Redbooks

ibm.com/redbooks

Temenos on IBM LinuxONE Best Practices Guide

Vic Cross

Ernest Horn

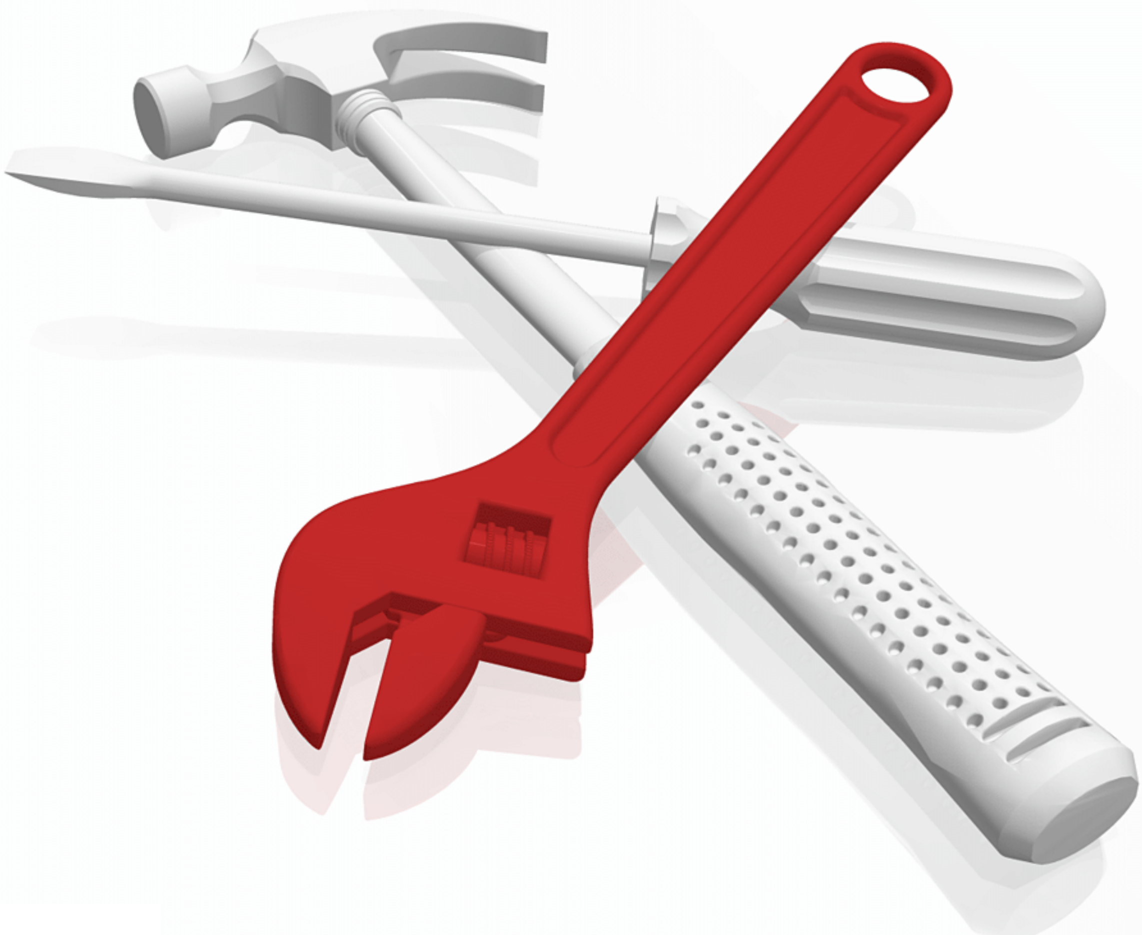
Colin Page

Jonathan Page

Robert Schulz

John Smith

Chris Vogan



IBM

Redbooks

IBM Redbooks

Temenos on IBM LinuxONE Best Practices Guide

February 2020

SG24-8462-00

Note: Before using this information and the product it supports, read the information in "Notices" on page vii.

First Edition (February 2020)

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.

COPYRIGHT LICENSE TEMENOS CONTENT:

Temenos contributed content and sections to this publication: Copyright Temenos Headquarters SA. Reprinted by Permission

Trademarks

Preface

The world's most successful banks run on IBM®, and increasingly IBM LinuxONE. Temenos, the global leader in banking software, has worked alongside IBM for many years on banking deployments of all sizes. This book marks an important milestone in that partnership. Temenos on IBM LinuxONE Best Practices Guide shows financial organizations how they can combine the power and flexibility of the Temenos solution with the IBM platform that is purpose built for the digital revolution.

Authors

This book was produced by a team of specialists from around the world working at IBM Redbooks Centers in Raleigh, NC and Montpellier, France.

Vic Cross is an IT Specialist living in Brisbane, Australia. Vic works with IBM Systems Lab Services, where he provides senior design and implementation expertise to IBM Z® and IBM LinuxONE projects across Asia-Pacific. He has 30 years of experience in general IT, 25 of which has been directly related to the IBM Z and IBM LinuxONE platforms and their antecedents. Vic holds a degree in Computer Science from Queensland University of Technology. He is especially interested in networking, security, and high availability. Vic has written and contributed to several IBM Redbooks® publications, including *Securing Your Cloud: IBM z/VM Security for IBM z Systems and LinuxONE*, SG24-8353 and *Linux on IBM eServer zSeries and S/390: Virtual Router Redundancy Protocol on VM Guest LANs*, REDP-3657.

Ernest Horn is the Worldwide LinuxONE Client Success Manager based out of Poughkeepsie, New York.   He has 37 years working at IBM. Systems Architecture is one of his areas of expertise, particularly on the IBM LinuxONE platform, which he has been involved with since its inception. He has consulted and implemented IBM LinuxONE environments for customers all over the world, several of which were new to the IBM LinuxONE platform. He has written two Redbooks: *IBM Wave Management Software* and *The LinuxONE Virtualization Cookbook*.

Colin Page is an Enterprise Architect based in Dubai, United Arab Emirates. He has over 30 years of experience in Banking and IBM Z technologies. He has worked at IBM for 24 years. His areas of expertise include IBM Db2®, IBM WebSphere®, Linux on z, systems architecture and various core banking and payment solutions. He presents on a variety of IBM Z based topics across the Middle East and Africa region. He has also written a number of white papers and Redbooks on data analytics and Banking solutions.

Jonathan Page is a senior technical writer with the Temenos Jumpstart team, based in the UK. He has 20 years' experience as a technical writer, producing documentation and technical marketing materials for a wide range of hardware and software organizations. Jonathan has an MA in Critical Theory from UEA in the UK and writes literary fiction in his spare time. His short stories have won a number of prizes over the years, including the Hay Writers Prize 2018.

Robert Schulz is a certified IT specialist in Austria. He has 30 years of experience in IBM Z for banking, government, and retail with a high degree of customer interactions from planning to implementation and support. His areas of expertise include all operating systems on IBM Z, networking, performance, and high availability solutions. He was also co-author on other IBM Z based Redbooks.

John Smith is the WW offering lead for Temenos on LinuxONE, based out of Brighton in the UK. He has 25 years of experience in Financial services field. John's degree is in marketing and has held many sales and management positions across a number of organizations, before he started at IBM in 2013. His main area of expertise is creating value propositions for IBM technology for clients and for independent software vendors within the Financial services sector.

Chris Vogan is a Solution Architect supporting Temenos on LinuxONE in Austin, Texas. He has 20 years of experience in IBM Z storage including 5 years of experience supporting banking clients. He holds a degree in Computer Science from Northern Illinois University. His areas of expertise include enterprise storage implementation and distribute storage solutions for IBM Z. He has authored several blog posts on sharing IBM z/OS® data and storage with distributed systems.

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

Robert Schulz, Certified IT Specialist and Author, Austria IBM
Robert was more than an author for this publication. His dedication, technology acumen and focused writing carried this book to publication.

Deana Coble, Project Leader for IBM Redbooks
IBM Redbooks, Raleigh Center

Rick Pekosh, Washington Systems Center - Storage, IBM DS8000® SME
Rick consulted with the team for the publication of this book.

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.

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 Facebook:

  http://www.facebook.com/IBMRedbooks

- Follow us on Twitter:

  http://twitter.com/ibmredbooks

- Look for us on LinkedIn:

  http://www.linkedin.com/groups?home=&gid=2130806

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

  https://www.redbooks.ibm.com/Redbooks.nsf/subscribe?OpenForm

- Stay current on recent Redbooks publications with RSS Feeds:

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

1

Introduction

The world's most successful banks run on IBM, and increasingly IBM LinuxONE. Temenos, the global leader in banking software, has worked alongside IBM for many years on banking deployments of all sizes. This book marks an important milestone in that partnership. Temenos on IBM LinuxONE Best Practices Guide shows financial organizations how they can combine the power and flexibility of the Temenos solution with the IBM platform that is purpose built for the digital revolution.

The purpose of this IBM Redbooks Publication is to:

- Introduce the Temenos solution and IBM LinuxOne
- Provide high-level design architecture
- Describe deployment best practices
- Provide a guide to getting started with Temenos on IBM LinuxONE

The following topics are covered in this chapter:

1.1 Introduction to Temenos and IBM LinuxONE

Temenos is the global leader in the provision of banking software, in both sales and industry ratings. When Temenos engages with larger financial organizations, Temenos recognize that these institutions need platforms that are both highly resilient and scalable for unknown growth. IBM LinuxONE has the heritage in Tier 1 banks and provides a secure platform for Temenos across Tier 1 to Tier 3 banks.

1.1.1 Temenos

Temenos is a world leader in banking software. Over 3000 clients across the globe, including 41 of the top 50 banks, use Temenos to deliver banking services to more than 500 million customers.

Temenos' objective is to provide financial organizations of all sizes with the software they need to thrive in the new era of Open Banking, instant payments and cloud. The integrated Temenos platform supports traditional Linux deployments.

Temenos' core banking solutions are centered around two products: Temenos Infinity and Temenos Transact. Both solutions give banks the most complete set of digital front office and core banking capabilities. Using the latest cloud-native, cloud-agnostic technology, banks rapidly and elastically scale, benefiting from the highest levels of security and multi-cloud resilience, generating significant infrastructure savings. Advanced API-first technology is coupled with leading design-led thinking and continuous deployment. As a result, banks are empowered to rapidly innovate, connecting to ecosystems and enabling developers to build in the morning and consume in the afternoon. These substantial benefits apply to banks whether they are running their software on-premises, on private or public clouds.

Temenos invests 20% of annual revenue in R&D to continue driving technological innovation for clients. Combined with the largest global community of banks, FinTechs, developers and partners in the financial industry, Temenos is leading the digital banking revolution.

Temenos products have helped their top performing clients achieve industry-leading cost-income ratios of 25.2% and returns on equity of 25.0%, which is twice the industry average.

Temenos Infinity and Temenos Transact

Temenos has evolved a product architecture that decouples front and back functionality, permitting separate, faster upgrades, scalability, and better elasticity. The products are native to the cloud and agnostic, Temenos is deployable on any platform. The bank's options remain open, whatever the future holds.

The modular design of both Infinity and Transact, where banking capability is deployable as separate, productized modules, allows Temenos to deliver capability on any scale within the bank's ecosystem.

The deployment of Infinity or Transact is different for every bank. Infinity's marketing and user modules can renew the bank's digital channels by powering a new mobile application. Infinity can also replace the bank's entire front office, through a phased roll-out that protects business continuity. Origination or onboarding can be rolled out first, for example, before the bank migrates its legacy channels and finally its core front office functionality.

Transact brings a similar flexibility to core banking. Transact can perform a set or subset of core processing services for a bank, ranging from retail banking to treasury and Islamic banking. At the same time, the unrivaled depth and range of functionality within Transact makes it the market leader in complete core banking renovations.

Figure 1-1 shows the Infinity and Transact product architecture.

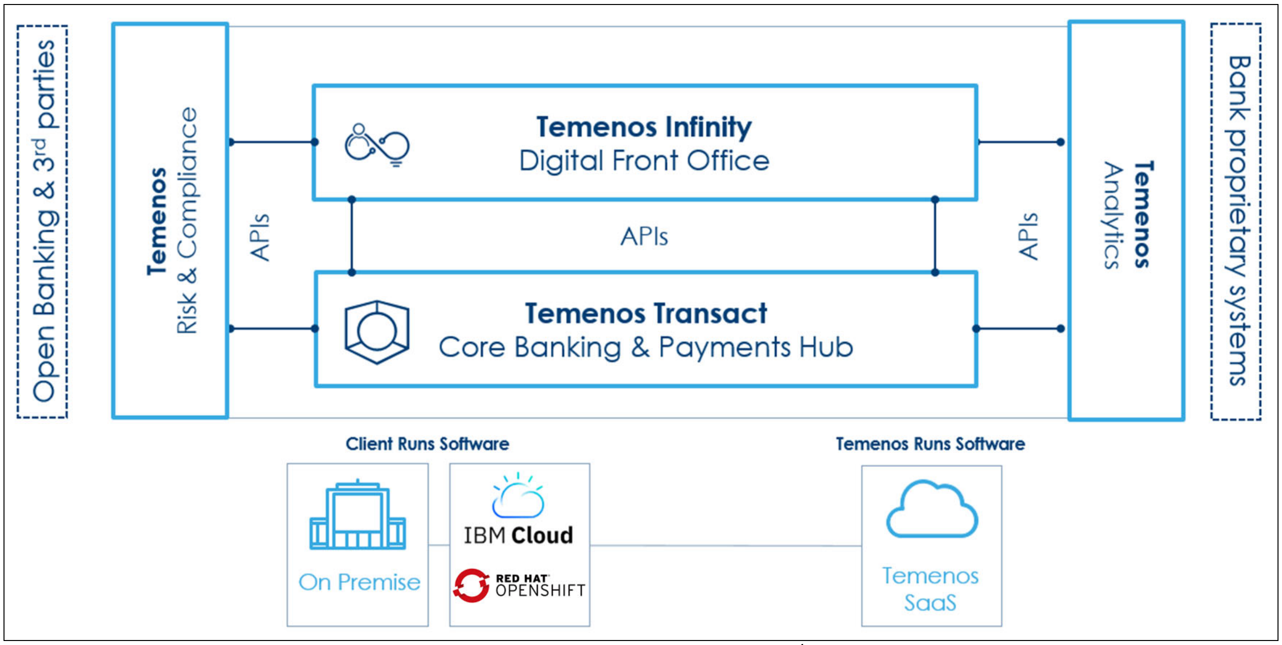


Figure 1-1 Temenos Infinity and Temenos Transact product architecture. [1]

Temenos stacks

Temenos uses preferred software stacks to help clients deploy Temenos banking software quickly. Although every customer is different and might want to use alternative software for certain tasks, the preferred stacks are an ideal starting point for implementations. The preferred stacks are tested and supported by Temenos Runbooks, technical approvals and other customer documentation. Figure 1-2 on page 4 shows four IBM variants on Temenos preferred software stacks, suitable for IBM LinuxONE on-premises deployments.

[1] Courtesy of Temenos Headquarters SA

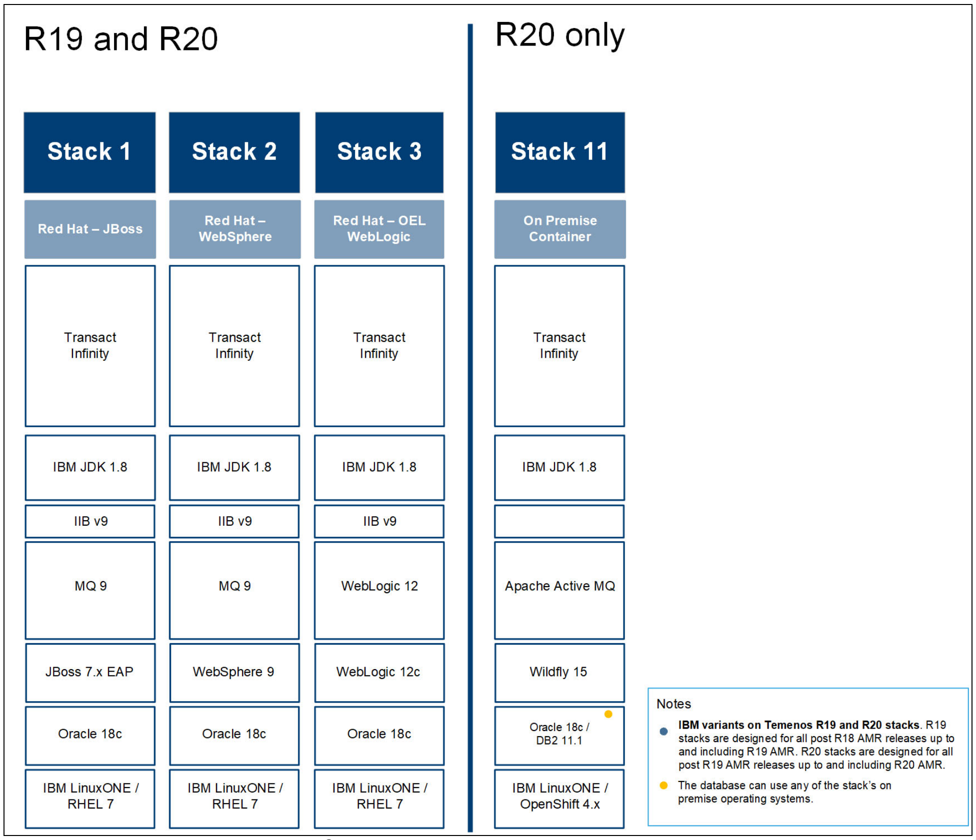


Figure 1-2 Temenos stacks. [2]

> **Note:** Temenos provides minor software releases every month, leading up to and including a major annual software release. For example, the R19 AMR (Annual Maintenance Release) includes all post R18 AMR releases up to and including R19 AMR.

1.1.2 IBM LinuxONE

IBM LinuxONE is designed with the rapidly evolving digital future in mind. A high performing, highly scalable enterprise-grade server that delivers mission critical Linux workloads. IBM LinuxONE operates at the intersection of open and enterprise computing. Annually, the platform drives 29 billion ATM transactions, 87% of all credit card transactions and 90% of all airline reservations. At the time of this publication, 44 out of the top 50 banks use IBM LinuxONE to drive their business forward.

Through an extensive ecosystem of partners, like Temenos, IBM LinuxONE delivers workloads in a common way across any on-premises or cloud architecture. That makes the platform an important stepping-stone for financial organizations that want to move away from simply adding digital capability towards a larger scale transformation. IBM calls this strategic evolution *chapter two* of the cloud or the cognitive enterprise. This is the point where the hybrid multicloud starts to deliver real performance and cost benefits to the organization.

IBM LinuxONE and Temenos' cloud native and cloud agnostic strategy both recognize the importance of choice, flexibility, scalability, security and low TCO to today's digital banks. See Figure 1-3.

[2] Courtesy of Temenos Headquarters SA

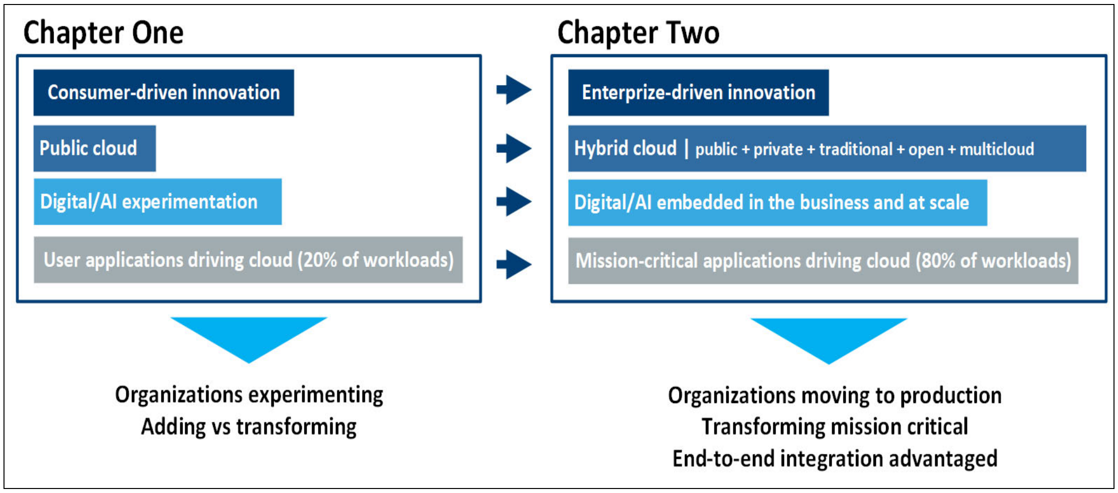


Figure 1-3 IBM Hybrid Cloud strategy: How organizations move to the hybrid cloud.

1.2 Temenos on IBM LinuxONE solves an industry challenge

Every financial organization's digital strategy is unique. For some — particularly challenger banks — launching an offering from the public cloud makes perfect sense. For other incumbent banks, migrating existing portfolios to the cloud might involve greater challenges (many banks have evolved complex IT ecosystems over time). Retail, corporate, and digital banking can run on entirely different software platforms, and critical applications can be sourced from dozens of technology vendors. Regulatory compliance can also introduce other obstacles that deprioritize the move to public cloud from the bank's IT plans.

Organizations can now take advantage of the performance enhancements, cost savings and improved customer experience offered by cloud technologies without sacrificing what already works well in their platform and architecture.

1.2.1 Why is Temenos on IBM LinuxONE

The majority of Temenos clients are operating their banking solutions on-premises currently and migrating to the cloud might be on the medium or long-term roadmap. IBM LinuxONE can help organizations implement digital transformations at their own pace, without limiting their ability to change or speed up their renovations, as the cloud matures. Consistent use of Continuous Integration and Continuous Delivery (CI/CD) with cloud platform using IBM Cloud™ Paks, Red Hat Ansible, and Red Hat OpenShift container platform guarantees commonality of deployment. That also makes it easier for Temenos clients to upgrade from TAFC to TAFJ and deliver workloads across any public cloud or internal architecture.

The core value proposition for running Temenos on IBM LinuxONE is the ability to:

- Consolidate strategic workloads (Scale up and Scale out)
- Deliver greater than 40% lower TCO over x86 architecture
- Deliver on core to digital strategy, inclusive of payments
- Scale capacity for future payments and digital growth
- Support regulatory requirements for high availability and minimal recovery times in the event of a disaster
- Deliver in various consumption models, whether on-premises, in a hybrid cloud, or on public cloud
- Keep banking secure through:

- Pervasive encryption to encrypt data at rest and data in flight. With On-chip Central Processor Assist for Cryptographic Functions (CPACF)
- Hardware Security Module (HSM) which provides tamper protection to meet regulations such as PCI DSS and PSD2
- Logical Partition with EAL5+ isolation to fully separate several workloads

IBM Secure Service Container (SSC) provides restricted administrator access and isolation for Linux workloads.

For more information about IBM LinuxONE, see the following link:

https://www.ibm.com/it-infrastructure/linuxone

1.2.2 IBM LinuxONE value for Temenos

The IBM LinuxONE system provides all the benefits of cloud; faster accessibility, greater scalability, high availability, and many more qualities of service.

IBM LinuxONE cores are more powerful than x86 cores. A combination of processor architecture, clock-speed, multiple cache levels, optimization, and I/O offloading differentiates this machine. Though security and scalability are the key differentiators of these platforms, the hardware also provides reliability and performance benefits for many important workloads.

IBM LinuxONE supports core banking and payment services as follows: 1) delivering higher transaction response times for the growing digital channels and 2) providing high-volume batch or interbank settlement services with speed, scale, security and data integrity.

Figure 1-4 shows the scalability from a single frame IBM LinuxONE III machine (up to 30 processors) to a 4-frame machine (max 190 processors).

Figure 1-4 1 to 4 frame IBM LinuxONE III.

With IBM LinuxONE you can do more with less. IBM LinuxONE is designed to run at near 100% utilization. This is in contrast to an x86 machine with a utilization of 50% or less. IBM LinuxONE is the ideal platform for workload consolidation. A financial benefit can be gained with software priced on a per core base. A fully configured IBM LinuxONE server (generation II) is able to run 2 million docker container.

Security is critical. With pervasive encryption configured using a tamper protected Hardware Security Module and IBM Secure Service Container, you are successful in protecting your data without the need to change application or database services. IBM Secure Service Container restricts administrator access to help prevent the misuse of privileged user credentials.

IBM LinuxONE has a long evolution in availability and reliability. This starts with a fully redundant architecture in the machine and reaches to highly available and highly automated cluster complexes. IBM Geographical Dispersed Parallel Sysplex® Virtual Appliance (IBM GDPS® VA) and IBM HyperSwap® provide the industry leading HA/DR capabilities. By configuring these automated services, you can minimize your RPO/RTO times to prevent data loss and minimize the recovery times in the event of planned or unplanned outages. For example, when applying maintenance to Temenos modules or updating middleware, you can update production services in real time without impacting the availability of the service. In the event of a disaster, system services combine to relocate the production workloads to the recovery site and instantiate the service in minutes without losing data. The IBM LinuxONE platform has the highest rated availability in the industry.

IBM LinuxONE is an enterprise-grade server especially designed to run Linux distributions. Its architecture is unique to address today's needs for security, performance, scalability, and resiliency.

IBM LinuxONE provides the following key features:

- Security
  - Pervasive encryption to encrypt data at rest and data in flight
  - On-chip Central Processor Assist for Cryptographic Functions (CPACF)
  - Crypto Express adapter with tamper protection to meet regulations like PCI DSS and HIPAA
  - Logical Partition isolation (LPAR) to fully separate several workloads, accredited to FIPS 102-Level 4
  - IBM Secure Service Container to provide restricted administrator access and isolation for container workloads
- Performance and scalability

  Dedicated high-performance processors for Linux (IFL), I/O (ZIIPs), Security (CPACF), Encryption (Crypto-Expres6S) and SMT/SIMD functionality, combined with advanced Channel (FICON/FCP), and redundant memory technologies (RAIM) to deliver the highest attributes for performance. This is managed within the server, ensuring that IBM LinuxONE scales both vertically and horizontally without interrupting applications and services.
- Reliability

  This IBM server family has been under development for over 50 years, bringing key engineering qualities to ensure the highest Mean-Time Between Failures (MTBF). Many Financial and Government institutions have relied upon this platform quality for uninterrupted services over many years.

IBM LinuxONE is becoming a key element in cloud strategies by integrating the quality of service features — namely Reliability, Availability, Security, Serviceability, and Data Integrity — to deliver the required business services without compromise. Temenos, as the core banking system, benefits from combining with the IBM LinuxONE platform. For example, encrypting all client data is not a new concept but it can be a slow process. Encryption with a high level of hardware support (IBM LinuxONE) makes it faster. Pervasive encryption methods encrypt all data at rest (data placed on storage) and in flight (data currently in memory or on wire). Hardware-assisted compression, which is provided by IBM LinuxONE III prior to the encryption process, delivers additional performance.

For more information about Temenos on IBM LinuxONE, read the following white paper, "*Leveraging IBM LinuxONE and Temenos Transact for Core Banking Solutions*," available at the following link:

https://www.ibm.com/downloads/cas/NEO7QNLJ

1.3 Lab environment testing of Temenos on IBM LinuxONE

Temenos has worked with IBM for many years, supporting various releases of their solutions on IBM platforms. In Spring 2017, they ported their Transact Core Banking R17 application to run on Red Hat Enterprise Linux on IBM LinuxONE for a specific customer opportunity. This was the TAFJ (full Java) release. No problems were encountered so the customer progressed with acquiring this solution and adopting IBM LinuxONE with a successful deployment of the application in 2H 2018.

In 2018, Temenos came to IBM's Montpellier facilities to complete a full technical evaluation, in a lab environment. Temenos evaluated Transact on IBM LinuxONE II platform for functional, performance, and platform-specific capabilities. Those tests are noted in the following list:

- Functional testing was conducted on the combined Transact R1802 running as an Oracle WebLogic Server 12.2.1.3.0 application using Oracle 12c R2 database platform on Red Hat 7.2 native LPARs on IBM LinuxONE II server with integrated I/O channels using IBM FICON® attached IBM DS8886 disk storage.
- Performance testing was conducted to assess the high-water mark of mixed online and batch retail banking workloads for peak volumes, alongside End of Day/End of Month batch and High-Volume Throughput (such as capitalization) workloads.
- IBM LinuxONE II-specific testing focused on the use of hardware encryption with CPACF and dm-crypt functionality to migrate the data volumes to fully encrypted disk volumes.

Temenos returned to the IBM Montpellier facilities in 2019 to evaluate, in a lab environment, the next release of Temenos Transact R1908 on IBM LinuxONE III platform. The 2019 tests built on what was learned during the 2018 testing and included additional unique features to IBM LinuxONE. Those tests are noted in the following list:

- Functional testing was conducted on the combined Transact R1908 running as an Oracle WebLogic Server 12.2.1.3.0 application using Oracle 19c database platform on Red Hat 7.6 Linux guests on IBM z/VM® 7.1 operating system on IBM LinuxONE III server with integrated I/O channels, using FCP attached to IBM DS8886 disk storage.
- IBM Spectrum Scale storage 4.2.3.18 was used to demonstrate the capabilities of using directly attached shared storage on IBM LinuxONE III to host high-volume workloads in conjunction with Temenos Transact.
- Performance testing was conducted on an Oracle 19c database — which was expanded to 7 TB with 50m accounts to assess the high-water mark of mixed online and batch retail banking workloads for peak volumes — alongside End of Day/End of Month batch and High-Volume Throughput (such as capitalization) workloads.
- IBM LinuxONE III-specific testing focused on the use of hardware encryption with CPACF and dm-crypt functionality to migrate the data volumes to fully encrypted disk volumes using AES-CBC encryption algorithm.

1.3.1 Lab environment results

> **Note:** This section and the book provide the configuration of our IBM lab test environment and some results. Results in your own lab environment can vary.

The lab environment engaged with IBM LinuxONE and Oracle SMEs, WW Java Technology SMEs, and Temenos UK and India development teams.

- The functional results demonstrated:

  - The colocated benefits of running both the application and database instances on the same Enterprise Class Linux Server
  - The reduction of network and virtualization services reducing I/O and CPU demands
  - Increased security, resilience, and data integrity of the transactions
- The performance results demonstrated:
  - Online workloads with greater transaction throughput per core, per JVM, or per thread than other platforms. Peak volume workloads showed a near 100% utilization of cores and significant TCO savings without impacting response times per transaction.
  - Batch workloads with far greater throughput and reduced elapsed/wait times for both End of Day and End of Month batch suites This reduction is a direct result of the pause-less garbage collection and JVM hardware instructions embedded in the IBM LinuxONE processors.
- The IBM LinuxONE-specific security results demonstrated:
  - Minimal CPU and elapsed-time overhead were observed with the use of dm-crypt and CPACF cryptographic accelerators built into IBM LinuxONE with the use of full disk encryption.

In summary, we can conclude that the Temenos Transact and Oracle DB on IBM LinuxONE outperforms other platforms (such as x86) by delivering benefits in terms of functionality, performance, security risks and total cost of ownership. This is achieved through these means:

- Reduce the number of processing cores for the Applications and DB.
- Dedicate hardware to accelerate encryption and Java workloads, thereby increasing throughput and decreasing wait times.
- Enable CPU cores to run at higher utilization rates without the reduction in response times seen on other platforms.

Figure 1-5 shows the example architecture used in our lab environment for testing Temenos on IBM LinuxONE.

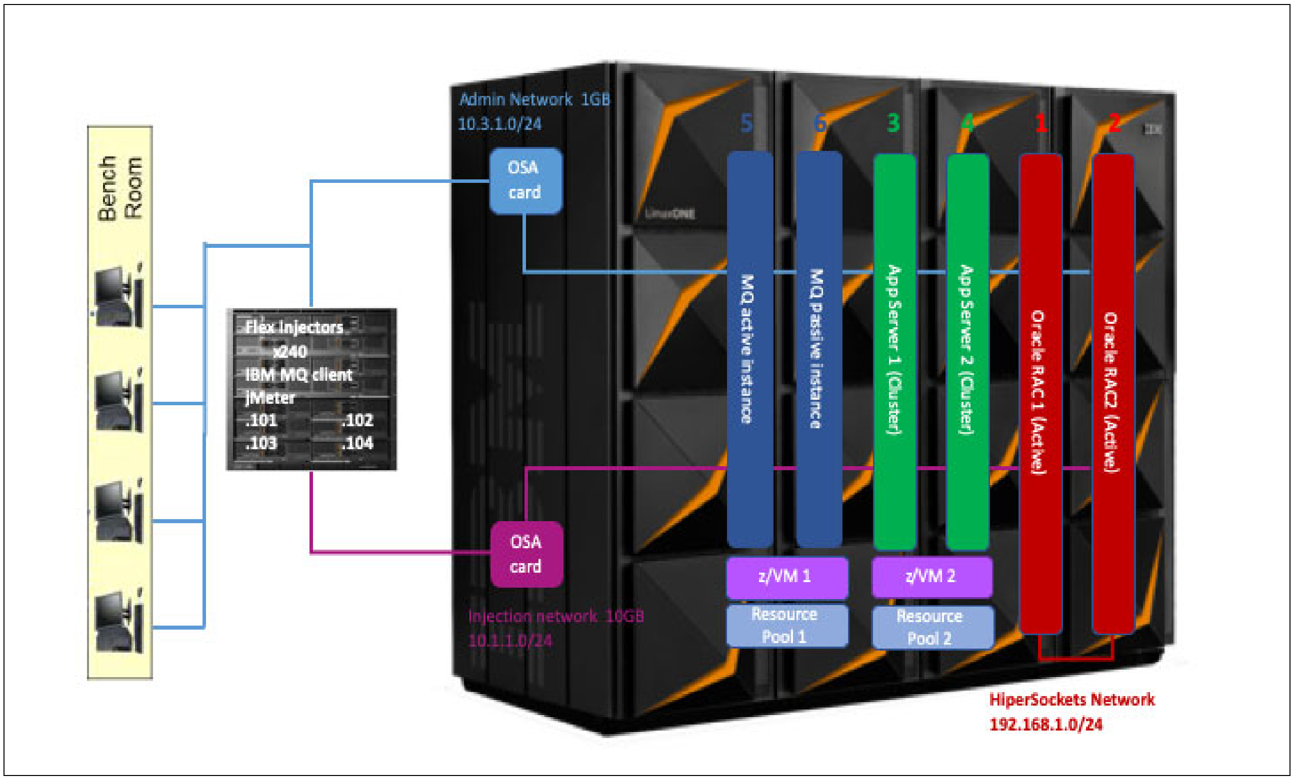


Figure 1-5 IBM LinuxONE III four frame configuration.

1.3.2 Hardware configuration

The lab environment used the server and configurations as described in this section.

IBM LinuxONE III with 63 IFL, and 1.5 TB RAM

On this server, 4 partitions (LPAR) had been defined as follows:

- 2 Native Linux LPAR for Oracle RAC database
- 1 LPAR hosting an Oracle WebLogic Server application server cluster
- 1 LPAR for z/VM, hosting 2 Linux guests for IBM MQ queue managers and Temenos lock manager

The LPARs were defined in PR/SM using Shared IFL and had a sufficient weight to ensure that all processors were configured in Vertical-High (both for z/VM and Native Linux). Using Vertical-High processing allows the best performance for IBM LinuxONE processors. It removes the constraint of having a dedicated processor, which makes changes on the LPAR less dynamic. In this lab environment, the IFL processors were used with and without SMT2 to minimize the number of physical cores defined, while providing sufficient bandwidth for applications with many threads.

IBM DS8886 with 512 GB of cache using two disks technologies

8.5 TB HPFE (High-Performance Flash Enclosure) and 20 TB HDD (Hard Disk Drive SAS 600 GB 15k RPM) were used, depending on the data stored.

In this configuration, database disks (managed by Oracle ASM) were stored on HPFE and Linux root file systems were stored on HDD. HDD was also used to store backups from the system using IBM FlashCopy® services. From a format perspective, RAID 5 was used for all the disk arrays. The SAN technology used 4 FCP/FICON Express 16S+ cards with switched connections, with 16 Gbps bandwidth each.

Network

In this lab environment, 3 types of network interfaces were used, depending on the network primary usage, as shown in Table 1-1.

Table 1-1 Network interfaces

Interface	Description
Administration: 1 Gbps Ethernet (OSA Express 7S)	This network was used for ssh connection (OSA Express 7S), monitoring, consoles, ftp, and so on.
Data network: 10 Gbps Ethernet (OSA Express 7S))	This network was used to transfer data between the injectors and the main components, and for the exchanges between the components of the solution: MQ, WebLogic Server, and database. In an IBM LinuxONE, OSA Express cards can be shared by several partitions, allowing a lower latency for the transfers.
Inter-node network: HiperSockets	This network was created through cross-memory technology called HiperSockets, available on IBM LinuxONE. This allows a low network latency, but the drawback is that this network is not reachable outside the box. Therefore, it is used here for the RAC Interconnect® network.

