-on-z?topic=channel-what-you-should-knowok

For more information about how to access, configure and use FCP Multipathing with Linux kernel, please access the link below:
IBM Documentation.

2.6 Applications

This section covers application exploitation in the IBM LinuxONE platform and introduces container technology, containerization, and containerized workloads. Also covered are the following topics: Middleware, Middleware types, Independent Software Vendors (ISVs) applications, and Databases supported.

2.6.1 Containers

Container images become containers at runtime. Available for Linux or other platforms such as Windows or Cloud, containerized software will always run the same, regardless of the infrastructure. Containers isolate software from its environment and ensure that it works uniformly despite differences, for instance between development and staging.

Containers are standard units of software that packages up code and all its dependencies so the application runs quickly and reliably from one computing environment to another. Container images are a lightweight, standalone, executable package of software that includes everything needed to run an application:

- code
- runtime
- system tools
- system libraries
- settings

Containers are a way to bundle and run applications. In a production environment, its required to manage the containers that run the applications and ensure that there is no downtime. For example, if a container goes down, another container needs to start.

Wouldn't it be easier if this behavior was handled by a system?

This is where Kubernetes will assist. Kubernetes provides a framework to run distributed systems resiliently. It manages applications scaling and failover as well as deployment patterns and much more.

2.6.2, “Containerization” shows the traditional virtualization implementation in comparison to the Containerized implementation based on a Container Engine.

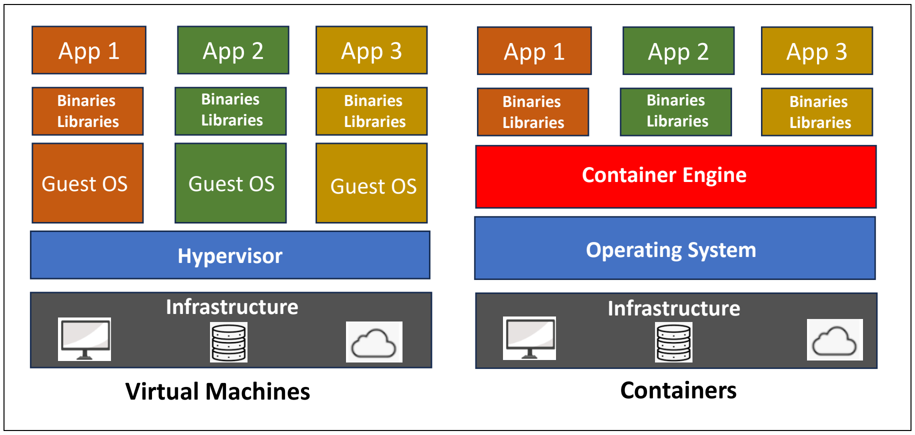

Figure 2-12 Virtualization versus Containerized environments

2.6.2 Containerization

Containerization is the packaging of software code with just the operating system (OS) libraries and dependencies required to run the code to create a single lightweight executable—called a container—that runs consistently on any infrastructure. More portable and resource-efficient than virtual machines (VMs), containers have become the *de facto* compute units of modern cloud-native applications.

Containerization allows developers to create and deploy applications faster and more securely. With traditional methods, code is developed in a specific computing environment which, when transferred to a new location, often results in bugs and errors. For example, when a developer transfers code from a desktop computer to a Virtual Machine (VM). Containerization eliminates this problem by bundling the application code together with the related configuration files, libraries, and dependencies required for it to run. This single package of software or “container” is abstracted away from the host operating system, and hence, it stands alone and becomes portable—able to run across any platform or cloud, free of issues.

Containers are often referred to as “lightweight,” meaning they share the machine’s operating system kernel and do not require the overhead of associating an operating system within each application. Containers are inherently smaller in capacity than a VM and require less start-up time, allowing far more containers to run on the same compute capacity as a single VM. This drives higher server efficiencies and, in turn, reduces server and licensing costs.

For additional information see “Containers in the enterprise”.

2.6.3 Containerized Workloads

Containerized workloads on IBM LinuxONE benefit from the platform's reliability, security, and scalability. With advanced virtualization capabilities and integration with container orchestration platforms, LinuxONE provides an excellent solution for running diverse containerized applications, including Linux-based workloads, microservices, cloud-native applications, and hybrid cloud deployments. Refer to 2.3.7, “Container Platforms” on page 48.

2.6.4 Middleware

Middleware is software that enables one or more kinds of communication or connectivity between applications or application components in a distributed network. By making it easier to connect applications that weren't designed to connect with one another, and providing functionality to connect them in intelligent ways, middleware streamlines application development and speeds time to market.

Middleware enables developers to build applications without having to create a custom integration every time they need to connect to application components (services or microservices), data sources, computing resources or devices.

It does this by providing services that enable different applications and services to communicate using common messaging frameworks such as JSON (JavaScript object notation), Representational State Transfer (REST), Extensible Markyp Language (XML) , Simple Object Access Protocol (SOAP), or web services. Typically, middleware also provides services that enable components written in multiple languages - such as Java, C++, PHP, and Python - to talk with each other.

Midleware Services

In addition to providing this work-saving interoperability, middleware also includes services that help developers:

- Configure and control connections and integrations.
  - Based on information in a client or front-end application request, middleware can customize the response from the back-end application or service. In a retailer's e-commerce application, middleware application logic can sort product search results from a back-end inventory database by nearest store location, based on the IP address or location information in the HTTP request header.
- Secure connections and data transfer.
  - Middleware typically establishes a secure connection from the front-end application to back-end data sources using Transport Layer Security (TSL) or another network security protocol. And it can provide authentication capabilities, challenging front-end application requests for credentials (username and password) or digital certificates.
- Manage traffic dynamically across distributed systems.
  - When application traffic spikes, enterprise middleware can scale to distribute client requests across multiple servers, on premises or in the cloud. And concurrent processing capabilities can prevent problems when multiple clients try to access the same back-end data source simultaneously.

Types of middleware

- **Message-oriented middleware (MOM)**
  - enables application components using different messaging protocols to communicate to exchange messages. In addition to translating - or transforming - messages between applications, MOM manages routing of the messages so they always get to the proper components in the in the proper order. Examples of MOM include message queues and message brokers.
- **Remote procedure call (RPC) middleware**

  - enables one application to trigger a procedure in another application - running on the same computer or on a different computer or network - as if both were part of the same application on the same computer.

- **Data or database middleware**
  - simplifies access to, and interaction with, back-end databases. Typically database middleware is some form of SQL database server.
- **API (application programming interface) middleware**
  - provides tools developers can use to create, expose and manage APIs for their applications - so that other developers can connect to them. Some API middleware includes tools for monetizing APIs - enabling other organizations to use them, at cost. Examples of API middleware include API management platforms, API gateways and API developer portals.
- **Object request broker (ORB) middleware**
  - acts as broker between a request from one application object or component, and the fulfillment of that request by another object or component on the distributed network. ORBs operate with the Common Object Request Broker Architecture (CORBA), which enables one software component to make a request of another without knowing where other is hosted, or what its UI looks like - the “brokering” handles this information during the exchange.
- **Transactional middleware**
  - provides services to support the execution of data transactions across a distributed network. The best-known transactional middleware are transaction processing monitors (TPMs), which ensure that transactions proceed from one step to the next - executing the data exchange, adding/changing/deleting data where needed, etc. - through to completion.
- **Asynchronous data streaming middleware**
  - replicates a data stream in an intermediate store, enabling data sharing between multiple applications. Apache Kafka is one of the best-known examples of middleware for real-time data streaming.
- **Device middleware**
  - provides a focused set of integration and connectivity capabilities for developing apps for a specific mobile OS.
- **Portal middleware**
  - provides tools and resources for integrating content and capabilities from various related applications 'at the glass' - or on a single screen - to create a single, composite application.
- **Robotics middleware**
  - simplifies the process of integrating robotic hardware, firmware and software from multiple manufacturers and locations.

2.6.5 ISV Applications

Independent Software Vendors (ISVs) develop applications that run and expand on an existing solution. IBM works closely with ISVs to ensure applications developed for LinuxONE platforms provide additional value to clients.

To explore the list of Independent Software Vendors (ISVs) for IBM LinuxONE, you can check out the IBM LinuxONE Partner Network (LPN) program. This program helps ISVs easily port, certify, and deploy applications on IBM LinuxONE. ISVs can also gain access to go-to-market resources and a rich set of learning tools for skill development 1.

2.6.6 Databases

IBM LinuxONE supports a variety of databases. IBM LinuxONE servers are designed for secure data serving and can run multiple Linux-based workloads that include Oracle Database 19c, Oracle WebLogic Server, open source, blockchain, and other Linux-based commercial software.

This article gives an overview of IBM LinuxONE support for open source databases like Mongo DB PostgreeSQL and MariaDB.

2.7 Management

Managing an IBM LinuxONE environment can be accomplished using several tools that depend on the virtualization environment. This section covers IBM Operations Manager, z/VM Centralized Manager, and IBM Infrastructure Suite for z/VM and Linux. The available KVM and Container Management tools are also discussed.

2.7.1 z/VM Hypervisor Management

IBM Operations Manager for z/VM

IBM Operations Manager for z/VM supports automated operational monitoring and management of z/VM virtual machines and Linux guests. It can help you address issues before they impact your service level agreements. Systems programmers and administrators can automate routine maintenance tasks in response to system alerts. Users can easily debug problems by viewing and interacting with consoles for service machines and Linux guests. Operators have the ability to better interpret messages and determine corrective actions.

Operations Manager for z/VM benefits:

- Automated response to issues
  - Automated actions can be created to address console messages on z/VM service machines and Linux guests.
- Advanced Monitoring
  - Monitoring for z/VM page space and spool files.
- Automate tasks
- Tasks can be performed at specific times using scheduled events without manual intervention.
- Rule-specific issues checking
  - z/VM and Linux guests are monitored based on pre-defined problem / issues rules you create.

z/VM Centralized Service Management

Managing multiple remote and local systems while coordinating levels of service among these systems can be difficult. New with z/VM 7.2 is a service management tool, z/VM Centralized Service Management (z/VM CSM).

When using Centralized Service Management, one system is designated as the principal system. This system uses the Shared File System (SFS) to manage service levels for a set of defined managed systems, regardless of their geographic location. The new *SERVMGR* command uses Virtual Machine Serviceability Enhancements Staged/Extended (VMSES/E) commands to apply service and local modifications, to build serviced content, and to drive the transport of the packaged service to the managed systems.

IBM Infrastructure Suite for z/VM and Linux

IBM Infrastructure Suite for z/VM and Linux is a single solution that provides multiple tools to monitor and manage z/VM and Linux virtual servers on IBM Z systems as part of both traditional and cloud infrastructure. It supports backup and recovery of the entire system at a reduced price.

The capabilities of IBM Infrastructure Suite for z/VM and Linux provide you with comprehensive insight to efficiently control and support your IBM z/VM and Linux on IBM Z systems environments.

- Products included with IBM Infrastructure Suite for z/VM and Linux:
  - IBM Tivoli® OMEGAMON® XE on z/VM and Linux
  - IBM Spectrum® Protect Extended Edition
  - IBM Operations Manager for z/VM
  - IBM Backup and Restore Manager for z/VM
  - IBM Tape Manager for z/VM (Optional)
  - ICIC - IBM Cloud® Infrastructure Center (Optional)

2.7.2 KVM Hypervisor Management

Various tolls exist for managing KVM Virtual Machines efficiently:

Libvirt:

Libvirt is a toolkit that provides a consistent and stable API for managing various virtualization solutions, including KVM. It offers command-line utilities (virsh) and APIs for management. See: KVM Management Tools.

Virt-manager:

Virt-manager is a desktop application for managing KVM virtual machines. It provides a user-friendly graphical interface to create, view, modify, and manage virtual machines. See: Good GUI for KVM.

Cockpit Web Console:

Cockpit is a web-based server management tool that includes support for managing KVM virtual machines. It offers a user-friendly web interface for VM management. See: Tecmint

WebVirtMgr:

WebVirtMgr is a web-based interface for managing KVM virtual machines. It provides a browser-accessible platform to create and control VMs. See: Reddit

2.7.3 Container Management

Container Management automates the creation, deployment and scaling of containers. It also facilitates the organization of containers. Container management allow systems to work more efficiently specially when the number of containers becomes too large for the IT team to handle. With Container management, IT teams can keep their environment more secure, and developers can explore its flexibility to create and deploy new apps and services.

Docker container management software and Kubernetes container management facilitates the orchestration, security and networking tasks. Container management software systems are available as both open source and proprietary commercial products.

2.7.4 Kubernetes

Kubernetes is an open source platform for managing containerized workloads and services, that facilitates both declarative configuration and automation. Kubernetes is portable, extensible, and has a large and rapidly growing ecosystem. Kubernetes services, support, and tools are widely available.

A list of some Kubernetes features include:

- Auto scaling
- Storage Orchestration
- Self healing
- Multi-Cloud and Hybrid Cloud support
- Load Balancing
- Applications rollout - (up-to-date)
- Community Support

For additional information about Kubernetes, see: https://kubernetes.io/docs/home/

2.7.5 Red Hat OpenShift Container Platform

Red Hat OpenShift Container Platform (RHOCP) is an example of Container Technology. RHOCP is a trusted Kubernetes enterprise platform that supports modern, hybrid-cloud application development and provides a consistent foundation for applications anywhere, across physical, virtual, private, and public clouds.

RHOCP provides an abstraction layer with the same experience regardless of the cloud deployment model (on-prem, public cloud, private cloud and hybrid cloud) and hardware architectures (x86, s390x, ppc64le and arm) and it is used as the foundation for how IBM distributes software in a containerized format.

Figure 2-13 shows the deployment options for RHOCP Container Platform on IBM LinuxONE.

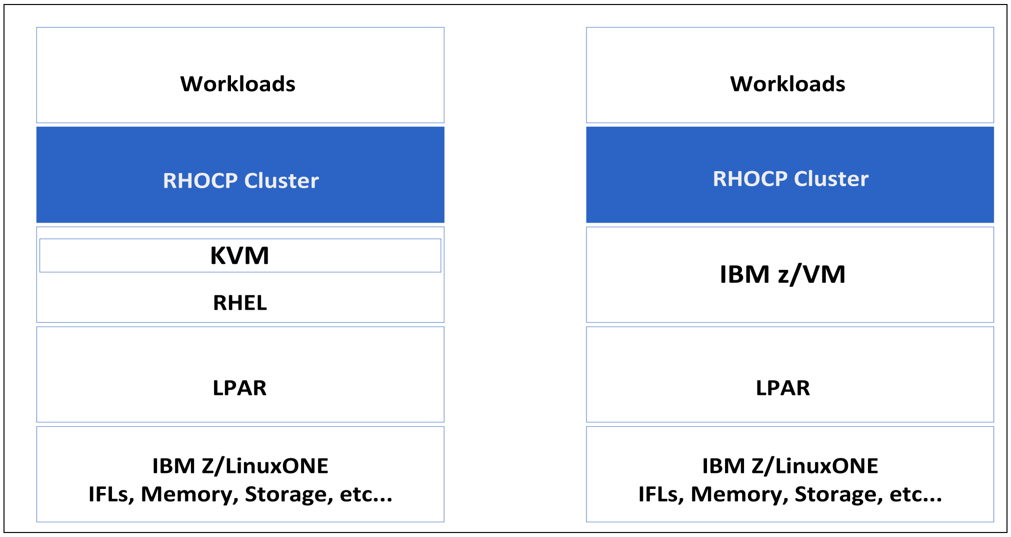

Figure 2-13 Deployment options for Red Hat OpenShift Container Platform on IBM LinuxONE

2.8 Automation

The IBM LinuxONE platform can exploit several Automation tools and processes depending on the chosen virtualization method. This section discusses Data Replication, Copy Services Manager, Automation with Linux HA, Ansible, IBM Cloud Infrastructure Center, and GDPS.

2.8.1 Data Replication

Synchronous data replication

Peer-to-Peer Remote Copy (PPRC) is a hardware solution which provides rapid and accurate disaster recovery as well as a solution to workload movement and device migration. Updates made on the primary DASD volumes are synchronously shadowed to the secondary DASD volumes. The local storage subsystem and the remote storage subsystem are connected through a communications link called a PPRC path. The protocol used to copy data using PPRC is Fibre Channel Protocol.

PPRC is configured in the Storage Subsystems using local administration capabilities. In the case of IBM DS8K, available ways of performing the configuration is: using its Hardware Management Console (HMC), HMC Command Line Interface (CLI) or Copy Services Manager CSM). See 2.8.2, “Copy Services Manager (CSM)”.

2.8.2 Copy Services Manager (CSM)

Copy Services Manager controls copy services in storage environments. Copy services are features that are used by storage systems such as IBM DS8000 to configure, manage, and monitor data-copy functions. Copy services include the point-in-time function - IBM FlashCopy®, Metro Mirror, Global Mirror, and Metro Global Mirror. Copy Services Manager can automate the administration and configuration of these services; and monitor and manage copy sessions.

You can use Copy Services Manager to complete the following data replication tasks and help reduce the downtime of critical applications:

- Plan for replication when you are provisioning storage
- Keep data on multiple related volumes consistent across storage systems for a planned or unplanned outage
- Monitor and track replication operations
- Automate the mapping of source volumes to target volumes

Starting with DS8000 Version 8.1, Copy Services Manager also comes preinstalled on the Hardware Management Console (HMC). Therefore, you can enable the Copy Services Manager software that is already on the hardware system. Doing so results in less setup time; and eliminates the need to maintain a separate server for Copy Services functions.

Copy Services Manager can also run on Linux on IBM Z and uses the Fiber Channel connection (FICON) to connect and to manage Storage Systems' count-key data (CKD) volumes.

2.8.3 Automation with Linux HA components

Using Open Source components, the resiliency of an environment can be automated to recover or switch workloads and guarantee continuous operation.

The major Open source components are Pacemaker and Corosync which are part of:

- RHEL HA
- SUSE HA
- ubuntu HA

Pacemaker, Corosync and additional HA components are listed below:

Pacemaker

Pacemaker is an open source high-availability cluster resource manager software that runs on a set of nodes. Together with Corosync, an open source group communication system that provides ordered communication delivery, cluster membership, quorum enforcement, and other features among the nodes, it helps detect component failures and orchestrate necessary failover procedures to minimize interruptions to applications.

Pacemaker can supervise and recover from failures within a cluster. The components of Pacemaker architecture are:

- **Cluster Information Base (CIB)**
  - CIB is the Pacemaker information daemon. It uses XML to distribute and synchronize current configuration and status from the Designated Coordinator (DC) The DC node is assigned by Pacemaker to store and report cluster state and actions to the other nodes using the CIB. There is a Cluster Information Base in each host instance.
  - The XML list of behaviors, directed by resource manager, informs policy engine
- **Cluster Resource Management daemon (CRMd**)
  - Pacemaker uses this daemon to route cluster resource actions. Resources managed by this daemon can be queried, moved, instantiated and changed when needed.
  - Communicates with Local Resource manager daemon on each cluster
  - The “Local Resource Manager” receives instructions from the CRMd and passes requests along to local resource agents (VirtualDomain, FileSystem, MailTo) general operations
  - Define one “Designated Focal Point” for the cluster

- The other hosts receive data from the **CRMd** via **corosync**
- The “Local Resource Manager” receives instructions from the CRMd and passes requests along to local resource agents (VirtualDomain, FileSystem, MailTo) general operations

- **Shoot the Other Node in the Head (STONITH)**
  - Is a Pacemaker fencing agent that detects if it loses contact with one of the nodes in the cluster.
  - If Pacemaker ”thinks” a node is down, STONITH will force it offline.
  - The Pacemaker fencing is implemented by STONITH.
  - STONITH forcely shuts down and fences nodes removing them from the cluster to maintain data integrity.
- **corosync**
  - corosync is a component/daemon that handles the core membership enrollment and the required communication with the members for cluster high availability for any Linux instance enrolled in an HA clusters.
  - corosync is a required component and a daemon in a Linux HA cluster.
  - Manages quorum rules and determination.
  - starts/stops the virtual machines.

Policy Engine

- Policy Engine is a software component that helps managing policies for clusters. Policy Engine defines what end users can do on a cluster and ensures that clusters can communicate. Any time a Kubernetes object is created, a policy evaluates and validates or mutates the request. Policies can apply across a namespace or different pods with a specific label in the cluster. Kubernetes policy engines block objects that could harm or affect the cluster if they don’t meet the policy’s requirements.
- Takes that list of behaviors and maps them to cluster’s current state

OpenSAF checkpoint APIs

- OpenSAF is a checkpoint service which provides mechanisms for check pointing and restoring an application state. It provides a series of APIs to allow applications registering for checkpoints and to perform restoration of applications.
  - SAF - Service Availability Forum – created the Service Availability Specifications.
  - OpenHPI - The Hardware Platform Interface (HPI) abstracts the differences between hardware implementations, providing a uniform interface to hardware features.
  - OpenAIS - The Application Interface Specification (AIS) specifies an interface that applications interchange information with the service availability middleware (i.e. CRM).

Heartbeat

- The **Heartbeat** of a node in a cluster it’s a signal that is sent between nodes to indicate that they are still alive and functioning properly. It is used to detect when a node has failed and to initiate failover procedures. The Heartbeat, in some cases, is used to monitor the health of a node and to initiate maintenance actions.
  - Uses messaging between nodes to make sure they are alive and available
  - Determines if an action is required when heartbeat stops after certain number of tries

Cluster-glue

- Cluster Glue is a set of libraries, tools and utilities used in the Heartbeat / Pacemaker cluster stack. In essence, Glue are the parts of the cluster stack that don't fit in anywhere else. Basically, it’s everything that is not messaging layer and not resource manager.

Resource-agents

- Resource agents are scripts that allow Pacemaker to manage any service it knows nothing about. They contain the logic for what to do when the cluster wishes to start, stop or check the health of a service.
  - A resource agent is an executable that manages a cluster resource. No formal definition of a cluster resource exists, other than "anything a cluster manages is a resource". Cluster resources can be as diverse as IP addresses, file systems, database services, and entire virtual machines — to name a few examples.
  - Resource agents run in clustered systems or remote.
  - Resource agents are able to start, stop or restart services.
  - Quorum ensures that the majority of a given cluster agrees on what the resource is
  - *Votequorum* as an interface for members to agree.
  - It also provides messaging for applications coordinating / operating across multiple members of a cluster.

2.8.4 Automation with Ansible

Ansible Automation Overview

Ansible is an open-source automation tool that simplifies IT tasks, including configuration management, application deployment, and orchestration. It allows you to automate routine tasks, reducing the risk of human error and improving system resiliency. LinuxONE is designed for high availability and security. Ansible complements these features by automating repetitive tasks, ensuring consistent configurations, and enabling rapid responses to system events.

With Red Hat Ansible Automation Platform running on IBM LinuxONE, we have an enterprise supported solution for automation on various levels and it can span all platfroms in the enterprise.

