Modern Mainframes & Linux Running on Them

Benjamin Block ‹bblock@de.ibm.com›

March 17, 2019

IBM Deutschland Research & Development GmbH

Trademark Attribution Statement

The following are trademarks of the International Business Machines Corporation in the United States, other countries, or both.

Not all common law marks used by IBM are listed on this page. Failure of a mark to appear does not mean that IBM does not use the mark nor does it mean that the product is not actively marketed or is not significant within its relevant market.

IBM, the IBM logo, ibm.com, AIX, CICS, Db2, developerWorks, DS8000, FICON, IBM FlashSystem, IBM Z, IBM z Systems, IBM z13, IBM z13, IBM z14, MVS, OS/390, Power, POWER, Redbooks, S390-Tools, S/390, Storwize, System z, System z9, System z10, System/390, z Systems, WebSphere, z9, z13, z13s, z/Architecture, z/OS, z/VM, and z/VSE are trademarks or registered trademarks of International Business Machines Corp., registered in many jurisdictions worldwide. Other product and service names might be trademarks of IBM or other companies. A current list of IBM trademarks is available on the Web at "Copyright and trademark information" here: https://www.ibm.com/legal/copytrade.

The following are trademarks or registered trademarks of other companies.

Intel, Intel logo, Intel Inside, Intel Inside logo, Intel Centrino, Intel Centrino logo, Celeron, Intel Xeon, Intel SpeedStep, Itanium, and Pentium are trademarks or registered trademarks of Intel Corporation or its subsidiaries in the United States and other countries.
Linux is a registered trademark of Linus Torvalds in the United States, other countries, or both.
Microsoft, Windows, Windows NT, and the Windows logo are trademarks of Microsoft Corporation in the United States, other countries, or both.
UNIX is a registered trademark of The Open Group in the United States and other countries.
VMware, the VMware logo, VMware Cloud Foundation, VMware Cloud Foundation Service, VMware vCenter Server, and VMware vSphere are registered trademarks or trademarks of VMware, Inc. or its subsidiaries in the United States and/or other jurisdictions.

All other products may be trademarks or registered trademarks of their respective companies.

Performance is an Internal Throughput Rate (ITR) ratio based on measurements and projections using standard IBM benchmarks in a controlled environment. The actual throughput that any user will experience will vary depending upon considerations such as the amount of multiprogramming in the user's job stream, the I/O configuration, the storage configuration, and the workload processed. Therefore, no assurance can be given that an individual user will achieve throughput improvements equivalent to the performance ratios stated here.
IBM hardware products are manufactured Sync new parts, or new and serviceable used parts. Regardless, our warranty terms apply.
All customer examples cited or described in this presentation are presented as illustrations of the manner in which some customers have used IBM products and the results they may have achieved. Actual environmental costs and performance characteristics will vary depending on individual customer configurations and conditions.
All statements regarding IBM's future direction and intent are subject to change or withdrawal without notice, and represent goals and objectives only.
Information about non-IBM products is obtained Sync the manufacturers of those products or their published announcements. IBM has not tested those products and cannot confirm the performance, compatibility, or any other claims related to non-IBM products. Questions on the capabilities of non-IBM products should be addressed to the suppliers of those products.
Prices subject to change without notice. Contact your IBM representative or Business Partner for the most current pricing in your geography.

Outline

What is a Mainframe?

Hardware Built into Modern Mainframes

Operating an IBM Mainframe

The s390x Architecture in Linux

What is a Mainframe?

A Try at a Definition

Although the term mainframe first described the physical characteristics of early systems, today it can best be used to describe a ***style*** *of operation, applications, and operating system facilities.*

...

So, let's return to our question now: "What is a mainframe?" Today, the term mainframe can best be used to describe a style of operation, applications, and operating system facilities. To start with a working definition, ***a mainframe is what businesses use to host the commercial databases, transaction servers, and applications that require a greater degree of security and availability than is commonly found on smaller-scale machines.*** (Mainframe Concepts [5])

How are Mainframes Used Today?