1.3.3 Software

Table 1-2 provides the software used in this lab environment.

Table 1-2 Products and software

Product	Software
Operating systems / Hypervisor	- Red Hat Enterprise Linux 7.6 (kernel-3.10.0-957.el7.s390x) - z/VM 7.1
Messaging system	- IBM MQ v9.1.0 - Spectrum Scale 4.2.3.18 (for MQ Active/Passive shared file system)
Application server	- Oracle WebLogic Server 12.2.1.3.0 - IBM Java JDK 8.0.5.35
Persistent shared storage	- Spectrum Scale 4.2.3.18 (for Temenos LOGS and Libraries)
Database	- Oracle DB 19c

1.3.4 Configuration (Logical architecture)

Table 1-3 provides information about the logical architecture.

Table 1-3 Logical architecture

Product	Description
Injectors	The injectors were x86-based machines used to generate and send the workload to the solution, which was hosted on the IBM LinuxONE platform. In our lab environment, they were mainly used during the online workload to send the financial messages to MQ and feed the Transact application.
MQ	MQ servers were used to receive the incoming messages representing the incoming workload. MQ was configured in an active/passive way, on 2 Linux guests under z/VM. Those 2 Linux were members of a Spectrum Scale cluster, so they can share MQ vital information (such as logs) and allow the active/passive behavior. **Note**: The passive MQ instance was also used to host an important component of the Transact architecture: the lock manager.
Oracle WebLogic Server	Oracle WebLogic Server cluster was hosting the Transact/TAFJ application and was applying the business logic to the incoming workload. During our tests, the cluster had up to 26 active members at the same time, all running the same deployed application. Each cluster member was a separate JVM with its own pool of resource (JDBC, JMS connections).
Database	The database layer was provided by Oracle RAC component, was an active/active trend. This means that both members were processing the incoming requests from the WebLogic Server application. The JDBC connection was established through the scan hostname, which was resolved using DNS to 3 IP addresses, as shown in Figure 1-5 on page 10.

1.3.5 Transaction mix

The online scenario was intended to simulate the processing of incoming payments by the Transact application. During this lab environment testing, the incoming workload was composed by 11 transaction types with the following split (as shown in Table 1-4).

Table 1-4 Transaction mix

Transaction	Transaction code	Percent of workload
Account Transaction Query	STMT.ENT.BOOK	37.50
Account Balance Query	ACCT.BAL.TODAY	37.50
Clearing Credit	CSM-CR	6
Clearing Debit	CSM-DB	6
Posting Cover Reservations	CSM-RESERVE	6
Money Transfer Account to Account	CSMACTRF	3
Payments	PAYMENT.ORDER	4

1.3.6 Encryption

The objective of this test was to evaluate how the Transact application can benefit from IBM LinuxONE encryption capabilities using the specific cryptographic engines. Two tests were done to compare the batch workload in clear and the batch workload with the DB disks encrypted using **`dm-crypt`**.

The aim was to compare a non-encrypted batch (T80516R1) with a batch with database disks encrypted (T80516R2).

When the CPU activity was compared between these, the behavior was roughly the same, as shown in Figure 1-6.

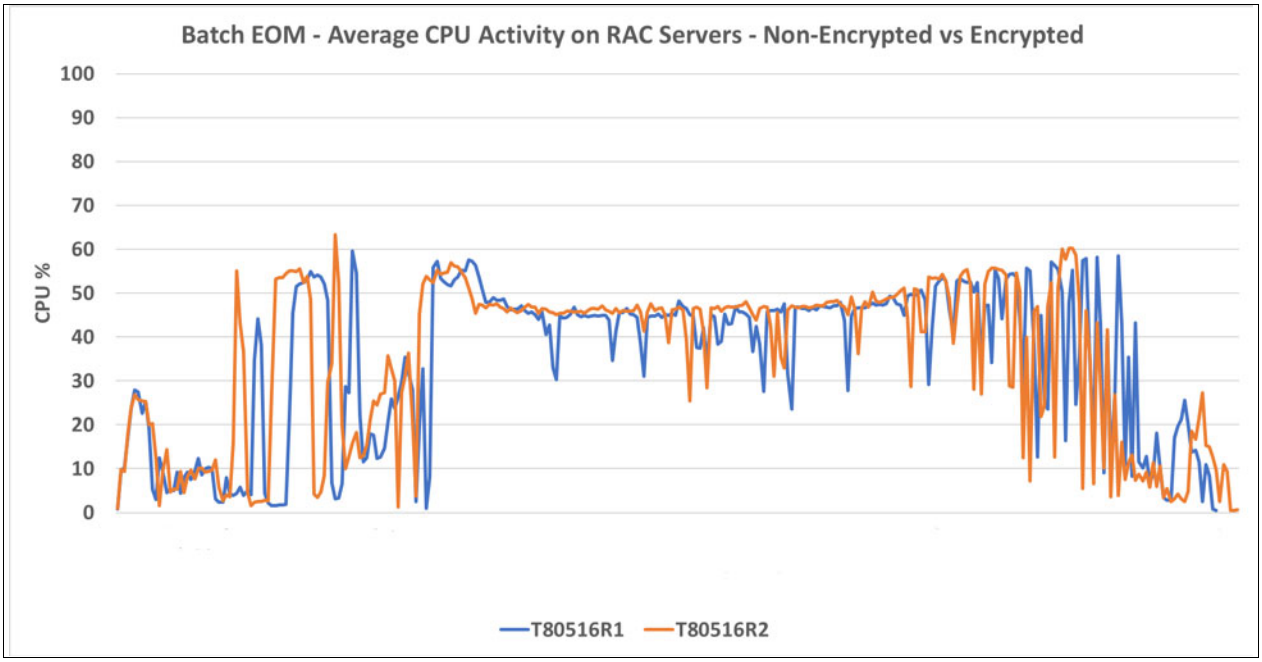


Figure 1-6 Batch EOM average cpu activity.

The encrypted scenario shows a difference of 2.06% CPU and 1.9% elapsed time.

To conclude, only the DB disks were encrypted because these disks were processing many I/O operations during the COB test. This encrypted scenario was consuming 2.06% more CPU than the clear one and the time elapsed to complete was 1.9% higher. This test shows that the IBM LinuxONE CPACF allows encryption of disk devices without significant performance impact.

1.4 Temenos modules supported/unsupported by IBM

IBM LinuxONE is fully tested and certified with the TAFJ framework. It supports all Temenos modules other than those noted in the following list:

- WealthSuite: Where IBM LinuxONE supports the database and uses x86 servers (provided by IBM) to deliver the communication and applications tier (IIB + AAA+ application).
- Analytics (formerly known as Temenos Insight®): A .Net-based application that is delivered on x86 servers, which are provided by IBM as part of an IBM LinuxONE solution.

1.5 Solution Details

Table 1-5 notes the deployments described in this IBM Redbooks Publication.

Table 1-5 Deployments and descriptions

Deployments	Description
Traditional on-premises	Vertical and Horizontal scaling
On-premises Cloud Native	OpenShift Container Platform (OCP)
	IBM Cloud Paks
IBM Public Cloud	IBM Hyper Protect
	Secure Service Containers (SSC)

All Solutions run the following middleware and applications:

- IBM MQ / Red Hat AMQ
- WebSphere/WebLogic Server/JBOSS
- IBM Integration Bus (IIB)
- IBM Streams
- Temenos TAFJ
- Temenos Transact (all modules)
- Temenos Infinity (all modules)
- Temenos Payments Hub

1.6 Financial Case

Running Temenos on IBM LinuxONE offers the following benefits:

- Greater than 40% lower total cost of ownership than x86
- Cost savings through commonality / standardization of deployment
- Cost effective scaling. Because IBM LinuxONE is designed for unknown levels of growth, an organization can scale without heavy investment in additional enterprise servers

To drive even more value from an IBM LinuxONE implementation, IBM offers a Temenos financial business case service that optimizes cost versus functional / non-functional requirements to create the correct solution for your organization. The service assesses not just the organization's core banking platform but the mission critical estate and applications that are dependent on it.

What is the value of it?

IBM LinuxONE provides other relevant benefits for cost reduction as noted in the following list:

- An end to end enterprise platform for all open software systems needed in the bank
- A digital acceleration platform - helps deliver on hybrid strategy for digitalization for the bank
- Has a broad ecosystem of all the leading FSS vendors from commercial and open standpoint with deep relationships with key vendors
- Allows consolidation away from lower end, less available Microsoft and x86 architecture to an open vendor approach or higher end enterprise vendors while at a lower cost
- Reduces enterprise middleware spending, while lowering costs and increasing availability, scalability, reliability, security

- Forms the foundations for a hybrid multicloud strategy with Red Hat OpenShift container platform and IBM Cloud Paks
- Delivers the highest levels of compliance available for financial services, whether on-premises or in cloud

1.7 Business and Technical Sales Contacts

The following individuals are available as business and technical contacts:

Contact at IBM:

John Smith

WW Offering Manager for Temenos | Linux Software Ecosystem Team

WW Offering Management, Ecosystem & Strategy for IBM LinuxONE

jsmith88@uk.ibm.com

Contact at Temenos:

Simon Henman

Temenos Product Manager, Benchmarking and Sizing | Temenos UK Ltd

shenman@temenos.com

To contact the Temenos sales team or any of the Temenos offices, use the following information:

Temenos Headquarters SA

2 Rue de l’Ecole-de-Chimie

CH - 1205 Geneva

Switzerland

Tel: + 41 (0) 22 708 1150

https://www.temenos.com/contact-us

2

Technology Overview

This chapter introduces you to a broad number of products that are common to Temenos on IBM LinuxONE installations. Concepts and considerations about these products are also included. Subsequent chapters present some of these products in a possible architecture and the final chapter presents specific deployment information for that architecture.

This chapter covers numerous products in the following areas:

2.1 Hardware

At the time of this publication, there are three versions of IBM LinuxONE available:

- The IBM LinuxONE Rockhopper II LR1 (3907) server is a single 19" frame supporting a single drawer of up to 30 processors.
- The LinuxONE Emperor II LMx (3906) server is dual 24" frame supporting up to 4 drawers with up to 170 processors.
- The latest IBM LinuxONE III LT1 (8561) server supports multiple 19" frames with up to 190 processors contained in five drawers with multiple I/O cage and memory configuration options.

An IBM LinuxONE machine is designed to be fully fault tolerant inside. It has built-in redundancy for all hardware components in the CPC. This redundancy includes aspects from dual cooling units to multiple entry points and to power the system. It is recommended that power is sourced from different locations in the data center to ensure that no single power interruption can cause an outage. Processor drawers also contain spare processors. In the event a processor fails, it is replaced automatically and concurrently, preventing any loss of processing power due to the failure. It also ensures the failure does not cause any system outage. The system memory in each processor drawer is RAIM memory, which is auto correcting if any memory module should fail. Most failures do not cause processing to stop on the system. When a failure occurs, errors are logged and the system contacts IBM so the hardware team can review the system. If needed, IBM dispatches onsite to replace the damaged component. Most replacement work is nondisruptive to the system and customer workloads.

On the external side, plan redundant cabling for power, I/O, and network.

There is additional spare capacity installed in the server that is not used. IBM offers several kinds of capacity records that allow you to use this additional capacity for defined periods of time. Such capacity-on-demand records can be used to address outages or special workload peaks (such as month end, year-end processing, or capitalization) from a period of 1 hour up to 90 days.

Additionally, there are more things to consider in planning for high availability (HA) and disaster recovery (DR). The concept of high availability (HA) and disaster recovery (DR) depends on the number of installed IBM LinuxONE servers. The following sections explain different scenarios.

2.1.1 IBM LinuxONE Central Processor Complex (CPC)

An IBM LinuxONE server needs an initial configuration before you are able to start the installation of an operating system. This initial configuration mainly includes the setup of the I/O configuration and the logical partitions. Consider and plan the redundancy of the cabling to the server, it starts from different power sources and reaches to multi-pathed I/O and the network.

PR/SM

Processor Resource and System Manager, or PR/SM, is the first-level virtualization software on the IBM LinuxONE system.

IODF/ML

The Input Output Definition File (IODF) will configure the IBM LinuxONE server hardware and define the partitions (LPARs) on the server.

CPACF

CPACF stands for Central Processor Assist for Cryptographic Functions. It enhances the instruction set of the IBM LinuxONE CPU, providing accelerated instructions for encryption and message digest functions (hashing). When combined with the Linux instruction set for DM-Crypt functionality, you can encrypt all file paths including cache and log files and Oracle ASM file structures to provide a fully encrypted file-based system.

2.1.2 LPARs

These logical partitions (LPARs) are defined directly in the hardware where an operating system runs natively without any additional hypervisor. This offers the maximum performance for an application or service but takes away some flexibility. Resources are more or less dedicated and not shareable. These LPARS are controlled by an internal management system called Processor Resource/Systems Manager (PR/SM).

2.1.3 Configuring with a single IBM LinuxONE server

A configuration with only one IBM LinuxONE server limits disaster-recovery abilities. However, that configuration can achieve limited high availability by defining multiple LPARs. For this high availability, it requires that you run at least two instances of each service and spread them over the LPARs. With such a configuration you can address planned and unplanned software outages. The IBM LinuxONE server offers built-in redundancy, any kind of hardware failure inside the machine is recovered automatically.However, a hardware outage, such as a power failure, impacts all the running services because there is no additional physical redundancy. In any case, we recommend that you use more than one storage subsystem and setup the replication technique that is offered by this storage subsystem.

Figure 2-1 shows a possible configuration using one IBM LinuxONE server.

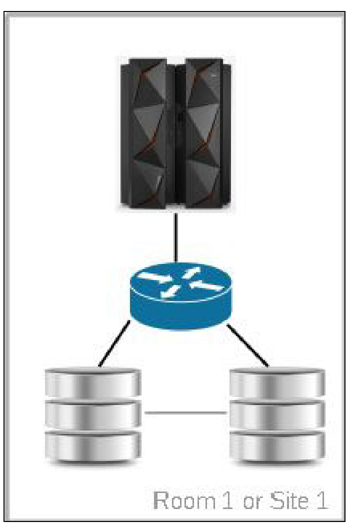


Figure 2-1 Single IBM LinuxONE server configuration.

2.1.4 Configuring with two IBM LinuxONE servers

This scenario extends the previous single configuration with the possibility of greater availability and disaster recovery. This two-server configuration can range from a simple cold standby to an active/active configuration. As a foundation, duplicate the components and locate them (at a minimum) in two different computing rooms, or (for maximum protection) at two different site locations. For clarity and use purposes, name one site as *primary* and the other site as *DR*. This naming purpose is especially important to the disk storage subsystem for your failover procedures. This naming can be diminished to each stage. You can verify usability by reverting the primary and DR sites during a testing stage. With the two-server configuration, you are able to spread all the I/Os over all the disk storage boxes and use the cache of each box.

Path length considerations

Calculate the length of the path between the IBM LinuxONE server and the storage subsystem and between the two storage subsystems. The length influences the round-trip time. Round-trip time is the amount of time between when a request is sent out and the corresponding answer is received. This includes the following events:

- The signal travels on the wire to the receiver
- The receiver gets the request, processes it and sends an answer
- The answer travels on the wire back to the requester

The shorter the round-trip time, the faster the process.

LPAR considerations

Ensure that the LPARS are spread across the machines, run more than one instance of each service and distribute them over the defined LPARs. This two-server configuration with these basics provides redundancy for both planned and unplanned outages in hardware and software. Figure 2-2 on page 20 shows a two-site configuration.

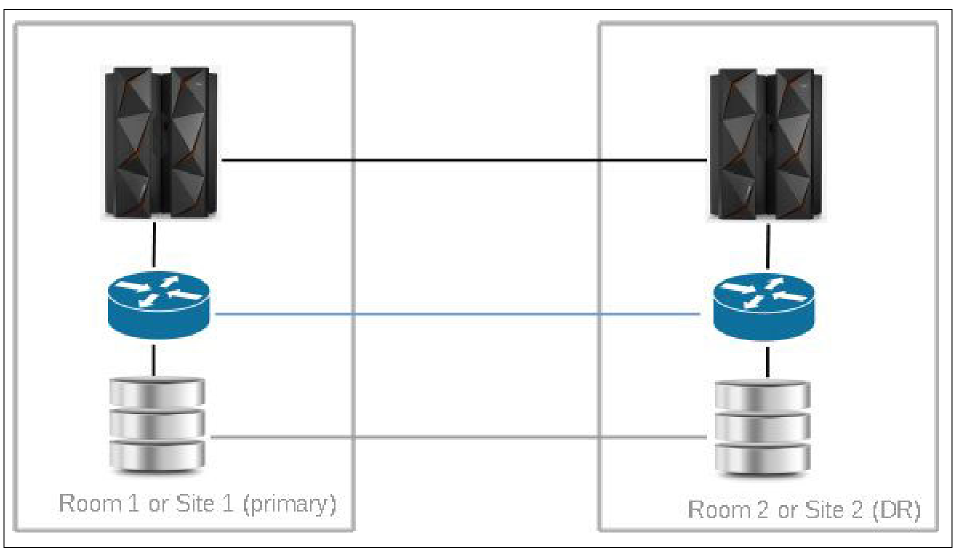


Figure 2-2 Two site configuration with full redundancy.

2.1.5 Configuring with three or more IBM LinuxONE servers

In addition to a two-machine scenario, a configuration with three or more IBM LinuxONE servers enables some quorum (automated decision making) in your environment. As in the previous scenario, you can also rotate the sites for each stage (such as production, test, quality, any other named stage). For the production stage, Room 1 is the primary site, Room 2 is the DR site, and the third machine (can be located in a separate room) is used for quorum. For the test stage, rotate the naming so that Room 2 is the primary site, *location quorum* is the DR site and Room 1 is acting as quorum. Rotate the naming further for a quality or pre-production stage. This ensures it is possible to use all the available resources (especially caches in the servers and storage subsystems) for the workload. The workload is spread over the storage subsystems. The same principle applies to the processor caches in the IBM LinuxONE servers.

> **Note:** A quorum is the majority of nodes in a cluster required to run the service. In an outage, the cluster needs to know what kind of outage happened and how to continue the service. A quorum decision is based on the predefined rules that an organization sets. It often requires some additional coding to find the correct decision for quorum or to reach a target state (`stop` or `start` services somewhere).
>
> Rules for a quorum node:
>
> - Use dedicated hardware for the quorum node and peripherals (network and disk). *Important:* Do not place the quorum node into one of the two physical servers
> - Set up the quorum for independence (different location, power, and so on)
> - Install an additional communication method for quorum decisions

A secondary less robust method is to define only one disk storage subsystem as primary. The disk storage for DR has some amount of cache installed but receives only write updates from the primary storage. In this case, the installed cache is not used by the workload. The cache in the primary storage needs to carry the full workload. The same principle applies to the processor caches in the IBM LinuxONE servers.

Quorum placement

The placement of the quorum site is important. It needs to be as independent as possible from the other sites utilizing separate power, an independent network and so on. Locating it at a different site location provides for stronger autonomy. Another option is to place the quorum in a separate computing room at one site to reduce dependency (shared cooling or power supply).

The quorum itself (remember it is reduced to stages) needs less capacity because it supervises only the main computing sites and addresses *split brain* situations. For example, if one of your computing sites has a failure, another *surviving* site takes over the workload automatically. However, in such a situation the surviving site does not know whether the failed site is completely down or if it has simply lost connectivity. Split brain situations are where both sites are fully available but have lost their intercommunication. Both sites have full access to the data but run in an uncoordinated way. This can lead to a damage in the data. In this situation, to get certainty, you need to ask a third party, such as the quorum site. The quorum site first checks its own connectivity to all sites. If it has lost connectivity too, the remaining site receives the quorum and does the takeover automatically. If the quorum site has connectivity to the failed site (a split-brain situation), there must be additional rules in place for how to continue the operation while ensuring data integrity. The quorum sites function is to find a decision, so it needs only limited capacity in terms of CPU power and disk space. A quorum cluster must always have an odd number of nodes. In case of a split-brain network failure between the cluster nodes, the portion that holds more than half of the nodes that the cluster initially had — including the quorum nodes — receives the quorum and continues to run. This portion is referred to as the *island*.

Figure 2-3 on page 22 shows a cluster including a quorum site with limited capacity.

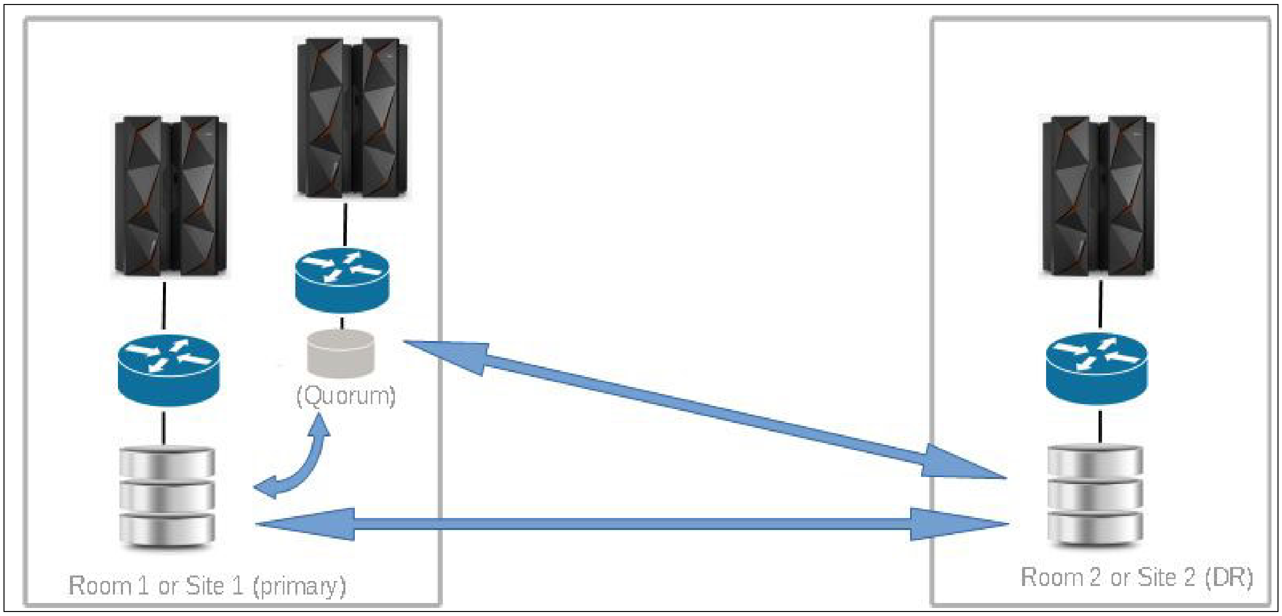


Figure 2-3 3 Site configuration with quorum site.

The functions that can use quorum are noted in the following list:

- Server-Time-Protocol (STP) can use a quorum (called Arbiter) to select the preferred time server
- A clustered file system (GPFS or Spectrum Scale) uses quorum to maintain data consistency in the event of a node failure

2.1.6 System partition configuration

The traditional method of configuring logical partitions and I/O on IBM LinuxONE uses the I/O Definition File (IODF). In this book, we describe running the machine in IODF mode and configuring the machine using IODFs.

> **Note:** More detail about I/O configuration on IBM LinuxONE can be found in 3.2, "Machine configuration on IBM LinuxONE" on page 55.

Dynamic Partition Manager

A new partition management system, known as Dynamic Partition Manager (DPM), has been introduced on IBM LinuxONE. DPM provides a complete interface for defining IBM LinuxONE partitions, assigning resources to them, and starting and stopping them. DPM is designed to require less upfront knowledge of the I/O configuration concepts than when using IODFs.

DPM supports most IBM LinuxONE hardware, including OSA Express and FICON Express adapters.

At the time of this publication, DPM does not provide support for two important IBM LinuxONE functions that are important to the reference architecture described in this book. These functions are:

- FICON channel-to-channel (CTC) devices
- The LPAR to run a GDPS Virtual Appliance

FICON channel-to-channel (CTC) devices are a point-to-point communication link using FICON, and are required for the z/VM SSI feature. This means that when an IBM LinuxONE machine is operating in DPM mode the z/VM SSI feature cannot be used on that machine.

The GDPS Virtual Appliance described in 2.4.6, “GDPS and Virtual Appliance (VA)” on page 40 requires a specific configuration of the LPAR the appliance runs in. DPM cannot configure an LPAR to support the GDPS Virtual Appliance.

Note: Using IODF or DPM mode does not change how operating systems run on IBM LinuxONE, it just changes the way that partition configuration is performed. The same underlying LPAR management firmware (PR/SM) is used in either mode. You can choose to use DPM in support of a Temenos implementation on IBM LinuxONE. But you will need to change aspects of the architecture to cater for functions that cannot be used as a result. The needed changes, such as substituting a different Linux clustering capability to make up for not have hypervisor clustering (z/VM SSI), are not described in this book.

2.1.7 Server-Time-Protocol (STP)

STP is designed to provide the capability for multiple servers to maintain time synchronization with each other and form a Coordinated Timing Network (CTN). CTN is a collection of servers that are time synchronized to a time value called Coordinated Server Time (CST). STP transmits timekeeping information in layers or stratums. The top level (Stratum 1) distributes time messages to the layer immediately below it (Stratum 2). Stratum 2, in turn, distributes time messages to Stratum 3. Through the Hardware Management Console (HMC) of the IBM LinuxONE server, STP can be synchronized to an external time source using Network Time Protocol (NTP). Figure 2-4 on page 23 shows a stratum hierarchy.

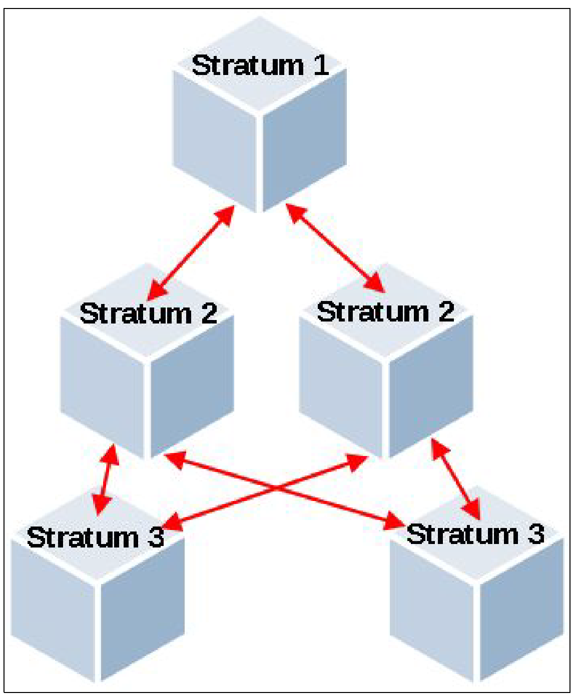


Figure 2-4 Stratum hierarchy.

STP is a proprietary protocol and requires dedicated adapters and extra cabling. The z/VM member and the SSI cluster are synchronized with the same time resource.

Linux reads the time from z/VM only at boot time. To keep the time in Linux accurate, it is recommended to additionally set up the time synchronization inside Linux using NTP.

2.1.8 Shared Memory Communication (SMC)

Shared Memory Communications (SMC) is an innovative communications protocol that provides optimized communications that allow applications (within separate operating system instances) to communicate directly though *shared memory*. There are two variables of SMC:

- SMC over RDMA (SMC-R) which provides host-to-host direct memory communications
- SMC Direct Memory Access (SMC-D) which provides memory-to-memory communications within a single physical server

TCP/IP is still used to establish the connection and provide security, however, SMC eliminates TCP/IP processing in the data path. The TCP/IP connection can be provided using OSA or HiperSockets connectivity. After the initial handshake is complete, communications then use sockets-based SMC.

SMC-R is a protocol solution that is based on sockets over RDMA and the Internet Engineering Task Force (IETF) Request for Comments (RFC) 7609 publication. It is confined to socket applications by using Transmission Control Protocol (TCP) sockets over IPv4 or IPv6. SMC-R solution enables TCP socket applications to transparently use RDMA, which enables direct, high-speed, low-latency, and memory-to-memory (peer-to-peer) communications.

Communicating peers, such as TCP/IP stacks, dynamically learn about the shared memory capability by using the traditional TCP/IP connection establishment flows. This process enables the TCP/IP stacks to switch from TCP/IP network flows to optimized direct memory access flows that use RDMA.

RDMA is available on standard Ethernet-based networks by using the RDMA over Converged Ethernet (RoCE) interface. The RoCE network protocol is an industry-standard initiative by the InfiniBand Trade Association. RoCE interface enables the use of both standard TCP/IP and RDMA solutions such as SMC-R over the same physical local area network (LAN) fabric.

Figure 2-5 shows SMC-R communication flow between two hosts. By using the TCP option, TCP synchronization operation determines whether both the hosts support SMC-R protocol solution, and then establishes the RoCE network.

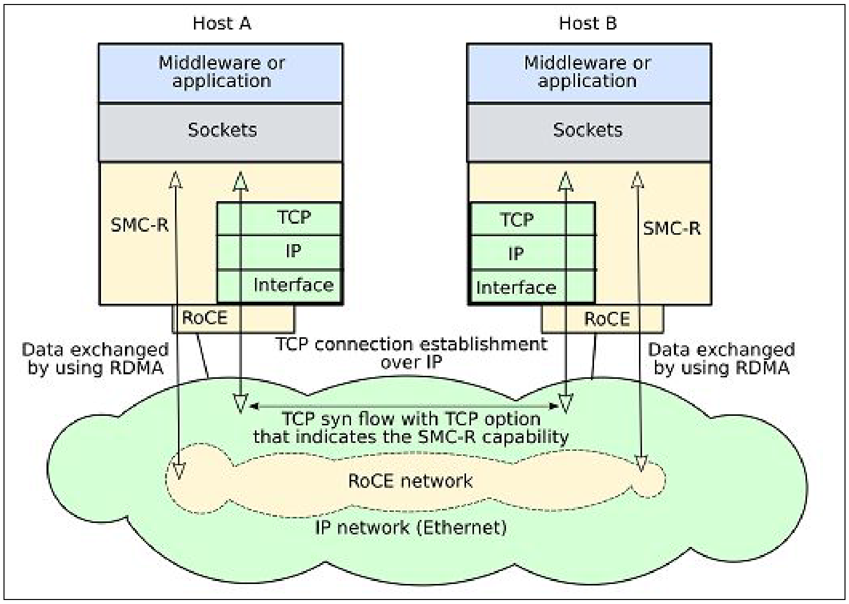


Figure 2-5 SMC-R communication flow between two hosts.

The same principle applies to SMC-D but without an RoCE adapter.

2.1.9 Guarded Storage Facility (GSF)

Speed can cost you customers, your reputation, and revenue. JAVA brings its own memory management but sometimes this memory must be inspected for obsolete objects, called garbage collection. At garbage collection time, all program threads need to be stopped simultaneously. With all the IBM LinuxONE Emperor II and IBM LinuxONE Rockhopper II models, IBM has introduced a unique hardware feature that is designed to greatly reduce garbage collection pause times. This feature also significantly improves response times for workloads that do a lot of garbage collection. This feature is not available on x86 servers. The feature is called the Guarded Storage Facility (GSF) and is enabled by specifying an additional option (`-Xgc:concurrentScavenge`) to the JVM. The GSF essentially enables garbage collection to run almost completely concurrently with the application threads, thus dramatically reducing the pause times and allowing the workloads to run smoothly.

Figure 2-6 on page 25 shows a test of the IBM Competitive Project Office team. The test compares the pause times, response times, and throughput for the same test workload running on an IBM LinuxONE server and running on Linux on an x86 server.

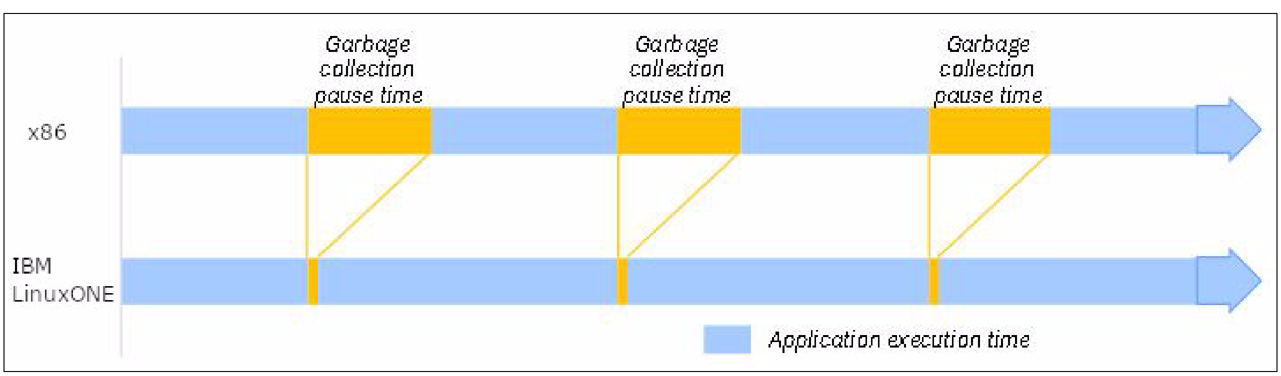


Figure 2-6 Garbage collection pause data: IBM LinuxONE as related to x86.

The Java garbage collection time on IBM LinuxONE was 92% lower than the compared x86 server under internal test conditions.

2.1.10 IBM LinuxONE III Integrated Accelerator for zEDC

The Integrated Accelerator for zEDC with the IBM LinuxONE III reduces the cost of storing, transporting, and processing data. It replaces the zEDC Express adapter with on-chip compression, providing increased throughput and capacity: up to 8 times faster application elapsed time with no additional CPU time compared to an IBM LinuxONE II with zEDC Express for compression/decompression. It requires enablement using Linux distribution, currently targeted for Red Hat Enterprise Linux v8.2 and SuSE Enterprise Linux 14.

2.1.11 Disk Storage

There are various storage subsystems that you can choose from. A suggested system is the IBM DS8900F family of enterprise class all-flash array (AFAs) disk systems. They offer unmatched performance, reliability, availability, interoperability, serviceability, and economics (total cost of ownership) in relation to its competition.

The DS8950F and DS8910F offer the same functional capabilities and share the same set of built-in caching algorithms, which enhances disk system performance. The models differ on available physical resources such as those noted in the following list:

- The internal POWER9 processors
- The number of processor cores
- Total system memory (cache) capacity
- Flash drive sizes
- Total usable capacity
- The number of host adapters (which are used for FICON I/O, Fibre Channel I/O, and Fibre Channel-based disk replication)
- Available zHyperLink (ultra-low latency direct host to disk I/O connectivity) ports

The IBM DS8900F family supports IBM LinuxONE hardware and operating systems, IBM Hypervisor z/VM and KVM. The supported native Linux operating systems include Red Hat, SuSE, and Ubuntu.

Both the DS8950F and DS8910F offer the same set of industry-leading Copy Services. These services can be implemented into unique 2-site, 3-site, and 4-site solutions to provide high availability, disaster recovery, and logical corruption protection. Copy Services include the following assets:

- IBM FlashCopy (point-in-time copy)
- Metro Mirror (synchronous copy)
- HyperSwap (dynamic automated failover between primary and secondary disk systems for planned and unplanned outages)
- Global Mirror (asynchronous copy)
- Metro Global Mirror (sync and async copy)
- Safeguarded Copy (logical corruption protection)

Copy Services replication management is available through IBM Copy Services Manager (application), IBM GDPS (services offering), or through custom scripting.