2.8.5 IBM Cloud Infrastructure Center (ICIC) automation

IBM ICIC is a software platform for managing the infrastructure of private clouds on IBM LinuxONE. The IBM Cloud Infrastructure Center is an IaaS offering that delivers an industry-standard user experience for the IaaS management of non-containerized and containerized workloads IBM LinuxONE.

2.8.6 Resiliency Automation for the infrastructure

With the requirements for Resiliency, different businesses have different requirements for their operational model, such as Business Continuity or Continuous Availability.

- **Business Continuity (BC)** -- Continuously operate and mask *planned* outages from end-users. It employs Non-disruptive hardware and software changes, non-disruptive configuration, software coexistence.

- **Continuous Availability (CA)** -- Deliver non-disruptive service to the end user seven days a week, 24 hours a day (no planned or unplanned outages).

The final goal is to provide Continuous Availability. That means it is very important to have capabilities of problem and failure detection as well as recovery automation processes, in case one of the layers or components are malfunctioning and need recovery actions.

2.8.7 Automation with GDPS

GDPS solutions are generally designed to:

- Deliver end-to-end application and data availability within a site or across multiple sites.
- Automate recovery procedures for near-continuous availability and disaster recovery.
- Provide extremely rapid RPO and RTO.
- Provide multiple, secure, point-in-time copies of production data to help protect from logical corruption threats.
- Monitor systems and storage that support open and IBM copy technology architectures, including IBM, Hitachi Vantara (Metro Mirror only for non-IBM storage).
- Simplify management tasks with an easy-to-use interface and a central point of control.

Multi-site environment

IBM GDPS technology provides a total business continuity solution for multi-site environments. The technologies also provides a collection of end-to-end automated disaster-recovery solutions on the IBM LinuxONE platform, each addressing a different set of IT resiliency goals that can be tailored to meet the recovery objectives for your business.

Depending on the solution, GDPS provides support for:

- Managing and monitoring the remote copy environment.
- Managing and monitoring the production environment.
- Data that spans more than one platform including z/OS, z/VM, KVM and Linux on Z
- Multiple disk subsystem vendors.
- Transparent and near-continuous application availability from disk failures.
- Automated and transparent fail-over and failback.

- GDPS extends the resource sharing, workload balancing, and continuous availability benefits of a parallel used environment. It also significantly enhances the capability of an enterprise to recover from disasters and other failures, and to manage planned exception conditions. As a result, you can achieve your own continuous-availability and disaster-recovery goals.

GDPS is a collection of several offerings, each addressing a different set of IT resiliency goals that can be tailored to meet the RPO and RTO for your business.

Each GDPS offering uses a combination of system and storage hardware or software-based replication and automation, and clustering software technologies:

- **Metro**
- Metro HyperSwap Manager
- **IBM Virtual Appliance**
- **Extended Disaster Recovery (xDR)**

- **Global - GM (also known as GM)**
- **Metro Global - MGM (also known as MGM**)
- Continuous Availability

In this publication we explore the LinuxONE applicable technologies: **Metro**, **Virtual Appliance, Extended Disaster Recovery (xDR), Global - GM and Metro Global - MGM**.

GDPS Metro

A near-CA and DR solution across two sites separated by metropolitan distances. The solution is based on the IBM Metro Mirror synchronous disk mirroring technology

Provides continuous availability, disaster recovery, and production system / sysplex resource management capabilities. Based on Metro Mirror synchronous disk mirroring technology, it can achieve RPO = 0. Typically the recovery time is less than one hour (RTO < 1 hour) following a complete site failure. Metro also supports HyperSwap to provide near-continuous disk availability after a disk failure.

GDPS Global - GM (also known as GM)

A DR solution across two regions that are separated by virtually unlimited distance. The solution is based on the IBM System Storage Global Mirror technology, which is a disk subsystem-based asynchronous form of remote copy.

GDPS Metro Global - MGM (also known as MGM)

A 3-site or a symmetrical 4-site configuration is supported:

- MGM 3-site
  - A 3-site solution that provides CA across two sites within metropolitan distances in one region and DR to a third site, in a second region, at virtually unlimited distances. It is based on a combination of the Metro Mirror and Global Mirror technologies.
- MGM 4-site
  - A symmetrical 4-site solution that is similar to the 3-site solution in that it provides CA within region and DR cross region. In addition, in the 4-site solution, the two regions are configured symmetrical so that the same levels of CA and DR protection is provided, no matter which region production runs in.

GDPS Virtual Appliance (VA)

The GDPS Virtual Appliance is a software appliance that is a self-contained OS / application / middleware / APIs / GUI all rolled into one IBM LinuxONE LPAR software image.

GDPS VA requires LinuxONE configuration to have a Processing Unit (PU) characterized as a Central Processor (CP). This required CP is ordered as a Feature Code and does not incur in any utilization charges.

GDPS VA supports both planned and unplanned situations, which helps to maximize application availability and provide business continuity. In particular, a Virtual Appliance solution can deliver the following capabilities:

- Near-continuous availability solution
- Disaster recovery (DR) solution across metropolitan distances
- Recovery time objective (RTO) less than an hour
- Recovery point objective (RPO) of zero

The Virtual Appliance can manage starting and stopping z/VM or KVM hypervisors, and thus all Linux guests running on the hypervisors. It is a fully integrated software solution for providing continuous availability and disaster recovery (CA/DR) protection to Linux on Z, zVM and KVM. If your Linux on Z production workloads have CA or DR requirements, then the Virtual Appliance can help you meet those requirements.

A Virtual Appliance environment is typically spread across two data centers (Site1 and Site2)[6] where the primary copy of the production disk is normally in Site1. The Appliance must have connectivity to all the Site1 and Site2 primary and secondary devices that it will manage. For availability reasons, the Virtual Appliance runs in Site2 on local disk that is not mirrored with Metro Mirror. This provides failure isolation for the appliance system to ensure that it is not impacted by failures that affect the production systems and remains available to automate any recovery action.

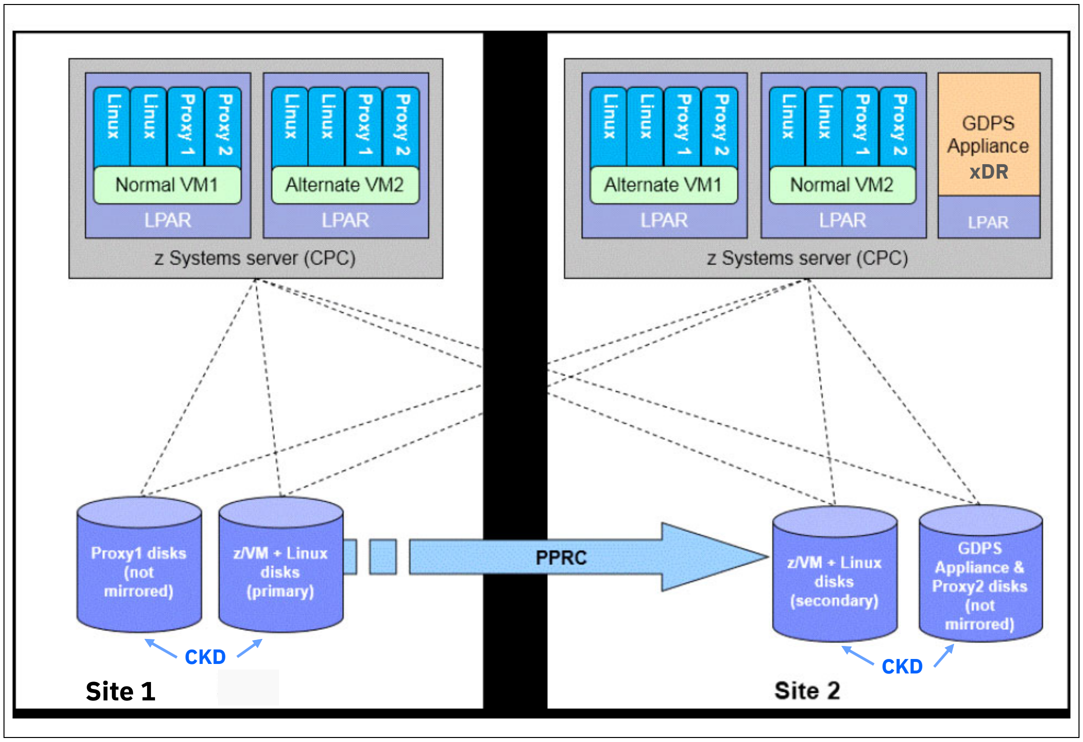

Figure 2-14 Typical GDPS Virtual Appliance Implementation

GDPS Virtual Appliance for LinuxONE

VA provides the following functions:

- Coordinated takeover in unplanned cases e.g. recovery from a node failure
- Coordinated takeover in planned cases for e.g. maintenance
- Coordinated planned and unplanned HyperSwap - (with z/VM)
- Freeze support
- Live Guest relocation with z/VM SSI
- Management of all disks —-> control mirroring
- Management of all LPARs --> start/stop

[6] VA may also be implemented in a two-system configuration installed on a single site.

– Graceful shutdown and start up of z/VM and Linux Guests

VA HyperSwap function

The Virtual Appliance delivers a powerful function known as HyperSwap. HyperSwap provides the ability to swap from using the primary devices in a mirrored configuration to using what had been the secondary devices, transparent to the production systems and applications using these devices.

Without HyperSwap, a transparent disk swap is not possible. All systems using the primary disk would need to be shut down (or might have failed, depending on the nature and scope of the failure) and would have to be re-IPLed using the secondary disks. Disk failures are often a single point of failure for the entire production environment.

xDR - Multiplatform Resiliency with VA

GDPS Virtual Appliance includes the Metro and the xDR capabilities. xDR function extends the near-continuous availability and disaster recovery capabilities provided by Metro to other platforms, or operating systems, running on LinuxONE servers.

xDR is the product that allows to communicate / manage: z/OS Proxy (z/OS systems monoplexes outside of the sysplex), z/VM, KVM, SSC(IDAA) and Linux in an LPAR. To provide these capabilities, they must run System Automation for Multi-platforms (SAMP) with the separetely licensed xDR feature.

The proxy nodes communicate commands from to z/VM, monitor the z/VM environment and communicate status and failure information back to the Virtual Appliance.

xDR for z/VM with VA

In each xDR-managed z/VM system, you must configure two special Linux guests, which are known as the *proxy* guests, as shown in Figure on page 68. One proxy node is configured on Site1 disk and the other is configured on Site 2 disk. The proxies are guests dedicated to providing communication and coordination with the Virtual Appliance. They must run SAMP with the separately licensed xDR feature.

xDR for KVM With VA

Multiplatform Resiliency for KVM uses the xDR protocol to communicate with an xDR KVM proxy to send Libvirt commands; therefore, xDR must be enabled to support KVM.

The xDR KVM Proxy is delivered as a Linux RPM for SLES or RHEL or a DEB package for Ubuntu. The proxy guest serves as the middleware for. It communicates commands from / to KVM, monitors the KVM environment, and communicates status information back to the Metro controlling system.

KVM does not provide a HyperSwap function. However, Metro coordinates planned and unplanned HyperSwap for Linux under z/VM and Linux under KVM CKD disks to maintain data integrity and control the shutdown and re-start in place of the KVM LPARs. For disk or site failures, Metro provides a coordinated Freeze for data consistency on CKD disks across KVM, z/VM and Linux LPAR.

xDR for z/OS - with IBM Z

In a scenario where client already has a z/OS Parallel Sysplex® environment running, Metro provides a function called “Metro Multiplatform Resiliency for IBM Z,” also referred to as cross-platform disaster recovery, or xDR.

This function is especially valuable for customers who share data and storage subsystems between z/OS and Linux z/VM guests on IBM Z or SUSE Linux running native on IBM Z LPARs. For example, an application server running on Linux on IBM Z and a database server running on z/OS.

With a multi-tiered architecture, there is a need to provide a coordinated near Continuous Availability/Disaster Recovery solution for both z/OS and Linux.

GDPS Metro can provide this capability when Linux is running as a z/VM guest or native. Using the HyperSwap function so that the virtual device associated with one real disk can be swapped transparently to another disk, HyperSwap can be used to switch to secondary disk storage subsystems mirrored Metro Mirror. If there is a hard failure of a storage device, coordinates the HyperSwap with z/OS for continuous availability spanning the multi-tiered application. HyperSwap is supported for ECKD and xDR managed FB disk.

For site failures, GDPS invokes the Freeze function for data consistency and rapid application restart, without the need for data recovery. HyperSwap can also be helpful in data migration scenarios to allow applications to migrate to new disk volumes without requiring them to be quiesced.

When using ECKD formatted disk, GDPS Metro can provide the reconfiguration capabilities for the Linux on IBM Z servers and data in the same manner as for z/OS systems and data. To support planned and unplanned outages, these functions have been extended to KVM with GDPS V4.5 and above, which provides the recovery actions such as the following examples:

- Re-IPL in place of failing operating system images.
- Heartbeat checking of Linux guests.
- Disk error detection.
- Data consistency with freeze functions across z/OS and Linux.
- Site takeover/failover of a complete production site.
- Single point of control to manage disk mirroring configurations.
- Coordinated recovery for planned and unplanned events.

Additional support is available for Linux running as a guest under z/VM. This includes:

- Re-IPL in place of failing operating system images.
- Ordered Linux node or cluster start-up and shut-down.
- Coordinated planned and unplanned HyperSwap of disk subsystems, transparent to the operating system images and applications using the disks.
- Transparent disk maintenance and failure recovery with HyperSwap across z/OS and Linux applications.

GDPS Global - MGM (also known as MGM)

GDPS provides disaster recovery and production system / sysplex resource management capabilities. Based on Global Mirror synchronous disk mirroring technology, it can achieve data loss down to 3 seconds (RPO = 3 seconds) and RTO < 1 hour. GDPS Global supports two sites separated by virtually unlimited distances with minimal impact to end-user response times.
Three- and four-site configurations are available such as combining GDPS Metro with Global - GM to form **GDPS MGM**.

MGM is capable of provide:

- Down to zero data loss.

- HyperSwap for disk availability.
- Protection from regional events.
- Minimal impact to end-user response time.

For more information about GDPS offerings, see: *IBM GDPS: An Introduction to Concepts and Capabilities*, SG24-6374.

Table 2.9 below shows the capabilities of various GDPS components applicable to LinuxONE resiliency scenarios discussed in detail in next chapter.

Table 1 GDPS components applicable to LinuxONE

Feature	**GDPS Metro**[a]	**GDPS VA**[b]	**GDPS GM**
Continuous Availability	Yes	Yes	No
Disaster Recovery	Yes	Yes	Yes
CA/DR protection against multiple failures	Yes	No	No
Max Supported distance	200 KM (fiber)	200 KM (fiber)	Virtually unlimited
Span of Control	Both sites	Both sites	Disk: both sites / Recovery site: CBU or LPARs
GDPS scripting	Yes	Yes	Yes
Monitoring, alerting and health checks	Yes	Yes No health checks	Yes
Query Services	Yes	No	Yes
Fixed Block disk	Yes	No	Yes
z/OS equivalent function for Linux on Z	Yes (Linux as z/MV Guest)	Yes (Linux as z/MV Guest)	Yes
GDPS GUI	Yes	Yes	Yes

[a] GDPS xDR is part of GDPS Metro.
[b] GDPS VA includes GDPS Metro capabilities.

2.8.8 Infrastructure placement

By design, the IBM LinuxONE platform is a highly resilient system. Through RAS design principles, the LinuxONE Architecture has built-in self-detection, error correction, and redundancy. The LinuxONE platform reduces single points of failure to deliver the best reliability of any enterprise system in the industry. Transparent processor sparing, and dynamic memory sparing enable concurrent maintenance and seamless scaling without downtime.

A such single-system environment has redundancy in the hardware and defines Resiliency based on the application and service availability.

For Resiliency reasons sensitive enterprises are using the IBM LinuxONE systems in a redundant deployment, in different topologies.

The naming that was used back in the years was diversified in the distributed world now.

- A first deployment with IBM LinuxONE and its corresponding distributed naming is:
  - **Hot – Cold ---- corresponds distributed naming Active – Passive**
- A second deployment with IBM LinuxONE and its corresponding distributed naming is:
  - **Hot – Warm --- that is known in distributed as Active – Idle**
- The most effective deployment for resiliency with IBM LinuxONE and its corresponding distributed naming is:
  - **Hot – Hot -- that is known in a distributed world as Active – Active**

2.8.9 Characteristics of a Hot – Cold (Active – Passive) deployment

In a such deployment we're talking about 2 datacenters with at least one LinuxONE server per datacenter and its associated Storage servers, network infrastructure and services.

The main datacenter, also named herein as PROD datacenter, is the one where main production runs and the DR - Disaster Recovery datacenter is the second datacenter.

The specialty in a such active-passive environment is that clients are acting for cost efficiency to keep the system in the DR datacenter shut down with active Storage servers in PROD and DR for permanent data replication. In case of a disaster – for resiliency – the system in the DR datacenter is activated and PROD is started to run in the DR datacenter.

The storage mirroring can be implemented on Storage server level and can be synchronous replication called Metro Mirror or it can be asynchronous, called Global Mirror.

2.8.10 Characteristics of a Hot – Warm (Active – Idle) deployment

In a such environment the PROD environment is running the entire production workload, but the DR datacenter has a system which is started and has some basic setup active such as hypervisors, replication software for data or special software for automatic fail-over such as IBM. The Storage servers in PROD and DR are setup for replication / mirroring of all data from PROD.

Similar to the first case, storage mirroring can be implemented on Storage server level and can be synchronous replication called Metro Mirror or it can be asynchronous, called Global Mirror.

For Container workloads the mirroring can also be implemented on a logical level if on the DR site, the storage replication software is actively running such as IBM Storage Fusion Data Foundation or IBM Storage Scale servers.

2.8.11 Characteristics of a Hot – Hot (Active – Active) deployment

The most effective resiliency can be reached in an environment were 2 sites are active at the same time and are setup to be able to take over the entire PROD workload whenever one site is failing or suffering a disaster.

For this environment it is necessary to have a Storage infrastructure that is able to share the storage between the 2 sites if the same applications are running at the same time in both sites.

A such scenario can be realized and is dependent on the type of workload that is used, whether it is traditional VM based workload or Container based workloads or even combined.

2.9 Building the related Deployment Model

There are different approaches for creating a deployment model with IBM LinuxONE. The section below presents the best practices for exploiting hardware and virtualization capabilities. We will extend the discussion around these concepts in Chapter 3.

2.9.1 Best Practices for Hardware and Virtualization:

- For a resilient environment, consider using multiple LPARs for the workload
- Dedicate Memory to the LPAR based on the size of the environment planned. When over committing memory in the hypervisor, make sure that you have the proper auxiliary memory available (paging in z/VM, swap in KVM) to avoid out of memory crashes
- Dedicate or share Core capacity according the LPAR weight definition
- Make use of at least 2 Network Cards for port aggregation and Link Aggregation Control Protocol (LACP). LACP requires the ports to be in promiscuous mode, and they must be dedicated to the partition
- Consider using LinuxONE internal networking with Hipersockets or shared network cards
- Make use of at least 2 I/O attachment cards (FICON or FCP)
- For the Virtualization in LPARs consider z/VM or KVM and Virtual Switch topology

2.9.2 Once storage, hardware, network and virtual infrastructure are covered, reflect on the workload availability

- Fail-over / DR partitions:
  - for when your entire partition needs to reappear in another location
- Mind volume access and crypto / key access
- VM duplication – having Virtual Machines (VM) or containers available to run workload in an emergency
  - Doesn’t avoid an outage, but having the spare VM may shorten the Mean Time to Repair (MTTR)
- “Dead guest relocation”
  - (shutdown and bring-up may be faster than guest mobility)

3

Deployment Scenarios for Resiliency

This chapter discusses possible LinuxONE implementation scenarios, beginning with a single site, and two Storage Subsystems.

Next, we expand the implementation to a two and three site scenarios. We also explore each scenario capabilities as well as possible variations in terms of their implemented resources.

This chapter contains the following topics:

LinuxONE Deployment models

3.1 Reliable base

3.1.1 Reliable base scenario details

This scenario is designed as the “starter” version of resiliency, with focus on vertical scaling. It contains:

- One Single Physical Site.
- One LinuxONE Systems.
- One Storage Subsystem.
- z/VM as hypervisor and ECKD[1] disks with HyperPAV.
- One set of software.

As shown in Figure 3-1, in this proposed scenario we have one IBM LinuxONE system running on a single physical site. The LinunxONE CPC is running z/VM with multiple Linux guests to support the workload. LinuxONE CPC is configured to use Single System Image (SSI) and IBM Operations Manager for z/VM.

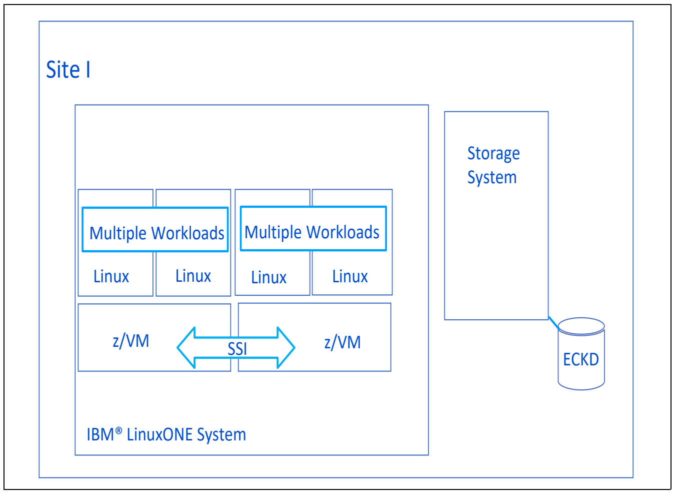

Figure 3-1 Reliable base implementation view

3.1.2 Resiliency level of the Reliable Base scenario - Summary

The workloads are running on a single LinuxONE machine:

- **Characteristics:**
  - Storage hardware and LinuxONE hardware failures not addressed by the RAS capabilities are both Single Points of Failure (SPOF).
    - Storage failover not available.
  - All the workloads will not run until and eventual LinuxONE hardware failure is addressed.
  - Recovery tasks are performed manually, the outage will likely extend to hours.
    - Inconsistent systems management.

[1] In this Chapter, ECKD- Extended Count Key Data and CKD- Count Key Data, are used interchangeably.

- **Benefits:**
  - Relies only on LinuxONE platform RAS (Reliability, Serviceability and Serviceability)
- **Improvements:**
  - Resiliency for a single IBM LinuxONE platform can be enhanced by using a clustering technology with data sharing across two or more Linux images.
  - Introduce an additional Store Subsystem to extend access to data in case of a hardware failure
  - Synchronous Copy with Peer to Peer Remote Copy (PPRC) and Hyperswap would improve recovery time and resiliency.
  - Develop / implement processes for handling problem and change management
  - Review or plan application and processes upgrades to ensure rapid failover can be accomplished
  - Implement / Exploit z/VM SSI and z/VM Operations Manager

3.2 Flexible site

3.2.1 Flexible site scenario details

This scenario expands upon the Reliable base by adding in redundancy for compute, storage, and network inside the single site. This allows for greater control of planned outages. It contains:

- One Single Physical Site.
- Two LinuxONE Systems.
- Two (or more) Storage Subsystems.
- z/VM as hypervisor and ECKD disks with HyperPAV.
- One set of software per system.
- z/VM Single System Image.