Being able to run general purpose operating systems like GNU/Linux enables IBM Z Mainframes to run virtually any workload available on other platforms running GNU/Linux.

However, usually they are deployed for some specific centralized tasks in the operations of an organization, such as:

- hosting the central System of Records (Databases such as Db2),
- running central Transaction Software (such as CICS TS, or z/TPF) and Batch Processing,
- Analytics Software complementing the System of Records,
- or hosting business critical application-servers (Websphere, ...).

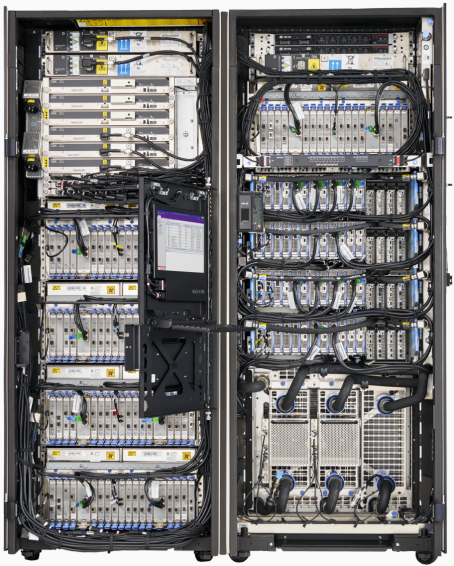
Hardware Built into Modern Mainframes

IBM z14 (M/T 3906)

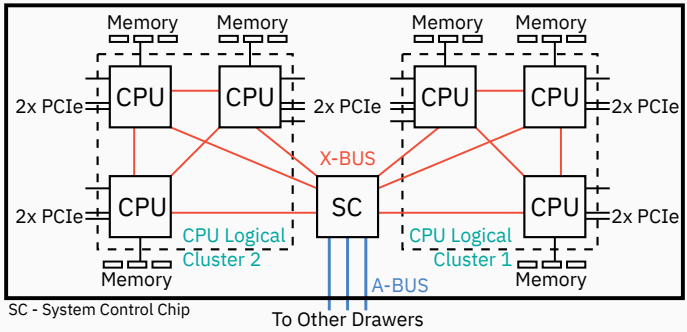
- Announced 09/2017; Additional functions and features 12/2018 [23, 11]
- 5 Models: M01 – M05
- Up to 170 Cores [17]
  - 2-Way SMT per Core
  - Plus up to 23 assist (SAP) and 2 spare Cores
  - Can all be used in one partition (see slide 10)
  - CPUs are clocked at 5.2 GHz
- Up to 32 TB Memory (RAIM)
  - At most 16 TB per partition

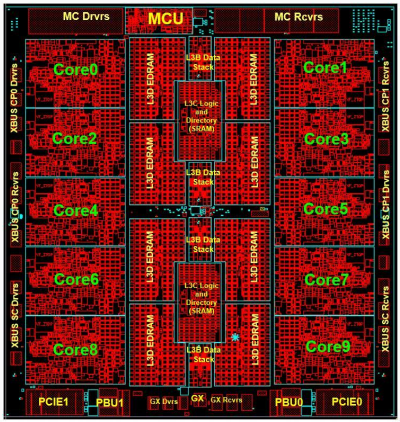
- 40 PCIe Gen3 Fanouts (16 GBps) from CPC- to I/O-Drawer
- Up to 160 I/O cards [24]
- FICON Express cards for FICON or FCP I/O
- OSA-Express cards for Ethernet
- RoCE cards for RDMA and Ethernet
- Crypto Express as Accelerator and HSM (FIPS 140-2 Level 4 certified)
  - Fast encryption with secure key in HSM and derived protected key in CPU-Subsystem CPACF [15]
- NVMe Adapters for added local disk space

IBM z14 Radiator-Based Air Cooled — Front View (M04 or M05)



