International Journal for Multidisciplinary Research (IJFMR)
E-ISSN: 2582-2160 ● Website: www.ijfmr.com ● Email: editor@ijfmr.com

IBM Mainframe & Z/OS: Advanced Insights from A Programmer's Perspective

Chandra Mouli Yalamanchili

chandu85@gmail.com

Abstract

Despite all the advancements in distributed technology, IBM mainframes still host a prominent percentage of transaction processing across different market segments, such as healthcare, airlines, financial, stock trading, etc., making IBM mainframe attractive from a prominent technology perspective for developers to adopt.

This paper explores and discusses the architecture of the IBM mainframe from a programmer's perspective to understand critical hardware and software components that support applications running on the mainframe.

This paper also explores some of the address spaces, such as coupling facility, TCP/IP stack, z/OS communication server, CICS TS, etc., that are critical from an application programming perspective.

This paper also examines the architectural similarities between virtual machines (VMs) and IBM mainframes and the historical progression of IBM mainframes, specifically z/OS.

Finally, this paper examines and discusses a simple COBOL CICS code example from the z/OS architecture perspective.

Author Keywords

IBM Mainframe; z/OS; Mainframe address space; CICS; Coupling facility; Mainframe programming; z/OS Communication Server.

INTRODUCTION

While other forms of computing have evolved over the years, the IBM mainframe still plays a vital role in the operation of several Fortune 1000 companies. It continues to be the core component for many enterprises in banking, finance, health care, insurance, utilities, government, and other business sectors.[1] Several factors make mainframes one form of computing that has been successful for many decades and continues to be successful. Below are a few of these factors:

- **Compatibility:** Despite the continuous evolution of the mainframe systems' hardware and software, the progression of the mainframe architecture has ensured that applications written in older versions are always supported without any issues.[1]
- **Reliability:** Most mainframe hardware and software components have self-checking and self-healing capabilities, providing a highly reliable platform for business applications.[1]
- **Availability:** Mainframe ssystems are built with high resilience at different hardware and software layers, making it possible to recover the system without impacting the applications.[1]
- **Serviceability:** Once again, the mainframe system's highly resilient design allows for seamless hardware or software upgrades for the business applications.[1]

- **Security:** Mainframe systems have robust and sophisticated security tools that provide users with role-based authority to protect critical systems and client data.[1]

In addition to these factors, the tight coupling between hardware and software components while maintaining resiliency at component level provides high processing capabilities and better system control. The combination of various critical components within mainframe architecture makes them successful systems, and this paper will explore some of these components of mainframe and z/OS from the programmer's perspective.

Mainframe Operating Systems

While this paper focuses on the z/OS operating system, IBM mainframes offer several other operating systems; below are the high-level details about different operating system options available within IBM mainframe.

- **z/OS** is the widely used operating system for mainframes and provides a stable, secure, and highly resilient environment for running applications.[1]
- **z/Virtual Machine** (z/VM®) is a hypervisor that creates a virtual machine that can host any other mainframe OS like distributed virtual machines.[1]
- **z/Virtual Storage Extended** (z/VSE) is a lighter version of z/OS that provides a smaller and less complex OS for batch and transaction processing.[1]
- **Linux for System z** allows different Linux distributions to run on a mainframe. Running Linux OS on a mainframe provides the benefits of high processing capability, being close to existing mainframe applications, and communicating with them using HiperSockets.[1]
- **z/TPF** is a special-purpose operating system for companies that process high transaction volumes, such as airline reservation systems and credit card companies.[1]

Below figure depicts the mainframe timeline starting from S/360 in 1964 till z/OS in 2000s.