To learn more about the DS8900F family, visit the Redbooks offerings available at the following link:

http://www.redbooks.ibm.com

From that site you can search various topics including DS8900F, IBM Easy Tier®, Thin Provisioning, Copy Services, Copy Services Manager, or GDPS. The IBM DS8000 publications of particular relevance are noted in the following list:

- IBM DS8900F Architecture and Implementation (SG24-8456-00)
  http://www.redbooks.ibm.com/redpieces/pdfs/sg248456.pdf
- IBM DS8000 Copy Services (SG24-8367-00)
  http://www.redbooks.ibm.com/redbooks/pdfs/sg248367.pdf
- IBM DS8000 Easy Tier (Updated for DS8000 R9.0) (REDP-4667-08)
  http://www.redbooks.ibm.com/redpapers/pdfs/redp4667.pdf
- IBM DS8880 Thin Provisioning (Updated for Release 8.5) (REDP-5343-01)
  http://www.redbooks.ibm.com/redpapers/pdfs/redp5343.pdf
- IBM Copy Services Manager Implementation Guide (SG24-8375-00)
  http://www.redbooks.ibm.com/redbooks/pdfs/sg248375.pdf
- IBM GDPS Family: An Introduction to Concepts and Capabilities (SG24-6374-14)
  http://www.redbooks.ibm.com/redbooks/pdfs/sg246374.pdf

Bridging industry standard protocols with IBM LinuxONE

In the past, IBM used a proprietary protocol and disk format to store data. This is different than the technology used by open systems to work with disk storage. IBM LinuxONE closes this gap and supports these two types of disk storage technologies:

- Open system storage which can be any kind of storage in the SAN and operates in 512-byte fixed block mode (FBA). It requires a Fibre Channel protocol (FCP) adapter.
- Extended Count Key Data (ECKD) format. ECKD is available only on enterprise-class storage (IBM DS8000 or equivalent) and requires a FICON channel adapter.

The channel adapters in the IBM LinuxONE machine operate either in FICON mode or in FCP mode. Volumes in ECKD format on an enterprise class storage subsystem offer some additional useful features.

FBA is more commonly associated with SAN or SCSI-based storage controllers and can be attached to IBM LinuxONE based systems. ECKD mode requires the use of IBM DS8000 or HPS/EMC Enterprise class storage controllers to provide direct attach storage performance. The channel adapters in the IBM LinuxONE machine are able to operate either in FICON mode or in FCP mode. The IBM LinuxONE III server introduced the use of FCP32S ports capable of 32gbs data transfer rates. FBA and ECKD storage can be attached and operated in parallel. When using GDPS VA for HA/DR between two or more sites, only ECKD storage mode is permitted.

Open system storage (SCSI or FBA storage)

Open system storage is any kind of SCSI storage available in your SAN. This starts from the IBM Storwize® family through the flash only systems including SAN switches and SAN Volume Controller (SVC). IBM LinuxONE can be attached to the SAN for participation in that network. It is easy to define volumes of any size but note that an ECKD volume has a limit in terms of volume size. This kind of storage is operating in 512-byte fixed block mode. The big advantage of using open systems storage in IBM LinuxONE is that it can be accessed also from the x86 environment or vice versa.All the management and monitoring tools and procedures that you previously set up still apply. LUNs on such a disk storage system can be passed directly to Linux guest or can be managed by z/VM itself using emulated device (EDEV). EDEV is limited 1 TB in size in z/VM. Open system storage is less expensive than enterprise class storage but also does not offer all the features of enterprise-class storage. Functions for point-in-time copy, space efficiency, and performance depend upon the model.

Extended Count Key Data (ECKD) and IBM DS8000

The IBM System Storage DS8000 series is a high-performance, high-capacity series of disk storage that supports continuous operations.

The latest and most advanced disk enterprise storage system in the DS8000 series is the IBM System Storage DS8800. It represents the latest in the series of high-performance and high-capacity disk storage systems. The DS8800 supports IBM POWER6+ processor technology to help support higher performance.

The DS8000 series support functions such as point-in-time copy functions with IBM FlashCopy, FlashCopy Space Efficient, and Remote Mirror and Copy functions with Metro Mirror, Global Copy, Global Mirror, Metro/Global Mirror, IBM z/OS Global Mirror, and z/OS Metro/Global Mirror. Easy Tier functions are supported also on DS8800 storage units.

All DS8000 series models consist of a storage unit and one or two management consoles, two being the recommended configuration. The graphical user interface (GUI) or the command-line interface (CLI) provide the ability to logically partition storage and use the built-in Copy Services functions. For high availability, the hardware components are redundant.

The IBM Tivoli® Key Lifecycle Manager (TKLM) software performs key management tasks for IBM encryption-enabled hardware, such as the DS8000 series. TKLM protects, stores, and maintains encryption keys that are used to encrypt information being written to, and decrypt information being read from, encryption-enabled disks. TKLM operates on a variety of operating systems.

With DS8000 support both ECKD and FBA disk format together. ECKD requires a FICON attachment. If you plan to run it on SAN, the SAN switches must support the FICON protocol.

Flash and Non-Volatile Memory Express (NVMe)

Flash is the ideal solution to meet growing performance demands and concurrently bringing speed, scalability and savings to your business. The portfolio ranges from additional flash expansions into a hybrid storage subsystem up to flash-only storage subsystems.

The advantages of IBM all-flash storage systems are that they are engineered to meet modern high-performance application requirements. Ultra-low latency, cost-effectiveness, operational efficiency, and mission-critical reliability are built into every product.

For building high-performance storage systems that have to efficiently deliver consistently low latency and high throughput for mixed enterprise workloads, there is no doubt that NVMe is a better core technology than SCSI. NVMe is specifically developed for flash storage, allowing it to make much more efficient use of flash performance and capacity. NVMe will be the I/O protocol of choice to support new emerging memory technologies like storage-class memory going forward. NVMe supports generally at least 50% lower latencies on a per-device basis and up to an order of magnitude higher bandwidth and throughput. IBM storage systems are expanded to use NVMe.

Parallel access volumes (PAV) and HyperPAV (HPAV)

Parallel access volumes (PAV) is the concept of using multiple devices or aliases to address a single ECKD disk device.

If there is no aliasing of disk devices then only one I/O transfer can be in progress to a device at a time. This is regardless of the actual capability of the storage server to handle concurrent access to devices. Parallel access volume exists for Linux on IBM LinuxONE in the form of

PAV and HyperPAV. PAV and HyperPAV are optional features that are available on the DS8000 series.

PAV is a static assignment for one or more aliases to a single base ECKD disk. HPAV is a dynamic assignment of an alias to a base disk during an I/O operation. HPAV follows the current workload needs and requires fewer alias devices compared to PAV. HPAV is recommended in most cases.

A suggested starting point for configuration is 8-16 HPAV devices configured per logical control unit in the DS8000.If PAV is used, as starting point, define the same number of PAV devices as for the base volumes (1:1 relation).

High Performance FICON (zHPF)

High Performance FICON (zHPF) is a channel I/O architecture that is designed to improve the execution of small block I/O requests. By using a Transport Control Word (TCW) the zHPF support facilitates the processing of an I/O request by the channel and the control unit. I/O operations that use TCWs are defined to be run in transport mode. A conversion routine translates a command mode channel program into a transport mode channel program. This makes zHPF support transparent for user applications. zHPF is a performance and reliability, availability, serviceability (RAS) enhancement of the IBM z/Architecture® and the FICON channel architecture.

The TCW enables multiple channel commands to be sent to the control unit as a single entity. The channel forwards a chain of commands and does not need to keep track of each single CCW. This leads to reduction in the FICON overhead and increases the maximum I/O rate on a channel. The performance improvement depends on your workload.

2.2 Operating systems

The IBM LinuxONE system is designed to run the Linux operating system and IBM z/VM. IBM has contributed to the open source community server Linux modules to maximize hardware features offered and supported only by IBM LinuxONE. IBM has partnered with three enterprise distributions Red Hat, SuSE, and Ubuntu to ensure that IBM LinuxONE remains a strategic platform from the distributions.

There are also non-enterprise distributions available for IBM LinuxONE (such as ClefOS (CentOS), openSUSE, and Fedora). These distributions do not provide enterprise support.

IBM also offers z/VM as a hypervisor for Linux on IBM LinuxONE. z/VM has been developing virtualization for over 40 years and throughout these decades IBM has modified z/VM to run Linux with the best possible performance.

2.2.1 Red Hat

Red Hat is a company based in the US and was founded in 1993. The first Red Hat distribution to port to the s390x architecture was RHEL 5 announced in December 2001. IBM LinuxONE is a key platform for Red Hat and IBM, and through the cooperation of both companies all the available hardware functions in Red Hat Enterprise Linux (RHEL) are enabled and supported. The code for this distribution is from the same source as for all the other platforms. The lifetime cycle of RHEL is 5.5 years full support and 3.5 years maintenance support with an annual update. After this period Red Hat offers Extended Life Cycle Support (ELS).

2.2.2 SuSE

SuSE was founded 1992 in Germany. Marist University ported the first Linux distribution for the s390x architecture in close cooperation with IBM and it was available near the end of 1999. Close to this date, SuSE announced the first commercial distribution for the s390x in October 2000. This SuSE Linux Enterprise Server (SLES) version was built from the same source code as for all the other platforms. The lifecyle of SLES is 10 years and an annual update with an optional extension of 3 further years.

2.2.3 Ubuntu

Ubuntu is the youngest distribution on this platform and was started with version 16.04 in April 2016, shortly after the announcement of IBM LinuxONE in 2015. Ubuntu is a distribution and is sponsored by Canonical. Canonical is founded 2004 in UK. Every even year Ubuntu announces Long Term Support (LTS) versions. They also provide interim versions. Only the LTS versions of Ubuntu are officially supported for IBM LinuxONE. The lifecycle for an LTS release is 5 years. Afterward, there are optional security maintenance extensions available.

2.2.4 z/VM

z/VM is an operating system developed by IBM and first announced in August 1972. It was designed to run workloads in a virtualized environment. At the time Linux was born, z/VM began to transform into a powerful hypervisor for Linux. z/VM is in continuously evolution to gain the most performance for Linux on IBM LinuxONE.

2.3 Software

The following sections discuss the required and optional software components needed to run Temenos on the IBM LinuxONE server. This section further discusses how to get the most from the facilities of an IBM LinuxONE server.

2.3.1 RACF

Resource Access Control Facility (IBM RACF®) is a software security product that controls access to protect information. This software also does the following actions:

- Controls what a person can do on the operating system and therefore protects resources
- Provides this security by identifying and verifying users
- Authorizes users to access protected resources
- Records and reports access attempts

2.3.2 IBM WebSphere Application Server

In this book, we are going to use IBM WebSphere Application Server. IBM WebSphere Application Server is a flexible, security-rich Java server runtime environment for enterprise applications. It supports microservices and standards-based programming models. It also delivers advanced performance, redundancy, and programming models. Broad support for enterprise-level security, integrated management and administrative tooling that ensures compliance with regulations, including FIIPS and GDPR. IBM WebSphere offers an integrated administration console to manage the applications inside. See the *Stack 2 WebSphere Runbook* for details about the installation of IBM WebSphere. For more information, Temenos customers can find the *Transact Runbook for Stack 2* on the Temenos customer portal.

2.3.3 Oracle WebLogic

Oracle WebLogic is an application server for building and deploying enterprise Java EE applications. Oracle WebLogic has support for new features for lowering cost of operations, improving performance, enhancing scalability, and supporting the Oracle Applications portfolio. WebLogic Server Java EE applications are based on standardized, modular components. WebLogic Server provides a complete set of services for those modules and handles many details of application behavior automatically, without requiring programming.

2.3.4 JBOSS EAP

Red Hat JBoss Enterprise Application Platform (EAP) is an application server. It works within its own integrated development environment, based on Apache Eclipse projects and other open source deployment tools. Developers can create and run complex Java applications that take advantage of the full Java EE software stack.

EAP is based on the open source application server project Wildfly. It in turn uses Enterprise JavaBeans APIs and containers to manage its transactions and business logic. Applications can be deployed on a variety of server situations, including physical, virtual, private, and public clouds.

2.3.5 IBM Java

IBM has been committed to Java on Z Systems® for more than a decade, focusing on performance improvements to the core Java Development Kit, and co-design of IBM Z systems hardware and IBM Java VM (JVM) technology. A large proportion of new instructions and hardware facilities introduced in the last four generations of the IBM Z process were co-designed with the IBM JVM team. What this means is that IBM LinuxONE uses hardware acceleration for the JVM. As a result, Java on z Systems consistently demonstrates about a 1.5x performance advantage over alternative platforms.

2.3.6 IBM MQ

IBM MQ can transport any type of data as messages, enabling businesses to build flexible, reusable architectures such as service-oriented architecture (SOA) environments. It works with a broad range of computing platforms, applications, web services, and communications protocols for security-rich message delivery. IBM MQ provides a communications layer for visibility and control of the flow of messages and data inside and outside your organization.

IBM MQ provides the following benefits:

- Versatile messaging integration from IBM LinuxONE to mobile that provides a single, robust messaging backbone for dynamic heterogeneous environments
- Message delivery with security-rich features that produce auditable results
- High-performance message transport to deliver data with improved speed and reliability
- Administrative features that simplify messaging management and reduce time spent using complex tools
- Open standards development tools that support extensibility and business growth

An application has a choice of programming interfaces, and programming languages to connect to IBM MQ.

IBM MQ is *messaging* and *queuing* middleware, with several modes of operation including *point-to-point*, *publish/subscribe*, and *file transfer*. Applications can publish messages to many subscribers over *multicast*.

The following list gives further details about some aspects of IBM MQ:

- Messaging

  Programs communicate by sending each other data in messages rather than by calling each other directly.

- Queuing

  Messages are placed on queues, so that programs can run independently of each other, at different speeds and times, in different locations, and without having a direct connection between them.

- Point-to-point

  Applications send messages to a queue, or to a list of queues. The sender must know the name of the destination, but do not have to know where it is.

- Publish/subscribe

  Applications publish a message on a topic, such as the result of a game played by a team. IBM MQ sends copies of the message to applications that subscribe to the results topic. They receive the message with the results of games played by the team. The publisher does not know the names of subscribers, or where they are located.

2.3.7 Databases

The IBM LinuxONE platform is one of the best systems to host databases. IBM LinuxONE has greater processing power than the current x86 processor. IBM LinuxONE is capable of Simultaneous Multi-Threading (SMT). This allows more instructions through the same processor.

Each virtual Linux guest can be assigned enough memory to allow most data transactions to happen in real memory. This provides for increased transaction speeds and allows the customer to reduce the number of virtual servers and processors needed to service database transactions.

Because of the ability to create Linux guests with large memory sizes, major cost savings in database software licensing can be realized. Performance gains are also noticed because you do not have to shard large databases across multiple systems.

There are some database options available for running Temenos Transact core banking database on IBM LinuxONE:

- IBM Db2®: A relational database for transactional processing. It is designed to have high availability, scalability with high performance. Other features such as IBM BLU® Acceleration® for in-memory column-organization and selective compression to optimize database performance. IBM Db2 is also optimizable for data analytics and AI processing.
- Oracle Database (Oracle DB): This is the preferred database for Temenos Transact on IBM LinuxONE with the tradition deployment architecture. Oracle Real Application Clusters (RAC) is used to ensure High Availability (HA) of the database in the event of an outage on one of the database nodes.

- PostgreSQL: PostgreSQL is an open source object-relational database system that is currently being certified by Temenos for use in the IBM public cloud. PostgreSQL has earned a strong reputation for its proven architecture, reliability, data integrity, feature set, and extensibility from the open source community. It was selected to be deployed as part of the IBM Hyper Protect DBaas offering. IBM Cloud Hyper Protect DBaaS for PostgreSQL is a LinuxONE powered cloud database solution for enterprise workloads with sensitive data. Hyper Protect DBaaS for PostgreSQL currently contains PostgreSQL major version 10.

2.4 Hypervisor choices

z/VM is the premier hypervisor for the IBM LinuxONE platform. It is deeply integrated with the IBM LinuxONE hardware and provides the highest level of guest OS support for hardware features.

Linux KVM is also supported on IBM LinuxONE. KVM is integrated into the Linux distribution of your choice. For Temenos workloads, they currently certify the use of Red Hat Enterprise Linux distribution only.

Whether you use z/VM or KVM (or a combination of both) is a choice you need to decide for your environment. There are some support considerations in making the decision. The following sections provide some information about the choices available.

2.4.1 z/VM as hypervisor

z/VM is the premier hypervisor for the IBM LinuxONE platform. It is deeply integrated with the IBM LinuxONE hardware and provides the highest level of guest OS support for hardware features.

The z/VM hypervisor provides deep integration with the IBM LinuxONE platform hardware and provides rich capabilities for system monitoring and accounting.

z/VM, as the hypervisor for your Linux systems, gives you the ability to share the resources in a granular way. It has a long history of sharing system resources with one to many Linux guests running in an LPAR. IBM z/VM was developed to give the Linux guest access to system resources with little hypervisor overhead. z/VM offers a feature to cluster up to four members to a Single System Image (SSI). A useful feature within an SSI cluster is Life Guest Relocation (LGR). This feature moves a running guest to another z/VM member without interruption. This enables you to do maintenance in z/VM without stopping your workload.

Note: LGR is not supported for use with Oracle.

Features and additional software products for z/VM

The following features provide additional functionality for z/VM:

- Performance Toolkit for VM: Provides enhanced capabilities for a z/VM systems programmer, system operator, or performance analyst to monitor and report performance data. Offered as a priced optional feature of z/VM, the Performance Toolkit for VM is derived from the FCON/ESA program (5788-LGA), which is not available in all countries.
- RACF Security Server for z/VM: Enables the protection of IBM LinuxONE resources by making access control decisions through resource managers. Granting access to only authorized users keeps your data safe and secure.

- Directory Maintenance Facility for z/VM (DirMaint): Provides efficient and secure interactive facilities for maintaining your z/VM system directory. z/VM system directory is the definition repository for all virtual machines. Directory management is simplified by DirMaint's command interface and automated facilities. DirMaint's error checking ensures that only valid changes are made to the directory, and that only authorized personnel make the requested changes.
- IBM Infrastructure Suite for z/VM and Linux: A single solution that provides multiple tools to monitor and manage z/VM and Linux on IBM LinuxONE. It supports backup and recovery of the entire system. 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 LinuxONE environments.

2.4.2 KVM

Kernel Virtual Machine uses the acronym KVM. KVM for IBM LinuxONE is an open source virtualization option for running Linux-centric workloads that use common Linux based tools and interfaces. KVM uses the full advantage of the robust scalability, reliability, and security that is inherent to IBM LinuxONE platform. The strengths of the IBM LinuxONE platform have been developed and refined over several decades to provide additional value to any type of IT-based services.

Using KVM as a hypervisor allows an organization to use existing Linux skills to support a virtualized environment. Though KVM provides flexibility in the choice of management tools that can be used, it does not provide as deep a platform integration.

KVM provides a facility for relocating virtual machines between KVM hosts.

Note: Oracle does not support their product for use with KVM on IBM LinuxONE, therefore it is not recommended to use KVM with any Oracle product in a production environment.

Hypervisor management with KVM

Management of KVM has become standardized around the libvirt system. Libvirt was originally developed to abstract the details of various hypervisors to provide a universal interface. A management tool that uses the libvirt API can manage any hypervisor that libvirt supports. Though libvirt does indeed support several hypervisors, it is most commonly used to manage KVM on Linux.

A libvirt-based manager is usually the first choice for managing KVM on IBM LinuxONE. There are a number of choices including virt-manager. Virt-manager is a Python-based graphical tool that allows some configuration of hypervisor resources (network connections, disk pools, and so on) and access to and control of virtual machines. A web-based utility called Kimchi uses libvirt and works with KVM on IBM LinuxONE and can be used to manage many aspects of hypervisor operation.

A KVM host is a Linux system that runs as a hypervisor. Therefore, managing the basic Linux aspects of a KVM host is essential. Normal Linux command-line utilities can be used for this, but there are other interesting options. One such option is **`Cockpit`**, which provides a HTML-based management interface for managing many aspects of a Linux server. Cockpit is installed by default in RHEL 8.

Virtual machine management

Libvirt is again the most common choice for virtual machine management with KVM. The first option is **`virsh`**, a command-line interface to libvirt that allows basic hypervisor and virtual machine management (including changing virtual machine configuration, if you know how to code the XML file format used by libvirt). Also, KVM virtual machines on IBM LinuxONE can be managed easily using recent versions of virt-manager, which is usually installed with **virt-viewer**, a tool for accessing the console of a KVM virtual machine.

Red Hat OpenStack provides another option for virtual machine management with KVM. This is more complicated however, as a complex set of services need to be configured on the KVM host to allow it to be managed by a Red Hat OpenStack orchestrator.

Virtual machine migration

KVM has a construct for relocating virtual machines between hypervisors, which it refers to as migration. Virtual machine migration is a developing area for KVM, which has several provisos and considerations for its use.

Migration of memory pages is done over TCP connections, over the standard network interfaces of the KVM host. By default, there is no encryption of this TCP connection, so the memory of the guest being migrated appears on the network in the clear. Also, problems in the network during migration interfere with it completing successfully. Conversely, the bandwidth used by the migration might interfere with other services on the network.

Open VSwitch (OvS) for KVM MacVTap

MacVTap is a new device driver meant to simplify virtualized bridged networking. It replaces the combination of the tun/tap and bridge drivers with a single module based on the **macvlan** device driver. A MacVTap endpoint is a character device that largely follows the tun/tap **ioctl** interface and can be used directly by KVM, qemu, and other hypervisors that support the tun/tap interface. The endpoint extends an existing network interface, the lower device, and has its own mac address on the same Ethernet segment. Typically, this is used to make both the guest and the host show up directly on the switch to which the host is connected.

A key difference between using a bridge or a MacVTap is that MacVTap connects directly to the network interface in the KVM host. This direct connection effectively shortens the codepath by bypassing much of the code and components in the KVM host associated with connecting to and using a software bridge. This shorter codepath usually improves throughput and reduces latencies to external systems.

KVM Linux bridge

KVM Linux bridge support enables connecting 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.

2.4.3 z/VM Single System Image (SSI)

A z/VM 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. A z/VM SSI cluster consists of up to four z/VM systems in an Inter-System Facility for Communications (ISFC) collection. ISFC is a function of z/VM nucleusthat provides communication services between transaction programs on interconnected z/VM systems. Each z/VM system is a member of the SSI cluster.

Figure 2-7 on page 36 shows the basic structure of a cluster with four members. The cluster is self-managed by the nucleus using ISFC messages that flow across channel-to-channel (CTC) devices between the members. All members can access shared DASD volumes, the same Ethernet LAN segments, and the same storage area networks (SANs).

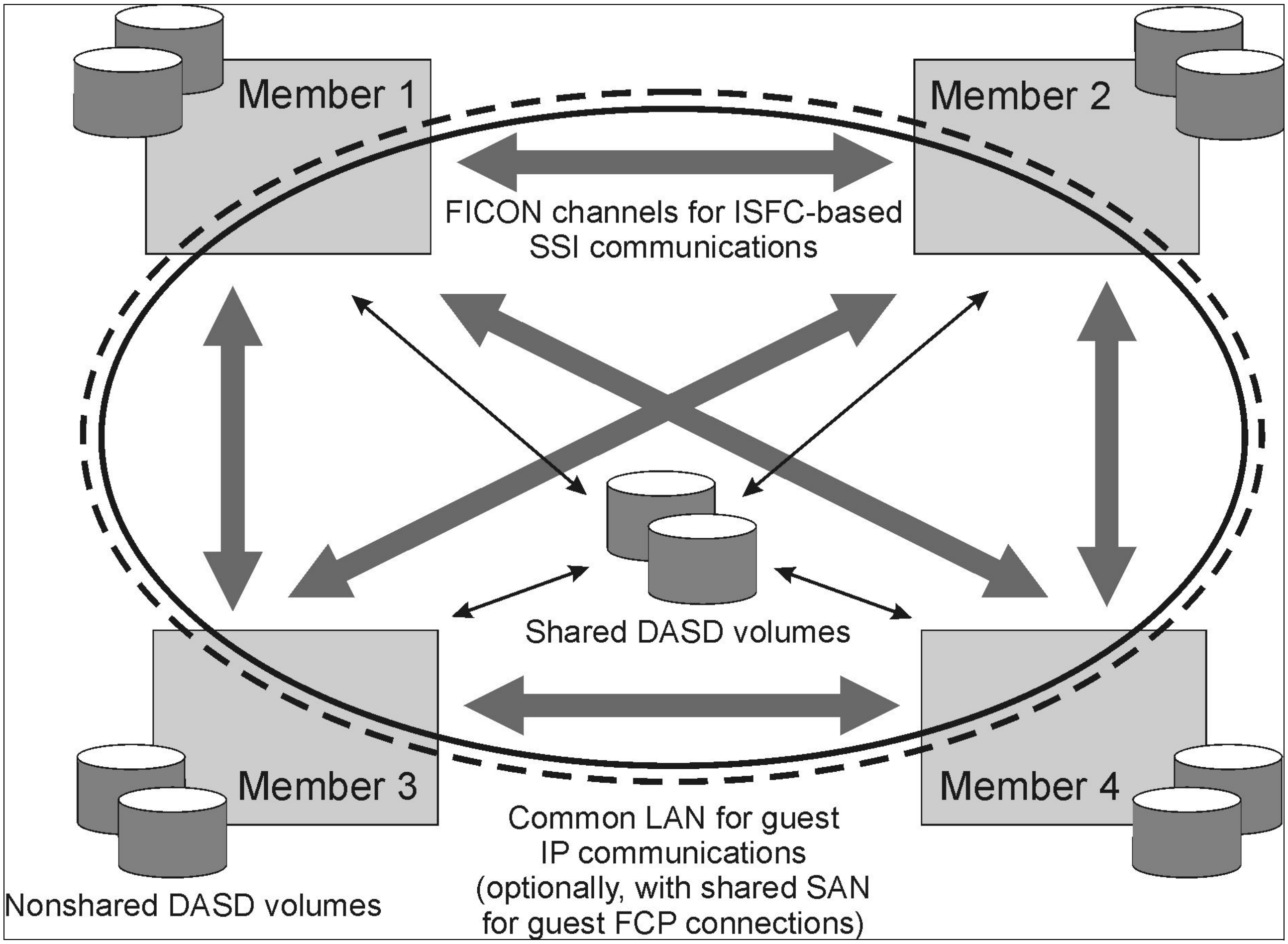


Figure 2-7 A four-member z/VM SSI cluster.

A useful feature in an SSI cluster is Life Guest Relocation (LGR). LGR is able to move a running Linux guest to another member in the cluster. This can be useful in the following situations:

- Memory or CPU member that is constrained

  You can either move more hardware resources to this member or move heavy or important guest systems off this member.

- Activate service to the z/VM hypervisor

  After the installation of fixes or a recommended service, you need to restart the z/VM member. LGR moves away the active Linux guests without taking them down. You are able to restart this z/VM member without stopping your services.

- Distribute Linux guests

  You can move away some Linux guests to prepare the member for a special workload, such as Ultimo. You can also distribute the guests to reach a nearly equal utilization of your z/VM members.

2.4.4 IBM Infrastructure Suite for z/VM and Linux

IBM provides a collection of operational monitoring and management tools for automating the backup, recovery, and performance of the IBM LinuxONE platform. It is highly recommended that clients include the use of this solution to support the qualities of service that running Temenos on this platform requires.

This solution is composed of the following products:

- 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 Wave for z/VM

IBM Tivoli OMEGAMON XE on z/VM and Linux

IBM Tivoli OMEGAMON XE on z/VM and Linux enables you to view z/VM data pulled from the Performance Toolkit for VM alongside views of Linux on IBM LinuxONE performance data. This multiple view capability helps you solve problems more quickly and manage complex environments more effectively.

Figure 2-8 on page 37 show a sample panel. All the panels can be tailored individually.

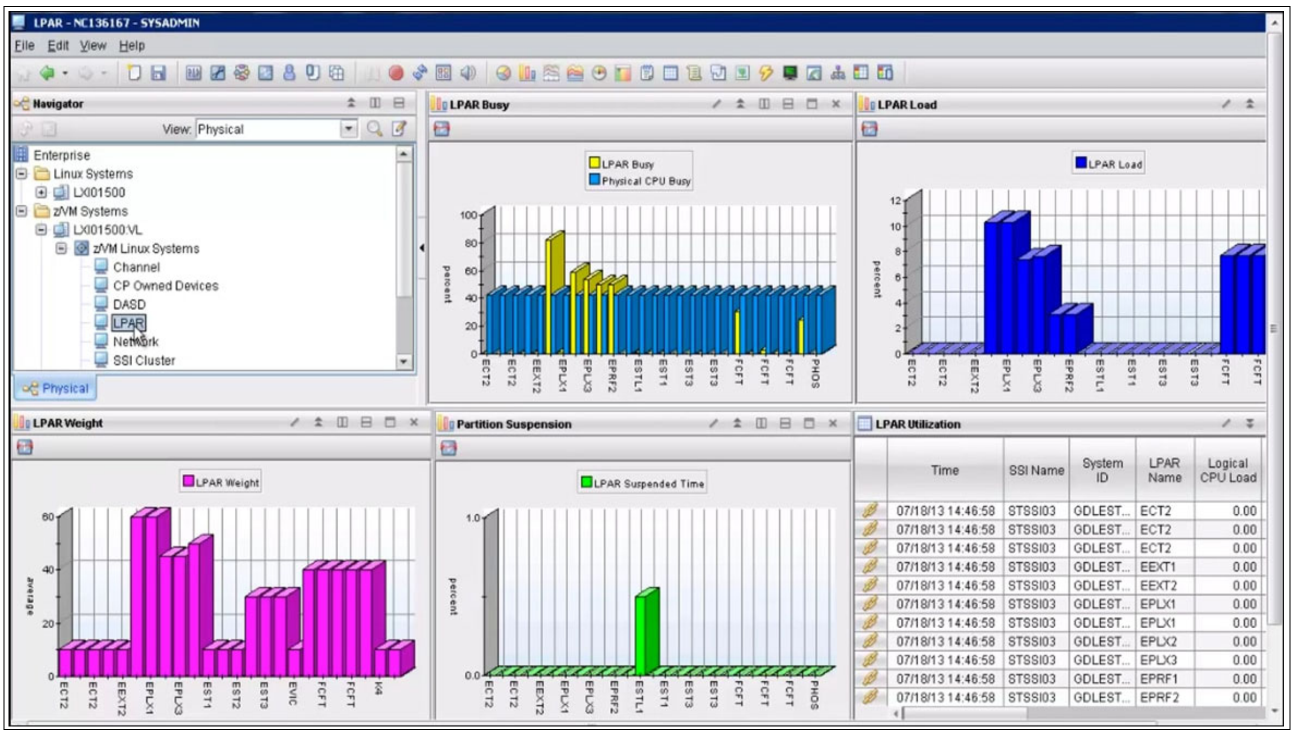


Figure 2-8 Sample window of IBM Tivoli OMEGAMON XE on z/VM and Linux.

IBM Spectrum Protect Extended Edition

IBM Spectrum Protect provides scalable data protection for physical file servers, applications, and virtual environments. Organizations can scale up to manage billions of objects per backup server. They can reduce backup infrastructure costs with built-in data efficiency capabilities and the ability to migrate data to tape, public cloud services and on-premises object storage. IBM Spectrum Protect can also be a data offload target for IBM Spectrum Protect Plus, providing an ability to use your existing investment for long-term data retention and disaster recovery.

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 can better interpret messages and determine corrective actions. Figure 2-9 on page 38 shows how Operations Manager for z/VM possible interactions.

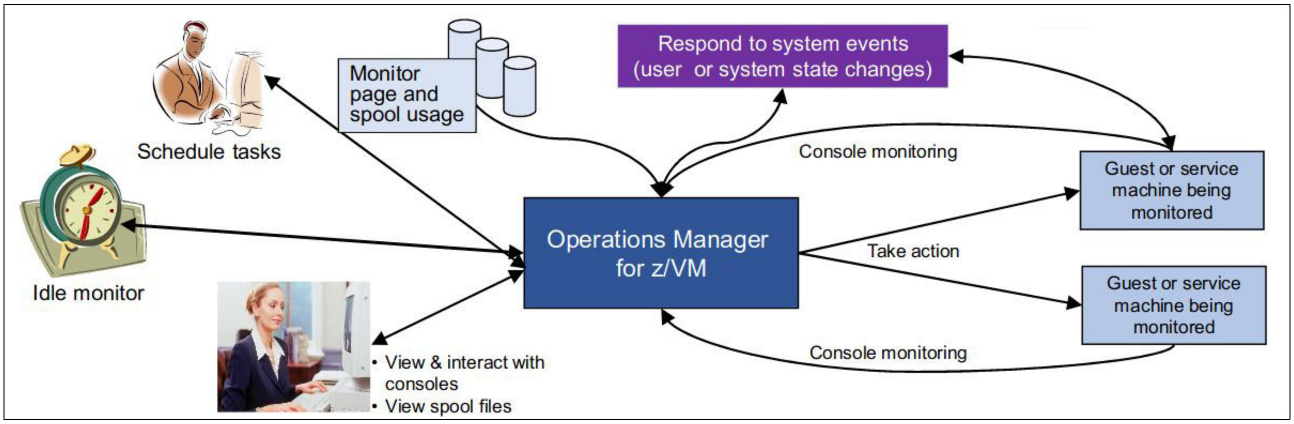


Figure 2-9 Operations Manager for z/VM.

IBM Backup and Restore Manager for z/VM

Back up and restore files and data on z/VM systems and images of non-z/VM guest systems such as Linux. IBM Backup and Restore Manager for z/VM makes your data available using simplified facilities for files, minidisks, Shared File System (SFS) file spaces and full system data backups and restores. It can perform backup and restore functions more efficiently by optimizing operations for each data type. It also provides a flexible file selection with support for wildcard characters and supports backups to disk, physical tape, or virtual tape. It is available as a single offering and as part of IBM Infrastructure Suite for z/VM and Linux.

IBM Wave for z/VM

IBM Wave for z/VM is intuitive virtualization management software that provides management, automation, administration, and provisioning of virtual servers, enabling automation of Linux virtual servers in a z/VM environment.

It helps simplify administration and management of Linux guests, is designed to integrate seamlessly with z/VM and Linux environments, and helps administrators view, organize, and manage system resources. System management is easily learned and reduces the dependence on z/VM experts.

You can define and control network, storage and communication devices, and view servers and storage utilization graphically with customizable views. It also allows you to monitor and manage your system from a single point of control. The low maintenance agentless discovery process detects servers, networks, storage, and more. You have the flexibility to instantly provision, clone, and activate virtual resources, optionally using scripts for additional customization. Routine management tasks are performed with ease. There is accountability built in as you can assign and delegate role-based administrative access with an audit trail of all activities performed. See Figure 2-10 on page 39.

Figure 2-10 Two screenshots from IBM Wave.

2.4.5 Geographical Dispersed Parallel Sysplex (GDPS)

Using GDPS allows a customer to achieve continuous availability for their IBM LinuxONE environment. GDPS ensures that if any System (CPC), LPAR, Hypervisor, Linux guest, or application fails, then GDPS automatically takes predefined actions to recover any outage and provides continuous availability for the business.

GDPS is an integrated, automated application and data availability solution designed to provide the capability to manage the storage subsystem(s) and remote copy configuration across heterogeneous platforms, automate Parallel Sysplex operational tasks, and perform failure recovery from a single point of control, thereby helping to improve application availability.

GDPS was initially designed for z/OS and extended to include z/VM and Linux running in z/VM. For IBM LinuxONE-only platforms, this solution has become known as GDPS VA (Virtual Appliance). GDPS is a collection of offerings each addressing a different set of IT resiliency goals that can be tailored to meet the recovery objectives for your business. Every one of the offerings uses a combination of services, server, and storage hardware and software-based replication and automation, and clustering software technologies to ensure that the solution fulfills your business objectives.

With IBM GDPS, you can be confident that your key business applications will be available when your employees, partners, and customers need them. Through proper planning and maximization of the IBM GDPS technology, enterprises can help protect their critical business applications from an unplanned or planned outage event.

When using ECKD formatted disk, GDPS Metro can provide the reconfiguration capabilities for Linux on IBM LinuxONE and data in the same manner as for z/OS systems and data. GDPS Metro can be used for planned and unplanned outages.

For a pure IBM LinuxONE environment, IBM offers GDPS Virtual Appliance (GDPS VA). It includes a predefined black boxed LPAR running z/OS with GDPS policies but requires no z/OS knowledge. In an IBM LinuxONE server, it will require that one processor is configured as a general-purpose engine (CP). GDPS VA runs with this limited CP capacity and it is the only authorized workload for this processor.

Figure 2-11 on page 40 show an IBM LinuxONE server with one LPAR running GDPS VA.