As shown in Figure 3-2 on page 78, in this proposed scenario we have two IBM LinuxONE systems running on a single physical site. The LinuxONE CPCs are running z/VM with multiple Linux guests to support the workload. LinuxONE CPCs are configured to use z/VM Single System Image (SSI) and IBM Operations Manager for z/VM.

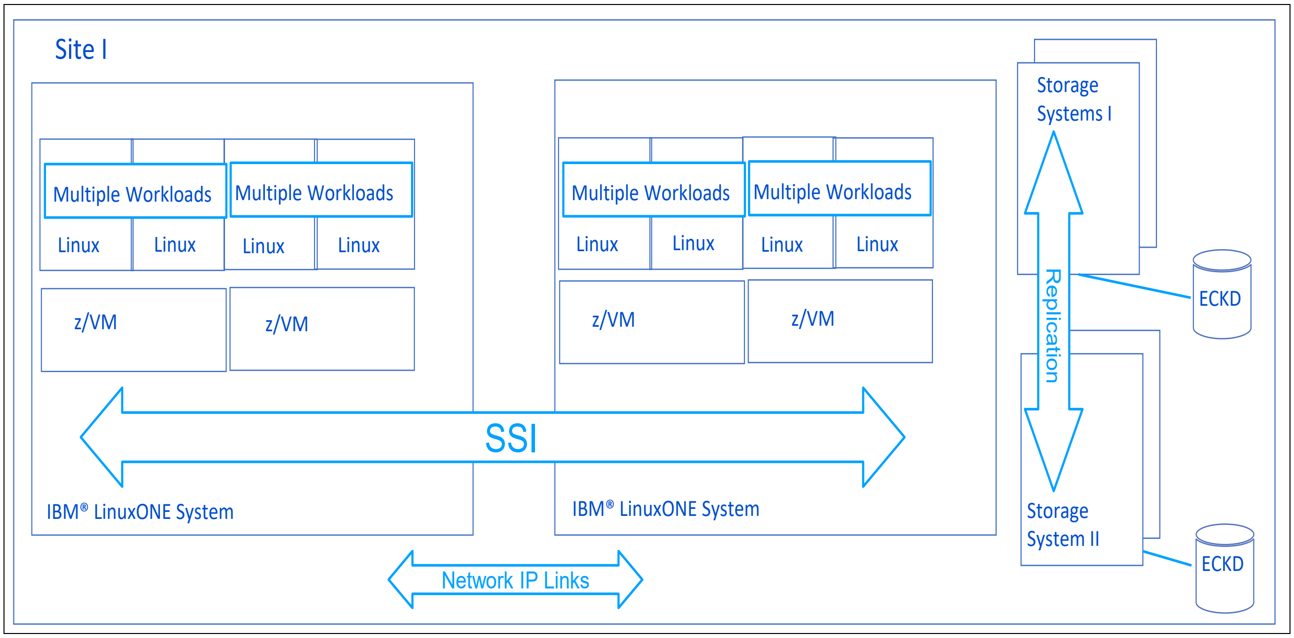

Figure 3-2 Flexible site implementation view

Note that running your workload on a single site constitutes a single point of failure. Any disturbance in the electrical facilities or in any other infrastructure capabilities at the site, will cause an unplanned outage.

The benefits this scenario brings are related to the fact that your data is fully replicated between the Storage Subsystems using synchronous data replication, which gets you covered in case of a Subsystem Storage malfunction, see 3.2.3, “Synchronous data replication” on page 80, and z/VM SSI which allows live migration of linux guests between the four z/VM LPARs.

Also, having two LinuxONE systems running the workload, allows for an extended resiliency. In case of a hardware or software failure in one of the footprints, the remaining active system can eventually run all the workload. The impact of a system failure will depend on how the workloads are split between both systems, and on the ability of the survival system to run them. Client can use Capacity on Demand or Flexible Capacity for Cyber Resiliency capabilities to bring the systems up to the required capacity to run the workload in one of the footprints.

In a more elaborated example shown in Figure 3-3 on page 79, we have four z/VMs and one Linux on Z LPARs defined, and three workload groups. One Kubernetes cluster group formed by partitions I, II and III while LPARs IV holds a Web Server, and partition V runs a Database workload. The Storage Subsystems data is being replicated with synchronous replication, eliminating the storage single point of failure. In the example, z/VM is running on Logical Partitions I, II, III and IV, while Logical Partitions V is running Linux on Z.

Note that the LPARs configuration is “mirrored” on both LinuxONE systems to share the workload and to facilitate the recovery process in a rare case of a system hardware or software failure.

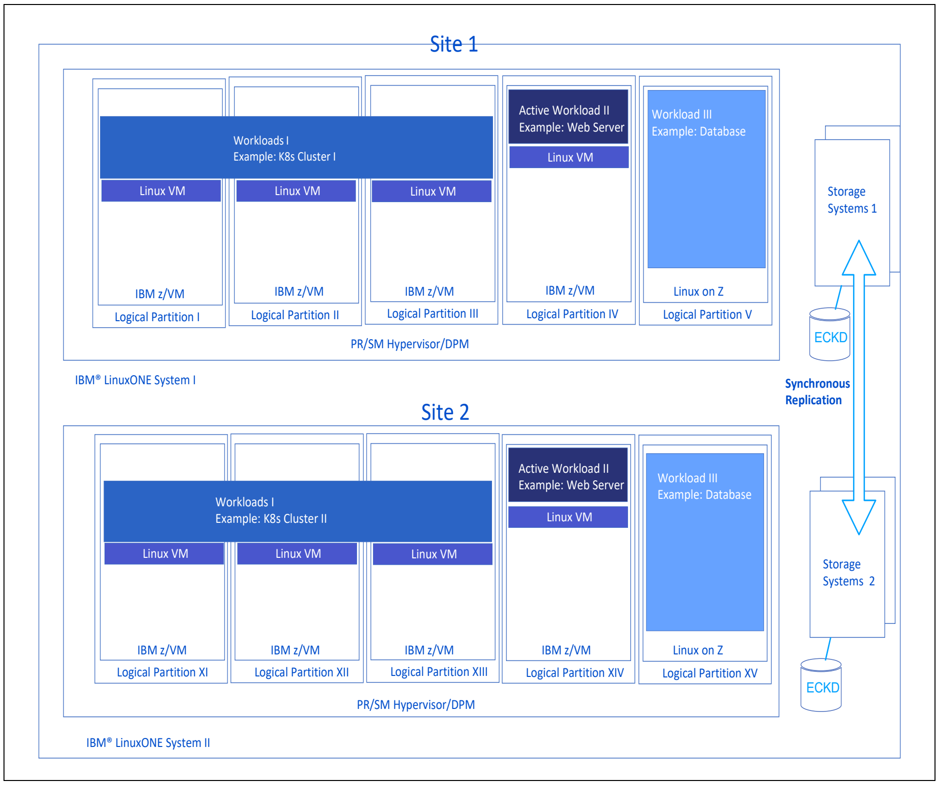

Figure 3-3 Flexible site reliable base detailed implementation example

3.2.2 z/VM PAV and HyperPAV

If the DASD subsystem supports PAV or HyperPAV, z/VM can use a real volume's and alias devices to run more than one guest I/O operation at a time to the volume. When z/VM uses alias devices, the guests' I/O operations tend not to interleave or block each other. In turn, the guests' I/O operations tend to experience reduced real-volume I/O queuing and reduced response time. Reduced response time yields guest applications that run more quickly.

For performance reasons, in this scenario we opted to use HyperPAV technology which allows concurrent I/O operations to a single disk volume.

Figure 3-4 on page 80 shows how HyperPAV is exploited by z/VM. Multiple I/O operations are initiated by the z/VM Guests and handled by the z/VM I/O Subsystems who is responsible for the selection of the Base device address and the available Aliases allowing parallel simultaneous operations to the same volume to take place. z/VM Control Program (CP) I/O Subsystem drives real concurrent Start SubChannel (SSCH) I/O operations to the Alias devices.

Both the base and the aliases devices addresses must be in the same subchannel set (SS) to be properly recognized by z/VM. Please refer to documentation at:
https://www.ibm.com/docs/en/zvm/7.3?topic=administration-multiple-subchannel-set-support

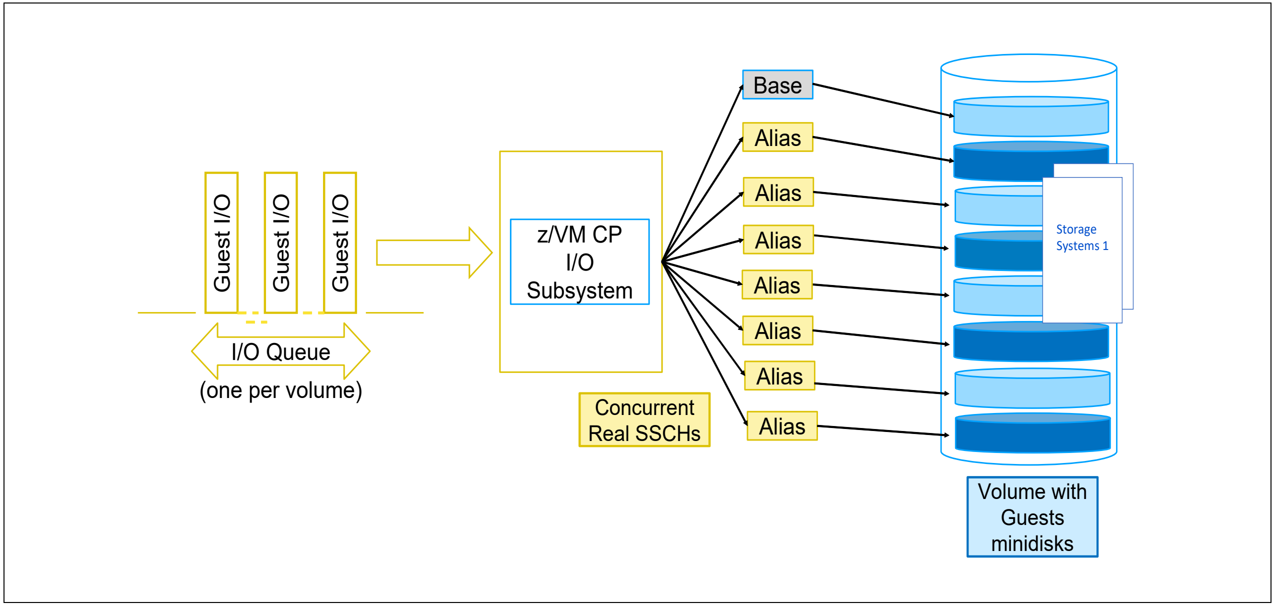

Figure 3-4 Multiple simultaneous disk operations using z/VM HyperPAV

3.2.3 Synchronous data replication

In the scenario depicted in Figure 3-2 on page 78 we show two Storage Subsystems. The two Storage subsystems are configured to use Peer to Peer Remote Copy (PPRC).

PPRC is a hardware solution which provides rapid and accurate disaster recovery as well as a solution to workload movement and device migration. Updates made on the primary DASD volumes are synchronously shadowed to the secondary DASD volumes. The local storage subsystem and the remote storage subsystem are connected through a communications link called a PPRC path. The protocol used to copy data using PPRC is Fibre Channel Protocol. PPRC is configured in the Storage Subsystems using local administration capabilities. In the case of IBM DS8K, one way of performing the configuration is using its Hot-wire Management Console (HMC) or using its Command Line Interface (CLI).

3.2.4 Additional Flexible site scenario considerations

In this scenario we used one Linux on Z partition and four z/VM Hypervisor partitions per LinuxONE system. Our considerations were explored based on the use of z/VM as our Hypervisor.

Using KVM instead of z/VM

If using KVM with ECKD disks, PAV or HyperPAV features can be enabled in your Storage Subsystem. It assigns unique IDs to its DASDs and manages the Alias devices. The following publication describes how to configure, prepare, and work with DASDs: Device Drivers, Features, and Commands, SC33-8411.

Using FBA[2] instead of ECKD

[2] In this publication the terms FB (Fixed Block) and FBA (Fixed Block Assignment) are used interchangeably.

Fixed-block architecture (FBA) is an IBM term for the hard disk drive layout in which each addressable block on the disk has the same size, utilizing 4 byte block numbers and a new set of command codes.

Besides supporting the emulation of CKD volumes, the DS8000 series Storage Server (for instance, and other vendors' storage servers) support the definition of FBA volumes. These are known as Logical Unit Numbers (LUNs) in a Storage Area Network (SAN).

Extended Count Key Data (ECKD) is a direct-access storage device (DASD) data recording format introduced in 1964, by IBM with its IBM System/360 and still being emulated today. It is a self-defining format with each data record represented by a Count Area that identifies the record and provides the number of bytes in an optional Key Area and an optional Data Area. ECKD blocks are usually addressed by CCHHR (CC=Cylinder; HH=Head <i.e. Track>; R = Record <i.e. Block>). This is in contrast to devices using fixed sector size or a separate format track.

As mentioned above, note that internally, all modern storage servers use FBA. E.g. the DS8000 storage servers emulate CKD DASD volumes but the underlying technology is all FBA.

If using KVM with FBA disks then IBM Storage Scale for clustering can be an alternative.IBM Storage Scale, based on technology from IBM General Parallel File System (herein referred to as IBM Storage Scale or GPFS), is a high performance shared-disk file management solution that provides fast, reliable access to data from multiple servers.

Applications can readily access files using standard file system interfaces, and the same file can be accessed concurrently from multiple servers and protocols. IBM Storage Scale is designed to provide high availability through advanced clustering technologies, dynamic file system management, and data replication. IBM Storage Scale can continue to provide data access even when the cluster experiences storage or server malfunctions.

Network

Network capabilities of the LinuxONE platform are discussed in 2.5, "LinuxONE Virtual Network" on page 51.

3.2.5 Resiliency level of the Flexible site scenario - Summary

The workload is running on a single site with multi LinuxONE machines

- **Characteristics:**
  - All the workloads can be moved between systems in case of hardware maintenance or outage.
  - Storage and Network failover not automated.
  - Manual recovery in the event of an outage.
  - Risk of entire datacenter outage.
- **Benefits:**
  - Data on disk being replicated between the two Storage Subsystems.
  - Hardware and Software are not Single Points of Failure (SPOF).
- **Improvements:**
  - Introduce GDPS to improve automation in case of a failure or system outage.
  - Exploit Hyperswap to improve recovery time.
  - Improve processes for handling problem and change management.

- Review, plan and test applications and processes for changes and upgrades to ensure a rapid failover can be accomplished.
- Implement / Exploit z/VM SSI and z/VM Operations Manager.

3.3 Multi-site resiliency

This category introduces one LinuxONE CPC and associated storage into a second data center, removing the limitation of physical site recovery from points of failure. This step is necessary on the journey to high availability and disaster recovery.

3.3.1 Multi-site scenario details

In the Multi-site scenario examples that follow we will use the following characteristics:

- Two Physical Sites in a Metro distance (up to 300 km with DS8870 Metro MIrror).
- One or more LinuxONE Systems per site.
- Two or more Storage Subsystems per site with synchronous replication.
  - Local and/or between sites replication.
  - z/VM as hypervisor and ECKD disks on both sites with HyperPAV.

Multi-Site resiliency scenario is basically the Flexible Site scenario with components replicated to a secondary site. Figure 3-5 shows the two sites basic components and Storage Subsystems data being replicated synchronously between the sites.

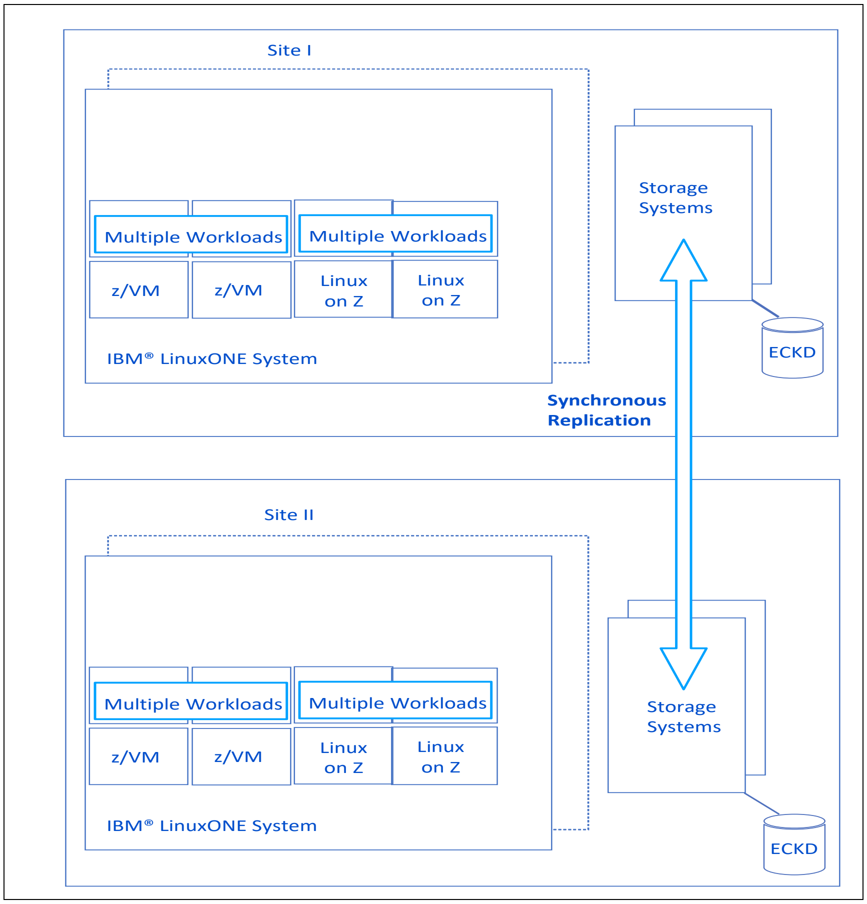

Figure 3-5 Multi-site Resilient® Scenario components

3.3.2 Revisiting Multi-Site applicable GDPS capabilities

GDPS VA

GDPS VA is exclusive to IBM LinuxONE. It includes GDPS Metro and GDPS xDR capabilities which are used in Scenario 2 Hot / Warm and Hot / Hot examples. See “GDPS Virtual Appliance (VA)” on page 67.

GDPS Metro

GDPS Metro also has the capability to manage the Multi-Target Metro Mirror configuration, extending PPRC management and HyperSwap capabilities to support two synchronous Metro Mirror relationships from a single primary volume. Each leg is tracked and managed independently. This provides additional data protection in the event of a disk subsystem failure or local disaster scenario. When using ECKD formatted disk, GDPS Metro can provide the reconfiguration capabilities for z/VM and its guests as well as Linux on IBM Z servers and data. To support planned and unplanned outages these functions have been extended to KVM on LinuxONE starting with GDPS V4.1.

GDPS Metro provides the following recovery actions:

- Re-IPL in place of failing operating system images.
- z/VM Live Guest Relocation management.
- Manage z/VM LPARs and z/VM guests, including Linux on Z.
- Heartbeat checking of Linux guests.
- Disk error detection.
- Data consistency with freeze functions across Linux.
- Site takeover/failover of a complete production site.
- Single point of control to manage disk mirroring configurations.
- Coordinated recovery for planned and unplanned events.

Additional support is available for Linux running as a guest under z/VM. This includes:

- Re-IPL in place of failing operating system images.
- Ordered Linux node or cluster start-up and shut-down.
- Coordinated planned and unplanned HyperSwap of disk subsystems, transparent to the operating system images and applications using the disks.
- Transparent disk maintenance and failure recovery with HyperSwap across Linux applications.

GDPS xDR

GDPS Metro provides management extensions for heterogeneous platforms (GDPS xDR) to be able to fully manage either z/VM systems and their guests, KVM and its guests, and Linux native running on LinuxONE environments, providing full end-to-end support including disk management with freeze and planned/unplanned HyperSwap support, systems management and monitoring. This support also applies to z/VM and KVM on FBA formatted disk. See GDPS xDR topics starting with “xDR for z/VM with VA” on page 69.

3.3.3 Multi-Site possible configurations

Because the LinuxONE CPCs and the Subsystem Storage are placed in two different physical sites, the reliability and availability is much improved compared to the previous scenarios. This is also a basic required scenario for Disaster Recovery (DR).

There are three configuration possibilities for setting up this two sites scenarios:

1. **Hot / Cold** - (active - passive):

   - refer to 2.8.9, “Characteristics of a Hot – Cold (Active – Passive) deployment” on page 72 for a description of this setup.
2. **Hot / Warm** - (active - idle):
   - refer to 2.8.10, “Characteristics of a Hot – Warm (Active – Idle) deployment” on page 72 for a description of this setup.
3. **Hot / Hot** - (active / active):
   - refer to 2.8.11, “Characteristics of a Hot – Hot (Active – Active) deployment” on page 72 for a description of this configuration.

3.3.4 Hot / Cold - (active - passive) configuration

In a Hot / Cold configuration, the secondary (DR) site can be a “dark” or ”dormant” site ready to be brought up in case the primary site fails. Data is being replicated between the sites.

Workload is not split and each site is configured to handle all operations. Manual intervention is normally required to activate additional resources and the partitions in the cold (passive site). Because of that, cold environment, often used in a DR situation, needs longer to get activated.

The LinuxONE CPC on the secondary site might have a temporary capacity record installed such as On-off-Capacity-on Demand (OOCoD), Capacity Back Up (CBU) or a Flexible Capacity for Cyber Resiliency, ready to be activated manually by the operations staff. See , “Capacity on Demand - (Temporary Upgrades)” on page 36

The objective of the temporary capacity records listed above is to bring the secondary site capacity to a level that would allow the partitions and workloads that were previously running on the primary site to be executed on the secondary site without any impact to the end-user or overall system performance. For more information about temporary capacity, refer to Capacity on Demand User’s Guide.

Figure 3-6 on page 85 shows a more elaborated example of an active - passive two sites implementation. Note that in this scenario the LPARS on Site 2 are defined but not activated. This is due to the fact that the total resources required by the defined LPARs, which are the same as the resources used by the primary site, may not be available till a Temporary Capacity record is activated. Activating a temporary capacity record in this scenario requires manual intervention using the LinuxONE HMC. Due to these manual interventions characteristics, the Recovery Time Objective (RTO) can be significantly increased.