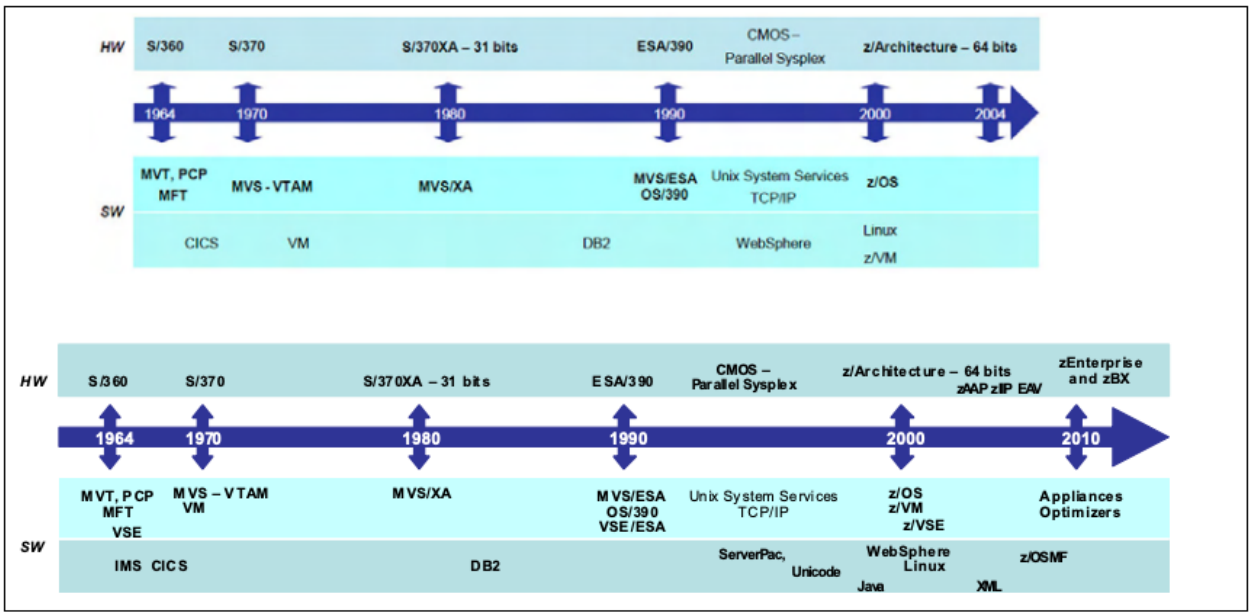

Figure 1: IBM Mainframe timeline[1]

IBM hardware Components

While the mainframe initially (back in the 1960s) had a single processor known as the central processing unit (CPU), today's mainframe hardware comes as an interconnected collection of hardware components

called a central processor complex (CPC) that includes main storage, central processors, timers, and channels.[1]

Below are some of the critical hardware components of the mainframe:

Central Processors: The Central processor contains the processors, memory, control circuits, and channel interfaces.[1]

Channels: Channels provide an independent data and control path between I/O devices and memory. Modern mainframes use ESCON (Enterprise Systems Connection) and FICON (Fiber Connection) Channels that connect to only one control unit or to a director that, in turn, connects to several control units. Channels are generally known as CHPIDs (Channel path identifiers) or PCHIDs (Physical channel identifiers).[1]

Control Units: Control units connect to I/O devices. They contain logic specific to the type of I/O device attached to them.[1]

Devices: Devices account for I/O devices like disk drives, tape drives, printers, communication interfaces, etc..[1]

LPAR: The mainframe can be partitioned into separate logical systems by sharing the resources available among these independent logical partitions (LPARs). The LPAR hypervisor, a standard Processor Resource/System Manager (PR/SM) feature on all mainframes, controls the sharing of resources among different LPARs. Each LPAR will operate like its own system and can host its own operating system.[1]

The system profile and IOCDS (I/O Configuration Data Set) hold the partitioning control specifications together. The IOCDS and profile reside inside the system's Support Element (SE) computer. Hardware Management Consoles (HMCs) are personal desktop computers that connect to SE to monitor and control mainframe microprocessors.[1]

Below is a simple depiction of how different LPARs share resources in the mainframe.