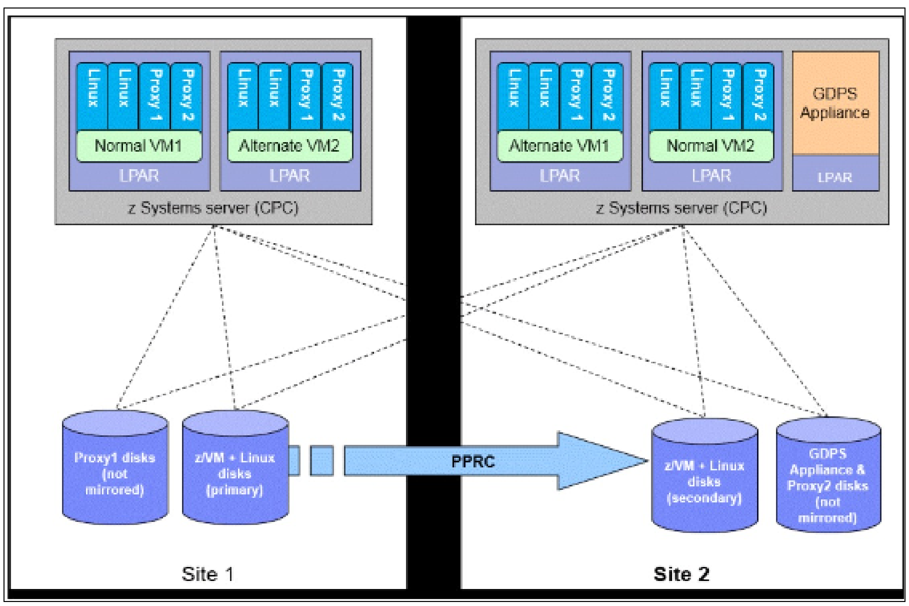


Figure 2-11 GDPS Virtual Appliance.

If you plan to implement GDPS and HyperSwap, the ECKD disk format is a requirement. GDPS can be a key component addressing your Recovery Point Objective (RPO) and Recovery Time Objective (RTO) requirements.

2.4.6 GDPS and Virtual Appliance (VA)

The GDPS Virtual Appliance supports both planned and unplanned situations, which helps to maximize application availability and provide business continuity. A GDPS 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 main objective of the GDPS Virtual Appliance is to provide these capabilities to clients using z/VM and Linux on IBM LinuxONE and do not have z/OS in their environments. The virtual appliance model that is used by this offering results in a solution that is easily managed and operated without requiring z/OS skills.

The functions provided by the GDPS Virtual Appliance fall into two categories: protecting your data and controlling the resources managed by GDPS. These functions include the following:

- Protecting your data:
  - Ensures the consistency of the secondary data if there is a disaster or suspected disaster, including the option to also ensure zero data loss
  - Transparent switching to the secondary disk using HyperSwap
  - Management of the remote copy configuration
- Controlling the resources managed by GDPS during normal operations, planned changes, and following a disaster:
  - Monitoring and managing the state of the production Linux for IBM LinuxONE guest images and LPARs (shutdown, activating, deactivating, IPL, and automated recovery)
  - Support for switching your disk, systems, or both to another site
  - User-customizable scripts that control the GDPS Virtual Appliance action workflow for planned and unplanned outage scenarios

For more information about GDPS and Virtual Appliance, visit the following link to download *"IBM GDPS: An introduction to Concepts and Capabilities:"*

https://www.redbooks.ibm.com/redbooks/pdfs/sg246374.pdf

2.4.7 HyperSwap

HyperSwap provides the ability to dynamically switch to secondary volumes without requiring applications to be quiesced. Typically done in 3–15 seconds in actual customer experience, this provides near-continuous data availability for planned actions and unplanned events.

HyperSwap can be used to switch transparently to secondary disk storage subsystems mirrored using Metro Mirror. If there is a hard failure of a storage device, GDPS coordinates the HyperSwap for continuous availability spanning the multi-tiered application. HyperSwap is supported for ECKD and xDR managed FB disk.

2.4.8 GDPS and HyperSwap

z/VM provides a HyperSwap function. With this capability, the virtual device associated with one real disk can be swapped transparently to another disk. GDPS coordinates planned and unplanned HyperSwap for z/VM disks, providing continuous data availability. For site failures, GDPS provides a coordinated *Freeze* for data consistency across all z/VM systems.

GDPS can perform a graceful shutdown of z/VM and its guests and perform hardware actions such as **LOAD** and **RESET** against the z/VM system's partition. GDPS supports taking a PSW restart dump of a z/VM system. Also, GDPS can manage CBU/OOCoD for IFLs and CPs on which z/VM systems are running.

2.4.9 Software in Linux

The software used when running Temenos Transact is specific and exact. In fact, there is only one specific version of the Linux operating system that is compatible for use. If an organization defers from the recommended list of software, Temenos can deny support.

The following components and minimum release levels are certified to run Temenos Transact.

- Red Hat Enterprise Linux 7
- Java 1.8
- IBM WebSphere MQ 9
- Application Server (noted in the following list)
- Oracle DB 12c

The Temenos Transact software is JAVA based and requires an application server to run. There are several options of application servers:

- IBM WebSphere 9
- Red Hat JBoss EAP
- Oracle WebLogic Server 12c (JDBC driver)

The Temenos Stack Runbooks provide more information about using Temenos stacks with different application servers. Temenos customers and partners can access the Runbooks through either of the following links:

- The Temenos Customer Support Portal: `https://tcsp.temenos.com/`
- The Temenos Partner Portal: `https://tpsp.temenos.com/`

IBM LinuxONE III has several hardware and firmware features for running Java-based workloads. These features would apply to Temenos applications running TAFJ. The performance benefits were based on using IBM SDK for Java 8 SR6 and are described in the informational graphic in Figure 2-12.

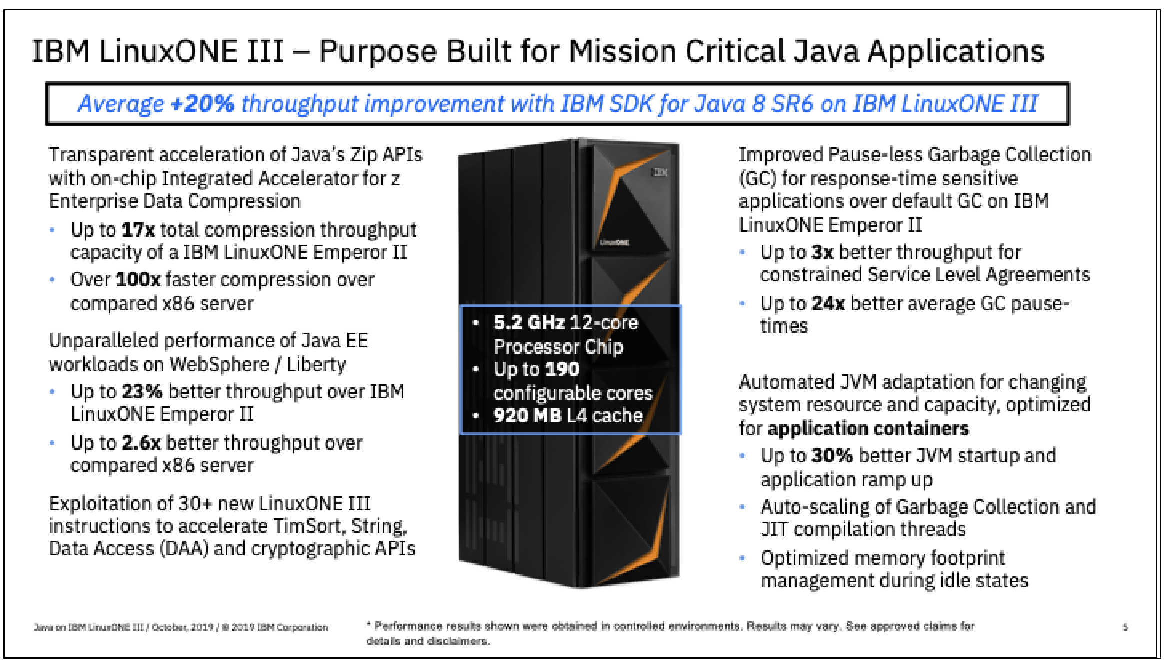


Figure 2-12 Java performance on IBM LinuxONE.

2.5 Temenos Infinity and Temenos Transact

Temenos' core banking solutions are centered around two products: Temenos Infinity and Temenos Transact. Both of these give banks the most complete set of digital front office and core banking capabilities. Using the latest cloud-native, cloud-agnostic technology, banks are able to rapidly and elastically scale, benefiting from the highest levels of security and multi-cloud resilience, generating significant infrastructure savings. Advanced API-first technology is coupled with leading design-led thinking and continuous deployment. As a result, banks are empowered to rapidly innovate, connecting to ecosystems and enabling developers to build in the morning and consume in the afternoon. These substantial benefits apply to banks whether they are running their software on-premises, on private or public clouds.

2.5.1 Temenos Infinity

Figure 2-13 illustrates the Temenos Infinity product architecture.

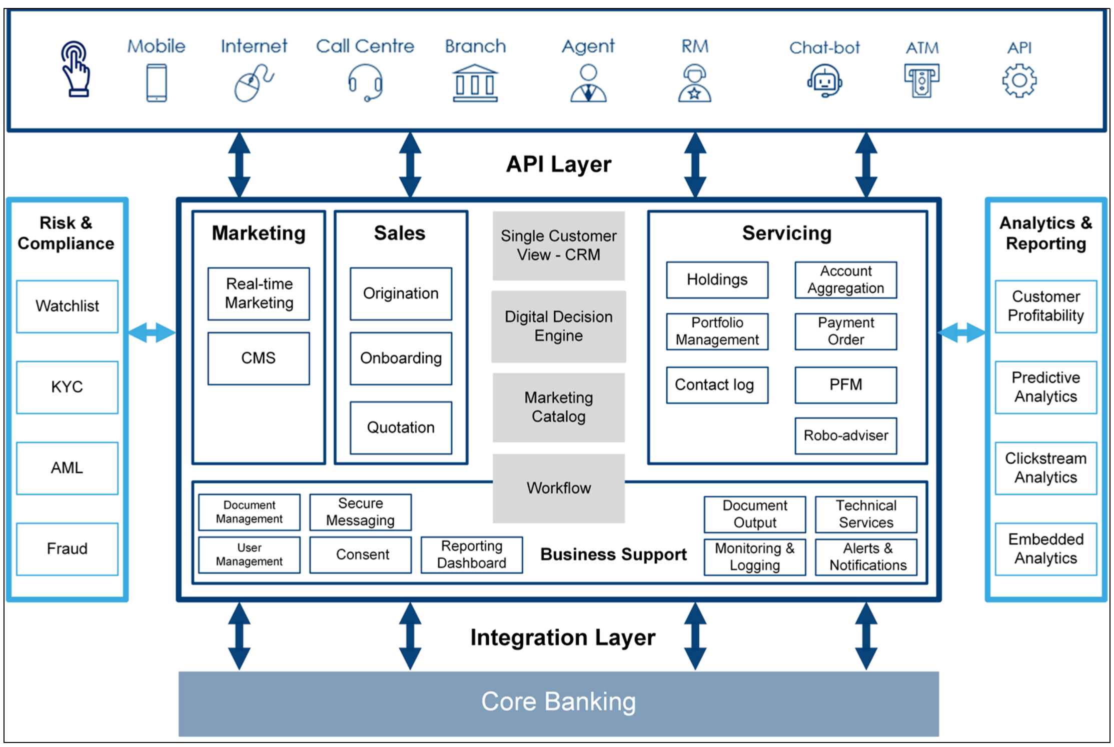


Figure 2-13 Temenos Infinity product architecture. [1]

Infinity uses APIs, rather than tight coupling, to connect to the bank's core banking platform, its peripheral systems and independent resources (such as Temenos Analytics or Risk and Compliance). This helps Infinity fit anywhere, and allows the bank to swap in, swap out or develop resources as its digital strategy evolves. Temenos open APIs (a readymade repository of over 700 customizable banking APIs) backed by a developer portal, makes it straightforward for banks to bring innovative third-party providers into the customer experience.

Infinity is design led, making it easier for banks to acquire, service, retain, and cross-sell to customers based on their needs. Infinity's design tools allow banks to quickly adjust or create

[1] Courtesy of Temenos Headquarters SA

workflows, products, and services. And its powerful AI and behavioral analytics capability helps the bank understand when and how to adjust its offering. One organization launched 47 products in a single year on Infinity, while another took just ten days to release CX changes, identified through analytics, to production.

Infinity's Real-Time Marketing capability helps demonstrate the product's emphasis on personalized customer journeys. Real-time events that are initiated by customers, the bank, or external parties, are subject to Infinity's digital decision engine. Every data source, from account activity and location to social media and biometrics, is analyzed to deliver the correct product offer through the correct channel to the correct person in real time.

Infinity's Origination solution applies a similar process to loans. Intelligent decisioning, a highly customizable workflow and smooth third-party integration are all designed to drive both customer satisfaction and build long lasting relationships with those customers.

Higher satisfaction rates are partly about speed, for example, one credit union achieved loan application times between 3-15 minutes, from application to funding. But the deeper explanation for Infinity's agility and flexibility lies in the decision to bring the marketing catalog, product details and banking processes out of Transact (the core banking product) and into a completely stand-alone front office.

Features

Temenos Infinity has the following features:

- Deployable in part or in whole, independent of the core banking system in use
- Easily integrated with other Temenos products, third-party providers and the bank's own peripheral systems
- Designed to offer a consistent customer experience across all banking capability, from acquisition and origination through to mobile banking and customer retention
- Design led, allowing banks to quickly adapt or create products and services that directly address customer needs
- Built to maximize analytics and AI to improve business agility and the customer experience

2.5.2 Temenos Transact

Figure 2-14 illustrates the Temenos Transact product architecture.

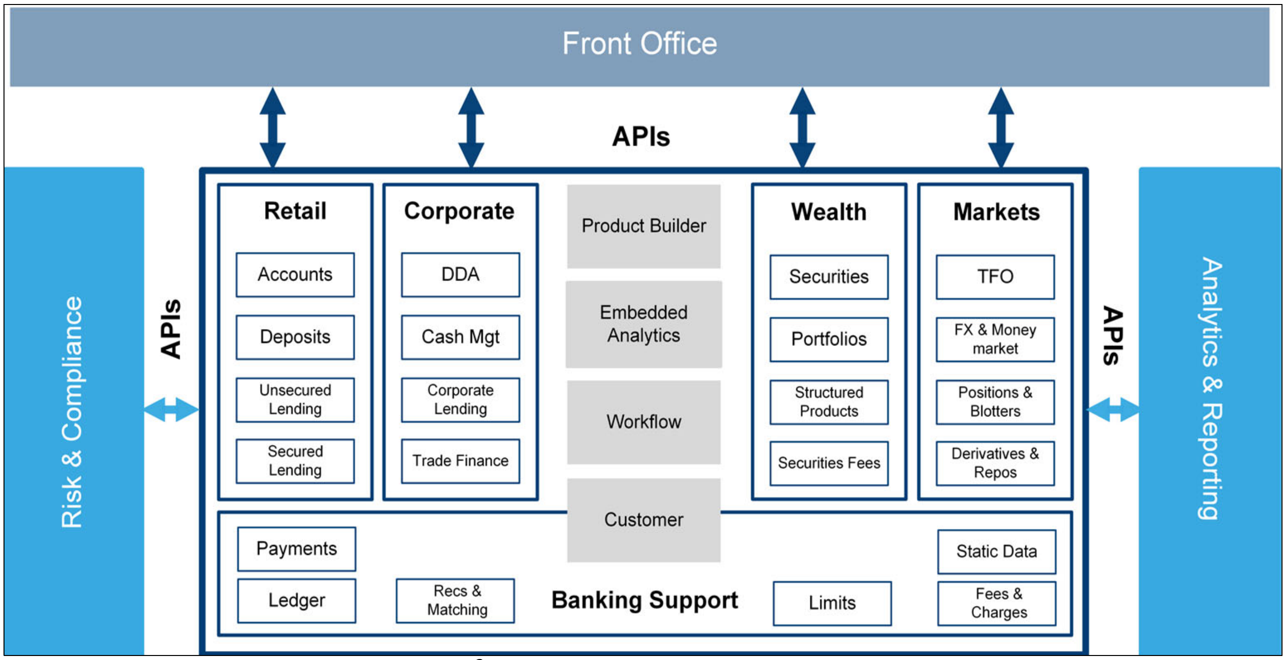


Figure 2-14 Transact product architecture. [2]

Since its creation over 25 years ago, Temenos has committed a significant proportion of its annual revenue to improving and expanding the functionality of Transact. For example, 148 enhancements were delivered to Transact in 2018 alone. Recent product announcements include Advanced Cash Pooling, Personalized Pricing and Fee Bundles, Capital Efficient Risk Limits, Analytics packs for the Retail, Corporate and Wealth sectors, and additional PSD2 and Customer Data Protection support.

Temenos has always emphasized its commitment to relentless innovation. But what this continuous development also recognizes is that banks need responsiveness in their core banking solution, not just their front office, to meet the challenges of Open Banking and the new technologies.

Transact, like Infinity, also operates on a cloud native, cloud agnostic technology platform. This doesn't commit organizations to a future in the cloud, of course, but it does allow banks to keep their options open, and maximize cloud technologies in on-premises deployments, and even avoid vendor lock in when the time is correct to move to the cloud.

The API led design of Transact allows banks to deploy the product independently of the front office. APIs make it easier to integrate Transact with the bank's wider ecosystem, including third-party providers, and even extend and modify the behavior of its banking capability. The introduction of open APIs, covering every aspect of core banking, together with a dedicated API Developer portal, helps banks maximize Fintech innovations and tailor their products to customer needs.

Temenos helps its core banking customers accelerate product development and shorten upgrade cycles still further through Temenos Continuous Deployment. This managed service applies the latest DevOps methodology to the design and delivery of functional innovations within the bank's implementation of Transact. Enhancements from both Temenos and the bank are automatically assembled, tested, and delivered in very short cycles, dramatically increasing the bank's ability to respond to both changes in the marketplace and new opportunities.

[2] Courtesy of Temenos Headquarters SA

Features

Transact has the following features:

- Offers the broadest and deepest set of core banking functionality available in the market today, from retail, mobile, and corporate through to treasury, payments and country or regional solutions
- Deploys independent of front office, on any cloud or on-premises platform
- Features open APIs and a dedicated API Developer portal, enabling banks to engage with external innovation teams and companies and to use their own product innovations
- Is continually updated with new product enhancements. Banks can improve product development times still further by using Temenos Continuous Deployment, a managed DevOps service

2.6 Planning phase and best practices

This section provides planning guidance and best practices for your consideration as you define what products best meet your organization’s needs. The following sections provide information for those purposes in the following categories:

- 2.6.1, “IBM LinuxONE” on page 46
- 2.6.2, “Network” on page 48
- 2.6.3, “z/VM networking” on page 49
- 2.6.4, “Inter-user communication vehicle (IUCV)” on page 50
- 2.6.5, “Backup and Restore” on page 50
- 2.6.6, “System Monitoring” on page 51

2.6.1 IBM LinuxONE

This section gives recommendations for the planning phase in regard to hardware.

The minimum recommendation is to have an IBM LinuxONE server configuration with two servers. This enables both high availability and disaster recovery functions. Install them with a location distance as described in 2.1, “Hardware” on page 18. The server configuration should contain sufficient resource capacity for both servers and for running production and non-production environments. These resources include processors, memory, I/O channels, networking products (such as SAN Switches), security firmware and both primary and secondary storage controllers and disks.

LPARs

The target is to run a z/VM SSI cluster with four members for production workload. Also, plan additional z/VM SSI clusters for your additional stages (test, pre-production, and so on). Distribute the LPARs equally across your IBM LinuxONE servers.

Each LPAR has a weight factor. The value for weight is finally defined in the share this LPAR receives. This calculation is not easy because it is a relative value. It is relative to the other active LPARs. The key messageis the phrase *active LPAR*. Any LPAR that is defined but not activated does not apply in this calculation. Sum the value for weight for all active LPARs. This result is 100%. Then, divide the LPAR weight by the sum and the result is the percentile of the machine this LPAR is able to receive. Note this is only a good approximation as there are more items influencing the share. Some of those factors are identified in the following list:

- Too few logical processors assigned to the LPAR which then cannot use all the share
- A processor-bound workload instead of an I/O-bound workload

- Dominant LPARs

Contact your IBM representative for a deeper discussion in this area.

Processor cores

Depending on the model of the IBM LinuxONE server, there is a maximum number of cores (IFL) available. The best way to get the maximum flexibility in the computing power is to define the cores as shared and define a number of shared cores to the LPARs. Do not dedicate cores to LPARs; you lose the computing power of such cores in the other LPARs if they are not fully utilized. Plan for Capacity on Demand (either Capacity BackUp (CBU) or On/Off Capacity on Demand (OOCoD)) in the activation profile based on the variability of your peak workloads and planned or unplanned downtimes. These logical processors are initially offline but they can be activated dynamically as your workload demands change. With these settings, you gain additional flexibility in the logical processor assignment.

> **Note:** Software licensing must be considered when defining shared IFLs, OOCoD, and CBU with products that are core licensed with the total number of IFLs available.

Each IFL has a pre-determined capacity based on the server model and capacity marker indicator. These processors run at 100% capacity all the time. By enabling Simultaneous Multi-Threading (SMT-2), you can increase the concurrent transaction processing capacity based on the individual task assignment. For Java based workloads (and according to Temenos testing in 2017), SMT-2 enablement increased transaction throughput by approximately 25%.

IBM LinuxONE has an internal function called HiperDispatch (HD) which attempts to utilize the cores as effectively as possible and to manage dispatching of the LPAR workload. HiperDispatch monitors the cache misses in the internal processor caches. A cache miss means this core waits and does nothing until the data is available in cache. The fewer cache misses that occur the more effective the core. The fewer LPARs a core has to work for, the fewer cache misses that occur. This is exactly the function of HD. HD tries to always dispatch an LPAR on the same physical cores. If this is not possible, HD parks a logical processor for a specific LPAR to reduce the number of LPARs one core is working for. HD is doing this calculation periodically. So, any change in the current workload (prime shift, off shift, peaks and so on) get incorporated in this calculation.

This calculation is called vertical assignment. The following list shows the four levels of vertical assignment ranked from highest to lowest performance for a core:

1. Vertical High
2. Vertical Medium
3. Vertical Low
4. Parked

Cores assigned to vertical high allows the best performance for using IBM LinuxONE processors.

Processor memory

The IBM LinuxONE server has a specific amount of memory installed. Calculate the memory requirements of your workload for each application instance and assign these portions of memory to each corresponding LPAR in the activation profile to the appropriate LPARs. z/VM offers the possibility to overcommit memory but do not apply this to your production environment(s).

Overcommitting memory can be considered and estimated for the non-production (dev/test/QA) environments. As a starting point, calculate the memory need of the applications or database systems and add a factor for your estimated growth.

It is also recommended to plan for reserved storage to each LPAR in the activation profile. Doing so enables you to add memory to or shift memory between LPARs without taking them down.

If you still have memory left at the end of your calculation, do not keep this memory unused in the machine. Assign this portion again to the LPARs. System memory that is not used by an application will be used for caching (both in z/VM and in Linux).

2.6.2 Network

IBM LinuxONE offers several types of external and internal network interfaces as noted in the following list:

- Open Systems Adapter (OSA)
- HiperSockets (HS)
- Share Memory Communication (SMC)

Open Systems Adapter (OSA)

OSA is the network interface for connecting to a LAN using Ethernet. The OSA Express family of hardware features provides TCP/IP connectivity over either fiber or copper Ethernet cabling (depending on the type of card). These interfaces can be used for both IPv4 and IPv6 traffic, and support a variety of Ethernet and IP functions such as Virtual LANs, Link Aggregation, and checksum offload. A single OSA Express card can be shared between multiple LPARs on an IBM LinuxONE server.

Both z/VM and Linux have the ability to group these interfaces. OSA also supports VLAN for network separation.

HiperSockets (HS)

HiperSockets provides a virtual high-speed low-latency network technology within the IBM LinuxONE server. LPARs in an IBM LinuxONE server can be attached to a HiperSockets channel, which functions like an IP network. Up to 32 HiperSockets networks can be created on an IBM LinuxONE server, with each HiperSockets network also providing VLAN support for further traffic separation if needed.

Another key feature of HiperSockets is support for large transmission frame sizes, which can allow for an IP Maximum Transmission Unit (MTU) of up to 56 kilobytes to be used.

Share Memory Communication (SMC)

A new type of network technology has developed for IBM LinuxONE based on Shared Memory Communications (SMC). Two types of SMC technology are available on IBM LinuxONE:

- SMC-R: RoCE Express features
- SMC-D: Internal Shared Memory (ISM)

All TCP connections start with the *three-way handshake*; the standard flow that starts a TCP connection. Systems that support SMC include extra information about the SMC support they each have in their handshake packets. If it is determined that they are both attached to the same SMC environment, they switch their data transfer from the Ethernet path used to start the connection to the SMC path that they negotiate. SMC reduces TCP/IP overhead by a direct point-to-point communication between two sockets.

SMC-R: RoCE Express features

The RoCE Express family of hardware features use Remote Direct Memory Access (RDMA) over Converged Ethernet (RoCE) to provide SMC over a physical Ethernet network.

SMC over this path is referred to as SMC-R.

> **Note:** The RoCE Express2 feature, available on IBM LinuxONE III, IBM LinuxONE Emperor II, and IBM LinuxONE Rockhopper II servers, can be used by Linux to support both standard Ethernet-based IP communication and SMC-R.

SMC-D: Internal Shared Memory (ISM)

Similar to HiperSockets, ISM is a virtual channel type that implements the SMC protocol within a single IBM LinuxONE server. This means that a hardware RoCE Express card is not required and the communication can be even faster than SMC-R.

SMC over the ISM path is referred to as SMC-D.

2.6.3 z/VM networking

z/VM also offers additional network facilities that can be used as stand-alone or in conjunction with the listed network facilities of an IBM LinuxONE server (noted in 2.6.2, “Network” on page 48).

z/VM VSWITCH

VSWITCH virtualizes an OSA by connecting a real OSA to it and defining a virtual NIC device to the guest. It allows easy and central management of the network connectivity for the guest and it offers several security functions such as network separation and isolation. This simple management is useful especially if you clone Linux guests or install them from templates. Each guest must be granted use of VSWITCH. This grant is granular and can be qualified down to VLANs or port numbers.

To achieve redundancy, VSWITCH can drive up to three OSA ports in active-backup mode.

z/VM Virtual Switch Link Aggregation

Link Aggregation (LAG) support for the virtual switch was first introduced in z/VM starting with z/VM Release 5.3. The virtual switch, configured in Ethernet mode, supports the aggregation of multiple OSA-Express features for external LAN connectivity. By supporting IEEE 802.3ad Link Aggregation, the combination of multiple OSA-Express features appears as one large link. The deployment of this type of configuration increases the virtual switch bandwidth and provides near seamless failover if a port becomes unavailable.

This support provides the ability to aggregate physical OSA-Express features for use by a single virtual switch (Exclusive mode) or by multiple virtual switches (Multi-VSwitch LAG mode).

Both Exclusive and Multi-VSwitch LAG configurations provide the same industry standard IEEE 802.3ad Link Aggregation protocol support with an external partner switch. Both LAG configurations are completely transparent to the z/VM guest hosted by a simulated NIC connected directly to the VSwitch.

2.6.4 Inter-user communication vehicle (IUCV)

z/VM uses IUCV for communication between virtual machines. IUCV needs to be authorized in the directory for each virtual machine.

Linux netiucv Driver

In Linux, there is a network driver called `netiucv` that uses this communication vehicle to provide a point-to-point IP network connection between two virtual machines. The `netiucv` driver is compatible with the `z/VM TCP/IP IUCV` driver. IUCV does not need a virtual network interface (like a virtual OSA or CTCs) to be defined, and it works cross-system within an SSI cluster.

For general networking, `netiucv` has been replaced by the z/VM Guest LAN and Virtual Switch. Because Guest LANs and VSwitches implement Ethernet-like networks, they are generally simpler to configure and more flexible. Note that `netiucv` links are point-to-point, so connecting many virtual machines using `netiucv` is cumbersome. However, there are still cases where `netiucv` can be used:

- Restricted Linux administrative access

  An extra Linux instance can be installed offering administrative access to all the other Linux systems in the z/VM SSI cluster using IUCV. This network is isolated from the LAN. You can also use this technique to create a secure transport between a Linux system and z/VM service virtual machines (such as the System Management API servers).

- Heartbeat network

  A heartbeat network is a private network within a cluster shared only by specific cluster nodes. It is used to monitor each individual node in the cluster and for coordination within the nodes. A heartbeat network needs to be as different and as separated as the primary LAN network. IUCV is a recommended choice as a fallback or emergency heartbeat because it does not require a physical interface and runs through the CTC connections within the SSI cluster.

Linux IUCV-based emergency console

IUCV can also be used to implement an emergency system console. This is particularly useful under z/VM as the default console through the z/VM virtual machine is a line-mode terminal interface with limited functionality. The IUCV-based console is not an IP networking connection, but it is analogous to a serial port console used on networking hardware and some embedded Linux systems.

Using the IUCV-based console system, one or more Linux guests can be configured to connect to all others using IUCV. On all the other guests, a terminal manager program (usually *agetty* or *mingetty*) is run against the `/dev/hvc0` device that represents the IUCV connection. From the management Linux guest, the `iucvconn` program is then used to connect over IUCV to the destination Linux console.

> **Note:** For more information about the IUCV-based console support, including full instructions on how to set it up, see the manual, "*How to Set up a Terminal Server Environment on z/VM*," SC34-2596.

2.6.5 Backup and Restore

Backup is an important task in preparing for situations beginning with single disk failures up to disaster recovery. The first questions about backup are similar to those in the following list:

- Which data do we need to be backup?
- Where do we perform the backup?
- Where do we store the backup?

Copy functions

Disk storage can have copy functions implemented but it depends on the type and model as to what functions are available. An IBM DS8000 provides advanced copy services including FlashCopy and Mirroring.

z/VM

The backup and restore of the z/VM hypervisor is based on the FICON attached disk.

Infrastructure Suite

IBM Backup and Restore Manager, part of the IBM Infrastructure Suite, is the product to manage this type of backup and restore. Backups can be performed in a periodic manner at the file or image level and as incremental or full backups.

If your environment does not use FICON attached disk, you need to think about how to back up and restore the z/VM hypervisor. There are several options available using built-in tools but there is no common best practice available. You need to have a discussion with your IBM representative to develop an individual solution.

Operating system

An operating system offers some built-in tools for backup purposes. A storage subsystem can offer some functionality that can be used to back up application user data. There are commercial solutions available that offer full and incremental backups with snapshots. Depending on the failure that you want to address and the time you have available before services are required up again (RTO), the appropriate technology must be chosen.

IBM Spectrum Protect

IBM Spectrum Protect provides scalable data protection for physical file servers, applications, and virtual environments. This product can be used to back up and restore Linux systems including their data. This suite provides both file-based clients and advanced agents for special purposes. IBM Spectrum Protect Suite is both available as a single product or as part of the IBM Infrastructure Suite.

2.6.6 System Monitoring

To monitor the z/VM system there are several products available to obtain insights and reports from the hypervisor:

- Performance Toolkit (see “Features and additional software products for z/VM” on page 33) as an optional priced feature of z/VM
- IBM OMEGAMON for z/VM (see “IBM Tivoli OMEGAMON XE on z/VM and Linux” on page 37) as a separate product or part of the IBM Infrastructure Suite

Both products can monitor Linux systems and the z/VM hypervisor. Linux distributions have commands and packages included that are used for monitoring. You can also buy other commercially available solutions.

3

Architecture

Running Temenos applications on IBM LinuxONE provides a robust Enterprise platform for mission critical banking services. In designing the correct solution, there are a number of architectural options. Choosing the correct path varies based on your own or your clients' architectural foundations, which are often influenced by budgetary constraints. Architectural workshops should be run to reach agreement about the correct ingredients and should encompass both the functional (application, database, system) and non-functional (availability, security, integrity, reliability) characteristics of your requirements.

In this chapter, a sample *Reference Architecture* is proposed and is based on a two server HA/DR configuration and additional components and decision points as appropriate.

The types of architecture are:

- Traditional on-premises, non-containerized solution
- IBM Cloud Hyper Protect/SSC
- Cloud native OpenShift/ICP on-premises

> **Note:** This chapter focuses on the Traditional on-premises, non-containerized solution and later updates to this book will include the other types of architecture including cloud native and on-premises cloud as they become available.

The following sections are covered in this chapter:

3.1 Traditional on-premises (non-containerized) architecture

Temenos Transact can be deployed in a variety of infrastructure environments. This chapter focuses on the Traditional on-premises, non-containerized solutions.

Perhaps unlike any other system architecture on which the Temenos applications can be installed, IBM LinuxONE provides alternatives for hypervisor and other aspects of deployment. The following paragraphs describe some of the architectural choices available on IBM LinuxONE and their considerations.

3.1.1 Key benefits of architecting a new solution instead of lift-and-shift

When migrating a deployed Temenos stack from another system architecture to IBM LinuxONE, it might be tempting to preserve the system layout currently implemented on the other system. This is certainly possible with IBM LinuxONE, by defining the same number of virtual instances and installing them with the same application structure as was previously deployed.

However, architecting a new solution specifically for IBM LinuxONE allows you to take advantage of the following important capabilities:

- Scalability, both horizontal and vertical
- Hypervisor clustering
- Reliability, Availability, and Serviceability

IBM LinuxONE scalability

The IBM LinuxONE server can scale vertically to large processing capacities. This scalability can be used to consolidate a number of physical machines of other hardware architectures to a smaller number of IBM LinuxONE servers. This simplifies the hardware topology of the installation by allowing more virtual instances to be deployed per IBM LinuxONE server.

On other architectures, the total number of instances deployed might be greater than the number required on IBM LinuxONE. A single virtual instance on IBM LinuxONE can scale vertically to support a greater transaction volume than is possible in a single instance on other platforms. Alternatively, you can decide to employ horizontal scaling at the virtual level and use the greater capacity per IBM LinuxONE footprint to deploy more virtual instances. This can provide more flexibility in workload management by lessening the impact of removing a single virtual instance from the pool of working instances.

Hypervisor clustering

The z/VM hypervisor on IBM LinuxONE provides a clustering technology known as Single System Image (SSI). SSI allows a group of z/VM systems to be managed as a single virtual compute environment. Most system definitions are shared between the members of the cluster, providing these benefits:

- Consistency in the system definition process: no need to replicate changes between systems as the systems all read the same configuration
- Single source for user directory: all definitions of the virtual instances are maintained in a single source location, again eliminating the need to replicate changes between systems
- Flexibility for deployment of virtual instances: allowing functions such as start and stop, live-relocate, and virtual instances between member z/VM systems

Other system architectures are more complex to manage in a clustered fashion, or approach hypervisor clustering in different ways that can adversely affect the workloads deployed or not provide the expected benefits.

Reliability, Availability, and Serviceability (RAS)

Other hardware platforms often require more physical systems than are actually needed to ensure that a failure does not affect operation. This means that, in normal operation, other hardware platforms are underutilized or oversized to withstand spikes in demand or system failures. It is also necessary to install additional equipment so that removing a system for maintenance (installation of new components, firmware, or OS patching) does not interrupt service.

An IBM LinuxONE server is designed to provide the highest levels of availability in the industry. First, the internal design of the system provides a high degree of redundancy to greatly reduce the likelihood that a component failure will affect the machine availability. Secondly, the IBM LinuxONE server provides functions that allow it to remain fully operational during service actions such as firmware updates. This means that in the majority of cases an IBM LinuxONE server does not have to be removed from service for hardware upgrades or firmware updates.

3.2 Machine configuration on IBM LinuxONE

On IBM LinuxONE the process of configuring the physical adapters and logical partitions, and resources (such as processors and memory allocation) is known as the I/O Definition process. There are two ways that this process can occur:

- The *traditional* method involving system configuration files known as the I/O Definition File (IODF) and the I/O configuration data set (IOCDS). This method also uses the Image Profile definitions on the Hardware Management Console (HMC).
- Dynamic Partition Manager (DPM) is a *new* configuration system and interface on the HMC that provides a graphical interface. The graphical method allows for configuring partitions, assigning network and storage resources to partitions, and assigning processor and memory resources.

> **Note:** Though DPM is simpler to use for newcomers to the IBM LinuxONE platforms, there are some limitations in supported configurations. Using the traditional IODF method ensures that partitions can utilize all hardware and software capabilities of the IBM LinuxONE platform. The recommended architecture, which uses z/VM SSI, requires IODF mode. This is because DPM is not able to configure the FICON CTC devices needed for SSI.

3.2.1 System configuration using IODF

An IODF is generated using an environment on z/VM known as Hardware Configuration Definition (HCD). A graphical Microsoft Windows based tool known as Hardware Configuration Manager (HCM) is used to generate the IODF. Then, HCD commands on z/VM are used to load the hardware portion of the IODF into the IODCS in the Service Element. This IOCDS is then used when a reset of the IBM LinuxONE system is performed.