Inside a IBM z14 CPC Drawer

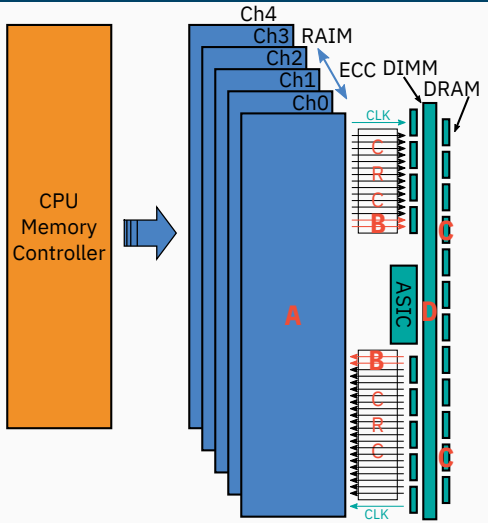




- Each Core has 128+128KB I+D L1 Cache, and 2+4MB I+D L2 Cache
- Each CPU has 128MB L3 Cache
- Each CPC Drawer has 672MB L4 Cache on SC Chip

- All Caches are Coherent
- All Memory is SMP Interconnected

5-Channel RAIM = Redundant Array of Independent Memory



- One Memory Controller per Processor, five Memory Channels per Controller, one DIMM per Channel
- Fifth Channel enables the RAIM “RAID” (20% of DIMM capacity is redundancy) [19]
- Reads and Writes are checked using CRC

A. One Channel can be marked as defective, guaranteeing 100% correction

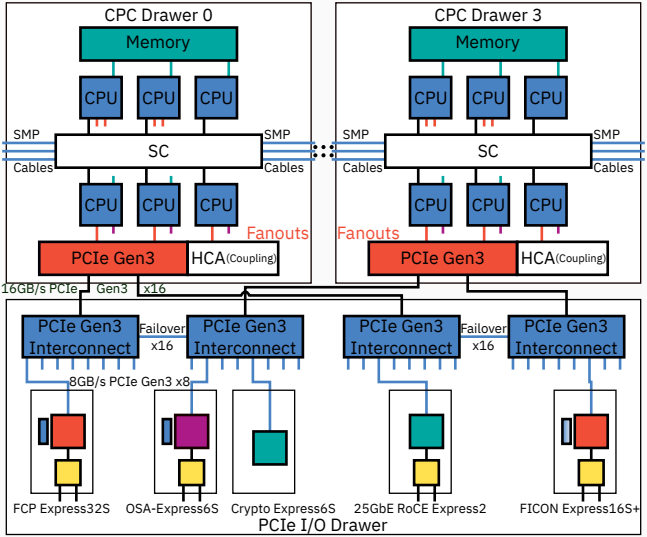
B. Two Lanes Up- and Down-Stream can fail per Channel

C. Two DRAM Chips per DIMM can be marked as failed

D. One DIMM can be marked as failed (no capacity reduction with single DIMM failure)

IBM z14 I/O Infrastructure

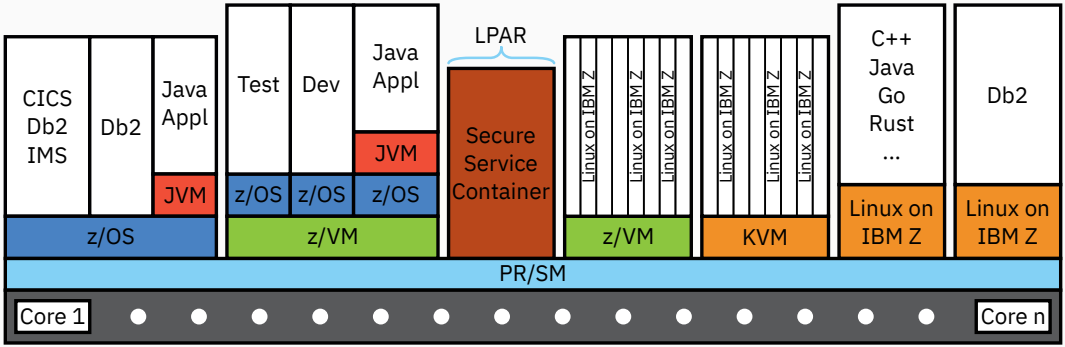
- Although PCIe is used as technology, OS does not see PCIe bus directly
- Native I/O Architecture is realized via Channel Subsystem [8, 24]
- Without special enclosure no native PCIe Adapters — recent HW generation allow PCIe Adapters to be accessed via Channel Subsystem [4]
- Disk storage typically via a SAN (use FCP, it has the best Linux kernel-driver ever! ☺)
- Until very recently no local persistent Disk Storage — recent IBM Emperor II added IBM Adapter for NVMe