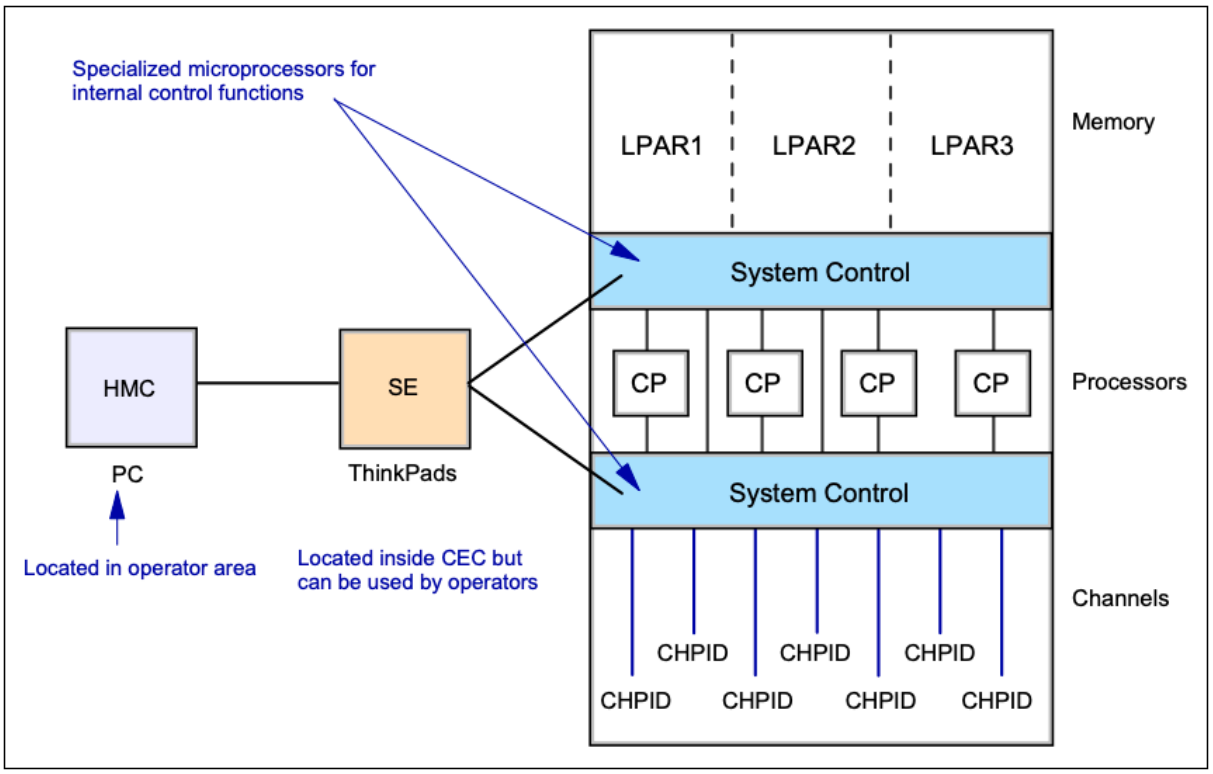

Figure 2: Depiction of LPAR and system control[1]

z/Architecture Processors

Below are several z/Architecture processors that are available to solve different purposes.

Central Processor (CP): This generic processor is available for general OS and application code.[1]

Integrated Facility for Linux (IFL): LPAR hosting Linux OS uses this processor. IFL is like CP but with a different cost model. Using IFL would lower the cost of the Linux workload on the mainframe.[1]

z Application Assist Processor (zAAP): This is like IFL but has several functions disabled, so it cannot execute full OS. However, if available, z/OS uses zAAP processors for the Java workload to reduce the software cost.[1]

z Integrated Information Processor (zIIP): Initially introduced as a specialized engine for processing eligible database workloads, but later repurposed for Java and other zIIP-eligible workloads.[1]

Below is an example of different LPARs hosting different operating systems and the respective types of processors available in z architecture.