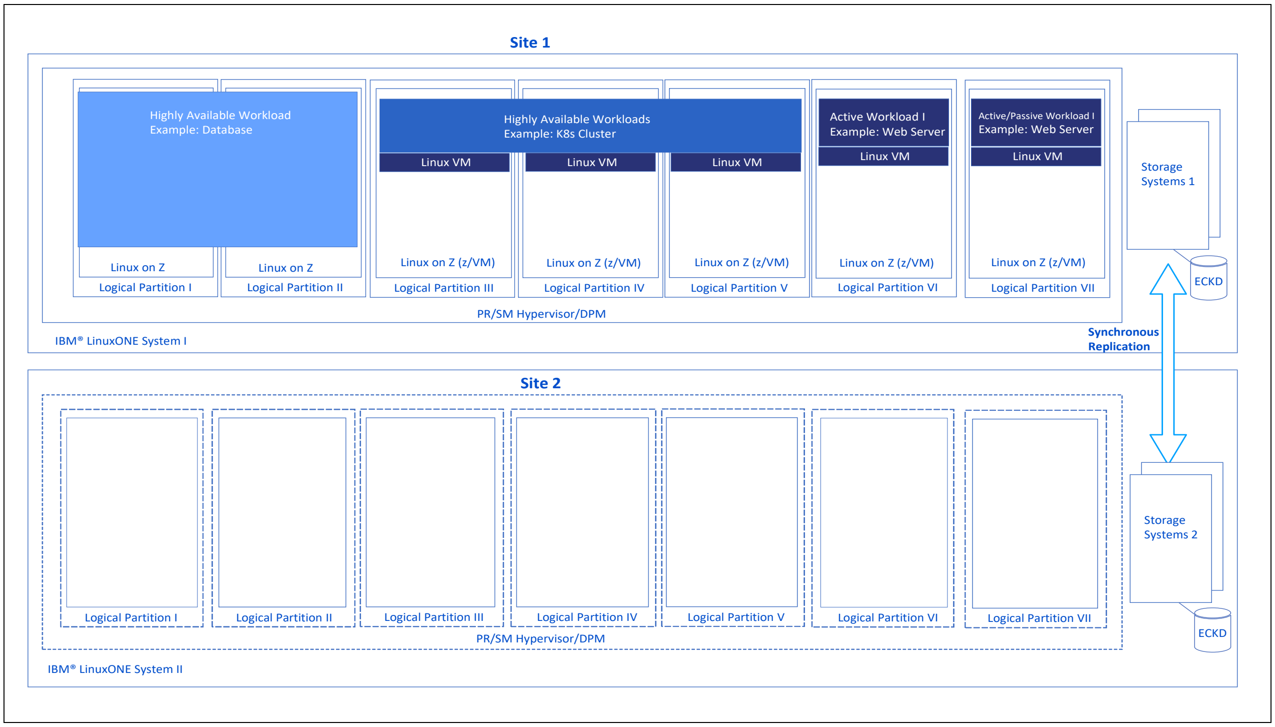

Figure 3-6 Hot - Cold (Active - Passive) two site implementation example

3.3.5 Hot / Warm - (active - idle) configuration

In a hot - warm (active - idle) environment, Site 1 is running the entire production workload, and Site 2 (DR) datacenter has a system which is started (powered up) and has some basic setup active such as hypervisors; support for replication of data or special software for automatic fail-over such as IBM GDPS VA. Site 2 has all the necessary resources to run the Site 1 workload. However all resources are in a “idle” state. Normally the Storage Subsystem servers in Site I and Site II are setup for replication / mirroring of all data from production.

In an event of Site 1 failure, Site 2 needs to be activated manually, via GDPS-VA automation or using user defined scripts.

Figure 3-7 on page 86 shows an implementation of a Hot / Warm (active - idle) environment.

The GDPS Virtual Appliance is a software appliance, meaning it is a self-contained OS/application/middleware/API/GUI, all rolled into one IBM LinuxONE LPAR software image. The GDPS Virtual Appliance can manage starting and stopping z/VM or KVM hypervisors, and thus all Linux guests running on these hypervisors. The GDPS Virtual Appliance also manages disk replication direction between your sites as shown in Figure 3-9 on page 87. The “sites” shown can be data centers separated by up to 124 miles (200 km), or different racks only separated by a few feet in the same datacenter. For disaster recovery though, distance between sites is an important item to be considered once it can save your business should there be a city or region-wide disaster.

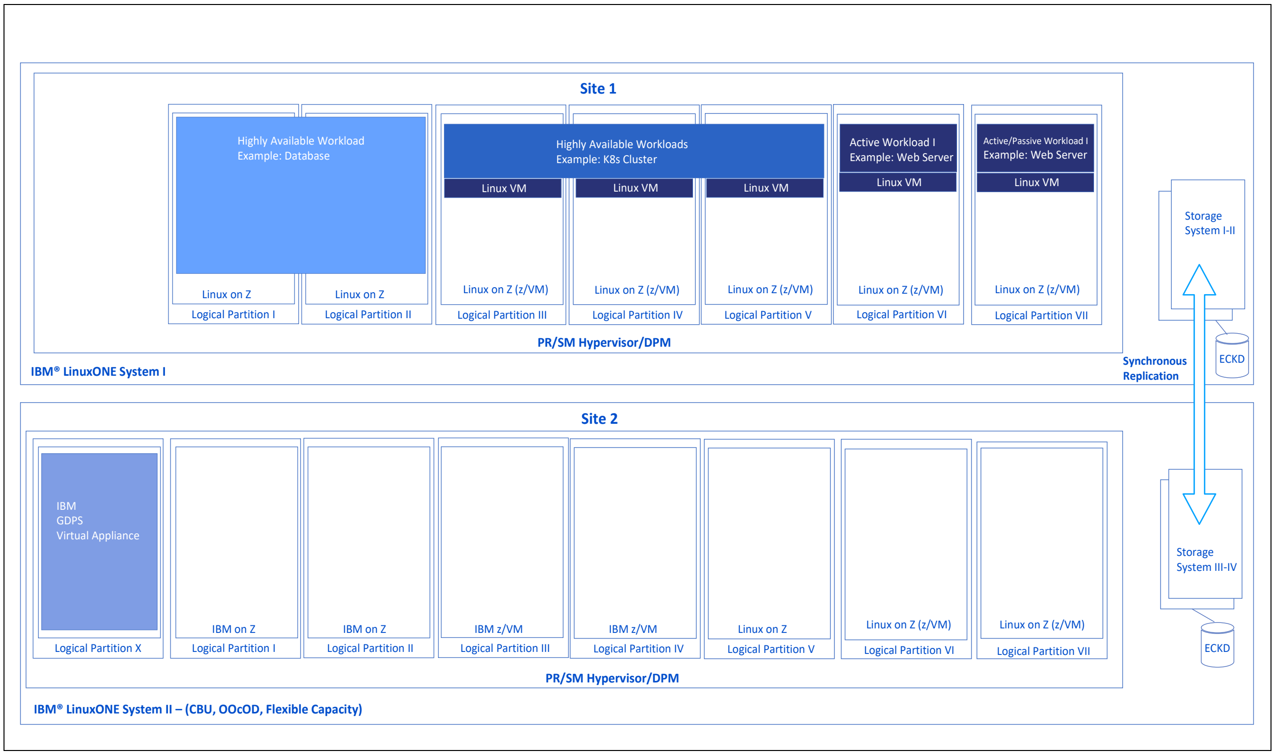

Figure 3-7 Hot - Warm (Active - Idle) two site implementation example

3.3.6 Hot / Hot - (active - active) configuration

In terms of resiliency, the most effective solution can be reached in an environment where the two sites are active at the same time, and they are configured to be able to take over the entire production environment workload whenever one of the sites fails or a disaster strikes. During normal operation, excess capacity at each site is consumed by lower priority work like development or test activities. Also, each site is configured with sufficient capacity to handle normal operations. In a failover situation, additional resources can be enabled and the entire workload runs on a single site.

Figure 3-8 on page 87 shows a more detailed two sites (active-active) implementation. The two Storage Controllers are synchronously replicated. This allows the sites to become the production site without compromising the information written (and mirrored) to the disks in both controllers in case one of the sites goes down. In a normal situation (both sites up) the workload can be shared between the CPCs on both sites.

Besides showing the workloads running on both sites, we added one partition (Logical Partition XII) that runs GDPS VA. For details about GDPS VA, see “GDPS Virtual Appliance (VA)” on page 67. Also note that on each z/VM image in Site 1 we added an xDR proxy LPAR. The xDR proxies are guests dedicated to provide communication and coordination between the z/VM and GDPS Virtual Appliance.

With active/active configurations you might have a router and a load balancer in front of the cluster to balance and split the incoming requests among the active nodes in the cluster. When a system fails, its service workload is migrated to an active node. When one active member fails, the resources are still running on the other active members, and the new incoming service is uninterrupted. Systems must have sufficient hardware resources to handle extra work in case of an outage of one system; or work must be prioritized and service restricted in case of a system failure.

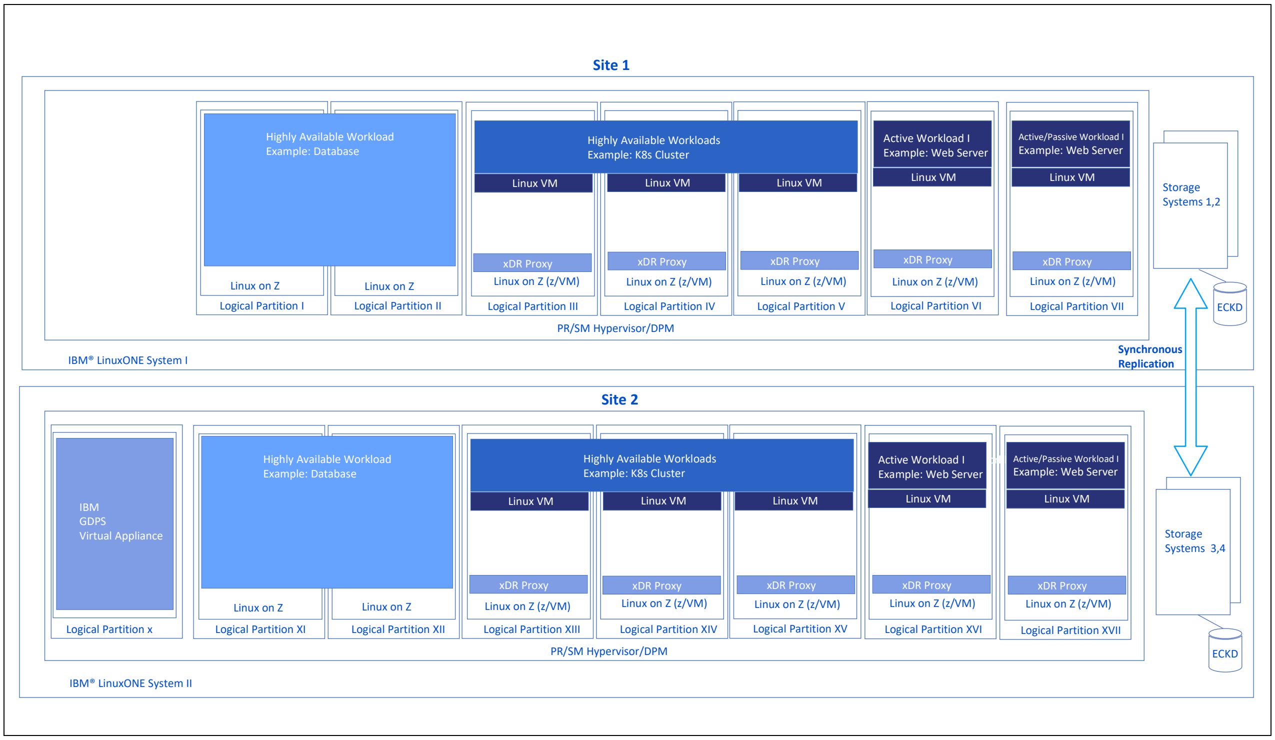

Figure 3-8 Two sites in an Active-Active configuration implementation example

Also, based on the Figure 3-8, note that GDPS VA partition is recommended to be placed in the secondary (or DR) site. The xDR Proxies partitions are present on both sites and on each z/VM partition.

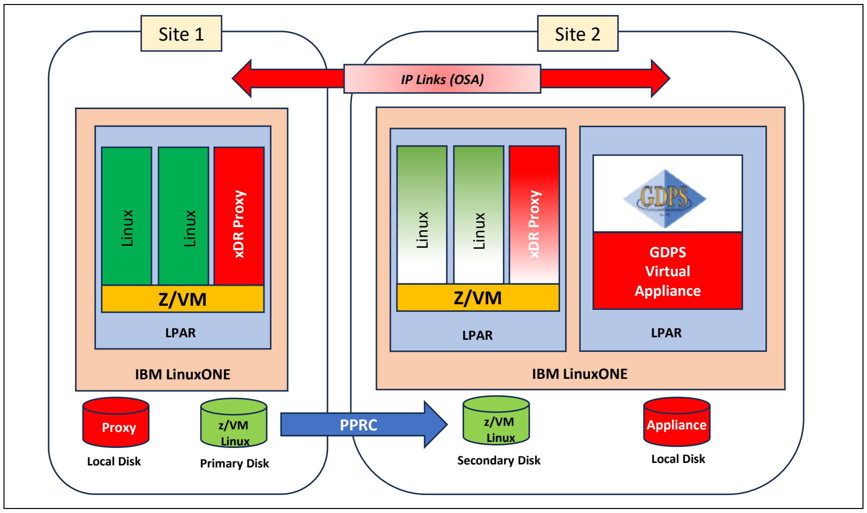

Figure 3-9 GDPS VA basic implementation diagram

With two sites, the exploitation of GDPS Virtual Appliance (VA) and GDPS Metro MultiPlatform Resiliency for IBM Z (GDPS xDR) can be configured to automate the process of transferring the workloads, swapping Storage Subsystems disk access, activating temporary records, restarting hypervisors such as z/VM and z/KVM and their respective guests, in order to keep production running in case of a site unexpected failure, disaster or planned and unplanned outage.

3.3.7 Implementing z/VM Single System Image (SSI)

z/VM SSI is discussed in section Figure 2-7 on page 46. SSI enhances the z/VM systems management, communications, disk management, device mapping, virtual machine definition management, installation, and service functions to enable multiple z/VM systems to share and coordinate resources within a Single System Image structure. This combination of enhanced functions provides the foundation that enables Live Guest Relocations (LGR), which is the ability for a Linux guest to be moved from one z/VM system to another within the SSI cluster.

z/VM SSI is applicable to both scenarios presented so far. It applies to a single site with one or more CPCs (Scenario 1), as well as to a dual site (Scenario 2) in a metro distance. In a single site or dual site. SSI requires communication between the z/VM images which is normally provided by Channel-to-Channel (CTC) connections.

3.3.8 HyperSwap Function

HyperSwap, available with the GDPS Virtual Appliance, provides the ability to transparently swap disks between two sites. The power of automation allows you to test and perfect the actions to be taken, either for planned or unplanned changes, thus minimizing or eliminating the risk of human error.

This is one of the offerings in the GDPS family, along with GDPS Metro, that offers the potential of zero data loss, and that can achieve the shortest recovery time objective, typically less than one hour following a complete site failure.

HyperSwap is also one of the only members of the GDPS family, again along with GDPS Metro, that is based on hardware replication and that provides the capability to manage the production LPARs.

3.3.9 PAV, HyperPAV and HyperSwap in z/VM

In a z/VM environment you can use HyperPav **and** HyperSwap but then HyperPavs need to be attached to system and not dedicated to a guest, This means that you need to use any guest as a non-exploiting operating system that is not configured to control the features of HyperPAV or has no knowledge of the HyperPAV architecture.

There is a trade-off between overall DASD I/O subsystems performance and the ability to perform a quick, dynamic and automated recovery process for a Storage Subsystem eventual failure.

- PAV and HyperPAV will benefit performance allowing multiple I/Os to be performed concurrently against one disk volume.
- HyperSwap will improve recovery time in case of a failure in the data replication process or even in a Storage Subsystem hardware incident.

With z/VM, use of virtual alias devices (PAV or HyperPAV) in a guest, whether dedicated or virtual, is considered an unsupported configuration for Hyperswap.

3.3.10 Resiliency level of Multi-site scenarios - Summary

- The workloads are running on a two (or more) LinuxONE machines at different sites.
  - **Characteristics:**
    - Requires higher investment in equipment and infrastructure.

   - Active-active requires that machines at both sites have extra capacity or temporary capacity available to be activated allowing them to support all workloads.
   - Exploits GDPS VA, GDPS xDR and GDPS Metro and HyperSwap.
- **Benefits:**
   - Provides near-continuous availability and continuous operations (active-active).
   - Minimal impact due to system and data outages - Redundancy in software, hardware and data.
   - Full data replication between sites and storage subsystems.
   - GDPS VA GUI allows configuration and monitoring of the environment without requiring additional operations skills.
   - With GDPS-VA and the selected scenario implementation, active-cold, active-warm or active-active, the failover process can be automated using service management processes.
   - In a failover scenario, impact can be minutes to hours depending on the level of detection and automation implemented.
- **Improvements:**
   - Proper workload distribution and routing between sites and resources (active-active).
   - Test and exercise all possible failure DR situations.
   - Determine the levels and frequency of maintenance of all installed equipment.
   - Improve processes for handling change and automated problem management.
   - Review or plan application updates to ensure rapid failover can be accomplished.
   - Continuous Availability.

3.4 Continuous Availability

3.4.1 Continuous Availability scenario details

In this example we will exploit the following characteristics:

- Multiple Sites in a Metro and Global distance.
- One or more LinuxONE Systems per site.
- Two or more Storage Subsystems with synchronous and asynchronous replication.
   - Local and/or remote replication between sites.
- z/VM as hypervisor and CKD disks in all sites with HyperPAV.
- KVM as hypervisor and CKD disks in all sites.
- Linux in LPAR with CKD disks.
- z/OS partition which is required for Global Mirror implementation.
- GDPS xDR.

3.4.2 Best Fit

The best fit for this implementation would be current IBM Z clients already running in a z/OS Parallel Sysplex environment, exploiting GDPS capabilities and willing to extend these capabilities to their workloads which are not running on their current IBM Z platform. Bring their distributed workloads to the IBM Z platform is not a difficult task and depending on the GDPS licenses they already own and explore, they might need just to add the GDPS xDR - Multiplatform Resiliency for their IBM Z systems GDPS licenses. Figure 3-10 on page 90 shows an environment where IBM Z is running z/OS in a Parallel Sysplex environment along with distributed Linux in LPAR, Native Linux, z/VM and KVM Linux guests running on LinuxONE servers. LinuxONE servers do not support running z/OS.

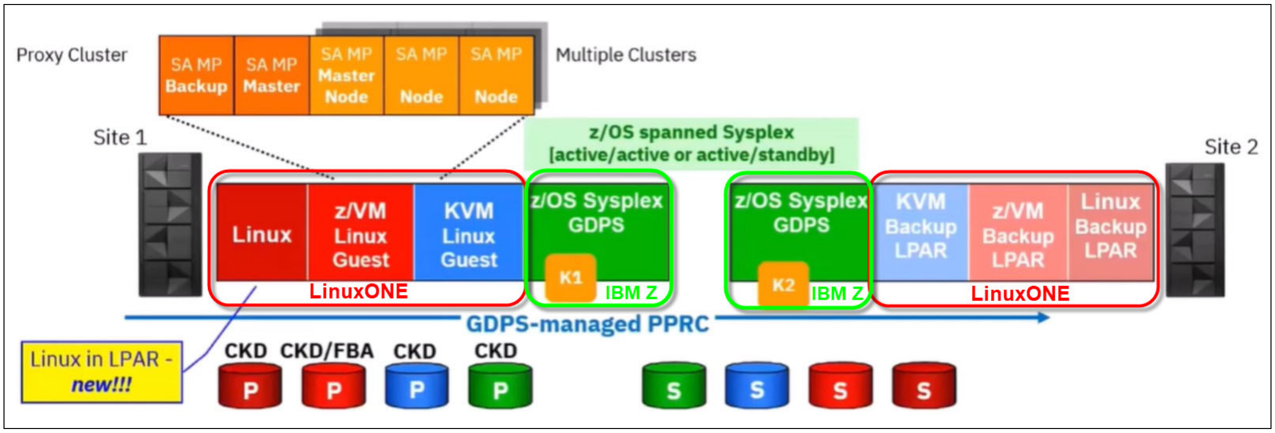

Figure 3-10 Example of two IBM Z systems running z/OS Sysplex, Linux on Z, KVM and z/VM

3.4.3 Revisiting GDPS Continuous Availability applicable capabilities

GDPS Metro

GDPS Metro also has the capability to manage the Multi-Target Metro Mirror configuration, extending PPRC management and HyperSwap capabilities to support two synchronous Metro Mirror relationships from a single primary volume. Note that HyperSwap is supported for ECKD and xDR managed FB disk. Each leg is tracked and managed independently. This provides additional data protection in the event of a disk subsystem failure or local disaster scenario.

GDPS Metro provides management extensions for heterogeneous platforms (xDR) to be able to fully manage either z/VM systems and their guests, Linux in LPAR and KVM on LinuxONE environments providing full end-to-end support including disk management with freeze and planned/unplanned HyperSwap support, systems management and monitoring. This support also applies to z/VM and KVM on FBA formatted disks.

As shown in Figure 3-10, GDPS Global requires two z/OS systems images, (K1and K2) to manage, monitor and execute recovery scripts.

GDPS 3-site solution

GPDS also supports 3-site configuration solutions by using a combination of Metro Mirror and Global Mirror solutions. A 3-site solution can combine the advantages of metropolitan distance business continuity and regional or long-distance disaster recovery. See Figure 3-11 on page 92.

GDPS Metro / Global Mirror / Metro Global Mirror

You can use GDPS Metro with GDPS Global Mirror to manage the configuration across all formats of data, as Global Mirror is not limited to IBM Z formatted data.

A typical configuration has the secondary disk from a Metro Mirror Remote Copy configuration which in turn becomes the primary disk for a Global Mirror Remote Copy pair. Data is replicated in a “cascading” fashion.

Metro Global Mirror is a method of continuous, remote data replication that operates between three sites that varying distances apart. Metro Global Mirror combines Metro Mirror synchronous copy and Global Mirror asynchronous copy into a single session, where the Metro Mirror target is the Global Mirror source. Using Metro Global Mirror and Metro Global Mirror with HyperSwap, your data exists on a second site that is less than 300 km away, and a third site that is more than 300 km away. Metro Global Mirror uses both Metro Mirror and Global Mirror Failover / Failback to switch the direction of the data flow. This ability enables you to run your business from the secondary or tertiary sites.

Fixed Block disk management

GDPS Metro, GDPS Global Mirror and Metro Global Mirror technology have been extended to manage a heterogeneous environment of IBM Z and distributed systems Logical Unit Numbers (LUNs), also known as Fixed Block disk, or FB disk. If installations share their disk subsystems between the IBM Z and distributed systems platforms, GDPS Metro, GDPS HM, and GDPS GM can manage the Metro Mirror and Global Mirror remote copy configurations, as well as FlashCopy for distributed systems storage. GDPS Metro and GDPS Global – GM are also designed to provide:

- A single point of control and management for both FB and ECKD disk replication.
- Freeze capability to protect the consistency group for synchronous replication.
- Data consistency across IBM Z and distributed environments for applications that span multiple tiers.
- HyperSwap capability for FB devices with native Linux on Z:

  > **Note:** If the client has zFBA in a shared ECKD Consistency Group and GDPS does a HyperSwap, GDPS resets all non-HyperSwap capable systems ("inhibited", Linux on Z, KVM, SSC) in case of an Unplanned HyperSwap (UHS). A Planned HyperSwap (PHS) will complain about those systems being up and running and a PHS will not be performed until all such systems are down.