Some degree of knowledge of I/O configuration on IBM LinuxONE is needed to perform this process. Understanding how to use the tools to create an I/O configuration and channel subsystem concepts is required to achieve a functional configuration.

Hardware Configuration and Definition (HCD)

HCD is the set of utilities used to create and manage IO Definition Files (IODFs).

On the z/OS operating system, HCD includes a rich Interactive System Productivity Facility (ISPF) interface for hardware administrators to manage IODFs. The ISPF interface for HCD is not provided on z/VM. So instead, a graphical Microsoft Windows-based tool called Hardware Configuration Manager (HCM) is used to interact with the HCD code in z/VM and perform IODF management tasks.

HCM is a client/server program that needs to have access to a z/VM host (over TCP/IP, to a server process called the HD Dispatcher). HCM also has a stand-alone mode that works separately from the Dispatcher. However, in the stand-alone mode, no changes can be made to IODFs.

The IODF process

The first step in updating a server's I/O configuration is to take the *production IODF* (the IODF that represents the machine's current operating configuration) and produce a *work IODF* from it. A production IODF cannot be edited, so it needs to be copied to make a new work IODF, which can be edited. Using HCM, the work IODF is customized with the changes that need to be made to the hardware configuration: adding or removing LPARs; adding, changing, or removing IBM LinuxONE server hardware; adding, changing, or removing disk subsystems; and so on.

After the changes are complete, the work IODF is converted to a new production IODF. This new production IODF can then be dynamically applied to the IBM LinuxONE server.

Stand-Alone I/O Configuration Program

When a new machine is installed, the first IODF has to be written to the machine using a limited-function version of some of the tools in HCD. This utility is called the Stand-Alone I/O Configuration Program (Stand-Alone IOCP) and is installed on every IBM LinuxONE system.

Stand-Alone IOCP is described in the IBM manual "Stand-Alone I/O Configuration Program User's Guide", SB10-7173-01.

Server's First IODF

Creation of the first IODF for an IBM LinuxONE server can be more complicated. As there is no operating system running on the server, how do we run HCD/HCM to create one?

If there is already an existing IBM LinuxONE server on which the IODF for the new machine can be created, the IODF for the new machine can be created there and then exported from the existing machine. Using Stand-Alone IOCP on the new machine, the IODF is written to the IOCDS of the new machine and can then be activated.

However, what if this machine is the first IBM LinuxONE server at your installation? In this case, Stand-Alone IOCP must be used to create a valid IODF. To make the process easier, rather than attempting to define the entire machine using this method a minimal IOCP deck defining a single LPAR and basic DASD can be used. This simple IOCP can be activated to make available a single system into which a z/VM system can be installed. This z/VM system is then used to download the HCM code to a workstation and start the HCD Dispatcher. HCM is then installed and used to create an IODF with more complete definitions of the system.

Note: An example of a minimal IOCP deck to perform this operation is provided in Appendix B, “Creating and working with the first IODF for the server” on page 107. The example lists important aspects and parts of the operation, enablement of the IOCP, and a success-verification example

Single IODF per installation

The data format of the IODF allows multiple IBM LinuxONE machines to be managed in a single IODF. This feature has distinct advantages over having separate IODFs, such as those noted in the following list:

- The visualization capabilities of HCM can be used to view the entire IBM LinuxONE installation at the same time
- Devices such as disk (DASD) subsystems, which usually attach to more than one server, can be managed more effectively
- A *wizard* in HCM can be used to configure CTC connections. The wizard can do this, though, only if both *sides* of the CTC link are present in the same IODF

When an IODF is written to the IOCDS of an IBM LinuxONE machine, HCD knows to write only the portions of the IODF that apply to the current machine.

I/O Configuration system roles

When multiple physical servers are in use, each physical server must at some point be able to access the IODF. Without using shared DASD, there is a possibility that each server might have a separate copy. With different copies of the IODF on different systems, it is important to always know which IODF is the *real* one.

We have created some definitions to describe the roles that various systems have in the I/O Definition process:

I/O Definition system
: This system is the one from which you do all of the HCM work of defining your I/O Configurations. This is the system you use to run the HCD Dispatcher when needed, and all of the work IODF files are kept there. As noted previously, there should be one I/O Definition system across your IBM LinuxONE environment.

I/O Managing system
: This system runs the HCD programs to dynamically activate a new IODF and to save the IOCDS. Each CPC requires at least one z/VM system to be the I/O Managing system. The I/O Definition system is also the I/O Managing system for the CPC on which it runs.

I/O Client system
: These are all the other z/VM systems in your IBM LinuxONE environment. These systems do not need a copy of the IODF, and they are not directly involved in the I/O definition process. When a dynamic I/O operation takes place (driven by the I/O Managing system), the channel subsystem signals the operating system about the status changes to devices in the configuration.

For backup and availability reasons, it is a good idea to back up or copy the IODF files (by default the files are kept on the A disk of the `CBDIODSP` user on the I/O Definition system). This allows another system to be used to create an IODF in an emergency.

Configuration system roles and SSI

z/VM SSI does not change the need for the roles described previously. However, it does simplify and reduce the number of systems that need to have their own copy of the IODF.

In an SSI cluster, the `PMAINT CF0` disk is common between the members of the cluster. This means that, if the I/O Managing systems for two CPCs are members of the same SSI cluster, those I/O Managing systems can share the same copy of the IODF. This reduces the number of IODF copies that exist across the IBM LinuxONE environment.

BAU IODF process

We recommend the following process for performing an update to the I/O definition in an IBM LinuxONE environment:

1. Plan the changes to be made, and gather required information (such as `PCHID/CHID` numbers, switch port IDs, and so on).
2. Log on to the **`CBDIODSP`** user on the I/O Definition system.
3. Start the HCD Dispatcher, using the **`CBDSDISP`** command.
4. On your workstation, start and log on to HCM.
5. Use the existing production IODF to create a new work IODF.
6. Open the work IODF in HCM.
7. Make whatever changes are required to the I/O configuration.
8. When changes are complete, build a new production IODF from the work IODF.
9. Transmit the new production IODF to any remote I/O Managing systems in the IBM LinuxONE environment.

   A variety of methods can be used to transmit the file:

   a. If Unsolicited File Transfer (UFT) has been set up on your z/VM systems, use the **`SENDFILE`** command with the **`UFTSYNC`** and **`NETDATA`** options to send the file to the spool of the I/O Managing system(s)

      `SENDFILE IODFxx PRODIODF A to CBDIODSP at` *`iomanager.`* `(UFTSYNC NETDATA`

      Where *iomanager* is the hostname or IP address of an I/O Managing system. The trailing '.' forces the command to treat the name as a TCP/IP hostname, which can be looked up using DNS or the **`ETC HOSTS`** file.

   b. Copy the file through a shared DASD volume;

   c. Use FTP, `IND$FILE`, or other file transfer method. Ensure that the record format of the IODF is preserved (it must be transferred as a binary file, with fixed record length of 4096 bytes).

10. If the I/O configuration is to be changed dynamically:

    a. Use HCD to test the activation of the new IODF on each IBM LinuxONE server that has I/O changes.

    b. If the test is successful, use HCD to activate the new IODF on each IBM LinuxONE server that has I/O changes.

11. Using HCD on each I/O Managing system, save the new IODF to the I/O configuration data set (IOCDS) on each IBM LinuxONE server Support Element.
12. Update either the `Active IOCDS` marker or the system `Reset` profile to indicate the new IOCDS slot for the next Power-on Reset (POR).

3.3 IBM LinuxONE LPAR Architecture

A system architecture implemented on IBM LinuxONE makes use of the Logical Partitioning (LPAR) capability of the server to create system images that operate separately from one another. These images can be used for different components of the architectures (for example, application and database tiers) or for different operational enclaves (for example Production, Test and Development, Stress testing, and so on). They can also be used to provide high availability.

The following paragraphs describe the layout of LPARs in the recommended architecture.

3.3.1 LPAR Layout on IBM LinuxONE CPCs

Based on the architecture diagram (shown in Figure 4-2 on page 87) the preferred design is to start with two IBM LinuxONE CPCs (which provide hardware redundancy). This allows hardware maintenance to occur on one CPC without impact to the production workload running on the other CPC.

The division and setup for logical partitions (LPARs) include the following aspects:

- Two LPARs for Core Banking Database

  There are a number of database solutions available for the TEMENOS Banking platform. When implementing any core banking database, high availability is key. Best practices suggest each IBM LinuxONE CPC have a z/VM LPAR with a minimum of two Linux guests hosting the core banking database. Isolating core banking databases in their own LPAR reduces the core licensing costs by dedicating the fewest number of IFLs to the core banking database.

  Oracle is the prevalent database used in Temenos deployments.

- Two LPARs for Non-Core Database Farm (this can include any databases needed for banking operations)

  In these LPARs, we can run the databases that support banking operations. These include credit card, mobile banking, and others. Each database will be running in a virtual Linux Guest running on the z/VM hypervisor.

- Four LPARs for Application servers

  The four LPARs run z/VM Hypervisor managed by a single system image (SSI). SSI allows sharing of virtual Linux guests across all four LPARs. Linux guests can be moved between any of the four LPAR clusters. This movement can be done by either of these methods:

  – Bringing a server down and then bringing that same server up on another LPAR
  – By issuing the Live Guest Relocation (LGR) command to move the guest to another LPAR without an outage of the Linux guest

SSI will also let you install maintenance once and push to the other LPARs in the SSI cluster.

Note: LGR is not supported for use with Oracle.

In this cluster, the Temenos Transact application server will run in each LPAR. This allows the banking workload to be balanced across all four LPARs. Each Temenos Transact application server can handle any of the banking requests.

3.4 Virtualization with z/VM

The z/VM hypervisor provides deep integration with the IBM LinuxONE platform hardware and provides rich capabilities for system monitoring and accounting.

z/VM was selected for this architecture to take advantage of several unique features of IBM LinuxONE. IBM LinuxONE uses these features to reduce downtime and system administration costs and are noted in the following list:

- GDPS for reduced and automatic failover in the event of an outage
- SSI clustering to manage the resources and maintenance of systems
- Live guest migration between LPARs or CPCs

z/VM provides a clustering capability known as Single System Image (SSI). This capability provides many alternatives for managing the virtual machines of a compute environment, including Live Guest Relocation (LGR). LGR provides a way for a running virtual machine to be relocated from one z/VM system to another, without downtime, to allow for planned maintenance.

How SSI helps virtual machine management

One of the important reasons SSI and LGR were developed was to improve availability of Linux systems and mitigate the impact of a planned outage of a z/VM system.

It is recommended that z/VM systems have Recommended Service Updates (RSUs) applied approximately every six months. When an RSU is applied to z/VM, it is usually necessary to restart the z/VM system. In addition, z/VM development uses a model known as *Continuous Delivery* to provide new z/VM features and functions in the service stream. If one of these new function System Programming Enhancements (SPEs) updates the z/VM Control Program, a restart of z/VM is required for those changes to take effect. Whenever z/VM is restarted, all of the virtual machines supported by that z/VM system must be shut down, causing an outage to service.

With SSI and LGR, instead of taking down the Linux virtual machines they can be relocated to another member of the SSI cluster. The z/VM system to be maintained can be restarted without any impact to service.

LGR is a highly reliable method of moving running guests between z/VM systems. Before you perform a relocation, test the operation to see whether any conditions might prevent the relocation from completing. Example 3-1 shows examples of testing two targets for relocations.

Example 3-1 VMRELOCATE TEST examples

```
vmrelo test zgenrt1 to asg1vm1
User ZGENRT1 is eligible for relocation to ASG1VM1
Ready;
vmrelo test zdb2w01 to asg1vm1
HCPRLH1940E ZDB2W01 is not relocatable for the following reason(s):
HCPRLI1997I ZDB2W01: Virtual machine device 2000 is associated with real device
2001 which has no EQID assigned to it
HCPRLI1997I ZDB2W01: Virtual machine device 2100 is associated with real device
2101 which has no EQID assigned to it
HCPRLL1813I ZDB2W01: Maximum pageable storage use (8256M) exceeds available
auxiliary paging space on destination (7211520K) by 1242624K
Ready(01940);
```

In the first example, all devices required by the guest to operate are available at the proposed destination system. In the second example, there are devices present on the virtual machine that **VMRELOCATE** cannot guarantee are equivalent on the destination system. Also, checks determined that there is not enough memory available on the destination system to support the guest to be moved.

When a guest is being relocated, its memory pages are transferred from the source to the destination over FICON CTC devices. FICON provides a large transfer bandwidth, and the CTC connections are not used for anything else in the system other than SSI. This means guest memory can be transferred quickly and safely.

User and security management

z/VM has built-in security and user management functions. The user directory contains the definitions for users (virtual machines) in the z/VM system and the resources defined to them: disks, CPUs, memory, and so on. The z/VM Control Program (CP) manages security functions such as isolation of user resources (enforcing the definitions in the user directory) and authorization of operator commands.

z/VM also provides interfaces to allow third-party programs to enhance these built-in functions. IBM provides two of these on the z/VM installation media as additional products that can be licensed for use:

- Directory Maintenance Facility for z/VM
- RACF Security Server for z/VM

Directory Maintenance Facility (DirMaint) simplifies management of the user directory of a z/VM system. RACF enhances the built-in security provided by CP to include mandatory access control, security labels and strong auditing capabilities.

Broadcom also provides products in this area, such as their CA VM:Manager suite, which includes both user and security management products.

When installed, configured, and activated, the directory manager takes responsibility for management of the system directory. A directory manager also helps (but does not eliminate) the issue of the clear text password.

Note: The directory manager might not remove the `USER DIRECT` file from `MAINT 2CC` for you. Usually the original `USER DIRECT` file is kept as a record of the original supplied directory source file, but this can lead to confusion.

We recommend that you perform the following actions if you use a directory manager:

- Rename the `USER DIRECT` file (to perhaps `USERORIG DIRECT`) to reinforce that the original file is not used for directory management
- Regularly export the managed directory source from your directory manager and store it on `MAINT 2CC` (perhaps as `USERDIRM DIRECT` if you use DirMaint). This file can be used as an emergency method of managing the directory in case the directory manager is unavailable. The DirMaint **USER** command can export the directory.

If you are not using an External Security Manager (such as IBM RACF), you can export this file with the user passwords in place. This helps its use as a directory backup, but it potentially exposes user passwords.

When a directory manager is used, it can manage user passwords. In the case of IBM DirMaint, z/VM users can have enough access to DirMaint to change their own passwords. Also, when a directory entry is queried using the DirMaint **REVIEW** command, a randomly-generated string is substituted for the password. However, it is still possible for privileged DirMaint users to set or query the password of any user. For this reason, the only completely effective way to protect against clear text passwords in the directory is to use an External Security Manager (such as IBM RACF).

3.4.1 z/VM installation

When installing z/VM on ECKD volumes, the preferred way is installing z/VM on 3390-9 volumes. z/VM 7.1 requires five 3390-9 volumes for the base installation (non-SSI or single member SSI installation) and an additional three 3390-9 volumes for each further SSI member.

The recommendation is to install z/VM as a two- or four-member SSI cluster with one or two z/VM members on each IBM LinuxONE server. You will be prompted to select an SSI or non-SSI installation during the installation.

If z/VM 7.1 is installed into an SSI, at least one extended count key data (ECKD) volume is necessary for the Persistent Data Record (PDR). If you plan to implement RACF, the database must be configured as being shared and at least two ECKD DASD volumes are necessary. Concurrent virtual and real reserve/release must always be used for the RACF database DASD when RACF is installed in an SSI.

3.4.2 z/VM SSI and relocation domains

The following paragraphs describe aspects of running the z/VM SSI feature in support of Linux guests.

FICON CTC

An SSI cluster requires CTC connections, always size them in a pair of two. If possible, use different paths for the cables. During normal operation, there is not much traffic on the CTC connection. LGR is dependent of the capacity of these channels, especially for large Linux guests. The more channels you have between the members, the faster a relocation of a guest completes. This is a valid reason to plan four to eight CTC connections between the IBM LinuxONE servers. Keep in mind, if you run only two machines, this cabling is not an obstacle. But if you plan to run three or four servers, the physical weight can become heavy as the connection must be point-to-point and you need *any-to-any* connectivity.

Note: FICON CTCs can be defined on switched CHPIDs, which can relieve the physical cable requirement. For example, by connecting CTC paths using a switched FICON fabric the same CHPIDs can be used to connect to multiple CPCs.

Also, FICON CTC control units can be defined on FICON CHPIDs that are also used for DASD. Sharing CHPIDs between DASD and CTCs can be workable for a development or test SSI cluster, this can further reduce the physical connectivity requirements.

Relocation domains

A relocation domain defines a set of members of an SSI cluster among which virtual machines can relocate freely. A domain can be used to define the subset of members of an SSI cluster to which a specific guest can be relocated. Relocation domains can be defined for business or technical reasons. For example, a domain can be defined having all of the architectural facilities necessary for a particular application, or a domain can be defined to allow access only to systems with a particular software tool. Whatever the reason for the definition of a domain, CP allows relocation among the members of the domain without any change to architectural characteristics or CP functionality as seen by the guest.

Architecture parity in a relocation domain

In a mixed environment (mixed IBM LinuxONE generations or z/VM levels) be cognizant of the architecture level. z/VM SSI supports cluster members running on any supported hardware. Also, during a z/VM upgrade using the `Upgrade In Place` feature, different z/VM versions or releases can be operating in the same cluster. For example, you can have z/VM 6.4 systems running on an IBM LinuxONE Emperor processor in the same SSI cluster as z/VM 7.1 systems running on an IBM LinuxONE III processor.

When a guest system logs on, z/VM assigns the maximum common subset of the available hardware and z/VM features for all members belonging to this relocation domain. This means that by default, in the configuration described previously, guests started on the IBM LinuxONE III server have access to only the architectural features of the IBM LinuxONE Emperor. There also can be z/VM functions that might not be presented to the guests under z/VM 7.1 because the cluster contains members running z/VM 6.4.

To avoid this, a relocation domain spanning only the z/VM systems running on the IBM LinuxONE III server are defined. Guests requiring the architectural capabilities of the IBM LinuxONE III or of z/VM 7.1 are assigned to that domain, and are permitted to execute only on the IBM LinuxONE III servers.

SSI topology in the recommended architecture

Our recommended architecture uses z/VM SSI to offer simpler manageability of the z/VM layer. The database LPARs are part of one SSI cluster and application-serving LPARs are part of another SSI cluster. Additional SSI clusters can also be employed for other workload types such as test, development, quality assurance and others.

3.4.3 z/VM memory management

z/VM effectively handles memory overcommitment. The factor for overcommitment depends on how your z/VM paging performs. This factor is calculated by the virtual memory (sum of all defined guests plus the shared segments) to the real memory available to the LPAR. For the production system, a value of 1.5:1 should be considered as a threshold to not pass. As a recommendation for production systems, plan the initial memory assignment for no overcommitment (overcommitment factor lower than 1:1) and with space to grow. For test and development systems, the value can reach up to 3:1.

3.4.4 z/VM paging

Paging is used to move memory pages out to disk in case of memory constraints. Sometimes z/VM also uses paging for reordering pages in memory. Normally, the system is sized for no paging. However, paging can still occur if memory is overcommitted, short-term memory is constrained, old 31-bit code is running and the required memory pages out of 31-bit addressability and so on.

Memory overcommitment

Virtual machines do not always use every single page of memory allocated to them. Some programs read in data during initialization but only rarely reference that memory during run time. A virtual machine with 64 GB of memory that runs a variety of programs can actually be actively using significantly less than the memory allocated to it.

z/VM employs many memory management techniques on behalf of virtual machines. One technique is to allocate a real memory page only to a virtual machine when the virtual machine references that page. For example, after our 64 GB virtual machine has booted it might have referenced only a few hundred MB of its assigned memory, so z/VM actually allocates only those few hundred MB to the virtual machine. As programs start and workload builds the guest uses more memory. In response, z/VM allocates it, but only at the time that the guest actually requires it. This technique allows z/VM to manage only the memory pages used by virtual machines, reducing its memory management overhead.

Another technique z/VM uses is a sophisticated least-recently-used (LRU) algorithm for paging. When the utilization of z/VM's real memory becomes high, the system starts to look for pages that can be paged-out to auxiliary storage (page volumes). To avoid thrashing, z/VM finds the least-recently-used guest pages and selects those for paging. Using a feature known as co-operative memory management (CMM), Linux can actually nominate pages it has itself paged out to its own swap devices. z/VM can then prioritize the paging of those pages that Linux can itself re-create, in a way that avoids the problem of *double-paging*.

These capabilities are why memory can be overcommitted on IBM LinuxONE to a higher degree with lower performance impact than on other platforms.

Paging subsystem tuning

Plan z/VM paging carefully to obtain the maximum performance using the following criteria:

- Monitor paging I/O rates. Excessive paging I/O means that virtual machines are thrashing, which can be due to insufficient real storage in z/VM
- Use fast disks for paging or enable tiering in your disk subsystem
- Leverage HyperPAV for paging devices and use fewer, larger devices

  Command: **SET PAGING ALIAS ON**

  Configuration file: **FEATURES ENABLE PAGING_ALIAS**
- If you do not use HyperPAV for paging, use these considerations:

  More smaller disk volumes are better than one large volume. As an example, use three volumes of type 3390-9 rather than one volume of type 3390-27. z/VM can then utilize three paging I/Os in parallel on the smaller volumes as opposed to only one I/O with the larger volume. The sum of all defined paging volumes is called paging space
- Continuously monitor your paging about usage (command **QUERY ALLOC PAGE** or **panel FCX109** in Performance Toolkit). z/VM crashes with **ABEND PGT004** if it runs out of paging space
- Monitor the Virtual-to-Real ratio, which reflects the amount of memory overcommitment in a z/VM system. A ratio of less than 1:1 means that the z/VM system has more memory than it needs. Above 1:1 means that some overcommitment is occurring
- Reserve or predefine slots for additional paging volumes in the z/VM system configuration file

Note these considerations when working with **AGELIST**, **EARYLWRITE**, and **KEEPSLOT**. It is important to save I/O for paging or paging space, especially for systems with a large amount of memory. **EARLYWRITE** specifies how the frame replenishment algorithm backs up page content to auxiliary storage (paging space). When `Yes` is specified, pages are backed up in advance of frame reclaim to maintain a pool of readily reclaimable frames. When `No` is specified, pages are backed up only when the system is in need of frames. **KEEPSLOT** indicates whether the auxiliary storage address (ASA) to which a page is written during frame replenishment should remain allocated when the page is later read and made resident. Specifying `Yes` preserves a copy of the page on the paging device and eliminates the need to rewrite the contents if the page is unchanged before the next steal operation. Keeping the slot might reduce the amount of paging I/O, but can result in more fragmentation on the device. See the *CP Planning and Administration Manual* from the z/VM documentation for details about **EARLYWRITE**. For environments where the overcommit level is low and large amounts of real memory are being used, you will want to consider disabling **EARLYWRITE** and **KEEPSLOT**.

See also the page space calculation guidelines that are located in the CP Planning and Administration Manual located at the following z/VM 7.1 library:

https://www-01.ibm.com/servers/resourcelink/svc0302a.nsf/pages/zVMV7R1Library?OpenDocument

3.4.5 z/VM dump space and spool

z/VM uses spool to hold several kinds of temporary (print output, transferred files, trace data, and so on) or shared data (such as Named Saved Systems and Discontiguous Saved Segments). Spool is a separate area in the system and needs disk space. For performance reasons, do not mix spool data with other data on the disk.

One important item is dump space. At IPL time, z/VM reserves a space in spool for a system dump. The size depends on the amount of memory in the LPAR. It is important to ensure that there is sufficient dump space in the spool.

The **SFPURGER** tool can be used to maintain the spool. If you use an automation capability (such as the Programmable Operator facility or IBM Operations Manager for z/VM) you can schedule regular runs of **SFPURGER** to keep spool usage well managed.

3.4.6 z/VM minidisk caching

z/VM offers a read-only caching of disk data. Per default, caching is permitted to utilize the whole memory. In some rare cases for workloads with a high read I/O rate, minidisk caching can utilize all the available memory and the guests are dropped from dispatching. To avoid this scenario, restrict minidisk caching to a maximum value. A viable starting point is about 10% to 25% of the available memory in the LPAR. The following command, in the **autostart** file, sets this restriction:

```
CP SET MDCACHE SYSTEM ON
CP SET MDCACHE STORAGE OM <max value>M
```

With command, **CP QUERY MDCACHE**, you can control the setting and the usage.

Deactivate minidisk caching for Linux swap disks. To do so, code **MINIOPT NOMDC** operands on the **MDISK** directory statement of the appropriate disk.

3.4.7 z/VM share

Share is the denotation for the amount of cpu processing a virtual machine receives. There are two different variations of share settings (absolute and relative share).

Relative share is factored similarly as the LPAR weight factor. The sum of the relative share of all active virtual machines in conjunction to the share setting of an individual virtual machine. Relative share ranges from 1-9999.

Absolute share is expressed in percent and defines a real portion of the available cpu capacity of the LPAR dedicated to a specific virtual machine. This portion of the cpu capacity is reserved for that virtual machine, as long as it can be consumed. The remaining piece, which cannot be consumed, is returned to the system for further distribution. It ranges from 0.1-100%. If the sum of absolute shares is greater than 99%, it will be normalized to 99%. Absolute share users are given resource first.

The default share is `RELATIVE 100` to each virtual machine. The value can be changed dynamically by the command **`CP SET SHARE`** or permanently at the user entry in the z/VM directory.

SHARE RELATIVE and multi-CPU guests

It is important to remember that the `SHARE` value is distributed across all of the virtual CPUs of a guest. This means that no matter how many virtual CPUs a guest has, if the `SHARE` value is not changed the guest gets the same amount of CPU.

To make sure that adding virtual CPUs actually results in extra CPU capacity to your virtual machines, make sure the `SHARE` value is increased when virtual CPUs are added.

3.4.8 z/VM External Security Manager (ESM)

The security and isolation mechanisms built into the z/VM Control Program (CP) to protect virtual machines from each other are extremely strong. These mechanisms are facilitated by various hardware mechanisms provided by the IBM LinuxONE server architecture. There are some areas where improvements can be made, such as those in the following list:

- Auditing of resource access successes and failures
- Queryable passwords for users and minidisks
- Complexity of managing command authority and delegation

z/VM allows the built-in security structure to be enhanced through the use of an External Security Manager (ESM). When an ESM is enabled on z/VM, various security decisions can be handled by the ESM rather than by CP. This allows for greater granularity of configuration, better auditing capability and the elimination of queryable passwords for resources.

The IBM Resource Access Control Facility for z/VM (RACF) is one ESM available for z/VM. It is a priced optional feature preinstalled on a z/VM system. Broadcom also offers ESM products for z/VM, such as CA ACF2 and CA VM:Secure.

Note: IBM **strongly** recommends the use of an ESM on all z/VM systems.

Common Criteria and the Secure Configuration

IBM undergoes evaluation of the IBM LinuxONE server hardware and z/VM against the Common Criteria. z/VM is evaluated against the Operating System Protection Profile (OSPP). This evaluation allows clients to be more confident that the IBM LinuxONE server with z/VM as the hypervisor is a highly secure platform for running critical workloads. z/VM has achieved an Evaluation Assurance Level (EAL) of 4+ (the *plus* indicates additional targets from the Labelled Security Protection Profile (LSPP) were included in the evaluation).

The evaluation process is performed against a specific configuration of z/VM which includes RACF. The configuration that IBM applies to the systems evaluated for Common Criteria certification is described in the z/VM manual "*z/VM: Secure Configuration Guide,*" document number SC24-6323-00. This document is located at the following link:

http://www.vm.ibm.com/library/710pdfs/71632300.pdf

By following the steps in this manual you can configure your z/VM system in a way that meets the standard evaluated for Common Criteria certification.

3.4.9 Memory of a Linux virtual machine

In a physical x86 system, the memory installed in a machine is often sized larger than required by the application. Linux uses this extra memory for *buffer cache* (disk blocks held in memory to avoid future I/O operations). This is considered a positive outcome as the memory cannot be used by any other system; using it to avoid I/O is better than letting it remain unused.

Virtualized x86 systems often retain the same memory usage patterns. Because memory is considered to be inexpensive, virtual machines are often configured with more memory than actually needed. This leads to accumulation of Linux buffer cache in virtual machines; on a typical x86 virtualized environment a large amount of memory is used up in such caching.

In z/VM the virtual machine is sized as small as possible, generally providing enough memory for the application to function well without allowing the same buffer cache accumulation as occurs on other platforms. Assigning a Linux virtual machine too much memory can allow too much cache to accumulate, which requires Linux and z/VM to maintain this memory. z/VM sees the working set of the user as being much larger than it actually needs to be to support the application, which can put unnecessary stress to z/VM paging.

Real memory is a shared resource. Caching disk pages in a Linux guest reduces memory available to other Linux guests. The IBM LinuxONE I/O subsystem provides extremely high I/O performance across a large number of virtual machines, so individual virtual machines do not need to keep disk buffer cache in an attempt to avoid I/O.

Linux: to swap or not to swap?

In general it is better to make sure that Linux does not swap, even if it means that z/VM has to page. This is because the algorithms and memory management techniques used by z/VM provide better performance than Linux swap.

This creates a tension in the best configuration approach to take. Linux needs enough memory for programs to work efficiently without incurring swapping, yet not so much memory that needless buffer cache accumulates.

One technology that can help is the z/VM Virtual Disk (VDISK). VDISK is a disk-in-memory technology that can be used by Linux as a swap device. The Linux guest is given one or two VDISK-based swap devices, and a memory size sufficient to cover the expected memory consumption of the workload. The guest is then monitored for any swap I/O. If swapping occurs, the performance penalty is small because it is a virtual disk I/O to memory instead of a real disk I/O. Like virtual machine memory, z/VM does not allocate memory to a VDISK until the virtual machine writes to it. So memory usage of a VDISK swap device is only slightly more than if the guest had the memory allocated directly. If the guest swaps, the nature of the activity can be measured to see whether the guest memory allocation needs to be increased (or if it was just a short-term usage bubble).

Using VDISK swap in Linux has an additional benefit. The disk space that normally is allocated to Linux as `swp` space can be allocated to z/VM instead to give greater capacity and performance to the z/VM paging subsystem.

Hotplug memory

Another memory management technique is the ability to dynamically add and remove memory from a Linux guest under z/VM, known as *hotplug memory*. Hotplug memory added to a Linux guest allows it to handle a workload spike or other situation that could result in a memory shortage.

We recommend that you use this feature carefully and sparingly. Importantly, do not configure large amounts of hotplug memory on small virtual machines. This is because the Linux kernel needs 64 bytes of memory to manage every 4 kB page of hotplug memory, so a large amount of memory gets used up simply to manage the ability to plug more memory. For example, configuring a guest with 1 TB of hotplug memory consumes 16 GB of the guest's memory. If the guest only had 32 GB of memory, half of its memory is used just to manage the hotplug memory.

When configuring hotplug memory, be aware of this management requirement. You might need to increase the base memory allocation of your Linux guests to make sure that applications can still operate effectively.

Monitoring memory on Linux

There are a number of places to monitor memory usage on Linux:

- Monitor memory usage using the **free** or **vmstat** commands, along with **/proc/meminfo**. This can provide summary through to detailed information about memory usage. **/proc/slabinfo** can provide further detail about kernel memory.
- Generally Linux does not suffer memory fragmentation issues, but longer server uptimes might lead to memory becoming fragmented over time. **/proc/buddyinfo** contains information about normal and kernel memory pools. Large numbers of pages in the small pools (order-3 and below) indicate memory fragmentation and possible performance issues for some programs (particularly kernel operations such as allocation of device driver buffers).

3.4.10 Simultaneous Multi-threading (SMT-2)

By using SMT, z/VM can optimize core resources for increased capacity and throughput. Its maximization of SMT enables z/VM to dispatch a guest (virtual) CPU or z/VM Control Program task on an individual thread (CPU) of an Integrated Facility for Linux (IFL) processor core.

SMT can be activated only from the operating system and requires a restart of z/VM. We recommend activating multithreading in z/VM by defining **MULTITHREADING ENABLE** in the system configuration file. The remaining defaults of this parameter set the maximum number of possible threads (currently two) for all processor types. This parameter also enables the command **CP SET MULTITHREAD** to switch multithreading back and forth dynamically without a restart.

3.4.11 z/VM CPU allocation

This section describes CPU allocation in z/VM.

Linux Guests

It is important that Linux is given enough opportunity to access the amount of CPU capacity needed for the work being done. However, allocating too much virtual CPU capacity can, in some cases, reduce performance.

IFL/CPU and memory resources

This section introduces how to manage allocating IFLs to your LPARs, and to your Linux guests within LPARs.

Symmetric Multi-Threading

Modern CPUs have sophisticated designs such as pipelining, instruction prefetch, out-of-order execution and more. These technologies are designed to keep the execution units of the CPU as busy as possible. Yet another way to keep the CPU busy is to provide more than one *queue* for instructions to enter the CPU. Symmetric Multi-Threading (SMT) provides this capability on IBM LinuxONE.

z/VM does not virtualize SMT for guests. Guest virtual processors in z/VM are single-thread processors. z/VM uses the threads provided by SMT-enabled CPUs to run more virtual CPUs against them.

On the IBM LinuxONE IFL, up to two instruction queues can be used (referred to as SMT-2). These multiple instruction queues are referred to as *threads*.

Two steps are required to enable SMT for a z/VM system. First, the LPAR needs to be defined to permit `SMT` mode. Second, z/VM must be configured to enable it. This is done using the **`MULTITHREAD`** keyword in the `SYSTEM CONFIG` file.

When z/VM is not enabled for SMT, logical processors are still referred to as processors. When SMT is enabled, z/VM creates a distinction between *cores* and *threads*, and treats threads in the same way as processors in non-SMT.

Logical, Physical, or Virtual CPUs/IFLs

It is important to make sure that CPU resources are assigned efficiently. As z/VM on IBM LinuxONE implements two levels of virtualization. It is vital to configure Linux and z/VM to work properly with the CPU resources of the system.

The following section introduces details about CPU configuration in IBM LinuxONE.

LPAR weight

IBM LinuxONE is capable of effectively controlling the CPU allocated to LPARs. In their respective Activation Profile, all LPARs are assigned a value called a *weight*. The weight is used by the LPAR management firmware to decide the relative importance of different LPARs.

> **Note:** When HiperDispatch is enabled, the weight is also used to determine the polarization of the logical IFLs assigned to an LPAR. More about HiperDispatch and its importance for Linux workloads is in the following section.

LPAR weight is usually used to favor CPU capacity toward your important workloads. For example, on a production IBM LinuxONE system, it is common to assign higher weight to production LPARs and lower weight to workloads that might be considered discretionary for that system (such as testing or development).

z/VM HiperDispatch

z/VM HiperDispatch feature uses the System Resource Manager (SRM) to control the dispatching of virtual CPUs on physical CPUs (scheduling virtual CPUs). The prime objective of z/VM HiperDispatch is to help virtual servers achieve enhanced performance from the IBM LinuxONE memory subsystem.

z/VM HiperDispatch works toward this objective by managing the partition and dispatching virtual CPUs in a way that takes into account the physical machine's organization (especially its memory caches). Therefore, depending upon the type of workload, this z/VM dispatching method can help to achieve enhanced performance on IBM LinuxONE hardware.

The processors of an IBM LinuxONE are physically placed in hardware in a hierarchical, layered fashion:

- CPU cores are fabricated together on chips, perhaps 10 or 12 cores to a chip, depending upon the model
- Chips are assembled onto nodes, perhaps three to six chips per node, again, depending upon model
- The nodes are then fitted into the machine's frame

To help improve data access times, IBM LinuxONE uses high-speed memory caches at important points in the CPU placement hierarchy:

- Each core has its own L1 and L2
- Each chip has its own L3
- Each node has its own L4
- Beyond L4 lies memory

One-way z/VM HiperDispatch tries to achieve its objective is by requesting that the PR/SM hypervisor provisions the LPAR in *vertical mode*. A vertical mode partition has the property that informs the PR/SM hypervisor to repeatedly attempt to run the partition's logical CPUs on the same physical cores (and to run other partitions' logical CPUs elsewhere). For this reason, the partition's workload benefits from having its memory references build up context in the caches. Therefore, the overall system behavior is more efficient.

z/VM works to assist HiperDispatch to achieve its objectives by repeatedly running the guests' virtual CPUs on the same logical CPUs. This strategy ensures guests experience the benefit of having their memory references build up context in the caches. This also enables the individual workloads to run more efficiently.