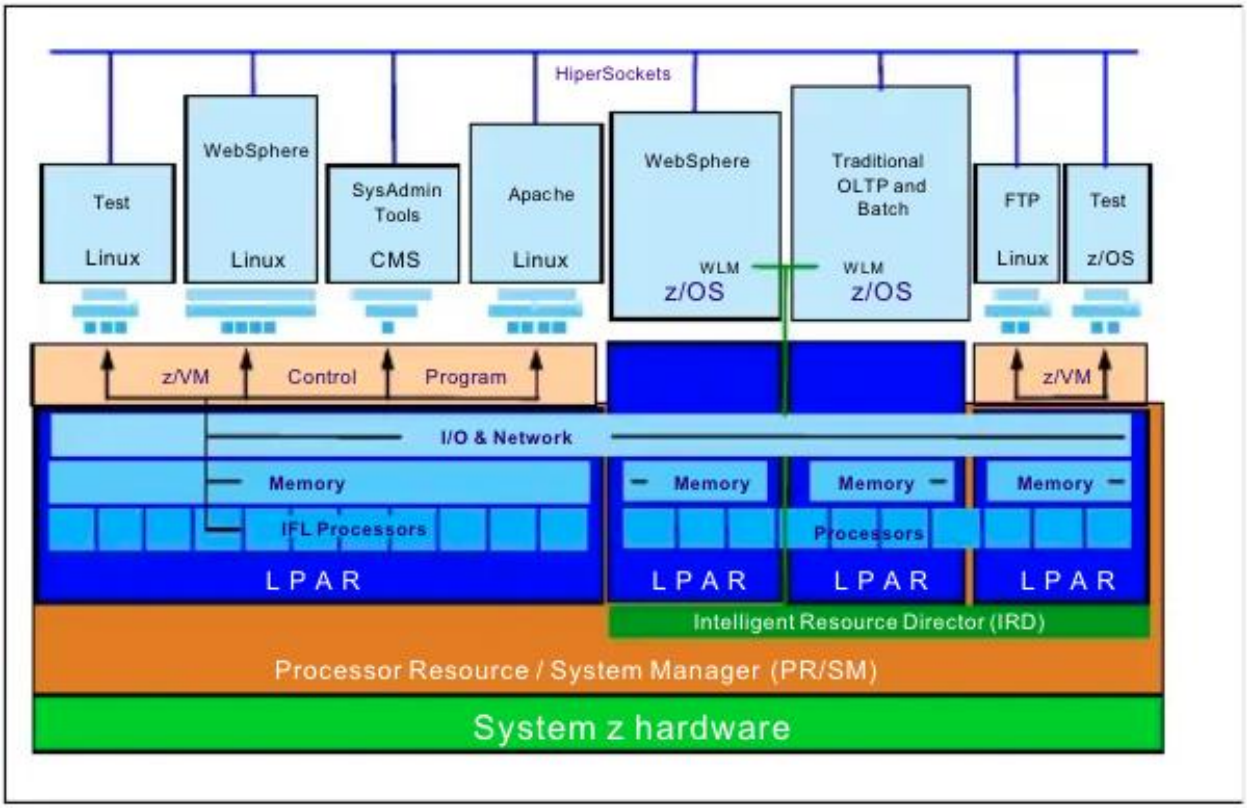

Figure 3: Depicting different LPARs with different Operating Systems and different processors [2]

Critical z/OS Components from developer perspective

MVS (Multiple Virtual Storage): Predecessor to z/OS and still a foundational component of modern z/OS. It manages critical functions of OS like hardware resources, memory, tasks, and users in a multi-programming environment.[3]

z/OS Communications Server: The z/OS Communication Server enables TCP/IP and SNA (Systems Network Architecture) communications. It is critical for network-based mainframe applications, enabling various client/server applications to communicate with the mainframe.[3]

Coupling Facility: The Coupling Facility is used primarily in Parallel Sysplex environments to ensure \ high resource availability. It enables multiple coupling-capable z/OS processors to share data and

workload in real-time by providing locking, caching, and list services. VSAM record-level sharing (RLS) is one such benefit of the coupling facility, allowing multiple systems to access the same VSAM file simultaneously.[3]