- GDPS Metro uses the DS8000 zFBA support.

Using Fixed Block disk management support allows GDPS to be a single point of control to manage business resiliency across multiple tiers in the infrastructure, improving cross-platform system management and business processes.

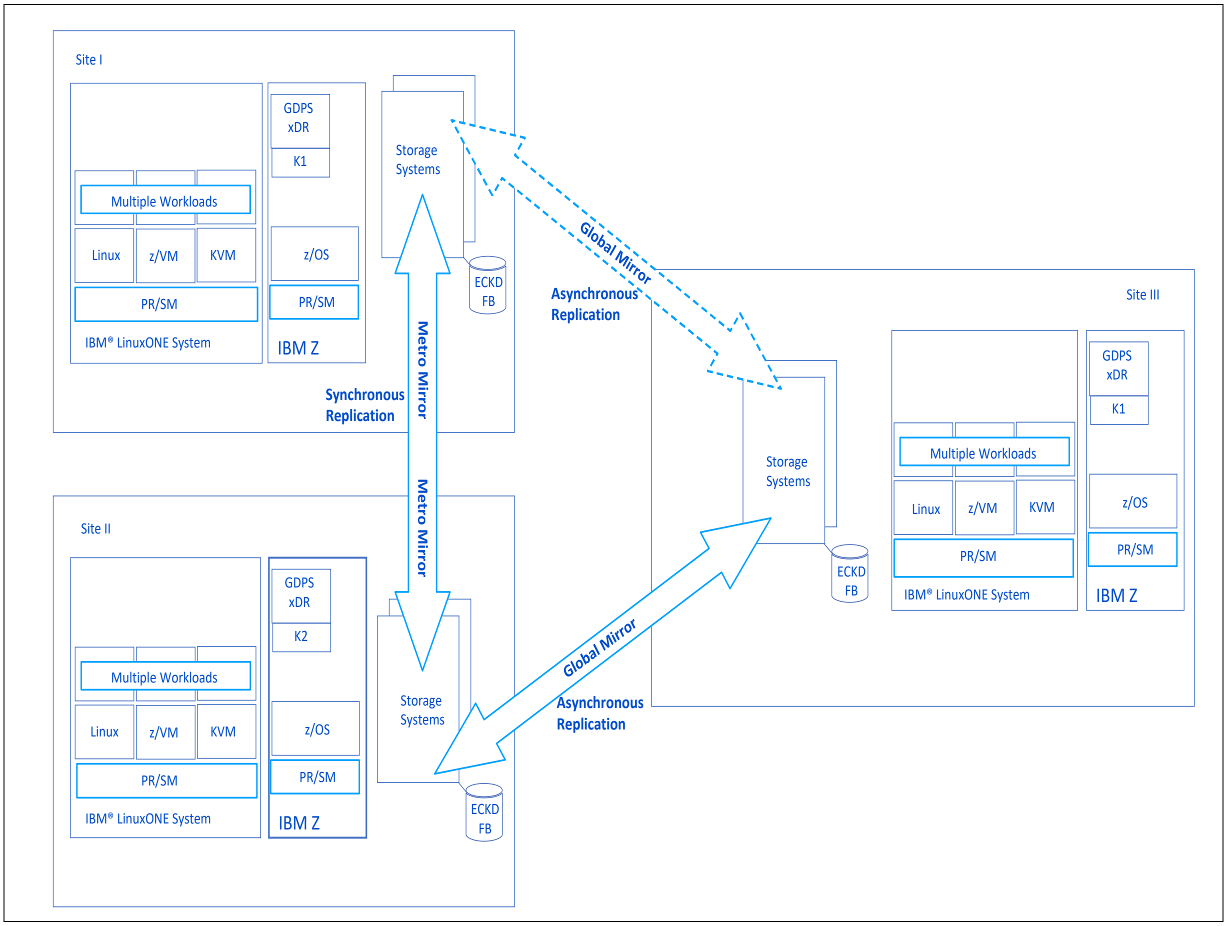

Figure 3-11 Multi-Site fault tolerant Option - Continuous availability implementation example

Figure 3-11 on page 92 shows a possible configuration of a Metro/Global Mirror environment. It is divided into a primary production site, Site I, a secondary production site, Site II, and a remote site, Site III. Between the primary and secondary production sites (Site I and Site II), Metro Mirror relationships can be configured in either direction. In both production sites, a Global Mirror is configured to the remote site. All connections are set up with redundant inter-switch links between fiber directors or switches, not shown in the figure.

To support this complex configuration two z/OS partitions are required. As LinuxONE cannot run z/OS, another IBM Z system footprint is required on each site. The IBM Z systems will hold the z/OS LPARs, GDPS (with xDR) as well as the K1 and K2 controlling systems.

The operating systems must run on servers that are connected to the same Hardware Management Console (HMC) Local Area Network (LAN) as the GDPS control system

3.4.4 Resiliency level of Continuous Availability - Summary

– **Characteristics:**

- Protects against local and regional planned and/or unplanned outages.
- Provides Continuous Availability of heterogeneous platforms with GDPS xDR.
- Effective and efficient Systems’ management.
- GDPS automated takeover (site switch / Hyperswap / restart).
- GDPS proxies on z/VM (multiple) and backing by a K-sys in a parallel sysplex.

- **Benefits:**
  - Rapid restart.
  - Multiple LPARs cloned.
  - Data sharing (GDPS Metro - Global Mirror.
  - GDPS Full Automation
  - Provisioning Spare Capacity - auto activation by GDPS.
- **Improvements:**
  - GDPS and SafeGuarded Copy for snapshots of system contents.
  - Restoration of previous copies of workload, dependent upon forensic investigation.

3.4.5 “Eight nines” 99.999999% Availability

Low downtimes as IBM LinuxONE Emperor 4 systems, with GDPS, IBM DS8000 series with Hyper Swap and running a Red Hat OpenShift Container Platform environment are designed to deliver 99.999999% availability.

Chapter four will explore a configuration that is designed to allow this extended availability.

Disclaimer: IBM internal data based on measurements and projections was used in calculating the expected value. Necessary components include IBM LinuxONE Emperor 4; IBM z/VM V7.3 systems collected in a single system image, each running RHOCP 4.14 or above; IBM Operations Manager; GDPS 4.6 for management of data recovery and virtual machine recovery across metro distance systems and storage, including Metro multi-site workload and GDPS Global; and IBM DS8000 series storage with IBM Hyper Swap. A MongoDB v4.2 workload was used. Necessary resiliency technology must be enabled, including z/VM single system image clustering, GDPS xDR Proxy for z/VM, and Red Hat OpenShift Data Foundation (ODF) 4.14 for management of local storage devices. Application-induced outages are not included in the above measurements. Other configurations (hardware or software) may provide different availability characteristics.

4

Resiliency for Red Hat Openshift Container Platform

This chapter describes Red Hat OpenShift Container Platform (RHOCP) resiliency when running on IBM z16 and IBM LinuxONE 4.

This chapter also discusses the means of availability calculation, IBM's differentiation and value-add to RHOCP workload, the required Resiliency configuration, potential points of planned and unplanned outage, technical mitigations and means around service interruption, disaster recovery, and cyber resiliency.

OpenShift Container Platform (OCP) provides continuous operation and service resiliency when running on IBM LinuxONE. By combining the Red Hat Open Container Platform clustering technology with enterprise-level compute (IBM LinuxONE) and storage (IBM DS8K), along with an appropriate virtual infrastructure (IBM PR/SM, IBM z/VM and associated automation, IBM GDPS), client workload can achieve over 99.999999% (eight nines) of uptime despite planned or unplanned outages.

This uptime can be achieved for data and enterprise services within the bounds of this configuration.

This chapter contains the following topics:

4.1 Resiliency for Red Hat OpenShift Container Platform

In this chapter we will use the Red Hat OpenShift Container Platform (RHOCP) running on IBM z16 or LinuxONE to discuss the means of availability calculation, IBM's differentiation and value-add to RHOCP workload, the required Resiliency configuration, potential points of planned and unplanned outage, technical mitigations and means around service interruption, disaster recovery, and cyber resiliency.

By combining the Red Hat Open Container Platform clustering technology with enterprise level compute (IBM LinuxONE) and storage (IBM DS8K), along with an appropriate virtual infrastructure, consisting of IBM PR/SM, z/VM and associated automation and IBM GDPS, client workload can achieve over 99.999999% (eight nines) of uptime, despite planned or unplanned outages.

4.1.1 Calculating Availability

Availability is calculated based upon service metrics, environment testing and solution modeling. Given compute redundancy, our availability ceiling is access to data on storage.

- Physical sites exist at 10^5 availability (99.999% uptime); at least two sites recommended
- The implementation requires redundancy and automation for compute (IBM z16 and IBM LinuxONE);
  - IBM z/VM; IBM GDPS; Red Hat OpenShift Container Platform
- IBM DS8K storage volumes are 10^6 availability (99.9999% uptime) on an individual basis
- When DS8K is managed by GDPS, we reach at least 10^8 for availability (99.999999% uptime)

Assumptions

The claims mentioned above are based on an optimum configuration for the calculation of availability for Openshift Container Platform workloads running on LinuxONE hardware.

- We assume proper configuration, whereupon all components (hardware/software) have appropriate back-up and redundancy to allow for outage mitigation and to ensure minimal performance impact, within client requirements
  - This includes ”free space” on systems (CPU, storage, memory) in case of planned outages and workload redistribution
  - This includes configuring z/VM and RHOCP for High Availability and redundancy – virtual resources are resources
- We assume that the greater datacenter ecosystem is configured for redundancy and high availability:
  - This includes services such as DNS and load balancing for applications running in RHOCP
  - This includes physical cabling for networking
  - This includes power/electricity, fire suppression, etc
- We assume a competent and non-malicious set of system administrators
  - This includes testing fixes on a development system, isolated from production workloads
- We assume a properly configured z/OS system running GDPS and xDR exists somewhere pertinent in the enterprise:
  - Parallel Sysplex Best Practices: http://www.redbooks.ibm.com/abstracts/sg247817.html?Open
  - GDPS Configuration:

    For GDPS and xDR configuration guidance, please refer to this RedBook publication:

IBM GDPS An Introduction to Concepts and Capabilities, SG24-6374, using this link: https://www.redbooks.ibm.com/redpieces/abstracts/sg246374.html

- We assume that all pertinent security guidelines have been followed with regards to configuration of the included implementation.

Threats

The resiliency model has been designed to withstand the following types of threats to service availability:

- Natural disasters
  - Multi-site configuration meant to support this, but flexibility depends on size/scope of disaster
- Planned outages to systems
- Power outages
  - Having backup generators and power failover should already be a physical availability consideration.
- Replication failures (non-malicious, non-user-error)
  - Restoration of previous copies of workload, dependent upon forensic investigation
  - Data integrity check following, for instance, a system upgrade

While security threats (cyber attack, malicious users, ransomware) are acknowledged potential threats, they are outside the scope of this particular calculation. Refer to “The Value of Virtualization Security” (https://www.vm.ibm.com/devpages/hugenbru/L1SECV22.PDF) for more information about IBM LinuxONE Security and cyber resiliency.

4.1.2 Differentiators

IBM LinuxONE hardware

designed for maximal resiliency using its RAS characteristics such as core sparing, Redundant Array of Independent Memory (RAIM) and critical parts redundancy, for instance Power Supplies, cooling components, service network, and more. LinuxONE provides Vertical Scalability of systems (more servers hosted; less hardware that can fail), and at given performance points, processor cores and memory can be added concurrently and transaction security can be achieved via on-chip symmetric encryption or shared Crypto Express domains.

IBM Storage hardware (DS8K)

built for high availability and, with Extended Count Key Data (ECKD) it allows for additional data integrity, redundancy and availability compared to SCSI devices.

IBM virtualization - PR/SM and z/VM

provide secure, scalable workload hosting designed to avoid planned outages.

IBM GDPS

used in support of data replication, mirroring and HyperSwap of disk volumes.

4.2 Availability and Resiliency for Linux workloads

This section aims to illustrate optimum configurations for the calculation of Availability and Resiliency for Linux based workloads on IBM LinuxONE hardware.

The workload under consideration is containerized applications (RHOCP) and Database workloads such as Oracle will have additional or alternate considerations.The included calculations work within the boundaries of virtual infrastructure and containerization, but do not include individual applications. While this implementation recommends that RHOCP container workloads be appropriately distributed, with load-balancing and redundancy for each, it does not evaluate resiliency of an application itself.

No claims are made specifically about zCX and z/OS-hosted workloads in this document.

As in all cases, this guidance is not a client mandate; clients may optimize their environments to meet their needs, including modifications to achieve even higher redundancy

4.2.1 Components of LinuxONE Resiliency

The components of this approach are shown in Figure 4-1 on page 99 and listed below:

- Two physical sites at regional distance
- At least two (2) Storage units (DS8K-series configured for ECKD
- One (1) IBM z16 machine per site for running z/OS 2.5, GDPS 4.5 (K-sys and xDR)
- LinuxONE machines, 2+ CPCs (at least one per site), each configured PR/SM mode (not DPM)
- z/VM V7.3 with virtual networking, HyperSwap, and Single System Image (SSI)
  - Minimum of three (3) z/VM partitions (out of 8 maximum) in an SSI cluster
  - z/VM V7.3, GA 09/2022, increases allowable SSI cluster to 8 members[1]
  - IBM Operations Manager for z/VM, V1.6
  - GDPS 4.5 xDR Proxies for z/VM
- RHOCP V4.10 (CoreOS) configured for containerization resiliency
  - Red Hat Openshift Data Foundation 4.10 (ODF) for local storage
  - System Automation for Multi-platforms (SA MP) for RHOCP Control Node and Compute Node
  - Control and Compute nodes should not be converged in a single guest

Considerations about the environment shown in Figure 4-1 on page 99:

- Versions:
  - RHOCP 4.10 runs on z14 - z16. Version 4.9 is optimized to run on z13® - z15®. It is not supported on z16.
  - ODF carries the same version number as RHOCP
  - Operations Manager V1.6 supports earlier releases of z/VM.
  - RHOCP 4.10 runs on IBM z13—z16. Version 4.9 is optimized to run on IBM z13—z15 but it is not supported to run on the IBM z16.
- Hardware:
  - IBM Z (z16) and LinuxONE (LinuxONE IV) machines must be configured to run in PR/SM mode - not DPM. This is a CPC-wide setting.

[1] Previous version of z/VM supported a maximum of four members in a Single System Image.

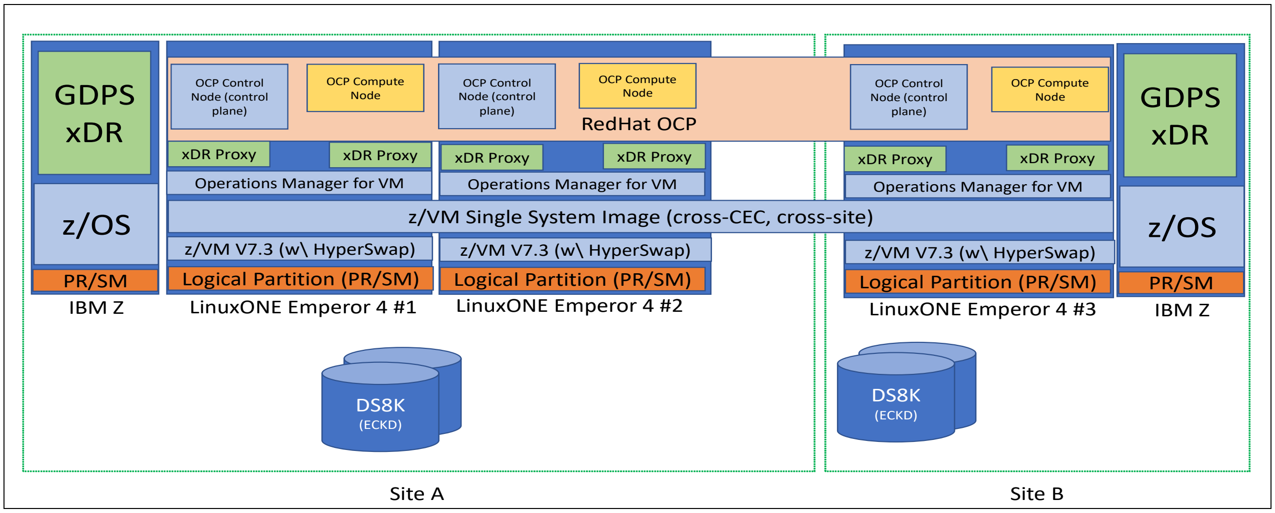

Figure 4-1 Structural Diagram

Figure 4-2 expands one or the z/VM partitions from the diagram shown in Figure 4-1. Some important considerations about the single partition view are:

- Reminder: RHOCP does not run under Linux on IBM Z; it runs under an IBM Z hypervisor (z/VM or KVM)
- RHEL-hosted workloads (or SLES, etc) are not under discussion in this example
- IBM hardware must be in PR/SM mode and not DPM mode
- z/VM is a requirement due to the necessity of HyperSwap. It will contain 2 xDR proxies.

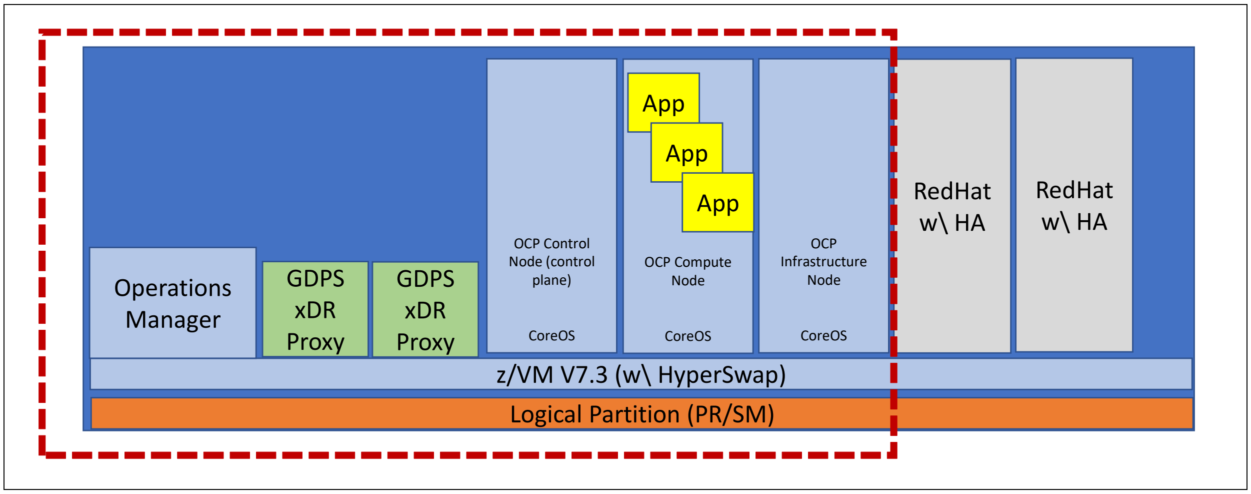

Figure 4-2 Single Partition, Interior view

4.2.2 Network

Network implementation is shown in Figure 4-3 on page 100.

The implementation has multiple (three) virtual devices at the CoreOS level connected to a z/VM Layer 2 Virtual Switch. Link aggregation is used to bind multiple OSA Ports across multiple adapters on a CPC.

Intra-VSWITCH link to collate connections across physical hardware in a SSI Cluster. Physical switches and cabling are used to bind sites together. Sites can be in a 200 km range fro Metro-Mirror (GDPS). This implementation allows for network traffic separation at virtual, logical and/or physical layers in accordance with security requirements.

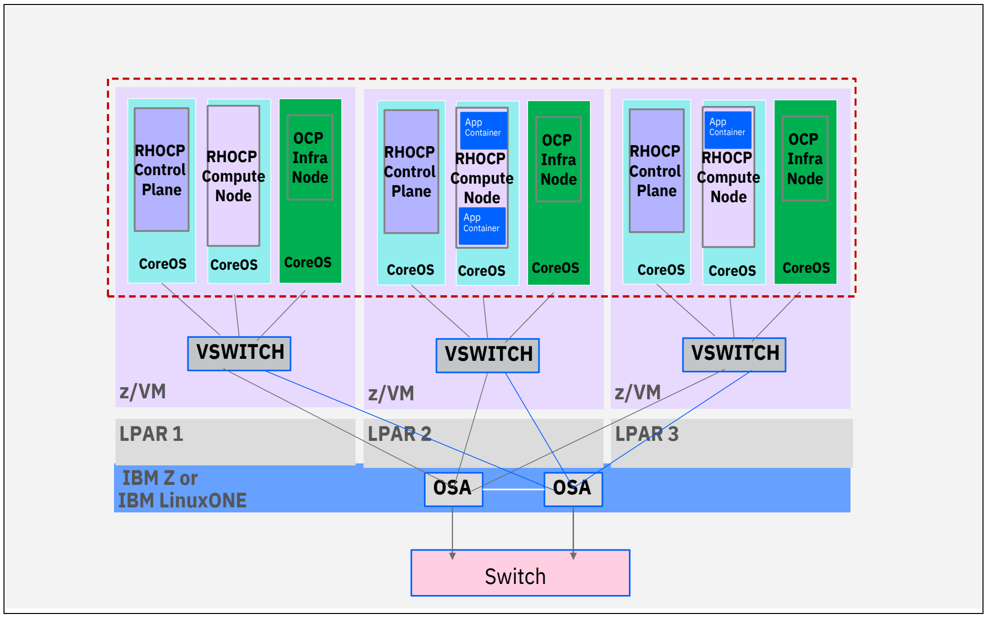

Figure 4-3 Networking diagram

4.2.3 Network Optimization

The network optimization is divided in three layers: RHOCP, z/VM and Physical. The considerations pertaining to each layer are described below.

- **RHOCP considerations**
  - etcd timeout tuning - This timeout determines how long a node will go without hearing a heartbeat before attempting to become leader. The latency for control heartbeat needs to be within acceptable parameters (heartbeat interval 100ms and election timeout 1000ms).

The *Heartbeat Interval* is the frequency with which the leader will notify followers that it is still the leader. For best practices, the parameter should be set around round-trip time between members. By default, etcd uses a 100ms heartbeat interval. The *Election Timeout* is how long a follower node will go without hearing a heartbeat before attempting to become leader itself. By default, etcd uses a 1000ms election timeout.

- **z/VM considerations**
  - analyze site-to-site.
  - Vswich congestion - add network prioritization to give control plane traffic first access.
  - measure link aggregation to determine optimal port access.
- **Physical**
  - GDPS communication IP-based; an outage there represents a catastrophic network problem. (arrange z/OS cross-site accordingly).
  - Network separation:
    - Separate VSWITCHs, or direct-attached OSAs, for sensitive guests.
    - VSWITCH VEPA mode to force traffic separation to physical switch.
  - Internet latency not solvable by IBM, but we can optimize local traffic to give sensitive communications a running head start.