3.4.12 z/VM configuration files

This section describes the following major configuration files:

- z/VM system configuration file **SYSTEM CONFIG**
- z/VM directory file **USER DIRECT**
- **Autostart** file

z/VM system configuration file SYSTEM CONFIG

This file is placed on the configuration disk CF0 of user **PMAINT** and carries all global z/VM system parameters. This file will be read only once at IPL time.

z/VM directory file USER DIRECT

After the z/VM installation, **USER DIRECT** is present on disk 2CC in user MAINT. This file has resource definitions about all virtual machines. After changing this file, it has to be compiled to update the system directory areas. This will be done by the command **DIRECTXA**. Take security precautions with this file because it contains clear text passwords!

If you run a directory management software such as IBM Directory Maintenance Facility for z/VM (DirMaint), this file is no longer used. See “User and security management” on page 61 for more information about using a directory manager.

Autostart file

You need to invoke commands every time the z/VM system starts. User AUTOLOG1 is automatically started after IPL. Inside AUTOLOG1 the file **PROFILE EXEC** on minidisk 191 is automatically executed. Every command in that file will be automatically invoked after the system starts.

One of the key things this process is used for is starting your Linux guests automatically after z/VM is started.

Note: IBM Wave for z/VM also uses the `AUTOLOG1` user for configuration of entities (such as z/VM VSwitches) managed by IBM Wave.

3.4.13 Product configuration files

System management products installed on z/VM will have their own configuration files. A few examples include:

- Files on various `DIRMAINT` disks, such as 155, 11F, and 1DF, for IBM DirMaint:
  - `CONFIGxx DATADVH`
  - `EXTENT CONTROL`
  - `AUTHFOR CONTROL`
  - Any customized `PROTODIR` files
- Files on `VMSYS:PERFSVM` (or minidisks if not installed to file pool) for IBM Performance Toolkit for z/VM:
  - `$PROFILE FCONX`
  - `FCONRMT SYSTEMS`
  - `FCONRMT AUTHORIZ`
  - `UCOMDIR NAMES`
- Various files on `OPMGRM1 198` for IBM Operations Manager for z/VM
- Backup and disk pool definition files for IBM Backup and Restore Manager for z/VM

Consult the product documentation for each of the products being customized for the role and correct contents for these files.

3.4.14 IBM Infrastructure Suite for z/VM and Linux

Section 2.4.4, “IBM Infrastructure Suite for z/VM and Linux” on page 36 previously described the key elements for monitoring and managing the IBM LinuxONE environments. Additional tools are proposed for advanced sites where dashboards and automation triggers present operations personnel with additional information about the status of the services and proposed actions.

First, for the base Infrastructure Suite, you need to install and configure **`DirMaint`** and Performance Toolkit. Then for the advanced tools, a suggested setup is to create five LPARs to host the following parts:

- IBM Wave UI Server (Wave)
- Tivoli Storage Manager Server (TSM)
- Tivoli Data Warehouse (TDW) with Warehouse Proxy and Summarization and Pruning Agents
- IBM Tivoli Monitoring (ITM) Servers: Tivoli Enterprise Portal Server (TEPS) and Tivoli Enterprise Management Server (TEMS)
- JazzSM server for Dashboard Application Services Hub (DASH) and Tivoli Common Reporting (TCR)

These five LPARs need to be set up only once for your enterprise. You can use any existing servers that meet the capacity requirements.

Before installing IBM Wave, check for the latest `fixpack` for IBM Wave and install it. The initial setup for IBM Wave is simple. All required setup in Linux and z/VM is done automatically by the installation scripts. IBM Wave has a granular role-based user model. Plan the roles in IBM Wave carefully according to your business needs.

3.5 Pervasive Encryption for data-at-rest

Protecting data-at-rest is an important aspect of security on IBM LinuxONE. Linux and z/VM support different aspects of Pervasive Encryption with two separate but important security capabilities. This section covers those capabilities in greater detail.

3.5.1 Data-at-rest protection on Linux: encrypted block devices

One of the key security capabilities of IBM LinuxONE is a highly secure way to encrypt disk devices. Protected key encryption uses both of the IBM LinuxONE hardware security features:

- The Crypto Express card, as a secure, tamper-evident master key storage repository
- The Central Processor Assist for Cryptographic Functions (CPACF), accelerated cryptographic instructions available to every CPU in the system

Protected key encryption uses an encryption key that is derived from a master key and kept within the Crypto Express card to generate a wrapped key that is stored in the Hardware System Area (HSA) of the IBM LinuxONE system. The key is used by the CPACF instructions to perform high-speed encryption and decryption of data, but it is not visible to the operating system in any way.

How IBM LinuxONE data-at-rest encryption works

When the **`paes`** cipher is used with IBM LinuxONE data at-rest encryption, the following protected volume options are available:

- The **`LUKS2`** format includes a header on the volume and a one-time formatting is required
- The **`LUKS2`** header is made up of multiple key slots. Each key slot contains key and cipher information
- The volume's secure key is wrapped by a key-encrypting key (which is derived from a passphrase or a **`keyfile7`**) and stored in a keyslot. The user must supply the correct passphrase to unlock the keyslot. A `keyfile` allows for the automatic unlocking of the keyslot

> **Note:** **`LUKS2`** format is the preferred option for IBM LinuxONE data at-rest encryption.

The plain format does not include a header on the volume and no formatting of the volume is required. However, the key must be stored in a file in the file system. The key and cipher information must be supplied with every volume open.

Creating a secure key

The process that is used to create a secure key for an **`LUKS2`** format volume is shown in Figure 3-1.

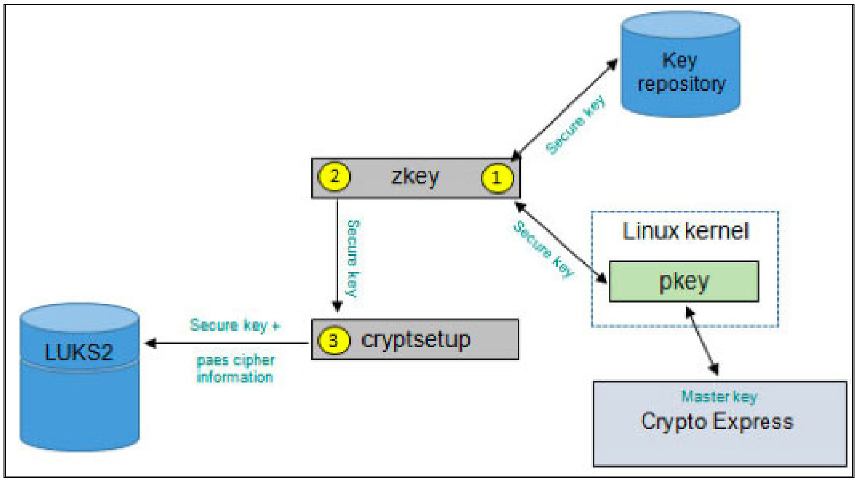


Figure 3-1 Create a secure key.

This process includes the following steps:

1. A secure key is created by using a **zkey** command. The zkey utility generates the secure key with the help of the pkey utility and an assigned Crypto Express adapter (with master key). The secure key is also stored in the key repository
2. The use of the **zkey** cryptsetup command generates output strings that are copied and pasted to the **cryptsetup** command to create the encrypted volume with the appropriate secure key
3. The **cryptsetup** utility formats the physical volume and writes the encrypted secure key and cipher information to the **LUKS2** header of the volume

Opening a LUKS2 formatted volume

The process that is used to open an LUKS2 formatted volume is shown in Figure 3-2 on page 73.

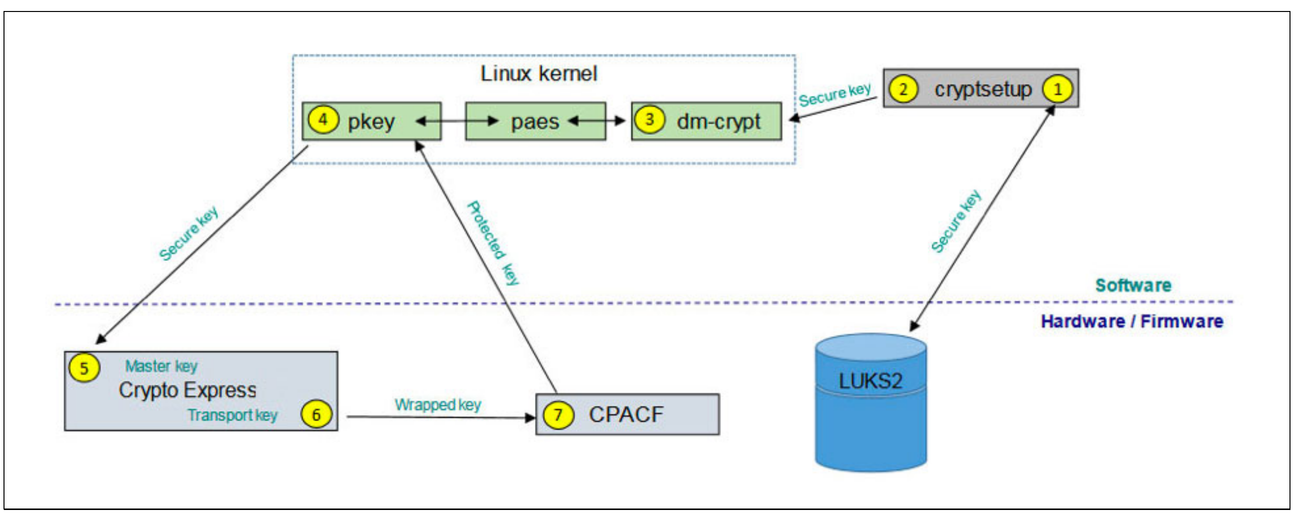


Figure 3-2 Opening a LUKS2 formatted volume.

This process includes the following steps:

1. The **cryptsetup** utility fetches the secure key from the LUKS2 header
2. The **cryptsetup** utility passes the secure key to `dm-crypt`

3. The `dm-crypt` passes the secure key to **paes** for conversion into a protected key by using **pkey**
4. The **pkey** module starts the process for converting the secure key to a protected key
5. The secure key is unwrapped by the CCA coprocessor in the Crypto Express adapter by using the master key
6. The unwrapped secure key (effective key) is rewrapped by using a transport key that is specific to the assigned domain ID
7. By using firmware, CPACF creates a protected key

3.5.2 Data-at-rest protection on z/VM: encrypted paging

For the most part, Linux running as a virtual machine is responsible for its own resources. The hypervisor does things to protect virtual machines from each other (such as protecting the memory allocated to the virtual machine from being accessed by another virtual machine). However, Linux manages the resources allocated to it. Currently, paging is the only operation that the hypervisor does that might result in exposure of a guest's resource (in this case, part of its memory).

Paging occurs when the z/VM system does not have enough physical memory available to satisfy a guest's request for memory. To obtain memory to meet the request, z/VM finds some currently allocated but not recently used memory and stores the contents onto persistent storage (a disk device). z/VM then reuses the memory to satisfy the guest's request.

When a paging operation occurs, the content of the memory pages is written to disk (to *paging volumes*). It is during this process that the possible exposure occurs. If the memory being paged-out happened to contain a password, the private key of a digital certificate, or other secret data, z/VM has stored that sensitive data onto a paging volume outside the control of Linux. Whatever protections were available to that memory while it was resident are no longer in effect.

To protect against this situation occurring, z/VM Encrypted Paging uses the advanced encryption capability of the IBM LinuxONE system to encrypt memory being paged out and decrypt it after the page-in operation. Encrypted Paging uses a temporary key (also known as an *ephemeral key*) which is generated each time a z/VM system is IPLed. If Encrypted Paging is enabled, pages are encrypted using the ephemeral key before they are written to the paging device.

3.6 Networking on IBM LinuxONE

IBM LinuxONE servers support a variety of hardware, software, and firmware networking technologies to support workloads.

Adding dedicated OSA ports to Linux guests can be ideal for use as interconnect interfaces for databases or clustered file systems. Using a dedicated OSA can reduce the path length to the interface, but you will need to decide your own method for providing failover.

Also, if you dedicate an OSA interface it can be used for only one IP network by default. You can use the Linux **8021q** module to provide VLANs, managed within Linux.

3.6.1 Ethernet technologies

Standard network connectivity is supported on IBM LinuxONE using two types of network technology:

- OSA Express features
- HiperSockets

OSA Express features

For optimal data transfer, use OSA with a speed of at least 10 Gb, and for redundancy plan them in a pair of two.

OSA Express cards can be used in conjunction with software networking facilities in z/VM and Linux (such as z/VM Virtual Switch and Open vSwitch in Linux). When used in conjunction, together they support connectivity for virtual machines in virtualized environments under z/VM.

HiperSockets

HiperSockets (HS) interconnects LPARs that are active on the same machine by doing a memory-to-memory transfer at processor speed. HS can be used for both TCP and UDP connections.

VSwitch

A z/VM Virtual Switch (VSwitch) will provide the network communication path for the Linux guests in the environment. Refer to sections 2.6.3, “z/VM networking” on page 49 and 3.6.3, “Connecting virtual machines to the network” for more information about VSwitch.

We recommend that a Port Group is used, for maximum load sharing and redundancy capability. The Link Aggregation Control Protocol (LACP) can enhance the usage of all the ports installed in the Port Group.

z/VM Virtual Switch also provides a capability called Multi-VSwitch Link Aggregation, also known as *Global Switch*. This allows the ports in a Port Group to be shared between several LPARs.

3.6.2 Shared Memory Communications (SMC)

SMC can be used only for TCP connections. It also requires connectivity through an OSA for the initial handshake. This OSA connectivity acts also as a fallback if SMC has problems initiating. SMC offers the same speed as HiperSockets but is more flexible because it also interconnects additional IBM LinuxONE servers.

SMC-R (RoCE) and SMC-D (ISM)

Using SMC, is recommended in any environment where there is extensive TCP traffic between systems in the IBM LinuxONE environment. This is the case whether using SMC over RoCE hardware or an ISM internal channel.

We anticipate significant throughput and latency improvement when enabling the communication between the database and the Transact application servers for SMC.

The RoCE Express card can be used by Linux for both SMC communication and for standard Ethernet traffic. At this time however, the RoCE Express card does not have the same level of availability as the OSA Express card (for example, firmware updates are disruptive). For this reason and at the time of this publication, we recommend the following condition. If RoCE

Express is being used for Linux, it should be used in addition to a standard OSA Express-based communications path (either direct-OSA or VSwitch). For more information about SMC, see 2.1.8, “Shared Memory Communication (SMC)” on page 24.

3.6.3 Connecting virtual machines to the network

Virtual machines (VMs) attach to the network using the physical and virtual networking technologies described previously. The following sections describe some of the ways that attaching VMs can be completed.

Dedicating devices to Linux under z/VM

z/VM can pass channel subsystem devices through to the guest virtual machine, which enables the guest to manage networking devices directly with its own kernel drivers.

> **Note:** This is the only way that HiperSockets can be used by a Linux guest under z/VM. For OSA Express, the z/VM Virtual Switch is an alternative. See “z/VM Virtual Switch” on page 76.

To allow a guest to access an OSA Express card or HiperSockets network directly, the z/VM ATTACH command is used to connect devices accessible by z/VM directly to the Linux virtual machine. If the Linux guest needs access to the network device at startup, use the **`DEDICATE`** directory control statement. This statement attaches the required devices to the Linux guest when it is logged on.

The adapter can still be shared with other LPARs on the IBM LinuxONE server. It can also be sharable with other Linux guests in the same LPAR. However, adapter sharing has a dependency. There must be enough subchannel devices defined in the channel subsystem to allow more than one Linux guest in the LPAR to use the adapter at the same time.

> **Note:** The way adapter sharing is done is different between the IODF mode and the DPM mode of the IBM LinuxONE server.

When you attach a Linux guest to an OSA Express adapter in this way, you need to consider how you will handle possible adapter or switch failures. Usually you attach at least two OSA Express adapters to the guest and use Linux channel bonding to provide interface redundancy. You can use either the Linux **`bonding`** driver or the newer Team **`softdev`** Linux driver for channel bonding. You have to repeat this configuration on every Linux guest. Managing this configuration across a large number of guests is challenging and is one reason this is *not* the preferred connection method for Linux guests.

z/VM Virtual Switch

A z/VM Virtual Switch can be used to attach Linux guests under z/VM to an Ethernet network. The guests are configured with one (or more) virtual OSA Express cards, which are then connected to a VSwitch. The VSwitch is in turn connected to one or more real OSA Express adapters. A z/VM Virtual Switch simplifies the configuration of a virtualized environment by handling much of the networking complexity on behalf of Linux guests.

A VSwitch can support IEEE 802.1Q Virtual LANs (VLANs). It can either manage VLAN tagging on behalf of a virtual machine or can let the virtual machine do its own VLAN support.

VSwitches also provide fault tolerance on behalf of virtual machines. This is provided either using a *warm standby* mode, or link aggregation mode using a Port Group. In the warm standby mode, up to three OSA Express ports are attached to a VSwitch with one carrying network traffic and the other two ready to take over in case of a failure. In the Port Group mode, up to eight OSA Express ports can be joined for link aggregation. This mode can use the IEEE 802.1AX (formerly 802.3ad) Link Aggregation Control Protocol (LACP). The two modes can actually be combined: a Port Group can be used as the main uplink for the VSwitch, with a further OSA port in the standard mode used as a further backup link.

A z/VM VSwitch can also provide isolation capability, using the Virtual Edge Port Aggregator (VEPA) mode. In this mode, the VSwitch no longer performs any switching between guests that are attached to it. Instead, all packets generated by a guest are transmitted out to the adjacent network switch by the OSA uplink. The adjacent switch must support Reflective Relay (also known as *hairpinning*) for guests attached to the VSwitch to communicate.

3.6.4 Connecting virtual machines to each other

There are times where virtual machines need specific communication paths to each other. The most common instance of this is clustered services requiring an interconnect or *heartbeat* connection (such as Oracle RAC, or IBM Spectrum Scale). It is possible to use the standard network interface used for providing service from the Linux guest. However, most cluster services stipulate that the interconnect should be a separate network dedicated to the purpose.

Any of the network technologies described in Section 3.6.3, “Connecting virtual machines to the network”, can be used for a cluster interconnect. Our architecture recommends the use of OSA Express adapters for cluster interconnect for the following reasons:

- Provides cluster connectivity between CPCs without changes
- Provides support for all protocols supported over Ethernet

Cross-CPC connectivity

HiperSockets is a natural first choice for use as a cluster interconnect: it is fast, and highly secure. It can be configured with a large MTU size, making it ideal as a database or file storage interconnect.

However, because HiperSockets exists only within a single CPC, it cannot be used when the systems being clustered span CPCs. If a HiperSockets-based cluster interconnect is implemented for nodes on a single CPC, the cluster is changed to a different interconnect technology if the nodes were to be split across CPCs.

When an OSA Express-based interconnect can be configured with a large MTU size (referred to as Ethernet *jumbo frames*), OSA Express is a good choice. This is because of the flexibility of OSA Express in being able to deploy cluster nodes across CPCs.

Protocol flexibility

The SMC networking technologies, SMC-D and SMC-R, can also be considered as cluster interconnect technologies. They offer high throughput with low CPU utilization. Unlike HiperSockets, the technology can be used between CPCs (SMC-R).

SMC can increase only the performance of TCP connections; therefore, it might not be usable for all cluster applications (Oracle RAC, for example, uses both TCP and UDP on the interconnect network). SMC operates as an adjunct to the standard network interface and not as a separate physical network. Because of this, it doesn’t meet the usual cluster interconnect requirement of being a logically and physically separate communication path.

3.7 DS8K Enterprise disk subsystem

Determining what type of storage your organization needs can depend on many factors at your site. The IBM LinuxONE can use FCP/SCSI, FICON ECKD or a mix of storage types. The storage decision should be made early on and with the future in consideration. Changing decisions later in the process can create longer migrations to another storage type. The architecture described in this book is based around FICON and ECKD storage, which is required for SSI and the high availability features that it brings.

There are two options available for which disk storage type to choose:

- A 512-byte fixed block open system storage based on the FCP protocol, which is the same storage as for the x86 platform. On this storage you need to define LUNs with the appropriate sizes, which can be found in the product documentations.
- ECKD storage, which requires an enterprise class storage subsystem (IBM DS8000) based on FICON protocol. ECKD volumes need to be defined in the storage subsystem. If the product documentation defined the disk size in GB or TB then you need to transform the sizes in number of cylinders or 3390 models.

The following section helps you to do the calculations for ECKD volume size.

3.7.1 ECKD volume size

ECKD is also known as IBM 3390 volume. The size of an ECKD volume is categorized into models and is counted in cylinders. One cylinder is 849,960 bytes. The base model is a 3390 model 1 (3390 M1 or 3390-1) and it has a size of about 946 MB. The 3390-1, 3309-2 and 3390-3 are no longer used (or only in rare cases). Table 3-1 shows the commonly used sizes for an ECKD volume.

Table 3-1 Commonly used 3390 sizes

Disk Type	Cylinder	Volume size
3390-9	10017	8.1 GB
3390-27	32760	27.8 GB
3390-54	65520	55.6 GB
ECKD EAV	up to 262668	up to 223 GB

The 3390-9 can be used for the operating system (especially for z/VM) and the other types for data. In an enterprise class storage, you find these volume models as predefined selections in the configuration dialog. However, you are not restricted to these specific sizes as you can define any number of cylinders within the limit of max 262668. The ECKD EAV is the extended addressability volume. This means there is no further specific type defined beyond 3390-54.

3.7.2 Disk mirroring

Independent of which storage is chosen, we recommend installing at least two identical storage subsystems and to set up the built mirroring technology and mirror all defined volumes. For this, IBM DS8000 offers Metro Mirror (the former name was Peer-to-Peer Remote Copy) function. Metro Mirror is a synchronous disk replication method and guaranties, at any time, an identical copy of your data. This allows you, at an outage of a disk subsystem, to restart immediately from the other disk subsystem or to immediately switch over to the other disk subsystem. This immediate response is dependent on the high availability functions that are implemented additionally (such as GDPS).

3.7.3 Which storage to use

When deciding between FBA storage and ECKD storage, there are several considerations to review in understanding what is best for your business needs.

FBA storage is more common because it is also usable as storage for the x86 platform. FBA storage has the following attributes:

- Any kind of SCSI disk storage can be used
- It fits into your already implemented monitoring environment
- It does not need any special hardware (SAN switches) or dedicated cabling
- It is less expensive as an enterprise class storage
- If you run IBM LinuxONE in DPM mode, FBA storage is the preferred storage
- Some functions are not available when compared to an enterprise class storage
- Multipathing must be done at the operating system level
- It has limits in scalability
- It does not support GDPS

ECKD or enterprise class storage is unique to the IBM LinuxONE architecture. ECKD storage has the following attributes:

- It supports all the functions available for disk storage systems
- It offers the most performance and scalability
- No additional driver is necessary for multi-pathing; it is implemented in the FICON protocol
- It is supported by GDPS
- It is more expensive when compared to FBS storage
- It requires enterprise class SAN switches
- FICON needs dedicated cabling

If you are considering running GDPS, you are required to use FICON and enterprise class storage. Otherwise, FBA storage is also a good option.

3.8 Temenos Transact

Temenos Infinity and Transact is a multi-module application suite supporting core banking, payments, Islamic fund and various other Retail and Commercial Banking services. Over recent years, this application framework has evolved to support more agile and flexible technologies such as Java and RESTful API services. The latest R19 and R20 releases referred to throughout this book are based on TAFJ (Temenos Application Framework for Java) and can be deployed within a range of Java application servers (such as IBM WebSphere, JBOSS, or Oracle WebLogic). This reduces the proprietary runtime components typically associated with the TAFC application versions. It also allows the application server to control the Enterprise Server processing, messaging, operations, and management features independently of the application instance(s).

This new application suite approach allows clients to integrate new modules and modify or update existing services without impacting the runtime services. It also reduces the development and testing effort required and appeals to the larger community of Java developers. This has also become the de-facto standard for Cloud-based adoption using containers (such as Docker and Podman) and orchestration technologies (for example, Kubernetes) based on Java frameworks.

In summary, the use of the latest Temenos TAFJ-based suite brings many functional advantages. This is based on the ability to exploit the latest Java, Cloud and associated runtime technologies, while allowing the non-functional architectural requirements such as Availability, Scalability, (Transaction) Reliability and Security to be fully exploited on the IBM LinuxONE platform.

The software used when running Temenos Transact is specific and exact. In fact, there is only one specific version of the Linux operating system that is compatible for use. If an organization defers from the recommended list of software, Temenos can deny support.

The following components and minimum release levels are certified to run Temenos Transact:

- Red Hat Enterprise Linux 7
- Java 1.8
- IBM WebSphere MQ 9
- Application Server (noted in the next list)
- Oracle DB 12c

The Temenos Transact software is JAVA based and requires an application server to run. There are several options of application servers:

- IBM WebSphere 9
- Red Hat JBoss EAP
- Oracle WebLogic Server 12c (JDBC driver)

The Temenos Stack Runbooks provide more information about using Temenos stacks with different application servers. Temenos customers and partners can access the Runbooks through either of the following links:

- The Temenos Customer Support Portal: https://tcsp.temenos.com/
- The Temenos Partner Portal: https://tpsp.temenos.com/

3.9 Red Hat Linux

Temenos supports only Red Hat Enterprise Linux (RHEL). IBM LinuxONE LPARs and guests, under z/VM, should be provisioned with RHEL Linux release for s390x. Depending on the LPAR and workload, IBM LinuxONE resources should be tuned specifically for each LPAR for best overall performance.

3.10 IBM WebSphere

In the traditional architecture described in this section, IBM WebSphere Application Server is deployed across four LPARS under z/VM on two CECs. IBM WebSphere Application Server is configured as a stand-alone server and all Temenos Transact components are installed on one instance of the IBM WebSphere Application Server per node. Deploying IBM WebSphere Network Deployment enables a central administration point of a cell that consists of multiple nodes and node groups in a distributed server configuration. The Database connection uses the JDBC driver. This driver is the only driver supported by Temenos Transact and Oracle on IBM LinuxONE.

3.11 Queuing with IBM MQ

No specific tuning or installation considerations are needed for IBM MQ on top of the Transact installation process. Queues should be defined as input shared and (where applicable) defined as persistent. Although failures of IBM MQ are rare, single points of failure should be avoided in any architecture and multiple MQ servers should be deployed.

MQ servers should be configured in an active/passive way, on two Linux systems (possibly guests under z/VM). IBM MQ requires shared storage (such as Spectrum Scale), so they can share MQ vital information (such as logs) and allow the active/passive behavior.

For the installation and configuration process of IBM MQ see *"Installing IBM MQ server on Linux,"* located at the following link:

https://www.ibm.com/support/knowledgecenter/en/SSFKSJ_8.0.0/com.ibm.mq.ins.doc/q008640_.htm

3.12 Oracle DB on IBM LinuxONE

Oracle Database (DB) is the preferred database for Temenos Transact on IBM LinuxONE with the tradition deployment architecture. Oracle Real Application Clusters (RAC) is used to ensure High Availability (HA) of the database in the event of an outage on one of the database nodes.

3.12.1 Native Linux or z/VM guest deployment

Oracle DB is supported on IBM LinuxONE either as a native installation on an LPAR or as a guest (virtual machine) under z/VM.

3.12.2 Oracle Grid Infrastructure

Oracle Grid Infrastructure is a prerequisite for Oracle Real application Cluster (RAC) and is a suite of software containing Oracle Clusterware and Oracle Automatic Storage Management (ASM).

3.12.3 Oracle Clusterware

Oracle Clusterware is part of the Oracle Grid Infrastructure suite and is required for Oracle Real Application Clusters (RAC). Oracle Clusterware is what allows the independent Linux Guests on the Production HA pair, shown in Figure 4-6 on page 91, to operate as a single database instance to the application and balance the database workloads.

Each DB Node is a stand-alone Linux server. However, Oracle Clusterware allows all Oracle RAC nodes to communicate with each other. Installation of Oracle DB or updates to it could happen across all DB nodes automatically.

Oracle Clusterware has additional shared storage requirements: a voting disk to record node membership and the Oracle Cluster Registry (OCR) for cluster configuration information.

3.12.4 Oracle Automatic Storage Management (ASM)

ASM is a volume manager and file system that groups storage devices into storage groups. ASM simplifies the management of storage devices by balancing the workload across disks in the disk group and exposes the file system interface for the Oracle database files. ASM is used alternatively to conventional volume managers, file systems and raw devices.

Some advantages of ASM are noted in the following list:

- Live add and remove of disk devices
- Ability to use external disk mirroring technology such as IBM Metro Mirror
- Automatically balancing database files across disk devices to eliminate hotspots

The use of ASM is optional and Oracle now supports Spectrum Storage (GPFS) on IBM LinuxONE as an alternative with Oracle RAC.

3.12.5 Oracle Real Application Clusters (RAC)

Oracle RAC is an optional feature from Oracle to provide a highly available scalable database for Temenos Transact on IBM LinuxONE. Oracle RAC is a clustered database that overcomes the limitations of traditional share nothing or share disk approach with a shared cache architecture that does not impact performance.

High availability of Oracle RAC is achieved by removing single points of failure of single node or single-server architectures with multi-node deployments while maintaining the operational efficiency of a single node database. Node failures do not affect the availability of the database because Oracle Clusterware migrates and balances DB traffic to the remaining nodes whether the outage was planned or unplanned. IBM LinuxONE can achieve high availability Oracle RAC clusters on a single IBM LinuxONE. This is done with the use of multiple LPARs to a server as individual DB nodes or with multiple IBM LinuxONE systems in a data center. See the architectural diagram shown in Figure 4-6 on page 91.

With IBM LinuxONE, scalability can be achieved in multiple ways. One way is to add compute capacity to existing Oracle DB LPARs by adding IFLs. A second way is to add additional Oracle DB LPARs to the environment. The ability for scaling the Oracle DB by adding IFLs one at a time is another unique feature of IBM LinuxONE. This ability can have a distinct CPU core savings instead of deploying an entire Linux Server that can have dozens of cores to your Oracle architecture.

Oracle RAC, in an active/active configuration, offers the lowest Recovery Time Objective (RTO). However, this mode is the most resource intensive. Better DB performance has been observed using Oracle RAC One Node. In the event of a failure, Oracle RAC One Node will relocate database services to the standby node automatically. Oracle RAC One Node is a great fit with the scale-up capability of IBM LinuxONE.

See the Oracle documentation for system prerequisites and detailed information for installation and operation at the following link:

https://docs.oracle.com/cd/E11882_01/install.112/e41962/toc.htm

3.12.6 GoldenGate for database replication

The described architecture outlines that storage replication is used for the production LPARs within a metro distance. Because of this configuration, Oracle's GoldenGate real-time database replication software is not required.

3.12.7 Use encrypted volumes for the database

Oracle offers Transparent Data Encryption (TDE) for Oracle DB. TDE can be configured to selectively and transparently encrypt and decrypt sensitive data. However, IBM LinuxONE has a built-in feature that is available to transparently encrypt and decrypt ALL data on the volume. This means your entire DB can be transparently encrypted and decrypted with little impact on DB performance or IFL consumption. The data report shown in Figure 1-6 on page 13 shows nearly 2% impact on transaction rates with Temenos Transact on fully encrypted volumes as compared to non-encrypted volumes. That is a profound advantage over other platforms when you can encrypt everything.

3.12.8 Oracle tuning on IBM LinuxONE

It is recommended to use the following guidance to get the most benefit of Oracle DB on IBM LinuxONE:

- Enabling large pages

  It is recommended for performance and availability reasons to implement Linux large pages for Oracle databases that are running on IBM LinuxONE systems. Linux large pages are beneficial for systems where the database's Oracle SGA is greater than 8 GB.

- Defining large frames

  Enabling large frames allows the operating system to work with memory frames of 1 MB (on IBM LinuxONE) rather than the default 4 K. This allows smaller page tables and more efficient Dynamic Address Translation. Enabling fixed large frames can save CPU cycles when looking for data in memory. In our testing, transparent huge pages were disabled to ensure that the 1 MB pool was assigned when specified. In our lab environment testing, two components of the Transact architecture benefited from the large frames: Java and Oracle.

- Disabling transparent HugePages with kernel parameter

  It is recommended for performance and stability reasons to disable transparent HugePages. Transparent HugePages are different than Linux large pages, which are still highly recommended to use. Use the following command to disable transparent HugePages:

  ```
  transparent_hugepage=never
  ```

- Increasing the Memory pool size

  Define the memory pool size for huge pages of 1 MB by adding kernel parameters, to do so use the following command:

  ```
  default_hugepagesz=1M hugepagesz=1M hugepages=<number of pages>
  ```

- Increase fcp queue depth

  To maximize the I/O capabilities within the Linux hosting Oracle database, set the **zfcp.queue_depth** kernel parameter to 256 to increase the default fcp queue size.

  You can check whether your system has transparent HugePages enabled by using the following command:

  ```
  cat /sys/kernel/mm/transparent_hugepage/enabled
  [always] madvise never
  ```

4

Temenos Deployment on IBM LinuxONE and IBM Public Cloud

This chapter provides an overview of a sample installation, deployment, tuning, and migration journey for Temenos on IBM LinuxONE.

The following topics are covered in this chapter:

The Temenos Runbook architecture described in this chapter is based on Stack 2, IBM Java 1.8, IBM MQ, WebSphere, and Oracle DB 18c. See Figure 4-1 on page 86.

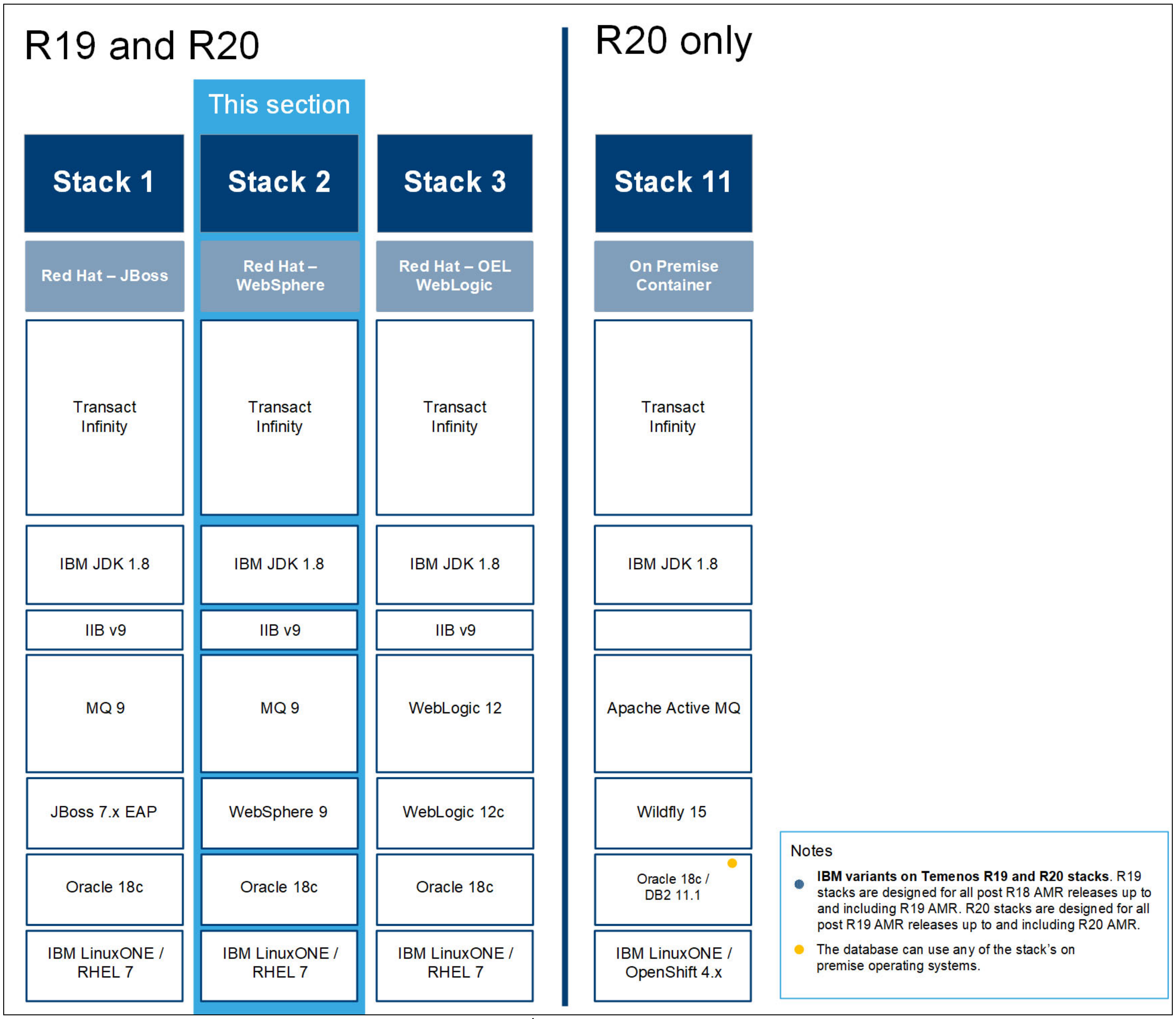


Figure 4-1 Stack 2 architecture used in this section. [1]

The standard solution that is presented in Figure 4-2 allows you to build a strong foundation for the future. This solution gives the customer the ability to provide maintenance to the base infrastructure with minimal impact to production. Building on the standard solution provides a pathway for the customer to continuous availability with other IBM products like GDPS and other storage mirroring solutions. Figure 4-2 shows the overall deployment architecture for a standard Temenos Solution on IBM LinuxONE Systems. The orange box represents the IBM LinuxONE III CPC.