DFSORT: DFSORT has been a standard tool on the mainframe to achieve fast and easy sorting, merging, copying, reporting, and analysis of business information.[3]

Distributed File Service: The zSeries File System (zFS) is a UNIX file system that contains files and directories. Like HFS (Hierarchical File System), it can be mounted into the z/OS UNIX hierarchy.[3]

HLASM (High-Level Assembler): HLASM is an integral part of z/OS that extends the basic assembler language to provide better development capabilities for building assembler language applications that run on z/OS.[3]

ISPF (Interactive System Productivity Facility): ISPF is a menu-driven interface for dataset management, application development, and system management on IBM mainframes.[3]

JES2/JES3 (Job Entry Subsystem 2/3): JES2 and JES3 are both responsible for managing batch jobs and output data by managing job scheduling and spooling. They both accept the submission of work to BCP (Base Control Program). JES2 exercises independent control over job processing, while JES3 exercises centralized control.[3]

Language Environment: The Language Environment provides the runtime environment for programs generated with C, C++, COBOL, Fortran, and PL/I. [3]

RMF (Resource Measurement Facility): This facility gathers resource performance and usage parameters and provides reports on various systems in the complex.[3]

SDSF (System Display and Search Facility): SDSF is an integral tool for monitoring, managing, and controlling z/OS systems. From developers' perspectives, it will help monitor user jobs and check system logs.[3]

TSO (Time Sharing Option) and TSO/E (TSO/Extensions): TSO and TSO/E are critical components of z/OS that provide an interactive environment for users to communicate with the system, execute programs, and perform various development tasks.[3]

z/OS UNIX: z/OS UNIX System Services (z/OS UNIX) provides the standard command interface familiar to interactive UNIX users.[3]

SMS (System Management Services): The System Management Services automates the data storage management. It handles functions like dataset allocation, volume selection, and data migration to ensure efficient data usage and storage.[3]

RACF (Resource Access Control Facility) and Security: RACF is the primary security framework for z/OS. It controls the user's access to all resources on the mainframe, such as datasets, programs, and system facilities, to ensure that only authorized users can access sensitive system parts.[3]

Simple Code Example

Below is a basic example of a COBOL program that handles a simple CICS (Customer Information Control System) transaction:

```
EXEC CICS RECEIVE INTO(INPUT-DATA)
LENGTH(INPUT-LENGTH)
END-EXEC.
MOVE 'Hello, CICS!' TO OUTPUT-DATA.
EXEC CICS SEND FROM(OUTPUT-DATA)
LENGTH(OUTPUT-LENGTH) END-EXEC.
```

This program demonstrates the fundamentals of writing a CICS transaction. These commands receive the user input and send back a simple greeting as a response to the user. CICS address spaces facilitate the interaction between the user and the program. Behind the scenes, several other address spaces come into play, like the Communication Server, which allows the user to communicate with the CICS transaction server through the 3270 terminals, and RACF, which allows the user to access the respective CICS transaction.

Conclusion

IBM mainframes, specifically z/OS, remain a powerful platform for enterprises. IBM Mainframe provides a perfect combination of hardware and software components to ensure high performance and reliability. z/OS offers several sophisticated tools for programmers, such as managing address spaces and utilizing system-critical components like Communication Sever and Coupling Facility. Most businesses rely on mainframe technology, making understanding z/OS architecture crucial for modern application development.

References

1. IBM Corporation, Introduction to the New Mainframe: z/OS Basics. IBM Redbooks, 2012. https://www.redbooks.ibm.com/redbooks/pdfs/sg246366.pdf
2. IBM Corporation, Practical Migration to Linux on System z. IBM Redbooks, 2010. https://www.redbooks.ibm.com/redbooks/pdfs/sg247727.pdf
3. IBM Corporation, z/OS Planning for Installation Version 2 Release 1. IBM documentation 2014. https://publibz.boulder.ibm.com/epubs/pdf/e0z3b102.pdf