4.2.4 Storage Optimization

- Apps
  - Apps less likely to need persistent/stateful data access
    - Implementation generally excludes apps, but we don't want to leave them hanging.
    - ODF can allow for configuration of apps to the use of persistent storage, instead of network-attached devices. This adds a logical layer with additional failover mechanisms (including maintaining multiple replicas of stored data).
    - GDPS can continue to manage ECKD volumes as pertinent; no need for other intervention.
- RHOCP and z/VM
  - Linux-hosted workloads running under z/VM will need persistent storage (ECKD on DS8K), attached to the guest itself and reflected to the local Linux OS.
    - Owned by z/VM, managed by GDPS.
    - HyperSwap of these volumes possible.
    - z/VM HyperPAV aliasing.
- OpenShift Data Foundation (ODF)
  - OpenShift Data Foundation is software-defined storage that is optimized for container environments. It runs as an operator on OpenShift Container Platform to provide highly integrated and simplified persistent storage management for containers.
    - ODF allows for Applications to communicate to persistent storage.
    - ODF adds replicas of persistent volumes, which are kept in sync and add to the overall resilience of stored data.
    - ECKD DASD allows us to restrict scope of Framework, eliminates need to configure
    - network-attached devices.

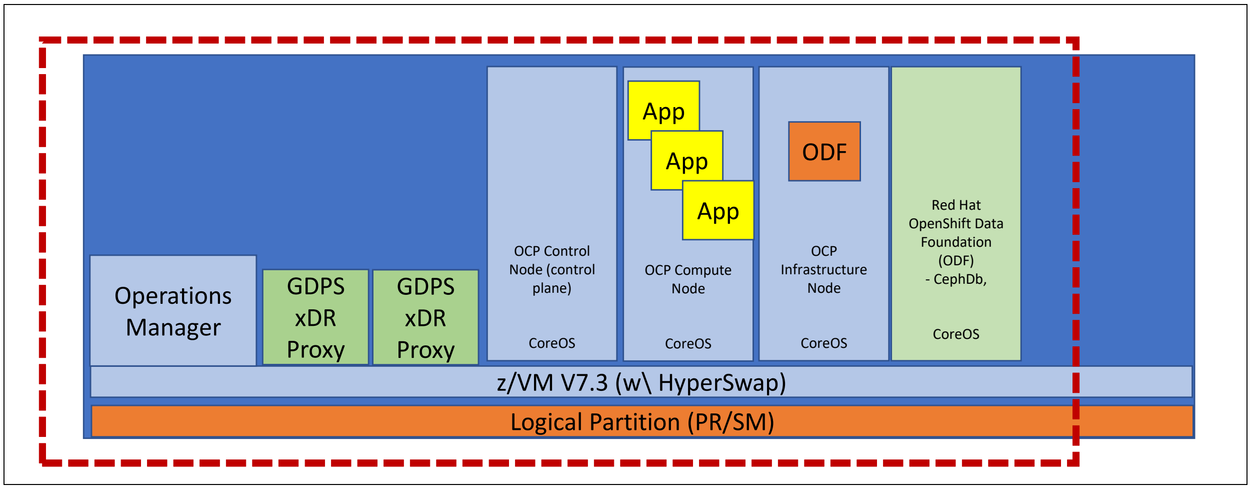

Figure 4-4 Single Partition with ODF

Hyperswap

- Refer to: "VA HyperSwap function" on page 69
  - HyperSwap is IBM's High Availability solution that provides active-active storage access to volumes shared across two physical sites. It is a high availability feature that provides dual-site, active-active access to a volume. This means that data is continuously available at both sites, improving the availability of your business. It is based on synchronous Peer to Peer Remote Copy (PPRC) technology for use within a single data center. Data is copied from the primary storage device to a secondary storage device. See "VA HyperSwap function" on page 69.

- GDPS Peer-to-Peer Remote Copy (GDPS/PPRC) HyperSwap is an IBM offering that non-disruptively swaps I/O activity from the primary volumes to the secondary volumes of PPRC mirrored pairs, when a problem occurs on any primary volume1. Automation is used to swap all mirrored volumes at the same time.

Figure 4-5 shows DS8K storage being mirrored between Site A and Site B using PPRC / Metro Mirror. Storage synchronous data replication (mirroring) is the foundational pre-requisite for the HyperSwap solution take place.

The blue lines connecting the partitions to the DS8K disks represent the "in-use" data access paths while the PPRC / Metro Mirror green arrow represents the required data synchronous replication capability between sites.

The gray dotted lines represent the alternate access paths to storage in an event of a HyperSwap. HyperSwap can be monitored and controlled by GDPS. See Figure 4-5.

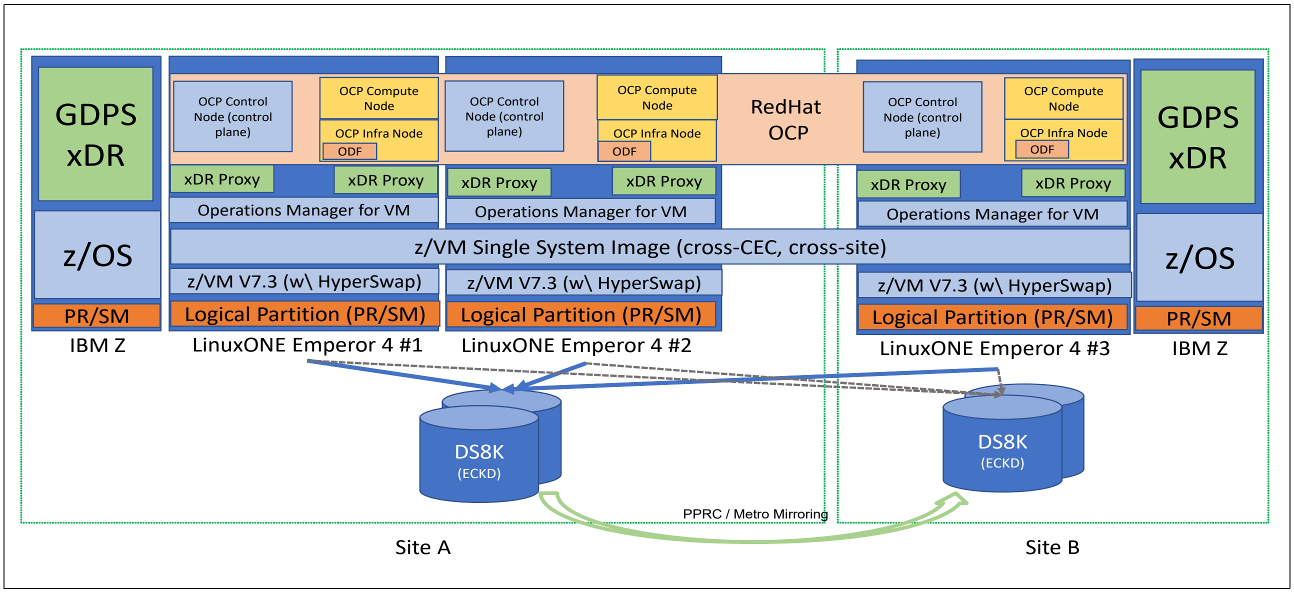

Figure 4-5 PPRC / Metro Mirror, GDPS xDR and HyperSwap between sites

4.3 Framework Analysis

This section discusses the Framework aspects in terms of:

- Points of Outage.
- Potential Technical Disruption.
- Threats.
- Mitigations and Workaround.
- Impacts in a Resilient / HA environment and Relative Mean Time To Repair (MTTR).

4.3.1 Points of Outage

Case 1: Storage Unit Failure

Failing Component	What happens	Mitigation, workaround, recovery
Storage unit fails: one DS8K physical device (not a volume, not a link) Figure 4-6	All data on storage unit is unavailable	• RAID and data striping to avoid integrity issues • GDPS doing mirroring to second local storage unit, PPRC to remote storage unit, and Cyber Vaulting via SafeGuard Copy
Impact: negligible. With HyperSwap, storage failover happens within 2-3 seconds. (z/VM Switch is quiesced at this time as well, so there may be a momentary network blip.)		
Time to recovery of the failing component: 2 hours to restore physical unit, 4 hours to uptime. Additional time to mirror current data to restored volume will be required. (Business impact not included in calculation.)		

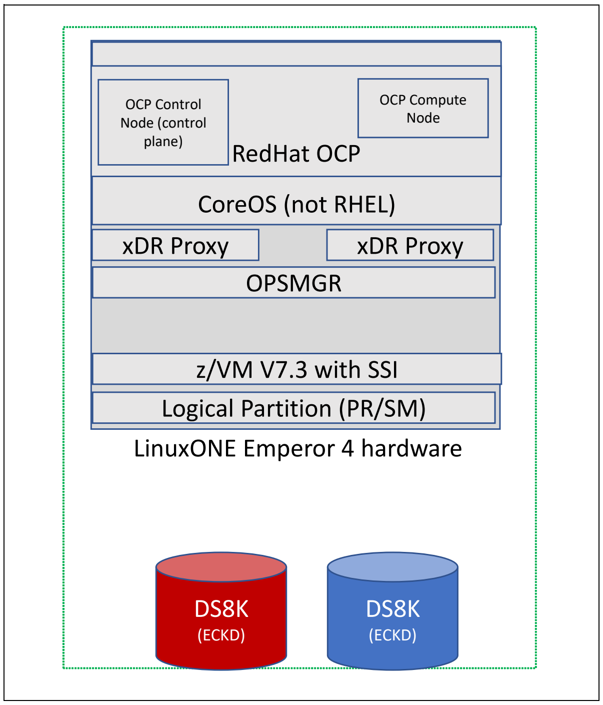

Figure 4-6 Storage unit failure

Case 2: Hardware unit outage

Failing Component	What happens	Mitigation, workaround, recovery
Hardware unit outage (one LinuxONE IV CPC in entirety) Figure 4-7	Compute for an entire system (1-n logical partitions and hosted workload) not available	• Core sparing for general workload bypass • RAIM to avoid memory corruption • Physical redundancy inside a CPC to mitigate problems
Impact: Negligible. Workload continues on other CPCs autonomically		
Time to recovery of the failing component: 4 hours typical for CPC repair. No data rebuild in this case (it's compute).		

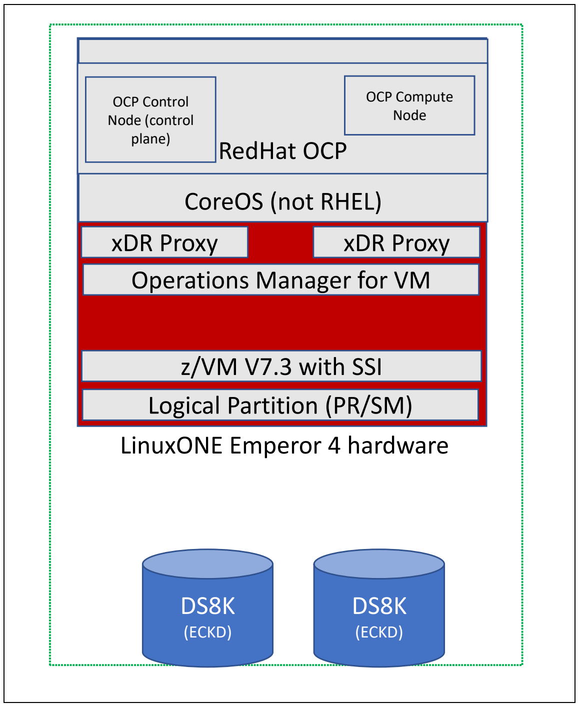
OCP Control Node (control plane)
OCP Compute Node
RedHat OCP
CoreOS (not RHEL)
xDR Proxy
xDR Proxy
Operations Manager for VM
z/VM V7.3 with SSI
Logical Partition (PR/SM)
LinuxONE Emperor 4 hardware
DS8K (ECKD)
DS8K (ECKD)

Figure 4-7 Hardware unit failure

Case 3: Logical Partition goes into a “Wait State” or has an outage

Failing Component	What happens	Mitigation, workaround, recovery
Logical partition takes wait-state or has outage Figure 4-8	Compute for one partition and hosted workload is gone	• SIE isolation to prevent impact to other workloads • Workload rescheduled on other Compute nodes (OpenShift)
Impact: negligible. Similar to hardware, z/VM and RHOCP workloads have the capacity to move to other partitions in the SSI / systems in the cluster.		
Time to recovery of the failing component: n/a. LPAR instantiation is near-immediate, and failure at the LPAR level almost always means “workload”; see following slides.		

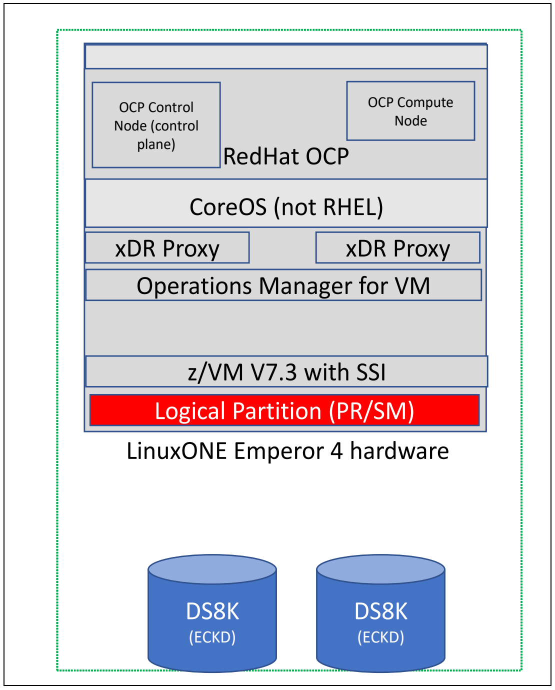
OCP Control Node (control plane)
OCP Compute Node
RedHat OCP
CoreOS (not RHEL)
xDR Proxy
xDR Proxy
Operations Manager for VM
z/VM V7.3 with SSI
Logical Partition (PR/SM)
LinuxONE Emperor 4 hardware
DS8K (ECKD)
DS8K (ECKD)

Figure 4-8 Logical Partition Outage

Case 4: One z/VM system has an outage

Failing Component	What happens	Mitigation, workaround, recovery
One z/VM system has outage Figure 4-9	Virtual infrastructure is gone (one logical partition)	• Operations Manager to automate general use-case recovery • SSI and relocation (if planned outage) • RHOCP rescheduling
Impact: negligible. SSI allows guest mobility to relocate workloads; RHOCP will reschedule work onto other compute.		
Time to recovery of the failing component: Five minutes to reactivate partition and IPL z/VM system. Recovery time past that will vary based upon the number of guests starting, and the size of the z/VM partition (which will have impacted the time to take a dump used in problem determination). IPL of z/VM system includes guests, virtual storage, GDPS xDR proxies, virtual networking, and RHOCP for that system. It will not include resync operations of various layers.		

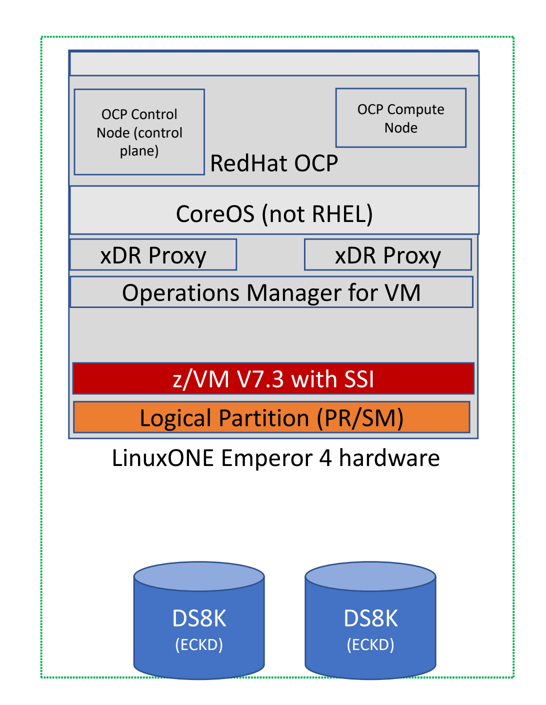
OCP Control Node (control plane)
OCP Compute Node
RedHat OCP
CoreOS (not RHEL)
xDR Proxy
xDR Proxy
Operations Manager for VM
z/VM V7.3 with SSI
Logical Partition (PR/SM)
LinuxONE Emperor 4 hardware
DS8K (ECKD)
DS8K (ECKD)

Figure 4-9 LPAR enters Wait State or Outage

Case 5: z/VM Virtual Network encounters problem

Failing Component	What happens	Mitigation, workaround, recovery
z/VM virtual networking encounters problem Figure 4-10	Traffic for z/VM or guests is disrupted, leading to problems with heartbeats and system assessment; disruption to client workload	• Link aggregation to collect multiple OSA ports into common logical channel • VSWITCH automatic failover (four controller nodes) in case of disrupted network operation • Inter-VSWITCH Link to coordinate across multiple z/VM systems
Impact: negligible, as link aggregation collates resources and failover prevents an outage.		
Time to recovery of the failing component: if physical networking, varies. If virtual networking, 1-2 hours to debug and rebuild.		

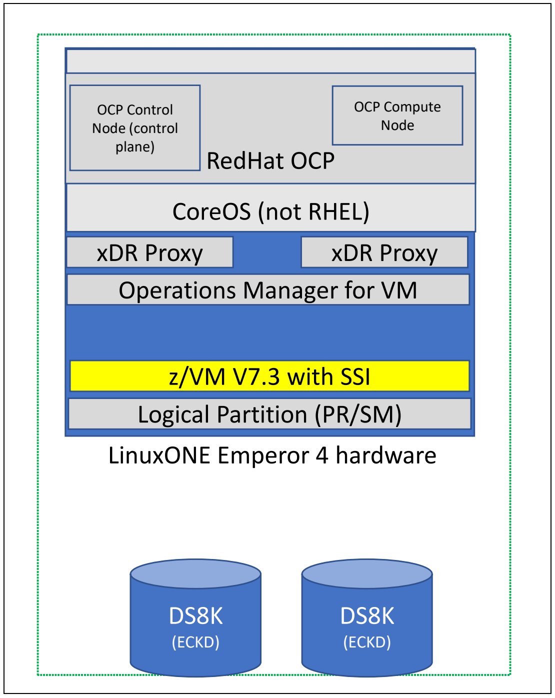
OCP Control Node (control plane)
OCP Compute Node
RedHat OCP
CoreOS (not RHEL)
xDR Proxy
xDR Proxy
Operations Manager for VM
z/VM V7.3 with SSI
Logical Partition (PR/SM)
LinuxONE Emperor 4 hardware
DS8K (ECKD)
DS8K (ECKD)

Figure 4-10 z/VM Network, I/O or SSI problem

Case 6: z/VM I/O links or channels take an outage

Failing Component	What happens	Mitigation, workaround, recovery
z/VM I/O links or channels take an outage Figure 4-10 on page 107	Access to storage via virtual infrastructure is disrupted; data cannot be read in or written out, potentially disrupting business	• Multiple channel paths for access to storage units • Operations Manager for local scripting to mitigate failures
Impact: negligible, as data remains available even with a failing virtual or logical device. (If the last channel path standing fails, a HyperSwap is triggered.)		
Time to recovery of the failing component: varies upon whether failure is physical or logical. (2-3 seconds if it's a reboot of a channel path; hours if there's a physical problem / cabling issue.)		

Case 7: One z/VM system has an outage

Failing Component	What happens	Mitigation, workaround, recovery
z/VM Single System Image has a one-node failure Figure 4-10 on page 107	Individual systems keep running, but guest mobility no longer feasible; potential for splitbrain problems in zoned workloads running under z/VM (RHOCP)	• Systems in SAFE mode to continue operating. (Manual intervention.) • SSI members don't STONITH • RHOCP should STONITH • SSI rejoins missing member(s) after repairs are made and state is reaffirmed
Impact: potential disruption to RHOCP quorum and workload dispatch if SAFE-mode system remains unavailable, but workload soon rebalanced.		
Time to recovery of the failing component: see "Planned Outages for z/VM." Recovery of one system will vary based upon damage; time to IPL remains constant.		

Case 8: z/VM SSI takes a hit to Persistent Data Record

Failing Component	What happens	Mitigation, workaround, recovery
z/VM Single System Image takes a hit to the Persistent Data Record Figure 4-11 on page 109	PDR volume maintains state of overall cluster and heartbeat tracking; if damaged, whole cluster moves into recovery mode	GDPS – the PDR is not handled separately from the rest of the GDPS consistency group.
Impact: minimal; HyperSwap takes 2-3 seconds. Note: PDR becomes a single point of failure until original PDR volume is restored.		
Time to recovery of the failing component: Once storage device is repaired (see Point of Outage 1), Metro Mirroring resumes active backup.		

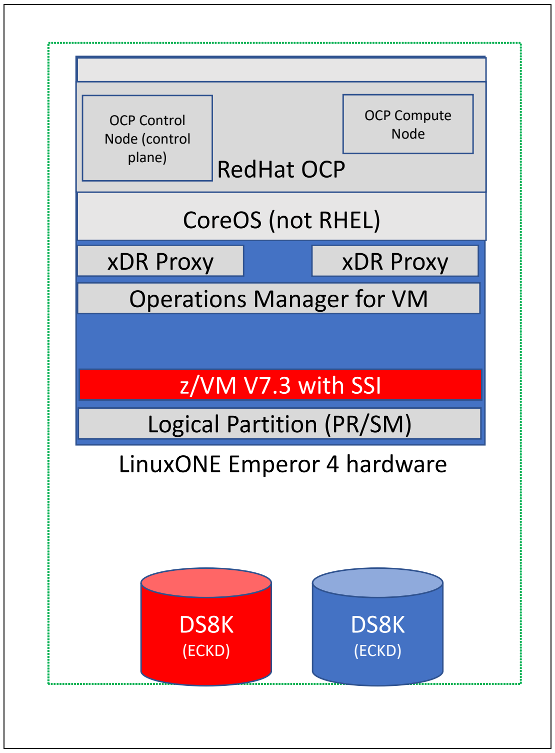
OCP Control Node (control plane)
OCP Compute Node
RedHat OCP
CoreOS (not RHEL)
xDR Proxy
xDR Proxy
Operations Manager for VM
z/VM V7.3 with SSI
Logical Partition (PR/SM)
LinuxONE Emperor 4 hardware
DS8K (ECKD)
DS8K (ECKD)

Figure 4-11 SSI Persistent Data Record takes a hit

Case 9: xDR Proxy failure

Failing Component	**What happens**	**Mitigation, workaround, recovery**
xDR Proxy fails Figure 4-12 on page 110	xDR satellite crashes on a given z/VM system, inhibiting	• Automatic failover to backup xDR satellite on z/VM system
Impact: none. xDR Proxy failover is immediate.		
Time to recovery of the failing component: 2-3 seconds, then 2-3 minutes to re-IPL xDR proxy guest.		