[1] Courtesy of Temenos Headquarters SA

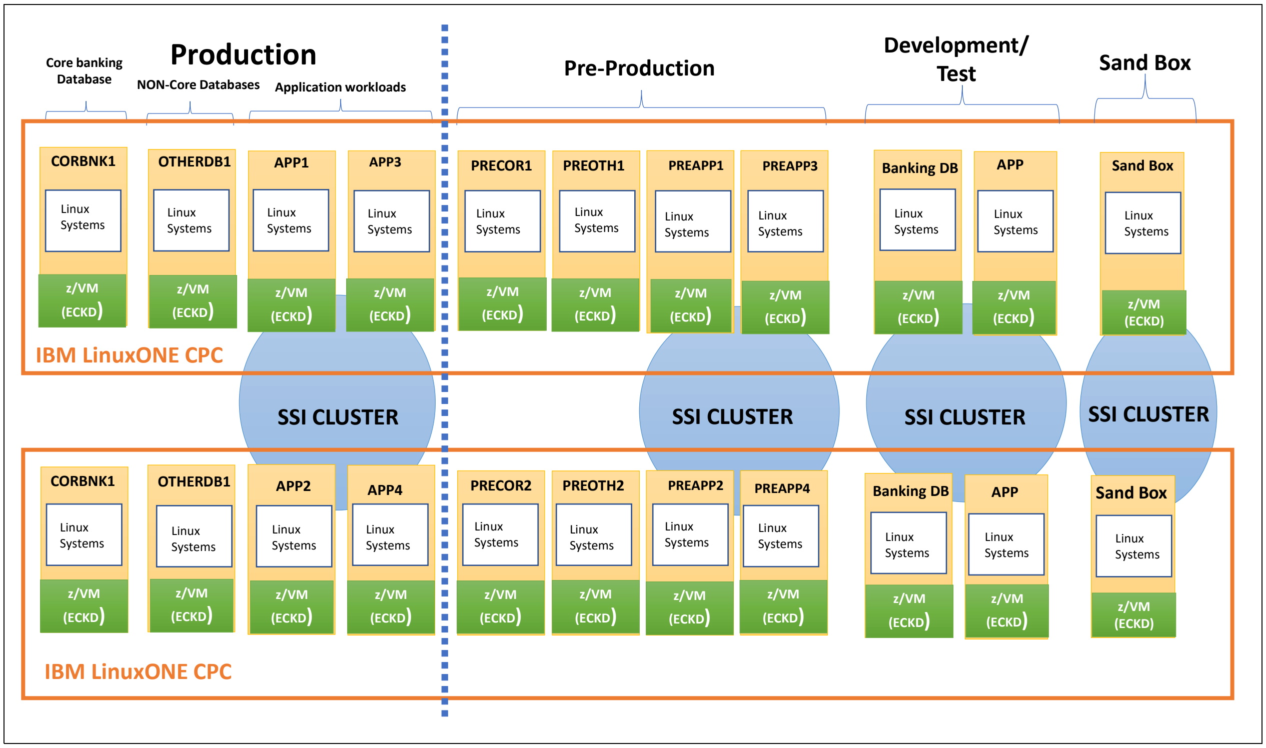


Figure 4-2 Standard Temenos Solution on IBM LinuxONE Systems.

When deploying any production hardware or application, it is important to ensure that there is no single point of failure. Considering this, always plan to have at least two or more of: hardware, Linux systems, applications systems, network equipment and connections, storage infrastructure and so on.

4.1 The installation journey for the IBM LinuxONE hardware

IBM engineers issue a *code 20* after they have completed unpacking, assembling, connecting power, and the initial power-up and general diagnostic testing for a new IBM system. When the system is classified as code 20, the machine's warranty start date is set and the system is considered yours.

Working together (the customer and the IBM engineer), the I/O configuration and logical partitions layout is created. The Input Output Definition File (IODF) configures the IBM LinuxONE server hardware and defines the partitions (LPARs) on the IBM LinuxONE. IBM Engineers use the IODF to create the mapping of where each I/O cable is to be plugged into the IBM LinuxONE.

Each LPAR is set up with real memory layout and the number of IFLs assigned to the LPAR. Working together, you and the IBM Team set up the system using IBM LinuxONE best practices and IBM LinuxONE and Temenos recommendations.

The following sections give a visual perspective and a high-level overview of the installation journey.

4.1.1 Sandbox LPARs - Sandbox environment

Figure 4-3 on page 88 shows the Sandbox environment.

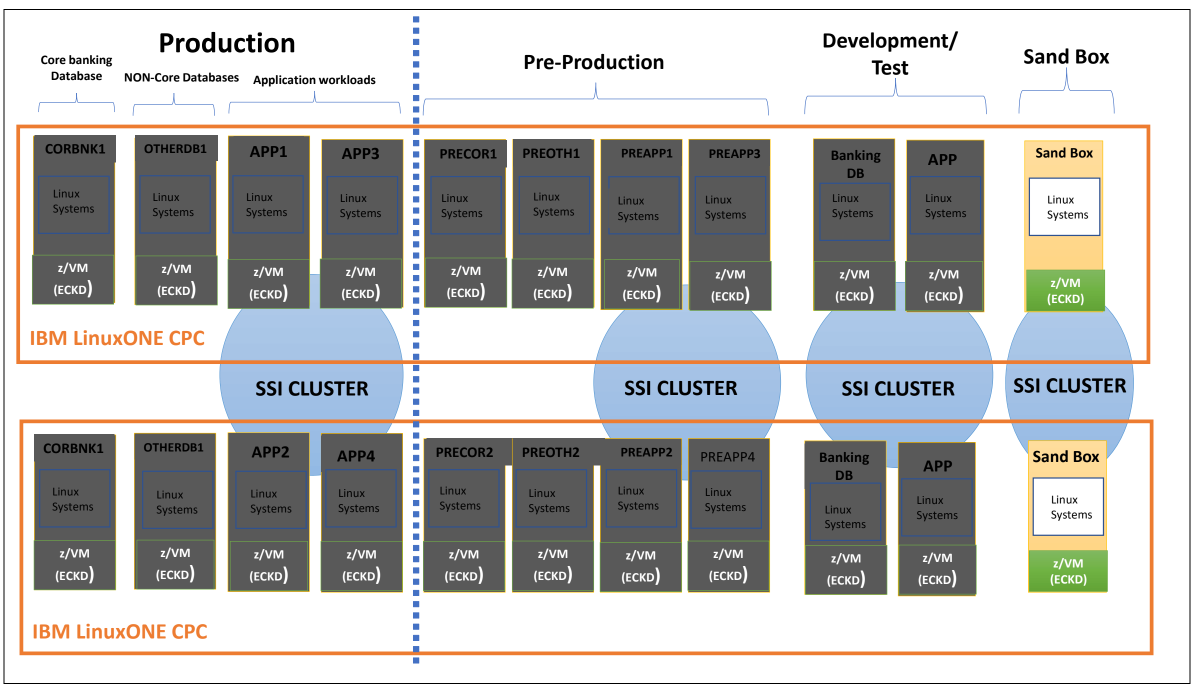


Figure 4-3 Sandbox environment.

We are now ready to install the Sandbox LPAR systems. IBM Hypervisor (IBM LinuxONE z/VM) is the first operating system that is made operational. These Sandbox systems are used to provide training and help in verifying that all network connections and hardware connections are working correctly. Each LPAR is set up as a Single System Image (SSI), so they are the same as all the other IBM LinuxONE z/VM LPARs. This provides the foundation where future IBM LinuxONE z/VM maintenance and upgrades are installed. This Sandbox environment provides assurance that the hardware and operating system are not negatively impacted when maintenance or upgrades are applied in the production environment. In addition, your first virtual Linux systems will be installed in the Sandbox environment, as should all Linux patches for the verification reason previously stated.

4.1.2 Development and Test environment

Figure 4-4 shows the Development and Test environment.

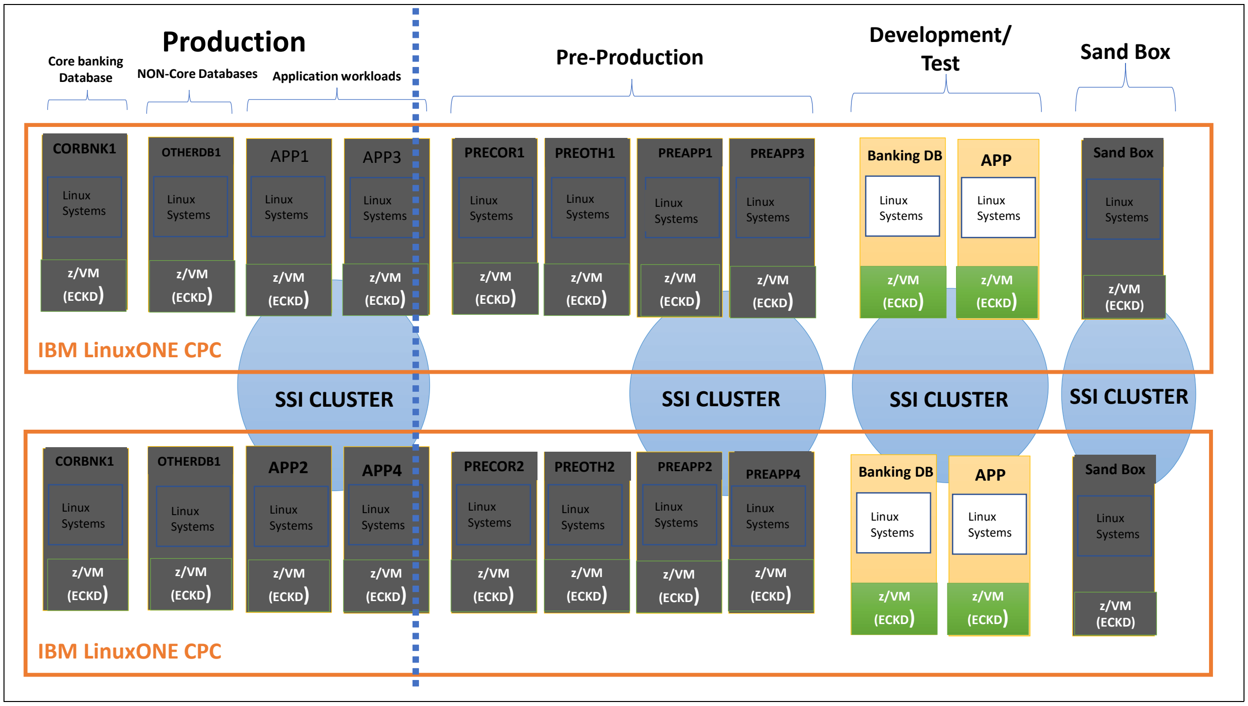


Figure 4-4 Development and Test environment.

The Development and Test environment defines the first set of LPARs used to create or develop Temenos and IBM LinuxONE systems on this platform. The Development or Test environment consists of four LPARs within a single SSI cluster with each LPAR running an IBM LinuxONE z/VM Hypervisor.

Two LPARs run the Temenos Application software, web services, and any other non-database software. The virtual Linux guests, running on these two LPARs, have many version or levels of applications and Linux operating systems installed on them. It is only on these virtual Linux guests where development and initial testing should occur.

The other two LPARs run both core and non-core banking databases. One of the benefits of running these databases only on these two LPARs is a reduction in database licensing costs. The segregation of development or test databases to their own two LPARs ensures that application development processes (running on the other Development or Test LPARs) can proceed unaffected by database workloads.

4.1.3 Pre-Production environment

Figure 4-5 on page 90 shows the Pre-Production environment.

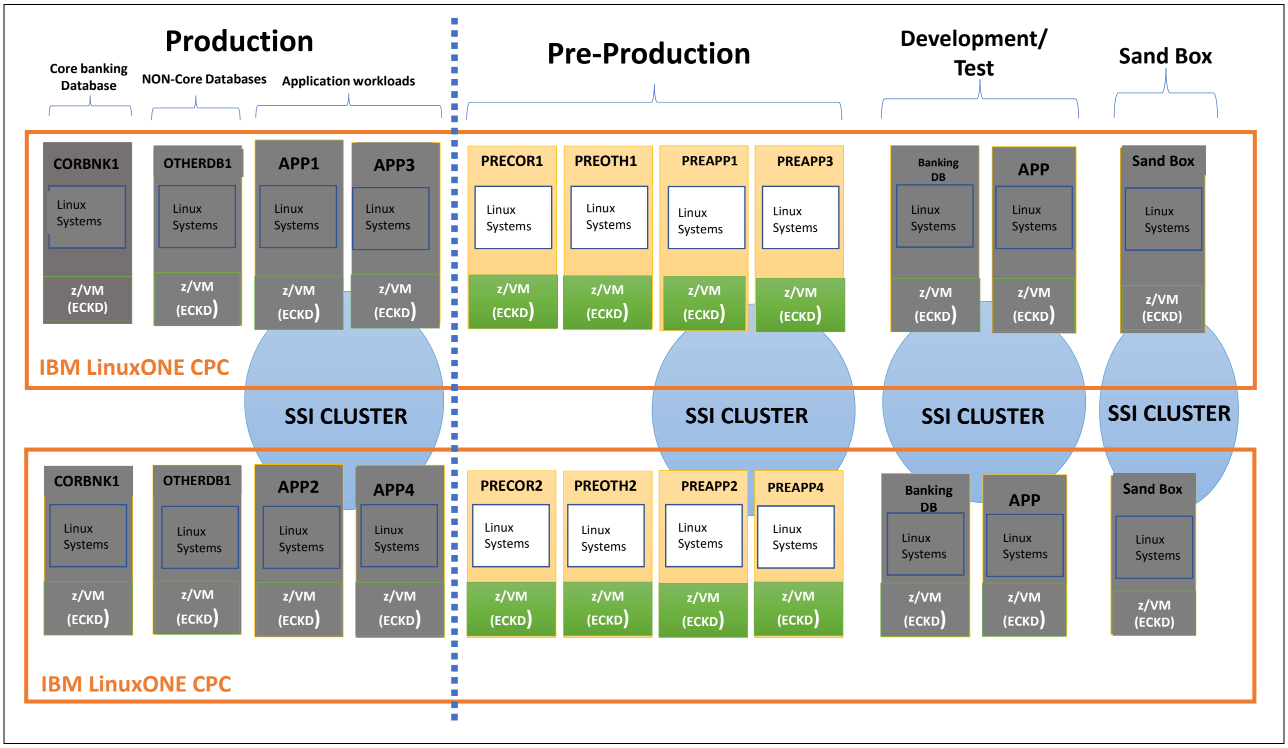


Figure 4-5 Pre-Production environment.

Within the systems environment hierarchy, the Pre-Production environment is second only to the Production environment. Pre-production systems provide for last chance verification of how any changes might affect production. This is a set of systems that mimic the real production environment. It is in this environment that errors or performance issues due to changes or updates can be caught. It is important that this environment is set up to replicate production as closely as possible.

In Figure 4-5, the Pre-Production environment has the following configuration:

- Two LPARs running the core banking databases
- Another two LPARs running non-core banking databases
- Another four application LPARs that are within a single SSI cluster

This configuration matches the Production environment setup also shown in Figure 4-5.

4.1.4 Production LPARs environment

Figure 4-6 on page 91 shows the Production environment.

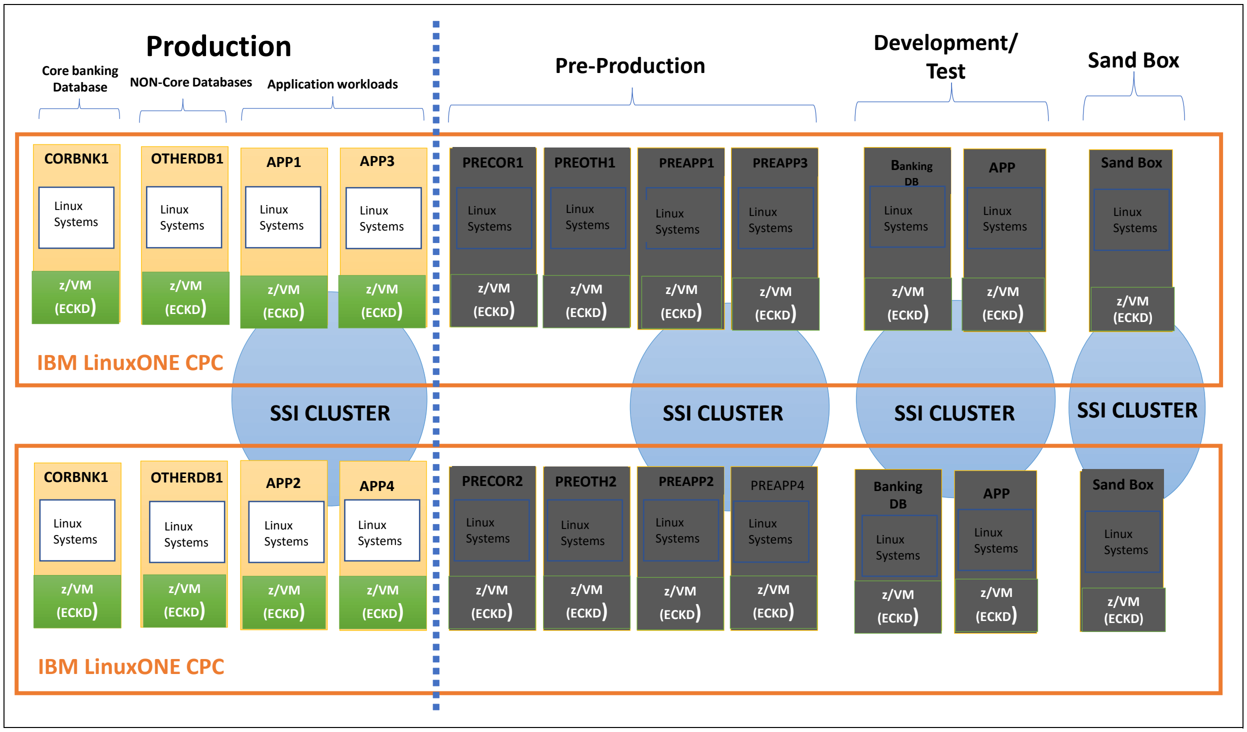


Figure 4-6 Production LPARs environment.

The work that has been done setting up Sandbox, Development and Test, and Pre-Production systems provides the foundation to create a Production environment that can take advantage of the IBM LinuxONE.

Clustering the virtual banking application Linux guests across four LPARs allows room for each LPAR to grow when workload increases. The database servers are split between core banking and non-core banking databases. This split of the databases provides savings in software licensing costs.

4.1.5 Disaster recovery

Figure 4-7 on page 92 shows a high-level storage disaster recovery layout.

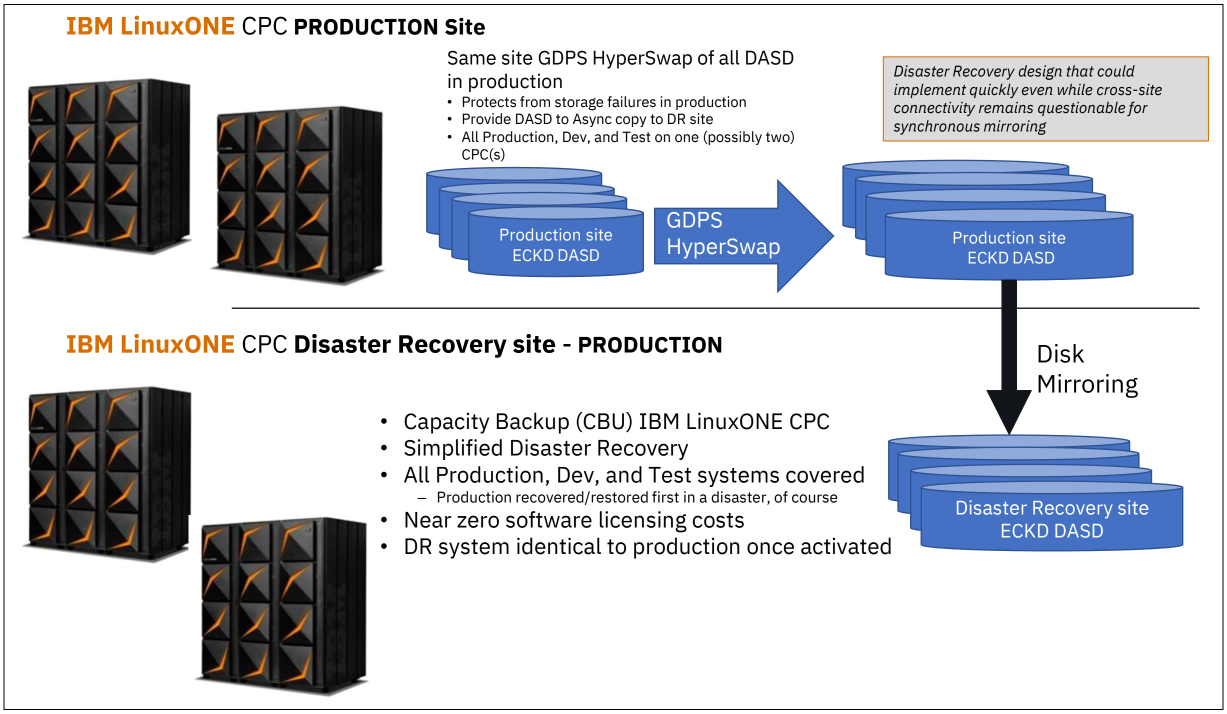


Figure 4-7 High level storage and disaster recovery layout.

IBM LinuxONE has a unique disaster recovery capability. Architecturally, every IBM LinuxONE is engineered to strict standards which ensures that no IBM LinuxONE differs architecturally from another; no matter the version. Because they are architecturally identical, any virtual Linux guest from any LPAR or IBM LinuxONE can run on any other LPAR or IBM LinuxONE as long as it has access to the same network, storage, or copy of the storage. That means no changes are needed for any virtual or native Linux guests to run on another IBM LinuxONE CPC. This portability does not exist on any other hardware platform. Therefore, any Linux guest, native or virtual, can be interchanged seamlessly with another IBM LinuxONE CPC.

The practice of instantaneous data storage mirroring between production and DR sites ensures any change, modification, or update applied in production to Linux guests is automatically replicated on the DR site.

Capacity Backup (CBU) processors are another unique and cost advantageous feature of the IBM LinuxOne offering. CBUs are processors that are available only on the DR system and are priced at a lower cost than production processors. These DR CBU CPCs are based on the permanent production configuration. They are not active while your production CPC is operational. As such, IBM software licensing and requisite fees apply only to those processors that are active (based on the CPC permanent configuration).

There can be additional fees for non-IBM software. In addition, some non-IBM software packages can require new license keys to take advantage of the additional capacity. Check with your software vendor for details.

Figure 4-8 shows how disaster recovery (DR) matches the production environment.

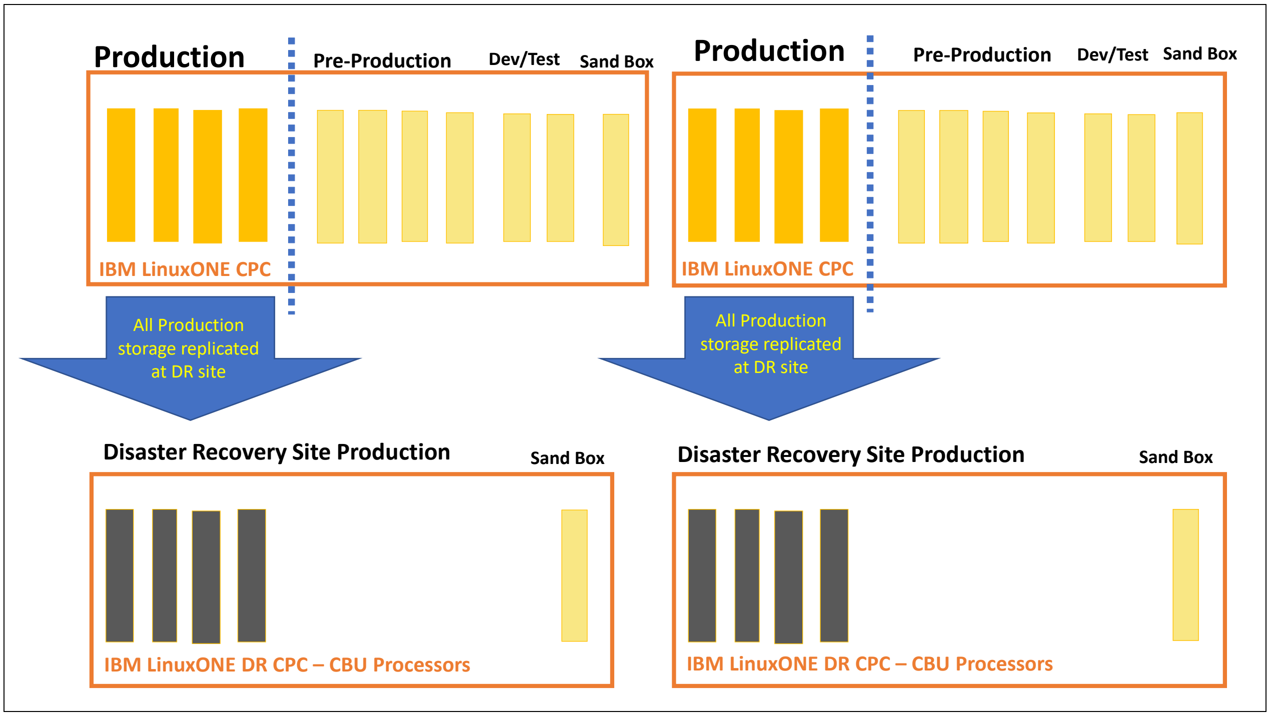


Figure 4-8 DR matching a production environment.

Disaster recovery (DR) CPCs match the production environments. This includes the number of processors, memory, network, and I/O configuration. You can design the DR site to handle only the production workload or you can build the DR site to handle both production and non-production workloads.

Figure 4-9 on page 93 shows the Sandbox flexibility in the DR environment.

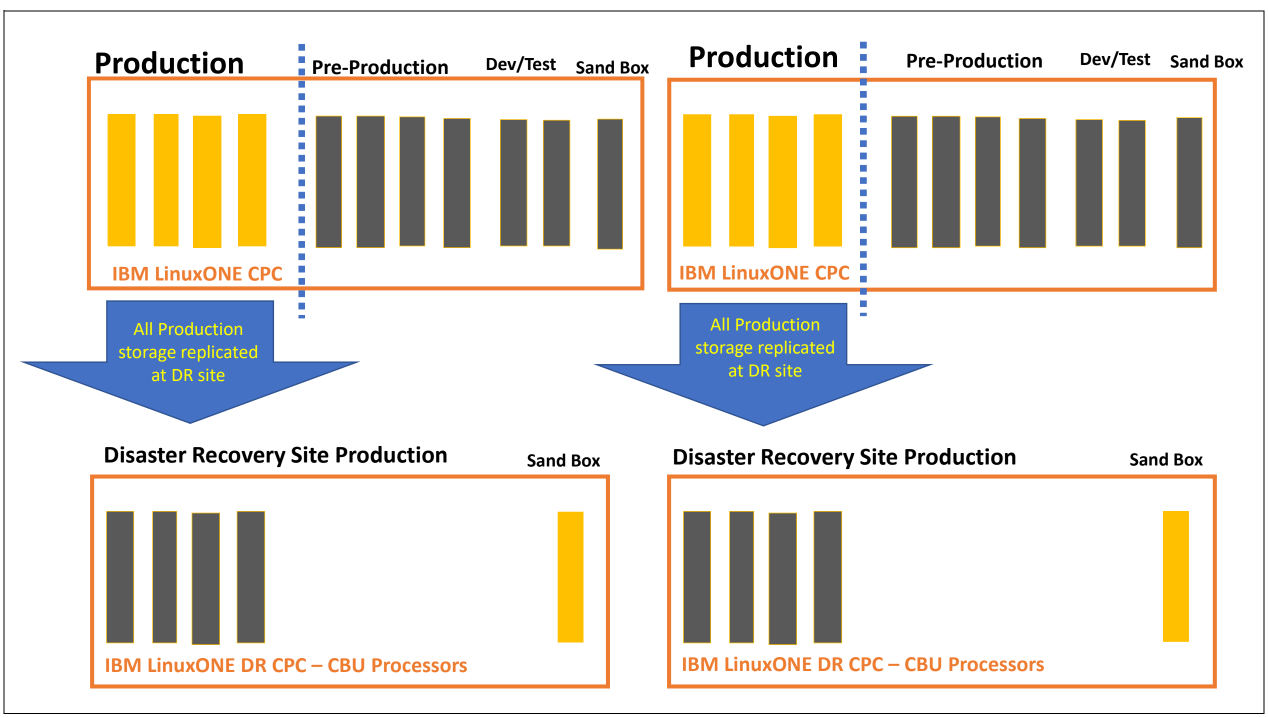


Figure 4-9 Flexible Sandbox capabilities in the DR environment.

The CPCs for each DR site have a small active LPAR (Sandbox). This LPAR is available to the support teams for test purposes to verify whether the network, mirrored storage, and the CPCs are ready to handle a disaster recovery.

Figure 4-10 on page 94 shows the process of engaged disaster recovery.

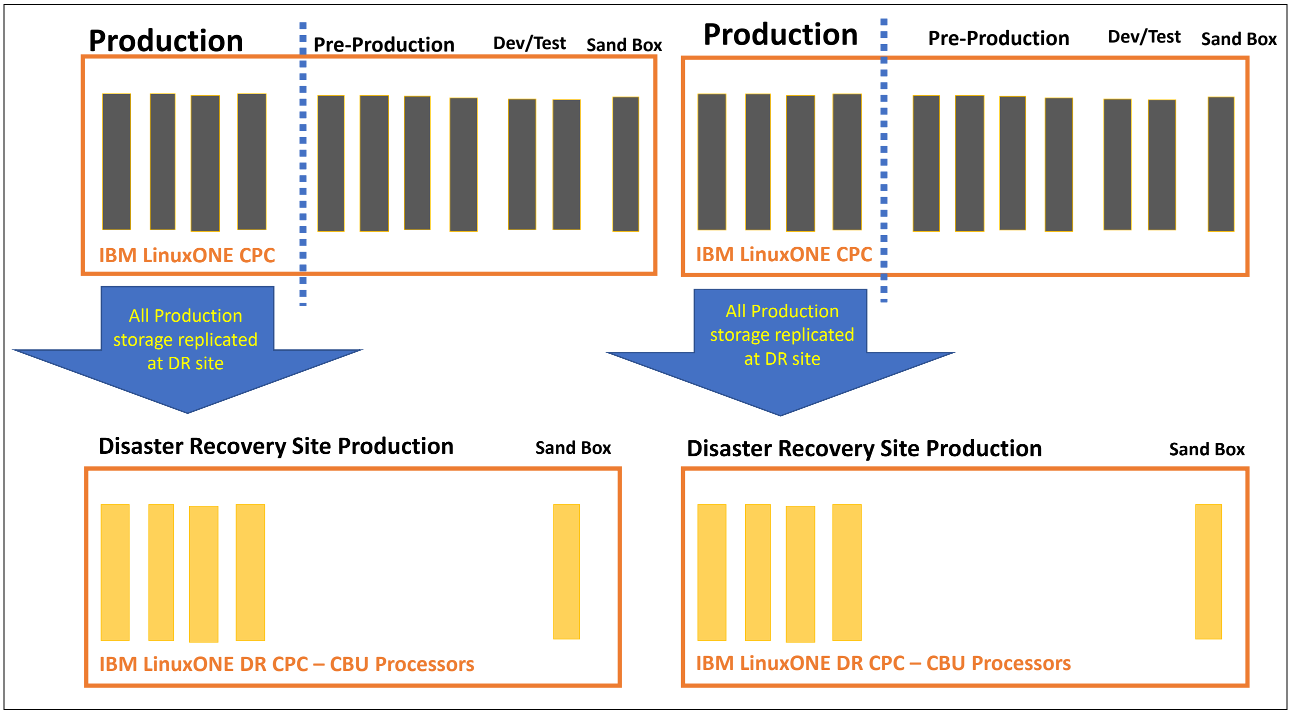


Figure 4-10 Disaster recovery process.

With this type of disaster recovery setup, a runbook can be created documenting the steps to transfer Production to the DR site. This runbook allows anyone in the Support Team structure to execute the process. It can be as simple as the process shown in the following steps:

1. Verify that all Production LPARs are down
2. Activate DR Production LPARs
3. Bring up DR Production systems (IPL systems)
4. Verify DR production virtual servers are active and ready to accept workloads

IMPORTANT: For DR site planning and setup purposes, all non-IBM equipment and workloads running in Production should also be replicated on the Disaster Recovery site. This ensures a complete and seamless recovery process.

Figure 4-11 shows the disaster recovery process with IBM GDPS Virtual Appliance.

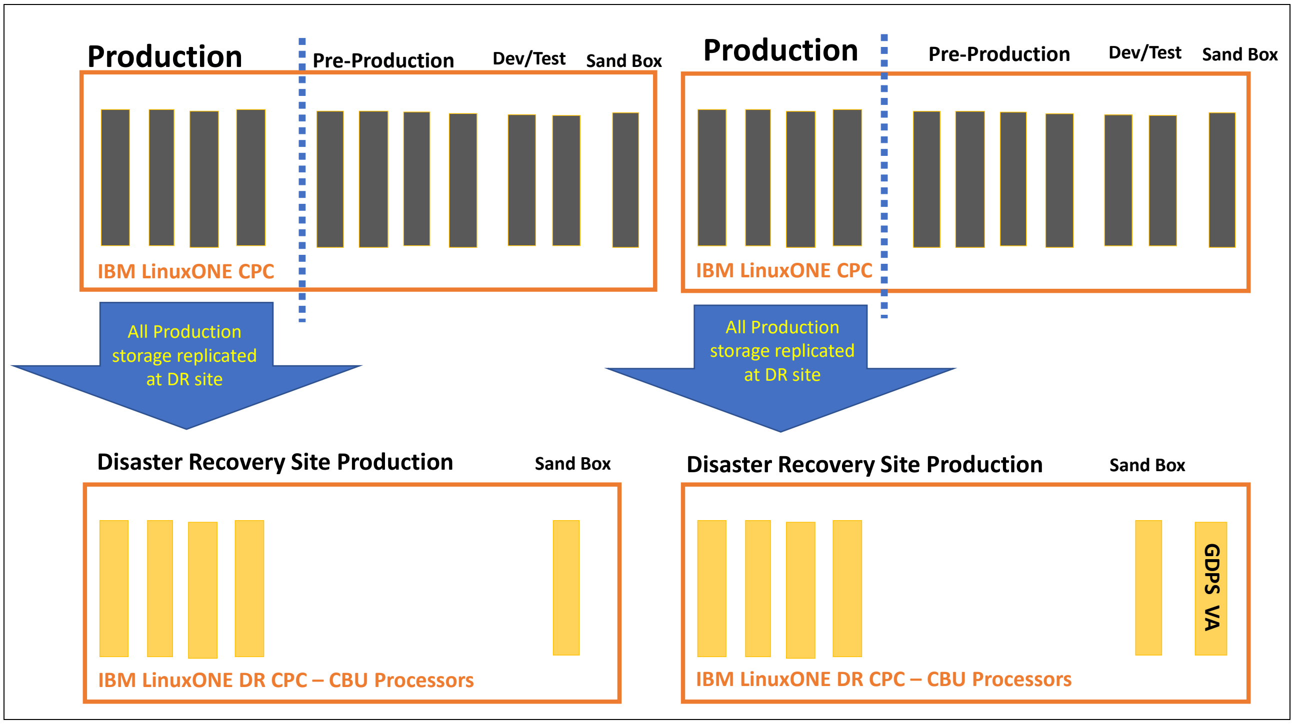


Figure 4-11 IBM GDPS Virtual Appliance.