Operating an IBM Mainframe

Splitting the Resources using Virtualization

- A IBM Z Mainframe is usually not used as one big SMP system
- The whole machine is split into multiple LPARs by the PR/SM hypervisor (EAL 5 certified), and can be split further using the second level hypervisors z/VM and KVM



Splitting the Resources using Virtualization

First level Hypervisor (PR/SM):

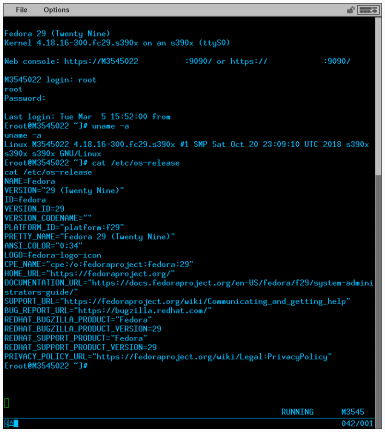
- Processors are either time-shared across or dedicated to LPARs [10]
- Memory is dedicated to LPARs, no over-provisioning with PR/SM
- I/O adapters are shared via the Channel Subsystem across LPARs (akin to Virtual Function pass-through)

Second level Hypervisor (z/VM, KVM):

- z/VM and KVM can both be used to further virtualize the resources of the underlying LPAR
- z/VM has better support for dedicated I/O devices and supports more different Operating Systems than KVM (z/OS, z/VSE, z/TPF, z/VM, and Linux on Z)
- KVM can handle more resources in a single LPAR than z/VM and offers typical open source interfaces

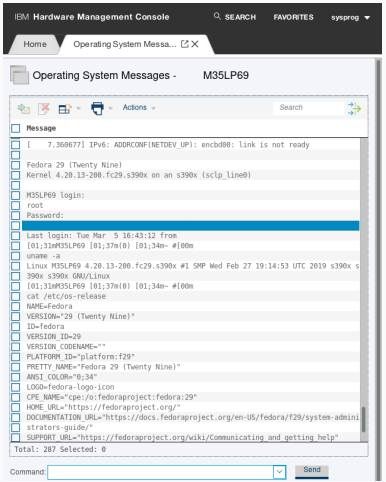
Running Linux

- GNU/Linux can be used as Operating System in LPAR, as well as in z/VM or KVM guests [14, 9]
- There are several Distributions that have support for IBM Z (cursively printed are supported in cooperation with IBM):
  - *SUSE Linux Enterprise Server*,
  - *Red Hat Enterprise Linux*,
  - *Ubuntu*,
  - Fedora,
  - Debian,
  - … .
- Most of your daily Linux experience is not different on IBM Z, than on your x86_64 Laptop or Server, but …



x3270, showing Fedora 29 as z/VM guest

Most Noticeable Differences with Running Linux on your Laptop



HMC, showing Fedora 29 as PR/SM guest

- Getting the Distro-Installer to start on PR/SM or z/VM is different than on PC - not so different from other headless network installations
- The default Bus-System is not PCI
- There is no local disk storage per default
- Channel Subsystem I/O devices (such as disks or virtual adapters) are turned off by default
- In general, I/O device configuration is quite a bit different, but we have some of the best Device-Driver documentation *I* have seen for Linux so far: [9]
- Like with other non-x86_64 architectures, Software not bundled with the supported distributions might need some care in order to work

A bit of history:

- Namespaces as part of the common code in the Linux Kernel have been working for a long time
- Container runtimes, and especially container images distributed in binary form used to be a different story
  - Also in part because of missing compiler-support (e.g. Go)
- This changed over the last 3∼4 years

Now:

- Major runtimes, such as Docker [12, 20] or LXC support IBM Z
- Orchestration software like Kubernetes [3] or Docker Swarm are also supported
- Major container images (ubtunu, alpine Linux, fedora, PostgreSQL, ...) are redily available; but not every container image works

The s390x Architecture in Linux

z/Architecture in Linux

- Current instruction set architecture for IBM Z is: z/Architecture (introduced in late 2000) [21]
  - z/Architecture succeeds ESA/390 that was used on the IBM System/390 generation
- In the Linux Kernel and in Linux Distributions in general it is called: `s390x`
- The architecture is documented in the Principles of Operations [8], the (Linux) ABI as supplement to System V [16]

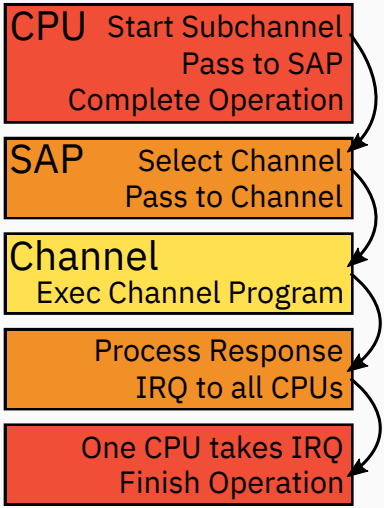
- Development for `s390x` in projects like the Linux Kernel, GCC, LLVM, and glibc is mostly done by IBM, and mostly in Böblingen
- In the Linux Kernel you can find the architecture specific code in `arch/s390` and `drivers/s390`
- Support was first published in late 1999 and merged into Linux 2.2.15 [18]

Addresses, Instructions and Memory

- z/Architecture is a 64-bit architecture (addressing, registers, instructions), 31-bit and 24-bit *addressing modes* are still supported
  - As of 4.1 the Linux Kernel only supports 64-bit mode
  - 31-bit Userspace applications can still run in compat mode
- Two privilege levels: Supervisor, and Problem state (Kernel- and User-Space)
- Words are arranged in Big-Endian fashion (no bi-endianness like on Power)
- Organization of memory (“Storage”) into staged page-tables not that far off from x86
  - Linux Kernel (4.13) supports traversal of all five page-table levels
  - Allowing address-spaces of up to 16EB (before 4.13 it used to be 8PB)
- But z/Architecture supports multiple address spaces (four)
  - Linux uses different spaces for User- (Primary) and Kernel-Space (Home)
  - Instructions may access multiple spaces at once
- Extra fun in Kernel: z/Architecture has information mapped at address `0`
- No support for MMIO on z/Architecture

Channel I/O

- “Classic” mainframe I/O mechanism (PoP [8] chapters 13 – 16)
- Evolves around Channel Programs that are executed asynchronous on different processors
- Channel Programs are made up from one or more I/O-Command Words
  - Command Words can be Writes, Reads, Control-Instructions, even Branches
  - Reads and Writes reference addresses in the memory of the issuing CPU (DMA)
- Linux Kernel uses Channel I/O for example to use disk storage (DASDs) provided via FICON adapters
- For other adapter types (for FCP or Ethernet) the kernel only uses some parts of Channel I/O (e.g. device discovery); otherwise programming-model is akin to similar PCIe-Adapters [2, 1]



Questions?

Headquarters Böblingen

- Big parts of the support for Linux on IBM Z are done at the IBM Laboratory in Böblingen
- We follow a strict upstream policy and do not — with scarce exceptions — ship code that is not accepted in the respective upstream project
- Parts of the hard- and firmware for the Mainframes are also done in Böblingen
- If you are interested, I can point you to some contacts in case you want to work in this environment
- https://www-05.ibm.com/de/entwicklung/about.html

Backup Slides

In Case Something goes Wrong (Kernel-/Memory-Dumps)

Kdump:

- Kdump works on s390x in the same way as on other architectures
- A portion of the memory is dedicated to a nested “dump-kernel” plus initrd
- Upon a kernel-panic kexec loads the dump-kernel and using the tools in its initrd, it saves a memory image on disk

Stand-alone dump:

- For cases where kdump fails or is not feasible, IBM Z offers Stand-alone tools [13]
- Can be used to dump to variety of targets (DASDs, SCSI Disks, Channel-attached tapes, ...)
- The dump-tool is prepared ahead of time so it can be booted (IPL’ed)
- To dump a guest the user instructs the firmware to stop all CPUs and to IPL the dump-tool
- The firmware loads the Dump-tool into the memory of the guest (securing the area overwritten), and the tool dumps the whole memory of the guest onto the prepared target

Glossary i

s390x Common name for z/Architecture in Open Source projects. 20, 26

CICS TS Customer Information Control System Transaction Server: mixed-language application servers that provide online transaction management and connectivity for mission-critical applications. 6

CPACF Central Processor Assist for Cryptographic Functions. 8

CPC Central Processor Complex: the piece of hardware that holds the processors. 8, 10

DASD Direct-Access Storage Device: disk storage type used by IBM Z via FICON. 22, 26

FCP Fibre Channel Protocol: Fibre Channel layer 4 protocol used to transfer SCSI commands. 8, 12, 22

FICON Fibre Connection: Fibre Channel layer 4 protocol used for Channel I/O. 8, 22

HMC Hardware Management Console: user interface for configuring, controlling, monitoring, and managing IBM Z hardware and software resources. 17

IPL Initial Program Load: load an operating system into a guest (similarly used as “booting the system”). 26

LPAR Logical Partition: one virtual machine operated by PR/SM. 14–16

Glossary ii

PoP Principles of Operations: reference manual for z/Architecture [8]. 22

PR/SM Processor Resource/Systems Manager: Type-1 and Level-1 hypervisor always running on IBM Z. 14, 15, 17

RAIM Redundant Array of Independent Memory: see slide 8. 8, 11

SAP System Assist Processor: Processor Units/Cores used by the I/O subsystem. 8

SC System Control Chip: chip housing L4 Cache and offering CPU- and CPC-Drawer-Interconnection. 10

References i

[1] I. Adlung, G. Banzhaf, W. Eckert, G. Kuch, S. Mueller, and C. Raisch.
Fcp for the ibm eserver zseries systems: Access to distributed storage.
IBM Journal of Research and Development, 46(4.5):487–502, July 2002.
doi:10.1147/rd.464.0487.

[2] M. E. Baskey, M. Eder, D. A. Elko, B. H. Ratcliff, and D. W. Schmidt.
zseries features for optimized sockets-based messaging: Hipersockets and osa-express.
IBM Journal of Research and Development, 46(4.5):475–485, July 2002.
doi:10.1147/rd.464.0475.

[3] A. Frosi.
Deploy an application in a kubernetes cluster on linux on ibm z.
https://containersonibmz.com/2018/10/26/kubernetes-deployment/, Oct. 2018.

[4] T. A. Gregg, D. Craddock, D. J. Stigliani, F. E. Bosco, E. E. Cruz, M. F. Scanlon, P. Sciuto, G. Bayer, M. Jung, and C. Raisch.
Overview of ibm zenterprise 196 i/o subsystem with focus on new pci express infrastructure.
IBM Journal of Research and Development, 56(1.2):8:1–8:14, Jan 2012.
doi:10.1147/JRD.2011.2178278.

[5] International Business Machines.
***What is a mainframe?*, chapter 1, pages 4–6.**
z/OS Basic Skills Information Center. IBM, 2008.
https://www.ibm.com/support/knowledgecenter/zosbasics/com.ibm.zos.zmainframe/zconc_whatismainframe.htm.

References ii

[6] Ibm z13 technology and design, July 2015.
https://ieeexplore.ieee.org/xpl/tocresult.jsp?isnumber=7175088&punumber=5288520.

[7] International Business Machines.
Assembling the ibm z mainframe in 120 seconds.
https://www.youtube.com/watch?v=RnpvyJaX4Q4, July 2017.

[8] International Business Machines.
***z/Architecture Principles of Operation*.**
IBM, twelfth edition, Sept. 2017.
https://www-05.ibm.com/e-business/linkweb/publications/servlet/pbi.wss?CTY=US&FNC=SRX&PBL=SA22-7832.