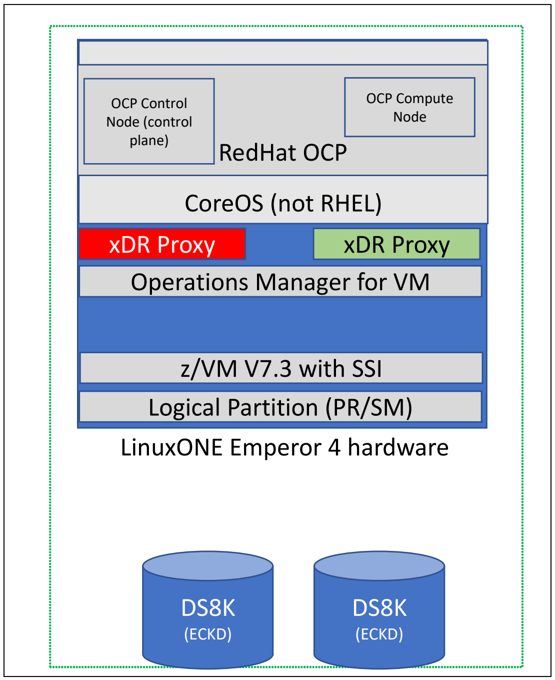
OCP Control Node (control plane)
OCP Compute Node
RedHat OCP
CoreOS (not RHEL)
xDR Proxy
xDR Proxy
Operations Manager for VM
z/VM V7.3 with SSI
Logical Partition (PR/SM)
LinuxONE Emperor 4 hardware
DS8K (ECKD)
DS8K (ECKD)

Figure 4-12 xDR Proxy Failure

Case 10: Network disruption - no xDR Proxy access to IBM Z GDPS

Failing Component	What happens	Mitigation, workaround, recovery
Networking disruption prevents xDR Proxy access to IBM Z GDPS installation Figure 4-13 on page 111	Commands from z/OS inhibited in terms of making backup	• If running z/OS in a nearby partition, not likely to be an issue (HiperSockets, local switches) • Refer to IBM Z Resiliency statements regarding z/OS
Impact: n/a. Since GDPS is IP-based, this only happens if an entire network is destroyed.		
Time to recovery of the failing component: n/a.		

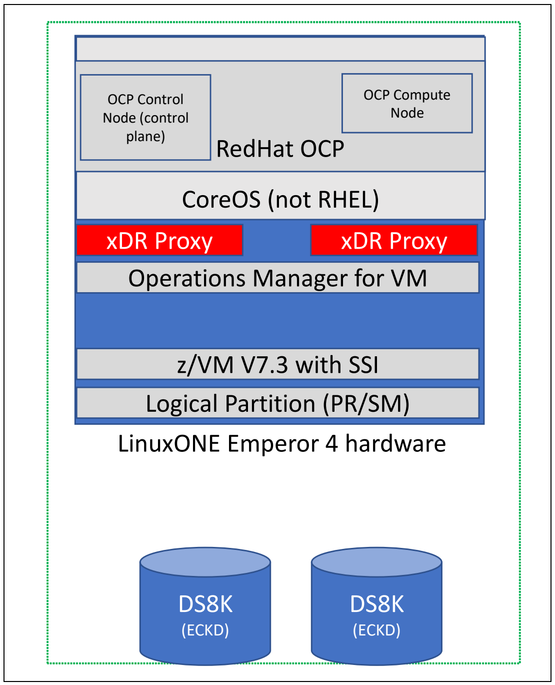
OCP Control Node (control plane)
OCP Compute Node
RedHat OCP
CoreOS (not RHEL)
xDR Proxy
xDR Proxy
Operations Manager for VM
z/VM V7.3 with SSI
Logical Partition (PR/SM)
LinuxONE Emperor 4 hardware
DS8K (ECKD)
DS8K (ECKD)

Figure 4-13 xDR connectivity issue

Case 11: CoreOS system crashes

Failing Component	What happens	Mitigation, workaround, recovery
CoreOS system crashes Figure 4-14 on page 112	z/VM system is running fine, but a Red Hat guest has taken a kernel panic, disrupting workload above	• RHOCP checks control status and reschedules workload if pertinent, • z/VM re-IPL as pertinent (automatable)
Impact: Temporary loss of compute related to Control or Compute; workload redistributed or restarted on other nodes as pertinent.		
Time to recovery of the failing component: n/a. There isn’t really a CoreOS instantiation independent of RHOCP operations, so this is not a distinct point of outage from RHOCP crash		

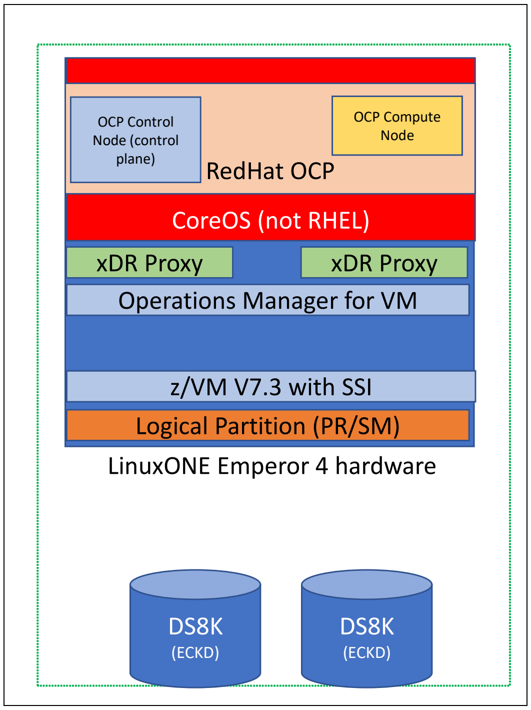
OCP Control Node (control plane)
OCP Compute Node
RedHat OCP
CoreOS (not RHEL)
xDR Proxy
xDR Proxy
Operations Manager for VM
z/VM V7.3 with SSI
Logical Partition (PR/SM)
LinuxONE Emperor 4 hardware
DS8K (ECKD)
DS8K (ECKD)

Figure 4-14 CoreOS System crash

Case 12: CoreOS networking failure inside Linux Guest

Failing Component	What happens	Mitigation, workaround, recovery
CoreOS networking failure inside Linux guest Figure 4-15 on page 113	Linux instance has networking device failure	• Workload rescheduled/restarted on other operational nodes. After network recovery node is accessible and workloads can be scheduled.
Impact: Temporary loss of compute related to Control or Compute; workload redistributed or restarted on other nodes as pertinent		
Time to recovery of the failing component: n/a. There isn't really a CoreOS instantiation independent of RHOCP operations, so this is not a distinct point of outage from RHOCP crash		

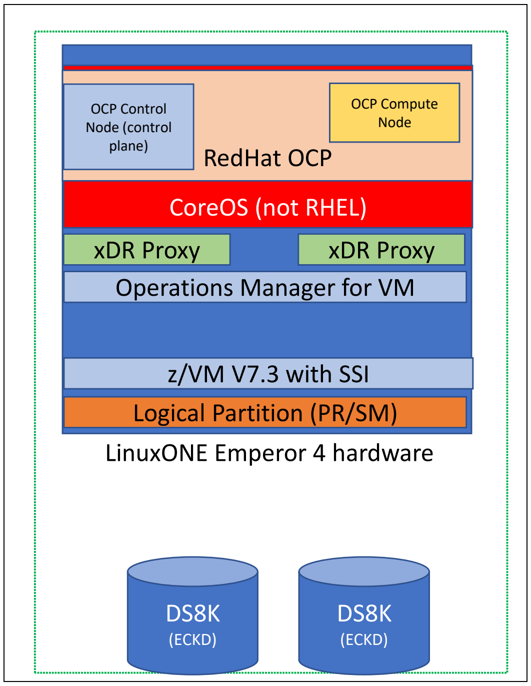
OCP Control Node (control plane)
OCP Compute Node
RedHat OCP
CoreOS (not RHEL)
xDR Proxy
xDR Proxy
Operations Manager for VM
z/VM V7.3 with SSI
Logical Partition (PR/SM)
LinuxONE Emperor 4 hardware
DS8K (ECKD)
DS8K (ECKD)

Figure 4-15 CoreOS Network failure inside Linux guest

Case 13: CoreOS loses storage devices

Failing Component	What happens	Mitigation, workaround, recovery
CoreOS loses storage devices Figure 4-15	Linux instance loses access to necessary storage for hosted applications	• Linux stays up and can save to the page cache. When device is recovered, store on disk • If guest crashed Reboot of guest after storage connection is re-established • If storage is invalid – reinstall node
Impact: Temporary loss of compute related to Control or Compute; workload redistributed or restarted on other nodes as pertinent		
Time to recovery of the failing component: n/a. There isn’t really a CoreOS instantiation independent of RHOCP operations, so this is not a distinct point of outage from RHOCP crash		

Case 14: RHOCP control-node failure

Failing Component	What happens	Mitigation, workaround, recovery
RHOCP Control-node failure Figure 4-16	Linux instance loses access to necessary storage for hosted applications	• Workload continues to operate, but cluster will show failure state. • Workload rescheduled on other operational nodes.
Impact: Temporary loss of compute related to Control for certain programs; workload redistributed or restarted on other nodes as pertinent.		
Time to recovery of the failing component: 30-60min, assuming automation and presence of backups. (System administrator needs to reinstall from scratch and have the new instantiation of the node rejoin the cluster. No data recovery per se.)		

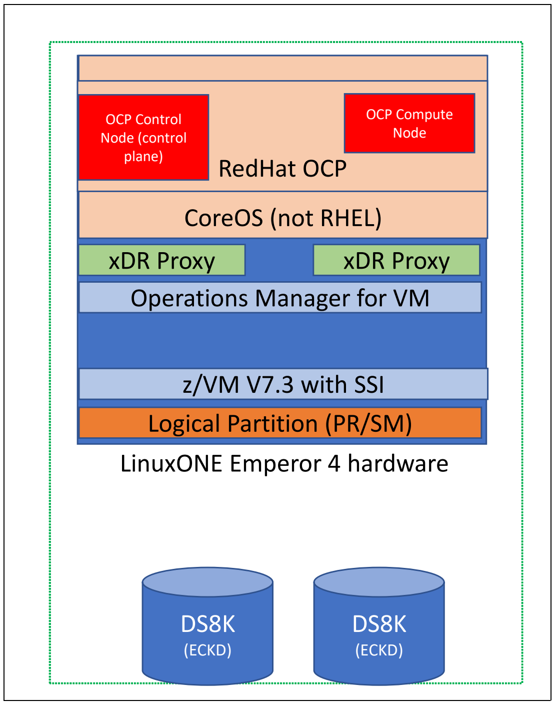
OCP Control Node (control plane)
OCP Compute Node
RedHat OCP
CoreOS (not RHEL)
xDR Proxy
xDR Proxy
Operations Manager for VM
z/VM V7.3 with SSI
Logical Partition (PR/SM)
LinuxONE Emperor 4 hardware
DS8K (ECKD)
DS8K (ECKD)

Figure 4-16 RHOCP Control-node failure

Case 15: RHOCP Compute node restart control-node failure

Failing Component	What happens	Mitigation, workaround, recovery
RHOCP Compute node restart Figure 4-16 on page 114		• workload rescheduled/ restarted on other operational nodes. After network recovery node is accessible and workloads can be scheduled
Impact: Temporary loss of compute related to Control for certain programs; workload redistributed or restarted on other nodes as pertinent.		
Time to recovery of the failing component: 30-60min, assuming automation and presence of backups. (System administrator needs to reinstall from scratch and have the new instantiation of the node rejoin the cluster. No data recovery per se)		

4.3.2 Planned Outages

This section discusses the application of service or new-release installation procedure as a potential disruption to availability, at each level of the LinuxONE availability model.

Case 1: Planned outage for CPS or Storage Unit

Failing Component	What happens	Mitigation, workaround, recovery
Planned outage for CPC or Storage Unit Figure 4-17 on page 116	Hardware down or firmware down for MCL application	• z/VM relocates work to alternate hardware; RHOCP redistributes compute to available nodes. **No outage**. • IBM Hyperswap points workload to alternate storage volume. **No outage.**
Recovery error on CPC or Storage Unit	Hardware remains down	• Workload persists on alternate systems until repair. **No further outage, because workload never rebalanced**

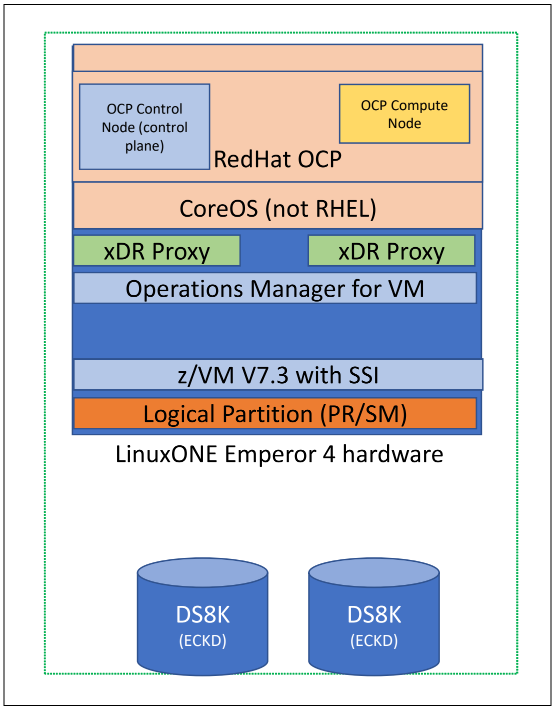
OCP Control Node (control plane)
OCP Compute Node
RedHat OCP
CoreOS (not RHEL)
xDR Proxy
xDR Proxy
Operations Manager for VM
z/VM V7.3 with SSI
Logical Partition (PR/SM)
LinuxONE Emperor 4 hardware
DS8K (ECKD)
DS8K (ECKD)

Figure 4-17 Planned Outage for CPC or Storage

Case 2: Planned outage for z/VM

<table>
<tr><th>Failing Component</th><th>What happens</th><th>Mitigation, workaround, recovery</th></tr>
<tr><td>Planned outage for z/VM
Figure 4-17</td><td>1 z/VM system down for PTF application</td><td>• z/VM admin (or GDPS) relocates work to alternate z/VM system(s). RHOCP redistributes compute to available nodes. <b>No outage</b></td></tr>
<tr><td colspan="3">• 4 hours to complete a new install of z/VM release throughout an SSI
• 5 minutes to install a service PTF on one member of an SSI.
• Note: assumes all planning has been done in advance.</td></tr>
<tr><td>Recovery error on CPC or Storage Unit</td><td>z/VM cluster goes into SAFE mode, isolates one system</td><td>• <b>No outage,</b> but communication problems with workload running in SAFE system until repair.</td></tr>
<tr><td colspan="3">• https://www.ibm.com/docs/en/zvm/7.3?topic=members-removing-service
• https://www.ibm.com/docs/en/zvm/7.3?topic=summaries-vmfrem-exec
• https://www.youtube.com/watch?v=ISN39CWEk7k</td></tr>
</table>

Case 3: Planned outage for RHOCP

Failing Component	What happens	Mitigation, workaround, recovery
Planned outage for RHOCP Figure 4-17 on page 116	Reboot of control plane. (Service applied in background.)	• **No outage.** • GDPS can initiate guest mobility of can be relocated (VMRELOCATE)
Recovery error on RHOCP	Problems with cluster (control, compute)	• On control nodes, no outage up to a single node down. If two control nodes down, the workloads would still run but on degraded mode until the recovery of the failed nodes [manual recovery process – review notes section] • On compute nodes, no outage as long other compute nodes are available, and the application is scaled up on all running compute nodes. [manual recovery process – review notes section]

Case 4: Planned outage for GDPS xDR

Failing Component	What happens	Mitigation, workaround, recovery
Planned outage for RHOCP Figure 4-17 on page 116	n/a	• **No outage.** • xDR proxies can be recycled one at a time after maintenance; system never without local agent. • Similarly, multiple z/OS K systems, update one at a time, never without GDPS xDR.)
Recovery error on GDPS xDR	Problems with storage synchronization, lack of failover support until mitigated	• Recycle / reboot • No known cases of this occurring.

4.4 Disaster Recovery

This section discusses scenarios such as loss of entire data center or regions due to natural disasters, power outages or human-generated catastrophes; also other events that can cause availability loss.

Generally, physical data centers (and full recovery thereof) are measured at five 9’s of availability (99.999%)

- Data center providers will have SLAs around availability and resources for same
- Calculating availability, based upon data centers around the world, is less reliable than for hardware/software.
- Standards vary in adoption and locality, but f.ex ANSO/BICSI 002-2019.
- Tiered approach based upon how much a client can tolerate (or afford).

For lower-extreme problems (water, electricity, air conditioning), it’s understood that data centers are meant to provide duplication of resource, back-up generators, alternate lines, disparate power grids (for maintenance / planned outages).

- Rate of failure understood to be low
- However, due to state of disaster, restoring the physical assets may be extremely difficult / impossible.

Having geographically dispersed data centers minimizes the chance of a single disaster hitting both data centers simultaneously.

- Having data replication via GDPS matters – whether or not the presented Framework is configured Active-Active.

Figure 4-18 shows an example of an entire site datacenter (Site B) outage.

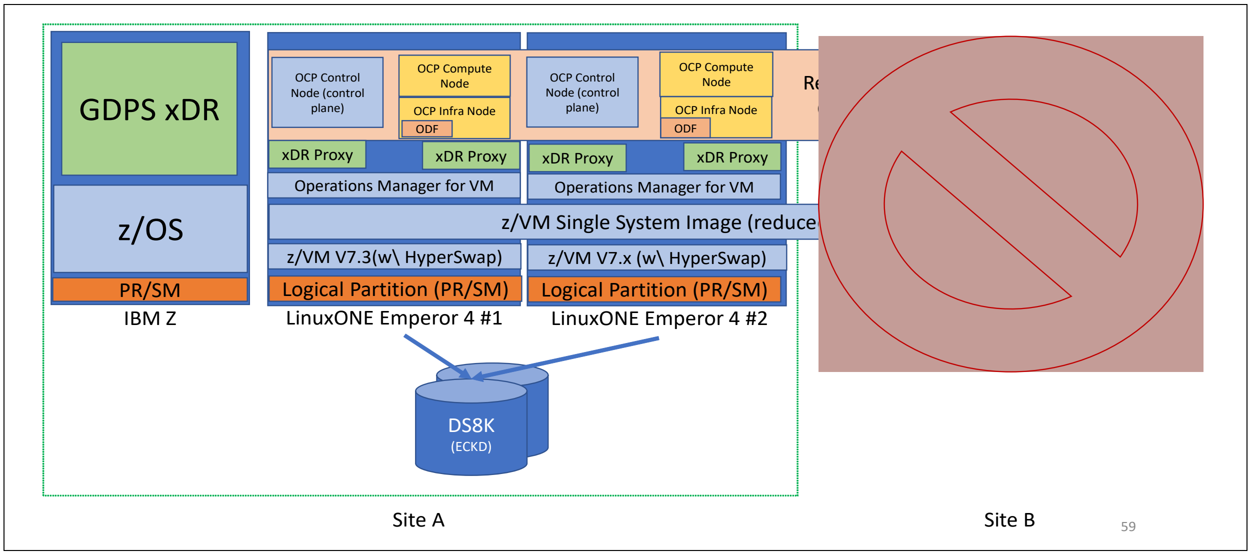

Figure 4-18 Site B has a datacenter outage

Case 1: Data Center Outage

Failing Component	What happens	Mitigation, workaround, recovery
One entire datacenter Figure 4-18, and Figure 4-19 on page 119	Loss of all compute, network, and storage in that building	- Per previous slides, redundant electrical, water, power, infrastructure to keep data center running - GDPS restarts virtual machines on surviving cluster members, or z/VM itself in new partitions - GDPS points compute to surviving storage - z/VM can restart virtual machines on surviving cluster members - RHOCP will redirect workload onto surviving compute nodes when it detects loss of Compute or Control virtual machines
Impact: reduced white space overhead for future failover, reduced redundancy opportunities for compute and storage; change in network latency patterns. - Since this is an unplanned outage, restarting multiple systems worth of workload may take time - RHOCP will restart applications (or failover to backups) as pertinent - z/VM Operations Manager will need to trigger VM IPL (minutes) **Time to recovery of the failing component:** depends on nature of disaster; somewhere between days, weeks, and years. Workload can continue, but at greater risk if other problems occur.		