IBM GDPS Virtual Appliance (IBM GDPS (VA)) is designed to facilitate near-continuous availability and disaster recovery by extending GDPS capabilities for IBM LinuxONE. It substantially reduces recovery time and the complexity associated with manual disaster recovery.

Virtual Appliance requires its own LPAR with a dedicated special processor.

4.2 Tuning

This section contains the Linux and Java tuning considerations to optimize Temenos Transact and its dependent software on IBM LinuxONE.

4.2.1 Linux on IBM LinuxONE

This section talks about IBM LinuxONE specifics for the Linux operating system.

Huge pages

Defining large frames allows the operating system to work with memory frames of 1 MB rather than the default 4 K. This allows smaller page tables and more efficient Dynamic Address Translation. Enabling fixed large frames can save CPU cycles when looking for data in memory. Disable the transparent huge pages (enabled per default) to ensure that the 1 MB pool is assigned when specified. Transparent huge pages tries to assign 2 MB pages until enough contiguous memory is available. The longer that the system is running, this effectiveness can diminish and the more that memory fragmentation occurs. Large page support entails support for the Linux `hugetlbfs` file system. To check whether 1 MB large pages are supported in your environment, issue the following command:

```
grep edat /proc/cpuinfo
```

```
features       : esan3 zarch stfle msa ldisp eimm dfp edat etf3eh highgprs te
```

An output line that lists **edat** as a feature indicates 1 MB large page support.

Defining huge pages with those kernel parameters allocates the memory as part of a pool. To monitor the pool usage, the information can be found using the **cat /proc/meminfo** output. The two components that can benefit from the large frames are Java and Oracle.

See the **USE_LARGE_PAGES** initialization parameter in the Oracle documentation to activate huge pages in the database.

The following kernel parameter in /etc/zipl.conf enables 1 MB large frames:

```
transparent_hugepage=never default_hugepagesz=1M hugepagesz=1M hugepages=<number
of pages allocated at boot>
```

Calculate the number of pages according to the application requirements. The number can be about three/fourths (3/4) of the memory to the instance.

4.2.2 JAVA virtual machine tuning

The IBM Java 1.8 package is the certified Java distribution for IBM LinuxONE with Temenos. Also, the IBM Java 1.8 JDK provides JIT compiler, which has shown in our lab environment to have a positive performance impact.

JVMs or Logical IFLs

In our lab, it was noticed that a 1-1 allocation of JVMs to physical IFLs was a sweet spot when tuning the Transact application for maximum throughput. Considerations must be made based on the transaction mix for your environment.

Shared class cache

Enable the use of a shared class cache between JVMs for AIT and JIT information.

Heap size

Set the minimum (option **-Xms**) and the maximum (option **-Xmx**) JAVA heap size as the same. This ensures the heap size does not change during run time. Make the heap size large enough to accommodate the requirements of your applications but small enough not to impact performance. A heap size that is too large can also impact performance. You can run out of memory or increase the time that it takes the system to clean up unreferenced objects in the heap. This process is known as Garbage Collection.

IBM LinuxONE III Integrated Accelerator for zEDC

This is a transparent exploitation of an integrated on-chip accelerator with no required setup. The prerequisite for using it is IBM Java 8 SR6.

Pause-less garbage collection

Pause-less Garbage Collection (GC) is a new GC mode in the 64-bit IBM SDK for Java 8 SR5. Its purpose is to reduce the impact of GC *stop-the-world* phases and improve the throughput and consistency of response times for Java applications. This technology leverages the new Guarded Storage Facility in IBM LinuxONE hardware to allow additional parallel execution of GC-related processing with application code. Pause-less Garbage Collection is particularly relevant for applications with strict response time Service Level Agreements (SLAs) or large Java heaps.

As seen in Figure 2-6 on page 25, the time where the program threads need to stop (during garbage collection) is massively reduced with the use of the guarded storage facility.

The Pause-less GC mode is not enabled by default. To enable the new Pause-less GC mode in your application, introduce **`-Xgc:concurrentScavenge`** to the JVM options.

Large pages

`JVM "-Xlp"` startup-option is set in the WebLogic Application server for the Temenos Transact application. This setting indicates that the JVM should use Large pages for the heap. If you use the SysV shared memory interface, which includes **`java -Xlp`**, you must adjust the shared memory allocation limits to match the workload requirements.

Garbage Collection Policy gencon

JAVA 7 gencon is the default policy. This policy introduces a nursery area where short living objects are placed. The default nursery area is relatively small. If your system needs to carry more objects, make that area larger.

Red Hat and Security Patches

Red Hat has made updated kernels available to address a number of security vulnerabilities. These patches are enabled by default because Red Hat prioritizes *ready for use* security. Speculative execution is a performance optimization technique which these updates change (both kernel and microcode) and can result in workload-specific performance degradation.

Customers who feel confident that their systems are well protected might want to disable some or all of the protection mechanisms. For more information about controlling the impact of microcode and security patches, read the following Red Hat article, which describes the vulnerabilities patched by Red Hat and how to disable some or all of these mitigations:

https://access.redhat.com/articles/3311301

4.3 Migrating Temenos from x86 to IBM LinuxONE

This section describes how we addressed some limitations of a Temenos application stack deployed on x86 by updating the architecture to the IBM LinuxONE and using its advantages.

The starting point

Our hypothetical initial installation used several x86 servers to implement database and application tiers of the solution. Key aspects to this installation included the following concepts:

- Large number of physical servers to be maintained
- Gaps in availability coverage due to poor manageability of virtual instances
- Physical connectivity (SAN, network) requirements
- Large number of virtual instances to operate and maintain

Step 1: IBM LinuxONE hardware

Having a large number of x86 servers is administratively burdensome and takes a large amount of data center resources (such as floor space, power consumption, networking and storage ports, cooling effort, and so on). In addition, software licensing is often based on server physical cores so the amount of x86 capacity required for a given workload can have significant licensing impact.

IBM LinuxONE addresses this by consolidation of the many x86 servers to two IBM LinuxONE servers. This provides a reduction in the physical server count and a reduction in the connectivity requirements.

As discussed previously, it might be possible to use a single IBM LinuxONE server. However, using two IBM LinuxONE servers provide greater flexibility in managing situations that require a server to be removed from service temporarily.

Step 2: Hypervisor

We use z/VM as the hypervisor, using the SSI function. This improves the manageability of virtual instances by eliminating the need to synchronize configuration details between *shadow* virtual instances. It also offers easier options for local recovery of virtual instances (restart on the same IBM LinuxONE server or restart on the other one) in the event of a restart being needed.

Running the members of the SSI cluster across the two IBM LinuxONE servers provides the maximum flexibility.

Step 3: Linux virtual instances

Rather than simply re-creating each virtual instance from the x86 environment, we use the superior vertical scalability of the IBM LinuxONE server and the z/VM hypervisor. This reduces the total number of virtual instances.

z/VM also allows a high degree of horizontal scalability by supporting large numbers of virtual instances per system. This provides the option of adjusting the number of instances to make sure that there were enough to prevent a noticeable impact to operation. For example, if a virtual instance needed to be removed from the environment for maintenance or in the event of a failure.

Step 4: Java

Migrating Java applications from one platform to another is easy compared to the migration effort required for C or C++ applications. Even though Java applications are operating system independent, the following implementation and distribution specifics need to be considered:

- Most of the Java distributions have their own Java virtual machine (JVM) implementations. There will be differences in the JVM switches. These switches are used to make the JVM and the Java application run as optimally as possible on that platform. Each JVM switch that is used in the source Java environment needs to verify for a similar switch in the target Java environment.
- Even though Java SE Developer Kits (JDKs) are expected to conform to common Java specifications, each distribution will have slight differences. These differences are in the helper classes that provide functions to implement specific Java application programming interfaces (APIs). If the application is written to conform to a particular Java distribution, the helper classes referenced in the application must be changed to refer to the new Java distribution classes.
- Special procedures must be followed to obtain the best application migration. One critical point is to update the JVM to the current stable version. The compatibility with earlier versions is significant and performance improvements benefit applications.
- Ensure that the just-in-time (`JIT`) compiler is enabled.
- Set the minimal heap size (`-Xms`) equal to the maximal heap size (`-Xmx`). The size of the heap size should always be less than the total of memory configured to the server.

Step 5: IBM WebSphere Application Server

IBM has ported many of its software products to IBM LinuxONE. The benefit to customers is that a migration from one platform to another is, in many cases, effortless. This is because many of these products share their code base across multiple platforms. This benefit is particularly the case for IBM WebSphere Application Server, which from Version 6, has had the same code base on Intel x86 and IBM LinuxONE. Thus simplifying migration considerably. You can use deployment manager and **`was-agent`** to deploy IBM WebSphere Application Server on the new IBM LinuxONE LPARs or Linux guests under z/VM. Generally, migrating from IBM products on distributed servers to the same IBM products on IBM LinuxONE is a relatively straightforward process.

For detailed guidance on migrating IBM WebSphere Application Server see the following link:

http://www.redbooks.ibm.com/redbooks/pdfs/sg248218.pdf

Step 6: Oracle database

In our recommended architecture the Oracle database is deployed with the Real Application Clusters (RAC) feature. This provides a highly available database tier to the Temenos application servers.

Deploying Oracle database in a z/VM SSI environment gives some choices for how the system can be configured. Oracle RAC One Node is a configuration of Oracle specifically designed to work with virtualized environments like z/VM. It can offer most of the availability benefits of full RAC without most of the cluster overhead. It does this by sharing some of the availability responsibility with the hypervisor. For example, being able to relocate a database guest from one z/VM system to another might be enough to provide database service levels high enough for your installation.

Step 7: TAFC to TAFJ Migration

A small but significant number of Temenos clients continue to run Transact on their C-based application framework (TAFC). However, Transact versions, greater than R18, are now deployed exclusively on their Java-based application framework (TAFJ). Organizations on TAFC will need to migrate to TAFJ to run Temenos software on IBM LinuxONE. Clients planning to upgrade from Temenos releases R14 and older require a two-step upgrade approach. This approach shifts to an intermediate release that supports TAFC and TAFJ before being able to upgrade to a current TAFJ release.

A typical migration consists of running the TAFC and TAFJ environments side by side during the migration process. Then a phased approach is used to upgrade the multiple parts of the core banking solution with the least amount of impact on the core banking operations. An important consideration is to ensure that all customizations and applications support the JDBC driver for connectivity to the database. This is the only driver supported by Temenos and IBM LinuxONE. See Figure 4-12.

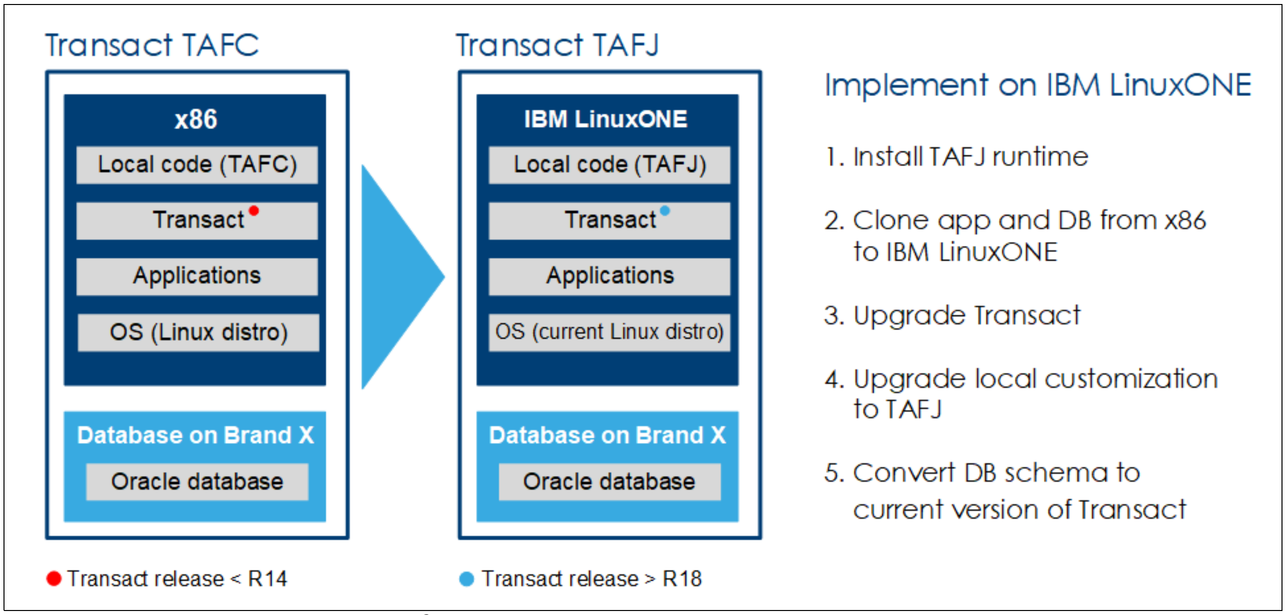


Figure 4-12 Migrating TAFC to TAFJ. [2]

Main points of a typical Temenos TAFC to TAFJ Migration

The following steps overview the main considerations when migrating from TAFC to TAFJ:

1. Install the desired Transact TAFJ version onto the IBM LinuxONE LPAR or Guest
2. Migrate Applications and DB from x86 to IBM LinuxONE
3. Port (applicable) C applications to run on IBM LinuxONE. If necessary, update applications to use JDBC. Run the Oracle **`DBUpdate`** conversion tool to migrate the existing database schema and data to a new target version or release of the Oracle DB
4. Deploy the new TAFJ compatible version of Transact on to a new IBM WebSphere Application Server running on IBM LinuxONE
5. Upgrade any specific customized modules to support the current Transact installation on IBM LinuxONE
6. Run the **`DBUpdate`** conversion process to update the database schema and data to support the current Transact installation on IBM LinuxONE

4.4 Temenos Transact certified Cloud Native deployment for IBM LinuxONE

IBM, Red Hat, and Temenos have designed the first on-premises cloud native stack (stack 11). This stack delivers a stepping stone to cloud for clients. This allows the delivery on an on-premises private cloud based on IBM LinuxONE with Red Hat OpenShift and IBM Cloud Paks. Figure 4-13 on page 101 shows Stack 11 for Temenos Transact cloud.

[2] Courtesy of Temenos Headquarters SA

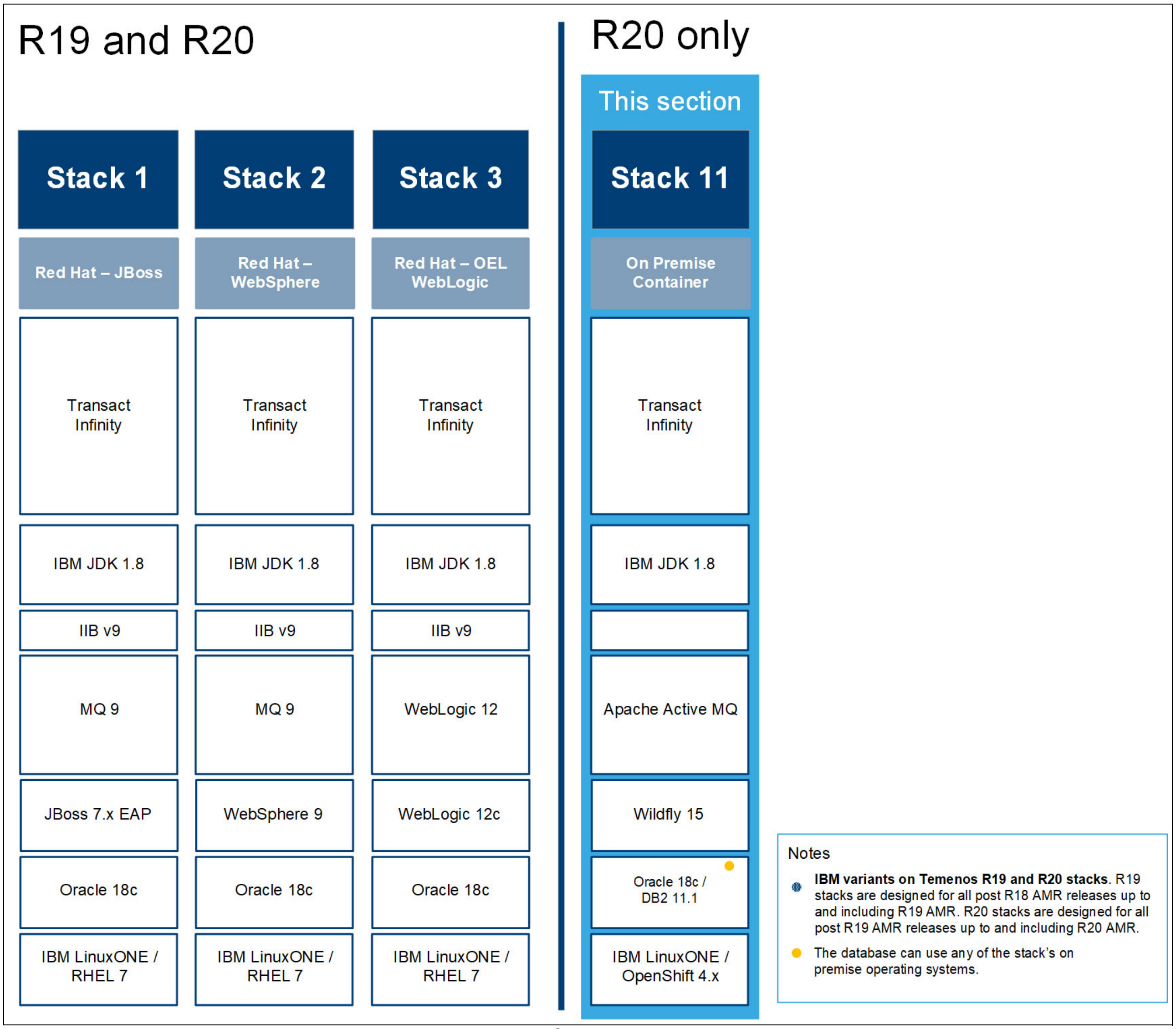


Figure 4-13 Stack 11 for Temenos Transact cloud. [3]

Figure 4-14 on page 102 shows one option for a cloud architecture.

[3] Courtesy of Temenos Headquarters SA

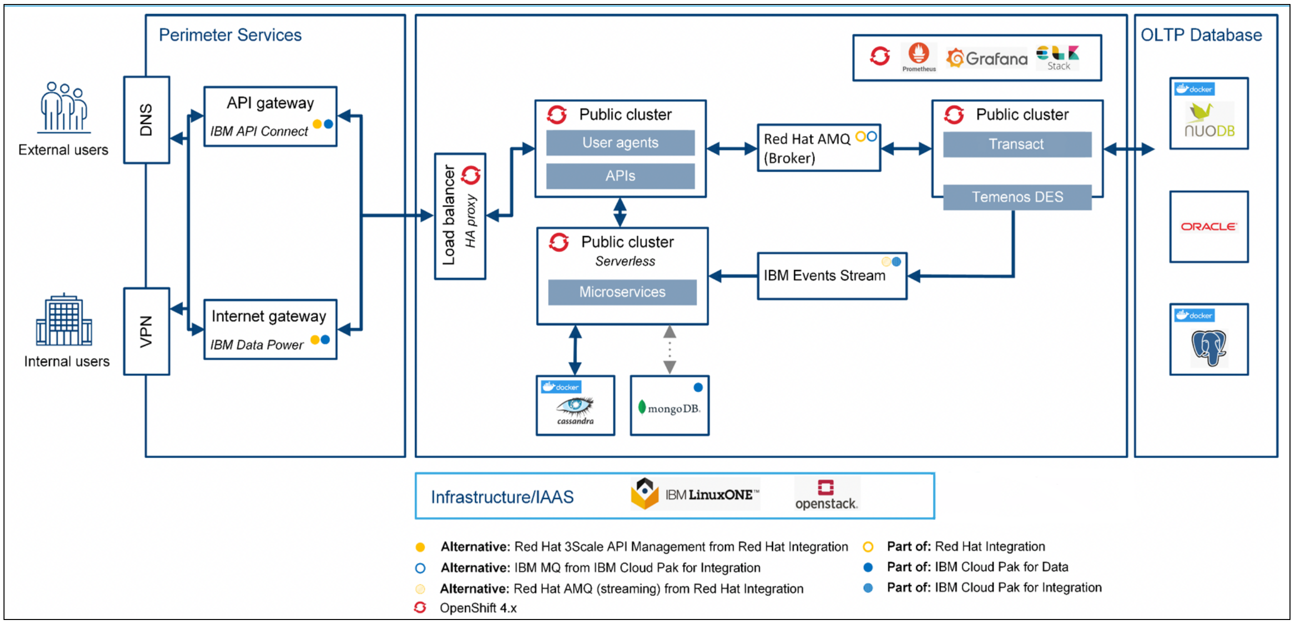


Figure 4-14 Cloud architecture.

This offering provides the highest levels of security and secure data residency for your core and delivers the benefits that are inherent with cloud native architecture.

This offering also allows for the concept of Hybrid cloud. Simultaneously maintaining the core data on-premises cloud native and using IBM Cloud (or other cloud providers) in a consistent and governed manner. This is achieved through the combination of Red Hat OpenShift and IBM Cloud Paks.

A possible use case is Temenos Transact deployed on IBM LinuxONE on-premises cloud native and Temenos Infinity on IBM Hyper Protect public cloud.

4.5 Temenos deployment options on IBM Hyper Protect public cloud

Temenos and IBM have tested Transact on IBM Hyper Protect DBaaS platform running PostgreSQL. PostgreSQL will be ready and certified as a backend database by May 2020.

For more information about this platform at the following link:

https://www.ibm.com/cloud/hyper-protect-dbaas

To discuss public cloud options, contact the following person:

John Smith

WW Offering Manager for Temenos | Linux Software Ecosystem Team

WW Offering Management, Ecosystem & Strategy for IBM LinuxONE

jsmith88@uk.ibm.com

A

Sample product and part IBDs and model numbers

This appendix provides a sample configuration to be used by IBM or certified Business Partners to license key components for the Temenos application to be hosted on IBM LinuxONE II (3907-LR1 model, introduced April 2018). This sample configuration is to help familiarize readers with a concept of the extent of equipment and software required as well as product codes (at the time of this publication). Configurations can vary from the one presented in this appendix.

Table A-1 provides a sample eConfig for a single instance of an IBM 3907-LR1 IBM Rockhopper LinuxONE II Server with 4 x IFLs, 832gb memory, Dynamic Partition Manager and key I/O technologies OSA-Express6S and FCP Express32S port channels.

Table A-1 Sample Configuration

Product	Description	Qty
3907-LR1	IBM LinuxONE Rockhopper II	1
16	HW for DPM	1
19	Manage FW Suite	1
33	Service Docs Optional Print	1
63	HMC Rack Mount	1
154	HMC Rack Keybd/Monitor/Mouse	1
173	PCIe fanout Gen3	2
174	Fanout Airflow PCIe	6
235	US English #103P	1
300	Model LR1 Air Cooled	1
401	PCIe Interconnect Gen3	2
425	OSA-Express6S 10 GbE SR	4

Product	Description	Qty
426	OSA-Express6S 1000BASE-T	4
439	FCP Express32S SX 2 ports	4
617	16U Reserved	1
622	Switchable PDU	4
623	Ethernet Switch	2
638	CPC Drawer Max24	1
641	CPC PSU	4
1021	STP Enablement	1
1064	IFL	4
1157	0-Way Processor A00	1
1040	A00 Capacity Marker	1
1628	64 GB Mem DIMM (5/feat)	4
1742	32 GB Memory Cap Incr>128 GB	24
3100	Lift Tool Kit	1
3101	Extension Ladder	1
3557	832 GB Memory	1
3863	CPACF Enablement	1
4001	PCIe+ I/O Drawer	1
7919	Bottom Exit Cabling	1
7943	32A/250V LSZH Cord	4
9883	19in Rack	1
9975	Height Reduce Ship	1

Table A-2 provides a sample IBM Software Bill of Materials (includes both systems software and middleware) based upon 4 x IFLs on IBM LinuxONE II (same as in the previous configuration).

Table A-2 Sample IBM Software Bill

Part number	Product description	Quantity per IFL	Chargeable unit
5741A09	z/VM Version 7	400	PVU
5741SNS	z/VM v7 Subscription & Support	400	PVU
5741A09	DirMaint Facility Feature Version 7	400	PVU
5741SNS	DirMaint Facility Feature S&S	400	PVU
5741A09	Performance Toolkit for z/VM Version 7	400	PVU

Part number	Product description	Quantity per IFL	Chargeable unit
5741SNS	Performance Toolkit for z/VM v7 S&S	400	PVU
5698IS2	Infrastructure Suite for z/VM	40	PVU
5698IS1	Infrastructure Suite for z/VM v7 S&S	40	PVU
Via Vendor	Red Hat Enterprise Linux (Annual license)	1	Per IFL
E0LNBLL	IBM MQ Advanced for Linux on z Systems PVU Annual SW S&S Renewal 12 Months	400	PVU
D1VL7LL	IBM Spectrum Scale SE for LoZ Server license Per 10 PVUs License + SW S&S 12 Months	40	PVU
E0NZ4LL	IBM Spectrum Scale SE for LoZ Server license per 10 PVU's Annual SW S&S Renewal 12 Months	40	PVU
D1VL4LL	IBM Spectrum Scale SE for LoZ Client license Per 10 PVUs License + SW S&S 12 Months	40	PVU
E0NZ3LL	IBM Spectrum Scale SE for LoZ Client license Per 10 PVUs Annual SW S&S Renewal 12 Months	40	PVU
Via Vendor	Oracle DB Enterprise for Linux on System z®	1	Per IFL
	Tivoli System Automation		
Via Vendor	Oracle Weblogic Application Server for Linux on System z License	1	Per IFL

B

Creating and working with the first IODF for the server

This appendix provides an example of how to create the server's first IODF and the following additional aspects:

- An example of a minimal IOCP deck to perform this operation
- Listed important aspects or parts of the operation
- Enabling the IOCP
- A success verification example of the process

The first IODF for the server

Creation of the first IODF for an IBM LinuxONE server can be complicated. Because there is no operating system running on the server, how do we run HCD/HCM to create one?

If there is already an existing IBM LinuxONE server on which the IODF for the new machine can be created, the IODF can be exported from the existing machine to be installed (using Standalone IOCP) on the new machine. However, what if this machine is the first IBM LinuxONE server at your installation? In that scenario, the Standalone IOCP must be used. Rather than attempting to do the initial definition for the entire machine using this method, a minimal IOCP deck defining a single LPAR and basic DASD can be used. This simple IOCP can be activated to make available a single system into which a z/VM system can be installed. This z/VM system is then used to download the HCM code to a workstation and start the HCD Dispatcher. HCM is installed and used to create an IODF with more complete definitions of the system.

An example of a minimal IOCP deck to perform this operation is shown in Example 4-1.

Example 4-1 Minimal IOCP deck

```
ID    MSG1='Initial IOCP ',                                    *
      MSG2='             ',SYSTEM=(3906,1)
RESOURCE PARTITION=((CSS(0),(START1,1),(*,2),(*,3),(*,4),(*,5)*
      ,(*,6),(*,7),(*,8),(*,9),(*,A),(*,B),(*,C),(*,D),(*,E),(*
      *,F)))
CHPID PATH=(CSS(0),20),SHARED,PARTITION=((START1),(=)),       *
      PCHID=1DC,TYPE=FCP
CHPID PATH=(CSS(0),21),SHARED,PARTITION=((START1),(=)),       *
      PCHID=121,TYPE=FCP
CHPID PATH=(CSS(0),30),SHARED,PARTITION=((START1),(=)),       *
      SWITCH=0B,PCHID=15C,TYPE=FC
CHPID PATH=(CSS(0),31),SHARED,PARTITION=((START1),(=)),       *
      SWITCH=0B,PCHID=1A0,TYPE=FC
CHPID PATH=(CSS(0),50),SHARED,PARTITION=((START1),(=)),       *
      PCHID=13C,TYPE=OSD
CNTLUNIT CUNUMBR=1001,PATH=((CSS(0),20)),UNIT=FCP
IODEVICE ADDRESS=(1000,16),UNITADD=00,CUNUMBR=(1001),UNIT=FCP
CNTLUNIT CUNUMBR=1101,PATH=((CSS(0),21)),UNIT=FCP
IODEVICE ADDRESS=(1100,16),UNITADD=00,CUNUMBR=(1101),UNIT=FCP
CNTLUNIT CUNUMBR=2000,                                         *
      PATH=((CSS(0),30,31)),                                   *
      UNITADD=((00,256)),LINK=((CSS(0),0101,0201)),            *
      CUADD=0,UNIT=2107
IODEVICE ADDRESS=(2000,208),UNITADD=00,CUNUMBR=(2000),         *
      STADET=Y,UNIT=3390B
IODEVICE ADDRESS=(20D0,48),UNITADD=D0,CUNUMBR=(2000),          *
      STADET=Y,UNIT=3390A
CNTLUNIT CUNUMBR=5001,PATH=((CSS(1),50)),UNIT=OSA
IODEVICE ADDRESS=(5000,30),UNITADD=00,CUNUMBR=(5001),UNIT=OSA
```

The important parts of this IOCP deck are as noted in the following list:

- The **SYSTEM** keyword on the **ID** macro indicates the machine type as 3906, which is an IBM LinuxONE Emperor II. This value is updated to suit the machine being installed.

- On the **RESOURCE** macro, a single LPAR named `START1` is being defined in logical channel subsystem 0. All other available partitions, in that CSS, are defined as reserved. It is not a requirement of IOCP to define all LPARs, in this scenario, a **RESOURCE** macro was copied from an existing IODF.
- Five channels are being defined: two for FICON (channels 20 and 21), two for FCP (channels 30 and 31), and one for an OSA Express card for networking (channel 50). It is necessary to have only one channel for each disk type, because redundancy is not important at this point. Also, only one instance of FICON or FCP is needed. (This example shows both for illustration. You choose one or the other, depending on the type of DASD on which you intend to install z/VM.) The **PCHID** values for all the **CHPID** macros must be updated to reflect the correct physical identifiers on the machine being defined.
- The two FCP channels have 16 devices defined, which is more than enough for the minimal configuration required.
- A single FICON DASD control unit is defined, reachable using the two FICON channels. The control unit **LINK** parameter indicates the path through the FICON fabric to reach the DASD. In this case, through FICON channel 30, we arrive at the DASD by leaving the FICON fabric from port 01 on switch 01. Through FICON channel 31, we arrive at the DASD by leaving the FICON fabric from port 01 on switch 02.
- The **IODEVICE** macros for the FICON DASD define a full complement of 256 devices. In this example, the first 208 devices (`00-CF`) are defined as base DASDs and the remaining 48 devices (`D0-FF`) are defined as PAV alias devices. Again this is just for illustration, as that many devices and the ability to multi-path is not a requirement at this time.
- The OSA Express device has been defined in QDIO mode (CHPID type OSD), which is the standard mode for TCP/IP networking. 30 usable device addresses have been defined, which is more than necessary for the initial configuration.

To enable this minimal IOCP, the **Input/output (I/O) Configuration** task is started from the Support Element.

> **Note:** The most convenient way to access the Support Element for this operation is to use the **Single Object Operations** function from the HMC.

If the machine is not already operating, a Power-on Reset (POR) is performed using the Diagnostic (D0) IOCDS. After the POR has completed, you can select one of the diagnostic LPARs and start the Input/output (I/O) Configuration task.

Four entries, labeled A0 to A3, are shown. D0 is also shown, but it is not user modifiable. These are the IOCDS slots, that contain the hardware portion of the I/O definition information.

To update and generate the minimal IOCP deck, select one of the **A-slots** and choose **Edit source** from the menu. When the edit window appears, copy and paste the minimal IOCP deck you have edited into the edit window. From the File menu click **Save**, and then close the editor window. You can then select **Build dataset** from the menu. After confirming your selection, the Standalone IOCP program is loaded into the Diagnostic LPAR and processes the IOCP deck. Progress messages appear in the status box. Ideally, the IOCP deck was successfully processed and the binary IOCDS slot has been updated with your configuration. If an error in processing occurred, the Standalone IOCP program updates the source file with comments to explain the error.

If Standalone IOCP successfully generated your deck, the source file is also updated with comments that provide some information. Example 4-2 shows an example of this information.

Example 4-2 IOCP comments after successful run

```
*ICP  ICP071I IOCP GENERATED A DYNAMIC TOKEN HOWEVER DYNAMIC I/O
*ICP          CHANGES ARE NOT POSSIBLE WITH THIS IOCDS
*ICP  ICP063I ERRORS=NO,  MESSAGES=YES, REPORTS PRINTED=NO,
*ICP          IOCDS WRITTEN=YES, IOCS WRITTEN=YES
*ICP  ICP073I IOCP VERSION 05.04.01
```

The most important part of this output is in `ICP063I`, where we see `IOCDS WRITTEN=YES`.

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

These IBM Redbooks publications provide additional information about topics in this book. Note some books are available only in soft copy.

- Practical Migration from x86 to LinuxONE

  http://www.redbooks.ibm.com/abstracts/sg248377.html?Open
- Oracle on LinuxONE

  http://www.redbooks.ibm.com/abstracts/sg248384.html
- Scale up for Linux on LinuxONE

  https://www.redbooks.ibm.com/abstracts/redp5540.html?Open
- Securing Your Cloud: IBM Security for LinuxONE

  https://www.redbooks.ibm.com/abstracts/sg248447.html?Open
- OpenShift OKD on IBM LinuxONE, Installation Guide

  http://www.redbooks.ibm.com/abstracts/redp5561.html?Open
- IBM DB2 with BLU Acceleration

  http://www.redbooks.ibm.com/abstracts/tips1204.html?Open
- WebSphere Application Server V8.5 Migration Guide

  http://www.redbooks.ibm.com/redbooks/pdfs/sg248218.pdf
- GDPS Virtual Appliance V1R1 Installation and Service Guide

  https://www.redbooks.ibm.com/redbooks/pdfs/sg246374.pdf
- Implementing IBM Spectrum Scale

  https://www.redbooks.ibm.com/redpapers/pdfs/redp5254.pdf
- IBM Spectrum Scale (GPFS) for Linux on z Systems

  https://www.redbooks.ibm.com/redpapers/pdfs/redp5308.pdf
- Best practices and Getting Started Guide for Oracle on IBM LinuxONE

  http://www.redbooks.ibm.com/redpieces/pdfs/redp5499.pdf
- Maximizing Security with LinuxONE

  https://www.redbooks.ibm.com/redpapers/pdfs/redp5535.pdf
- Hyper Protect Services

  https://www.ibm.com/cloud/hyper-protect-services

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

The Temenos Stack Runbooks provide more information about using Temenos stacks with different application servers. Temenos customers and partners can access the Runbooks through either of the following links:

- The Temenos Customer Support Portal: https://tcsp.temenos.com/
- The Temenos Partner Portal: https://tpsp.temenos.com/

Online resources

- Leveraging IBM LinuxONE and Temenos Transact for Core Banking Solutions

  https://www.ibm.com/downloads/cas/NEO7QNLJ
- LinuxONE for Dummies

  https://www.ibm.com/downloads/cas/LBOVYYJJ
- Installing IBM MQ server on Linux

  https://www.ibm.com/support/knowledgecenter/en/SSFKSJ_8.0.0/com.ibm.mq.ins.doc/q008640_.htm

Help from IBM

IBM Support and downloads

ibm.com/support

IBM Global Services

ibm.com/services

Redbooks

Temenos on IBM LinuxONE Best Practices Guide

SG24-8462-00

ISBN 0738458457

IBM

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

Redbooks

Temenos on IBM LinuxONE Best Practices Guide

SG24-8462-00

ISBN 0738458457

IBM

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

Redbooks

Temenos on IBM LinuxONE Best Practices Guide

SG24-8462-00

ISBN 0738458457

IBM

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

Redbooks

Temenos on IBM LinuxONE Best Practices Guide

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

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

Redbooks

Temenos on IBM LinuxONE Best Practices Guide

SG24-8462-00

ISBN 0738458457

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

Redbooks

Temenos on IBM LinuxONE Best Practices Guide

SG24-8462-00

ISBN 0738458457

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

SG24-8462-00

ISBN 0738458457

Printed in U.S.A.

Get connected

Redbooks®

ibm.com/redbooks