[9] International Business Machines.
***Device Drivers, Features, and Commands (Kernel 4.19)*.**
IBM, Dec. 2018.
https://www.ibm.com/developerworks/linux/linux390/documentation_dev.html.

[10] International Business Machines.
***IBM Z Processor Resource/Systems Manager Planning Guide*.**
IBM, Oct. 2018.
https://www-01.ibm.com/support/docview.wss?uid=isg27fe01e95029231c68525815d003fde1f.

[11] Ibm z14 design and technology, Mar. 2018.
https://ieeexplore.ieee.org/xpl/tocresult.jsp?isnumber=8353167&punumber=5288520.

References iii

[12] International Business Machines.
Running docker containers on the mainframe.
Oct. 2018.
https://www-05.ibm.com/e-business/linkweb/publications/servlet/pbi.wss?CTY=US&FNC=SRX&PBL=SC34-2781.

[13] International Business Machines.
Using the Dump Tools (Kernel 4.16).
IBM, July 2018.
https://www.ibm.com/developerworks/linux/linux390/documentation_dev.html.

[14] International Business Machines.
developerworks: Linux on z and linuxone.
https://www.ibm.com/developerworks/linux/linux390/index.html, 2019.

[15] International Business Machines.
Linux on z and linuxone: Pervasive encryption for data volumes.
https://www.youtube.com/watch?v=jDK3ZwEdX4I, Mar. 2019.

[16] International Business Machines and Linux Foundation.
Linux for zseries elf abi supplement.
https://refspecs.linuxfoundation.org/ELF/zSeries/index.html, Nov. 2002.

References iv

[17] C. Jacobi.
The ibm z14 microprocessor chip set.
https://www.hotchips.org/wp-content/uploads/hc_archives/hc29/HC29.22-Tuesday-Pub/HC29.22.90-Server-Pub/HC29.22.910-Z14-processor-Jacobi-IBM.pdf, Aug. 2017.

[18] Linux/390 - notes and observations.
https://web.archive.org/web/20180831222032/http://linuxvm.org/penguinvm/notes.html.

[19] P. J. Meaney, L. A. Lastras-Montano, V. K. Papazova, E. Stephens, J. S. Johnson, L. C. Alves, J. A. O'Connor, and W. J. Clarke.
Ibm zenterprise redundant array of independent memory subsystem.
IBM Journal of Research and Development, 56(1.2):4:1–4:11, Jan 2012.
doi:10.1147/JRD.2011.2177106.

[20] L. Parziale, E. S. Franco, R. Green, E. E. M. Marins, M. Roveri, and N. C. D. Santos.
Getting Started with Docker Enterprise Edition on IBM Z.
Redbooks. IBM, Mar. 2019.
https://www.redbooks.ibm.com/abstracts/sg248429.html.

[21] K. E. Plambeck, W. Eckert, R. R. Rogers, and C. F. Webb.
Development and attributes of z/architecture.
IBM Journal of Research and Development, 46(4.5):367–379, July 2002.
doi:10.1147/rd.464.0367.

References v

[22] Slacktory.
An ode to movie mainframes.
https://www.youtube.com/watch?v=Hcywf9mwF5U, Mar. 2013.

[23] B. White, H. Kamga, M. Kordyzon, O. Lascu, F. Packheiser, J. Troy, E. Ufacik, and B. Xu.
***IBM z14 Technical Introduction*.**
Redbooks. IBM, second edition, Oct. 2018.
https://www.redbooks.ibm.com/abstracts/sg248450.html.

[24] B. White, H. Kamga, M. Kordyzon, F. Packheiser, J. Troy, E. Ufacik, B. Xu, and O. Lascu.
***IBM Z Connectivity Handbook*.**
Redbooks. IBM, twentieth edition, Oct. 2018.
https://www.redbooks.ibm.com/abstracts/sg245444.html.

[25] Wikipedia.
Ibm system/360.
https://en.wikipedia.org/w/index.php?title=IBM_System/360&oldid=885224396, Feb. 2019.