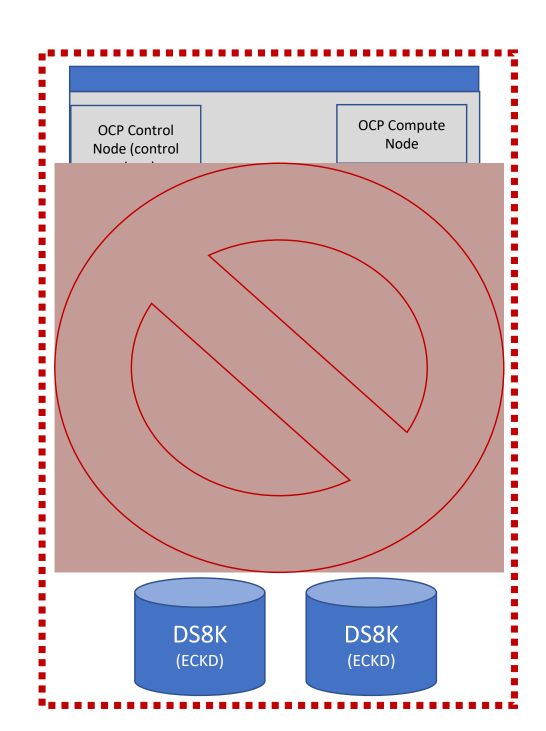

Figure 4-19 Data Center Outage

The framework covered in this chapter illustrates continuous operation of RHOCP workload in the context of ~3 partitions, backed by GDPS xDR:

- A robust Disaster Recovery Plan will often have reserve logical partitions available Framework does not consume three full CPCs
- Other partitions will be under use for other work, development work, test work, etc
- Given the right System Configuration, a z/VM SSI member can be IPL'd in a new location

Based upon availability and DR requirements, consider having a failover plan for z/VM SSI member nodes:

- This mitigates the damage another outage might incur
- This lessens the time an SSI may be running closer to a memory 'ceiling'
- Partition failover allows RHOCP to run at "full strength" while physical site is under repair

4.5 For more information...

Getting Started with IBM Z Resiliency:

https://www.redbooks.ibm.com/redbooks/pdfs/sg248446.pdf

End-to-End High Availability Solution for Linux on IBM Z:
https://www.redbooks.ibm.com/redbooks/pdfs/sg248233.pdf

Building an OpenShift Environment on IBM Z:
https://www.redbooks.ibm.com/redbooks/pdfs/sg248515.pdf

IBM GDPS: Introduction to Concept and Capabilities:
https://www.redbooks.ibm.com/redbooks/pdfs/sg246374.pdf

Linux-HA project Open source:
https://open-source-guide.com/en/Solutions/Infrastructure/High-availability/Linux-ha

Suse SLES 11 SP2 High Availability Guide:
https://suse.com/documentation/sle_ha/pdfdoc/book_sleha/book_sleha.pdf

Tivoli System Automation for Multiplatforms:
https://www-01.ibm.com/software/tivoli/products/sys-auto-multi

Solutions Assurance Team has some whitepapers and redpapers (around z/VM, around Linux HA, around OCP):
https://www.ibm.com/docs/en/linux-on-systems?topic=assurance-solution-papers

Abbreviations and acronyms

4700	400 m
ADDI	Application Discovery and Delivery Intelligence
AES	Advanced Encryption Standard
AESKW	AES Key Warp
AI	Artificial Intelligence
AID	Adapter ID
AIU	Accelerator for AI
ANSI	American National Standards Institute
AORs	application-owning regions
AP	Adjunct Processor
API	an application programming interface
APN	AusPayNet
APXA	AP extended addressing
ASCE	Address Space Control Element
ASHRAE	Air-Conditioning Engineers
ASICs	application-specific integrated circuits
ATLAS	Automatically Tuned Linear Algebra Software
AWLC	Advanced Workload License Charges
BCD	Binary Coded Decimal
BCD	binary-coded decimal
BCP	Base Control Program
BEAR	Breaking Event Address Register
BHT	branch history table
BMC	Base Management Card
BMC	Baseboard Management Card
BMC	Baseboard Management Cards
BMC	Baseboard Management Controllers
BMCs	Base Management Cards
BMCs	Baseboard Management Cards
BMCs	Baseboard Management Controllers
BNC	Bayonet Neill-Concelman
BTB	branch target buffer
BTS	Backup Time Server
CA	certificate authority
CAC	Common Access Card
CAT	Crypto Analytics Monitor
CAT	Crypto Analytics Tool
CBR	concurrent book repair
CBU	Capacity Back Up
CBU	Capacity BackUp
CBU	Capacity Backup
CBU	Capacity Backup Upgrade
CCA	Common Cryptographic Architecture
CCW	channel command word
CCWs	channel program
CDM	Concurrent Driver Maintenance
CDP	Common Data Provider
CDR	Concurrent Drawer Repair
CDR	Concurrent Drawer Replacement
CDR	concurrent drawer replacement
CDU	Concurrent Driver Upgrade
CEE	Converged Enhanced Ethernet
CEX6	Crypto Express7S (CEX7) and CryptoExpress6S
CEX7	Crypto Express8S (CEX8), Crypto Express7S
CEX8	Crypto Express8S
CEX8A	Crypto Express7S Accelerator
CEX8S	Crypto Express8S
CF	Coupling Facility
CF	Coupling facility
CF	coupling facility
CFCC	Coupling Facility Control Code

CHIM	Cryptographic Hardware Initialization and Maintenance
CHPID	channel-path identifier
CHPIDs	channel path identifiers
CHPIDs	channel-path identifiers
CI	control interval
CIM	Common Information Model
CISC	Complex Instruction Set Computer
CIU	Customer Initiated Upgrade
CKDS	Cryptographic Key Data Set
CLI	command line-interface
CMAC	Cipher-based Message Authentication Code
CMLC	Country Multiplex License Charges
CMP	Country Multiplex Pricing
CMPSC	Co-processor Compression Enhancements
CMT	CHPID Mapping Tool
CMT	CHPID mapping tool
CP	Central Processor
CP	central processor
CPACF	CP Assist for Cryptographic Function
CPACF	CP Assist for Cryptographic Functions
CPACF	Central Processor Assist for Cryptographic Function
CPC	central processing complex
CPC	central processor complex
CPC1	CPC drawer
CPD	Capacity Provisioning Domain
CPE	Capacity for Planned Event
CPI	cycles per instruction
CPM	Capacity Provisioning Manager
CPMC	Capacity Provisioning Management Console
CPP	Capacity Provisioning Policy
CPUMF	CPU Measurement Facility
CPs	central processors
CRC	Cyclic Redundancy Checking
CRO	Continuous Reliable Operation
CRT	Chinese Remainder Theorem
CS	Communications Server
CSM	Copy Services Manager
CSS	channel subsystem
CSSs	channel subsystems
CTB	Changing Target Buffer
CTC	communication across channels, channel-to-channel
CTN	Coordinated Time Network
CTN	Coordinated Timing Network
CTS	Current Time Server
CUs	control units
CX4	ConnectX4
CoD	Capacity on Demand
CoD	Capacity on-Demand
CoD	capacity on demand
CoP	Co-Processor
DASD	direct access storage device
DAT	Dynamic Address Translation
DAT	dynamic address translation
DBR	Device-based routing
DBaaS	Data Base as a Service
DCIM	Data Center Infrastructure Management
DCM	Dual Chip Module
DCM	dual chip module
DDS	Distributed Data Server
DED	dedicated
DES	Data Encryption Standard
DFP	decimal floating point
DFSMS	Data Facility Storage Management Subsystem
DHCP	Dynamic Host Configuration Protocol
DIF	Data Integrity Field
DIMM	Dual inline memory module
DK	Deutsche Kreditwirtschaft
DLC	Deep Learning Compiler
DMD	Dynamic Memory Downgrade

DMR	dynamic memory relocation
DNS	domain name server
DPM	Dynamic Partition Manager
DPM	Dynamic Partition Manger
DPM	Dynamic Partition Mode
DRAM	dynamic random access memory
DRDA	Distributed Relational Database Architecture
DRNG	Deterministic Random Number Generation
DRNG	Deterministic random number generator
DSP	Data Secure Platform
DSR	dynamic storage reconfiguration
EA	Extended addressability
EAV	extended address volume
EBA	enhanced book availability
EBR	example, Brocade's exchange-based routing
EC	engineering change
ECAR	Enhanced Console Assisted Recovery
ECC	Elliptic Curve Cryptography
ECC	Elliptic curves cryptography
ECC	Error Correction Code
ECC	error-correction code
ECDH	Elliptic Curve Diffie-Hellman
ECDSA	Elliptic Curve Digital Signature Algorithm
EDA	Enhanced Drawer Availability
EDA	enhanced drawer availability
EDM	Enhanced Driver Maintenance
EF	extended format
ELS	Extended Link Service
EMV	Europay, MasterCard, and Visa
EP11	Enterprise PKCS #11
EP11	Enterprise PKCS #11
ETS	External Time Source
ETS	external time source
EUV	extreme ultraviolet
EXCPVR	Executing Fixed Channel Programs in Real Storage
FB	fixed block
FC	Fibre Channel
FC	feature code
FCP	Fibre Channel Protocol
FEC	Forward Error Correction
FFDC	First Failure Data Capture
FICON	Fiber Connection
FICON	Fibre Channel connection
FICON	Fibre Connection
FIDR	FICON Dynamic Routing
FIDs	Functions ID
FIPS	Federal Information Processing Standard
FIPS	Federal Information Processing Standards
FLOGI	fabric login
FPE	Format Preserving Encryption
FPRs	file overlays the floating-point registers
FQC	Fiber Quick Connector
FRU	Field Replaceable Unit
FRU	field-replaceable unit
FSPs	Flexible Support Processors
FWLC	Flat Workload License Charges
FXU	Fixed-point unit
GBIC	German Banking Industry Commission
GCM	Galois Counter Mode
GCT	Global Completion Table
GDE	Guardium® Data Encryption
GRS	global resource serialization
GSF	Guarded Storage Facility
GTps	gigatransfers per second
GUI	graphical user interfaces
GVRP	GARP VLAN Registration Protocol
GbE	Gigabit Ethernet
Gbps	gigabits per second

HADR	high availability and disaster recovery
HCD	hardware configuration definition
HCM	Hardware Configuration Manager
HDD	hard disk drive
HFP	Hexadecimal Floating Point
HIS	Hardware Instrumentation Services
HMA	Hardware Management Appliance
HMAC	Hash-based Message Authentication Code
HMC	Hardware Management Console
HMCs	Hardware Management Consoles
HPVS	Hyper Protect Virtual Servers
HSA	Hardware System Area
HSA	hardware system area
HSM	Hardware Security Module
HSM	hardware security module
HTAP	hybrid transaction and analytic processing
IBM	International Business Machines Corporation
IC	Internal Coupling
ICA	Integrated Coupling Adapter
ICA	integrated communication adapter
ICF	Internal Coupling Facility
ICF	internal coupling facility
ICM	Instruction cache and merge
ICSF	Integrated Cryptographic Service Facility
IDAA	IBM Db2® Analytics Accelerator
IDU	Instruction decode unit
IEP	Instruction Execution Protection
IFB	Instruction fetch and branch
IFL	Integrated Facility for Linux
IFP	Integrated Firmware Processor
IFP	integrated firmware processor
IFU	Instruction fetching unit
ILP	instruction-level parallelism
IML	initial machine load
IMPP	Installation Manual for Physical Planning
IMS	IBM Information Management System
IOCDS	in I/O configuration data set
IOCDS	input/output configuration data set
IOCP	I/O configuration program
IOCS	I/O configuration source
IODF	IPL, an I/O definition file
IODFs	I/O definition files
IP	Internet Protocol
IPC	Instructions Per Cycle
IPL	Initial Program Load
IPL	initial program load
IPLA	International Program License Agreement
IPSec	Internet Protocol Security
IPv4	Internet Protocol version 4
IPv6	Internet Protocol version 6
IRD	Intelligent Resource Director
ISA	Instruction Set Architecture
ISL	inter-switch link
ISLs	inter-switch links
ISM	Internal Shared Memory
ISO	International Organization for Standardization
ISU	Instruction sequence unit
ISV	including independent software vendor
ITRRs	internal throughput rate ratios
IUs	information units
IWQ	Inbound workload queuing
JRE	Java Runtime Environment
JVM	Java virtual machine
KCV	Key Check Value
KMM	Keyboard Mouse Monitor
L1	level 1
L2	level 2
L3	Level 3
L4	Level 4

LAG	Link Aggregation Port Group
LAN	local area network
LBA	Logical Block Address
LCSS	logical channel subsystem
LCSSs	logical channel subsystems
LCUs	logical control units
LDAP	Lightweight Directory Access Protocol
LGR	live guest relocation
LIC	Licensed Internal Code
LICC	Licensed Internal Control Code
LICCC	Licensed Internal Code Configuration Control
LPAR	logical partition
LR	Long Reach
LRECL	length
LRU	least recently used
LSPR	Large System Performance Reference
LSU	Load-store unit
LUKS	Linux Unified Key Setup
MA	Modulo arithmetic
MAC	Media Access Control
MAC	message authentication code
MACs	message authentication codes
MCFs	more Microcode Fixes
MCI	Model Capacity Identifier
MCI	Model capacity identifier
MCL	Microcode Change Level
MCLs	Microcode Change Levels
MCU	Memory Control Unit
MCUs	memory control units
MCUs	memory controller units
MD5	message-digest
MDIX	medium-dependent interface crossover
ME	Modulus-Exponent
MES	Miscellaneous equipment specification
MESI	Modified, Exclusive, Shared, Invalid
MFA	Multi-Factor Authentication
MIDAW	Modified Indirect Data Address Word
MIF	multiple image facility
MIPS	millions of instructions per second
MLC	monthly license charge
MM	Multi Mode
MM	multimode
MPCI	Model Permanent Capacity Identifier
MPO	multifiber push-on
MRP	Mod_Raised_to Power
MSA	Message-Security Assist
MSS	Multiple Subchannel Set
MSS	multiple subchannel sets
MSU	millions of service units
MSUs	millions of service units
MTCI	Model Temporary Capacity Identifier
MzNALC	Multiplex License Charges (CMLC), Multiplex zNALC
NIAP	National Information Assurance Partnership
NICs	Network Interface Cards
NNPA	new Neural Networks Processing Assist
NPIV	N_Port ID Virtualization
NTP	Network Time Protocol
NXU	Nest Accelerator Unit
NXU	Nest Compression Accelerator
NXU	nest compression accelerator
OAT	OSA Address Table
OLM	optimized latency mode
OLS	Offline Signal
OLTP	optimizations for online transaction processing
OMI	Open Memory Interface
ONNX	Open Neural Network Exchange
OOCoD	On/Off Capacity on Demand
OOO	Out-of-order
ORB	operation request block

OSA	Open Systems Adapter
OSC	Operand Store Compare
OSC	Oscillator card
OTC	one-time-charge
OoO	Out-of-Order
OxID	originator exchange ID
P2PE	Point to Point Encryption
PAV	parallel access volume
PBR	port-based routing
PBU	PCIe Bridge Unit
PC	Pervasive Core unit
PCHID	physical channel ID
PCHID	physical channel identifier
PCHIDs	physical channel IDs
PCI	Payment Card Industry
PCI	Peripheral Component Express
PCI	Processor Capacity Index
PCIe	Peripheral Component Interconnect Express
PCIeCC	PCIe Cryptographic Co-processor
PCIeCC	PCIe Cryptographic Coprocessor
PCIeCC	Peripheral Component Interconnect Express (PCIe) cryptographic coprocessors
PCIs	program controlled interrupts
PDSE	partitioned data set extended
PDU	Power Distribution Unit
PE	Processing Elements
PER	permanent
PFC	Priority flow control
PHT	Pattern History Table
PHT	pattern history table
PI	Performance Index
PIV	Personal Identity Verification
PKCS	Public Key Cryptography Standards
PKD	Public Key Decrypt
PKE	Public Key Encrypt
PNG	Prime Number Generator
POL	point of load
POL	point-of-load
POR	Power On Reset
POR	power-on reset
POR	power-on-reset
PPC	Processor Power Control
PPC	Processor Power Control Cards
PPC	Processor Power Control cards
PPCs	Processor Power Control Cards
PPRC	Peer-to-Peer Remote Copy
PPS	Pulse Per Second
PPS	pulse per second
PRNG	Pseudo random number generator
PRNG	Pseudo-Random Number Generator
PSIFB	Parallel Sysplex InfiniBand
PSLC	Parallel Sysplex License Charges
PSLC	Parallel Sysplex license charge
PSP	Preventive Service Planning
PSU	Power Supply Units
PSUs	Power Supply Units
PSUs	power supply units
PSW	Program Status Word
PT	Processor Tiles
PTF	program temporary fix
PTFs	program temporary fixes
PTP	Precision Time Protocol
PTS	PIN® Transaction
PTS	PIN Transaction Security
PTS	Preferred Time Server
PTS	Primary Time Server
PU	Processing Unit
PU	Processor Unit
PU	Processor unit
PU	processor unit
QSC	Quantum-Safe Cryptography
RADIUS	Remote Authentication Dia-In User Service

RADIUS	Remote Authentication Dia-in User Service
RAID	Redundant Array of Independent Disks
RAIM	Redundant Array of Independent Memory
RAIM	redundant array of independent memory
RAS	reliability, availability, and serviceability
RAS	reliability, availability, serviceability
RAs	Repair Actions
RCL	Remote Code Load
RDMA	Remote Direct Memory Access
RDP	Read Diagnostic Parameters
RG	Resource Group
RG	resource group
RG3	resource group 3
RGs	resource groups
RHEL	Red Hat Enterprise Linux
RI	Runtime Instrumentation
RII	Redundant I/O Interconnect
RII	redundant I/O interconnect
RMF	Resource Measurement Facility
RNG	Random Number Generator
RNI	Relative Nest Intensity
RNI	Relative nest intensity
RNID	request node identification data
RSF	Remote Support Facility
RSM	Real Storage Manager
RSU	reconfigurable storage unit
RTM	Recovery Termination Manager
RU	recovery unit
RaP	Report a Problem
RoCE	RDMA over CEE
Rx	Receive
SA	System Automation
SADMP	standalone dump
SAN	Storage Area Network
SAN	storage area network
SAP	system assist processor

SAPs	System Assist Processors
SC	System Controller
SCM	Storage Class Memory
SCRT	subcapacity reporting tool
SDM	system data mover
SDSF	System Display and Search Facility
SE	Support Element
SEs	Support Elements
SFP	Small Form-Factor Pluggable
SFP	Special Function Processors
SHA	Secure Hash Algorithm
SHR	shared
SIGA	Signal Adapter
SIMD	Single Instruction Multiple Data
SIMD	Single-instruction multiple-data
SLES	SUSE Linux Enterprise Server
SMC	Shared Memory Communication
SMCv2	SMC Version 2
SMF	System Management Facilities
SMP	symmetric multiprocessing
SMP	symmetric multiprocessor
SMP9	symmetric multiprocessing
SMT	simultaneous multithreading
SN	serial number
SNMP	Simple Network Management Protocol
SOO	Single Object Operations
SOOs	Single Object Operations
SORTL	SORT LISTS

SR	Short Reach
SR	short reach
SRAM	static random access memory
SRB	Service Request Block
SRB	System recovery Boost
SS	subchannel set
SS1	subchannel set 1
SS2	subchannel set 2

SS3	subchannel set 3
SSC	Secure Service Container
SSH	Secure Shell
SSI	Single System Image
SSI	subsystem interface
SSID	subchannel set ID
SSL	Secure Sockets Layer
SSL	Sockets Layer cryptographic protocol
SSR	System Service Representative
SSRs	support IBM service support representatives
STIDP	Store CPU ID
STP	Server Time Protocol
STSI	Store System Information
SYID	system identifier
SYSIB	system information block
SYSIB	system-information block
SoD	Statement of Direction
TCP	Transmission Control Protocol
TDES	Triple Data Encryption Standard
TDES	Triple-DES
TDES	triple-length (192-bit) Triple-DES
TER	temporary
TFP	Tailored Fit Pricing
TFP	Taylor Fit Pricing
TKE	Trusted Key Entry
TLB	Transaction Lookaside Buffer
TLB	translation lookaside buffer
TLBs	translation lookaside buffers
TLS	Transport Layer Security
TM	Telum
TOTP	Time-based One-time Password
TPM	Trusted Platform Module
TRNG	True Random Number Generation
TRNG	True random number generator
TSAD	Transmit System Availability Data
TTOs	Technology Transition Offerings
TX	Transactional Execution
Telum	two PU chips
Tx	Transmit
UDP	User Datagram Protocol
UDX	User Defined Extension
UDX	User-defined extension
UFD	USB flash memory drive
UTP	unshielded twisted pair
VCHID	virtual CHPID
VF	virtual function
VFM	Virtual Flash Memory
VFPE	Visa Format Preserving Encryption
VFs	Virtual Functions
VFs	virtual functions
VIPA	virtual Internet Protocol address
VMAC	virtual MAC
VPD	vital product data
VPN	virtual private network
VRMs	Voltage Regulator Modules
WBI	WebSphere® Business Integration
WDM	Wavelength Division Multiplexer
WLC	Workload License Charges
WLM	Workload Manager
WMLz	Watson Machine Learning for z/OS
WSAPI	Web Services API
WWPN	Worldwide port name
WWPNs	worldwide port names
iPDU	Intelligent Power Distribution Unit-based power
iPDUs	intelligent Power Distribution Units
iPDUs	intelligent power distribution units
zAAPs	Z Application Assist Processors
zBNA	Z Batch Network Analyzer
zBNA	z Systems® Batch Network Analyzer
zCX	z/OS Container Extensions
zDAC	z/OS Discovery and Auto Configuration
zEDC	z Enterprise Data Compression
zEDC	zEnterprise® Data Compression
zELC	zSeries Entry License Charges
zELC	zSystems Entry License Charges

zERT	z/OS encryption readiness technology
zFS	z/OS file system
zHPF	Z
zIIP	Z Integrated Information Processor
zIIPs	Z Integrated Information Processors
zIIPs	z Integrated Information Processors
zNALC	z New Application License Charges
zPCR	Z
zSCC	Z Security and Compliance Center

Related publications

The publications listed in this section are considered particularly suitable for a more detailed discussion of the topics covered in this book.

IBM Redbooks

The following IBM Redbooks publications provide additional information about the topics in this document. Note that some publications referenced in this list might be available in softcopy only.

- *IBM GDPS: An Introduction to Concepts and Capabilities,* SG24-6374
- *IBM z16 Configuration Setup,* SG24-8960-01
- *IBM Z Connectivity Handbook,* SG24-5444
- *Getting Started with IBM Z Resiliency,* SG24-8446

You can search for, view, download or order these documents and other Redbooks, Redpapers, Web Docs, draft and additional materials, at the following website:

ibm.com/redbooks

Other publications

These publications are also relevant as further information sources:

- Hardware Management Console (HMC) Operations Guide Version 2.16.0.
  See IBM Resource Link (requires IBM ID authentication).
- IBM Hardware Management Console Help

Online resources

The IBM Resource Link for documentation and tools website is also relevant as another information source:

http://www.ibm.com/servers/resourcelink

Help from IBM

IBM Support and downloads

ibm.com/support

To determine the spine width of a book, you divide the paper PPI into the number of pages in the book. An example is a 250 page book using Plainfield opaque 50# smooth which has a PPI of 526. Divided 250 by 526 which equals a spine width of .4752". In this case, you would use the .5” spine. Now select the Spine width for the book and hide the others: **Special>Conditional Text>Show/Hide>SpineSize(-->Hide:)>Set** . Move the changed Conditional text settings to all files in your book by opening the book file with the spine.fm still open and **File>Import>Formats** the Conditional Text Settings (ONLY!) to the book files.

Redbooks

IBM LinuxONE Resiliency

SG24-8544-00

ISBN DocISBN

(1.5” spine)
1.5”<-> 1.998”
789 <->1051 pages

Redbooks

IBM LinuxONE Resiliency

SG24-8544-00

ISBN DocISBN

(1.0” spine)
0.875”<->1.498”
460 <-> 788 pages

Redbooks

IBM LinuxONE Resiliency

SG24-8544-00

ISBN DocISBN

(0.5” spine)
0.475”<->0.873”
250 <-> 459 pages

Redbooks

IBM LinuxONE Resiliency

(0.2”spine)
0.17”<->0.473”
90<->249 pages

(0.1”spine)
0.1”<->0.169”
53<->89 pages

To determine the spine width of a book, you divide the paper PPI into the number of pages in the book. An example is a 250 page book using Plainfield opaque 50# smooth which has a PPI of 526. Divided 250 by 526 which equals a spine width of .4752". In this case, you would use the .5" spine. Now select the Spine width for the book and hide the others: **Special>Conditional Text>Show/Hide>SpineSize(-->Hide:)>Set** . Move the changed Conditional text settings to all files in your book by opening the book file with the spine.fm still open and **File>Import>Formats** the Conditional Text Settings (ONLY!) to the book files.

Redbooks

IBM LinuxONE Resiliency

SG24-8544-00

ISBN DocISBN

IBM

(2.5" spine)
2.5"<->nnn.n"
1315<-> nnnn pages

Redbooks

IBM LinuxONE Resiliency

SG24-8544-00

ISBN DocISBN

IBM

(2.0" spine)
2.0" <-> 2.498"
1052 <-> 1314 pages

SG24-8544-00

ISBN DocISBN

Printed in U.S.A.