

# z/OS Masterclass

Alexander Brown

June 25, 2012

## 0.1 The New Mainframe

Run as much as possible, as long as possible without disrupting service.

100% CPU all of the time.

Run multiple, but isolated, operating systems concurrently.

Optimized for I/O.

96 Processors (spares, OS and Hardware specific), 3TB memory.

RAS – Reliability, Availability, Serviceability.

z/OS – zero (downtime)/OS.

Software as reliable as customers expect on zOS

Shared Everywhere - Centralised Control.

Goal mode processing - transaction needs to finished within *xms* - WLM (processor manager).

### 0.1.1 Batch Job versus Real Time Transactions

Batch – Input data -> Application Program -> Output.

Real-time – Query -> Application Program -> Reply.

SNA – System Network Architecture (before TCP/IP).

VTAM – Virtual Telecommunications Application ...

### 0.1.2 Error codes

IWASD - Information, Warning, Application, ?, Disaster.

...DFH – CICS

### 0.1.3 Mainframe OSes

**z/OS (MVS)**

Trimodal - switch between 24, 31 and 64bit.

**z/VSE Virtual Storage Extended**

Cut down version of z/OS

JCL different.

**z/TPF Transaction Processing Factory**

Around 50 customers use this. Used to be aircraft control system. Tuned to be ultra-fast.

**z/VM**

Hypervisor.

Two types:

z/BM – z/OS, etc.

PRISM - Schedule processors and h/w

Two basic components:  
Control Program (CP).  
Conversational Monitor System (CMS), a single-user OS.  
CP creates multiple VMs from real h/w resources.  
Appears as if each VM has dedicated use of shared resources.

### **zLinux**

Uses ASCII not EBCDIC.

## **0.2 Hardware Software and LPARs (Logical Partitions)**

Devices addressed by device, but today is virtualized.

FIO control layer uses a control file IOCDS that translates physical IO addresses into device numbers.

ESCON and FICON switch from CP to peripheral devices.

Sysplex are groups of LPARs.

CF (coupling Facility) is a high speed memory area which Sysplexes can share.

Data sets: Lists, Locks and Cache

Share CPUs over LPARs if wanted.

LPARs are an images of an OS, try and share as much as possible.

Controlled mostly via HNC.

PR/SM - Hypervisor.

Up to 60(+?) LPARs.

can have a native LPAR on z/OS.

LPARs are interpendant of each other but resources are shared.

Can run different versions.

### **0.2.1 Processors**

General Central Processor (CP): Standards applications and workloads.

System Assist Processor (SAP): Schedule I/O Operations.

Integrated Facility for Linux (IFL): Used exclusively by a Linux LPAR/Linux on VM.

zOS Application Assist Processor (zAAP): Provides for Java and XML workload offload.

zOS Integrated Information Processor (zIIP): Used to optimize certain database workload functions and XML processing.

Integrated Coupling Facility (ICF): Used exclusively by the Coupling Facility Control Code (CFCC) providing resource and data sharing.

Spares: Used in the event of a processor failure.

## **On Demand**

CBU – Capacity Back Up

CUoD – On/Off Capacity Upgrade on Demand

SubCapacity Licensing Charges.

### **0.2.2 Clustering**

Has been done for several years. Basic sharing don't using DASD.

CTC (channel to channel)/GRS (Global Resource S..) rings. Exclusive Locks and Memory Locks handled by a GRS.

RACF – User permissions and password controller.

## **0.3 Parallel Sysplex**

Tightly coupled LPARs, co-operating.

Coupling Facility and CFCC allows data to be shared between LPARs quickly.

Up to 32 LPARs (8 bytes).

Backwards compatible, so long as no new features are used.

Clock shared. z/OS has an instruction to return the Clock. Guaranteed unique clock.

8 byte clock, now 12 bytes, but not all products use the clock format.

### **0.3.1 Exploiters of the CF**

Authorized applications (CICS, MQ).

IMS DB

DB2

VSAM

## **CICS**

CICS-Sysplex (CICSplex).

CICS connectivity can be outside of a sysplex.

## **0.4 z/OS - An “Overview”**

### **0.4.1 What is z/OS?**

64-bit OS, most widely used on mainframes.

Ideally suited to large workloads for concurrent users. I/O intensive computing.

Architecture - z/Architecture.

Easily back out of changes.

## 0.4.2 Overview of z/OS internals

Comprises modules, system programs (macros), system components.

Uses a program status word (PSW) = A way to do jumps (either within the same context, or to a different context) easily very easily. Jumps load in a PSW (cleaner than normal).

Control block = A structure.

Manages 3 kinds of storage (memory):

Real storage or Central Storage (CSTOR): Actually memory

Auxiliary storage: copying out to storage

Virtual storage: what a program sees (allocation).

### Control Block

a.k.a.: A struct.

4 types:

System-related, Resource-related, Job-related, Task-related,

3 most commonly used:

TCB (Task Control Block) - A thread (similar to a POSIX thread). z/OS sees things in terms of address spaces, and within an address space you can have many TCBs.

Service request block (SRB) - “Special” thread, injects SRB into an address space to do a simple task.

Address Space Control Block (ASCB) - Defines an address space.

### Physical Storage

Two types of storage:

CSTOR located on the mainframe itself. Real memory.

Auxiliary storage which is external to the mainframe itself (disk drives, tapes, etc.). (Paging?).

CSTOR is accessed synchronously (direct).

Aux is accessed asynchronously (copied from disk to CSTOR or visa versa).

Virtual storage is an “illusion” created through the OS management of Real and Aux memory.

Running portions of a program are kept in real storage. Anything that can't fit in real storage is shipped to aux storage in page data sets (data sets are effectively files).

Users and programs are allocated an address space.

### Synchronous Cross Memory

This is a way that it can design things to make things really clean and fast and clever. Currently only z/OS.

Allows one instruction refer to multiple address spaces. Rather than having to call the OS to go and do something, there are some low level instructions which allow you to do this stuff.

- Allows data to be moved between address spaces.
- Allows invocation of programs from different address spaces.
- Allows instructions to access data from a different address space.
- Used a lot by CICS.

## **Dynamic Address Translation**

i.e.: Memory Management Unit

## **Virtual Storage**

1mb segments composed of 4kb pages.

- Paging: transfer of pages between aux and real storage.

Accessing memory that doesn't exist an interrupt (called a Page Fault) is signaled and the OS notifies. If it should exist, get the page from aux and fiddle the pointer.

- A 4k chunk of real memory is a frame.

- A 4k chunk of disk space is a slot.

- Virtual storage pages are contiguous with real storage pages.

## **Address Spaces**

\*NIX - Fork a process, exactly the same as the original, reading comes from the original until writing. Anything else is done through system calls.

- z/OS sets up address spaces with lots of information.

## **Page Steal**

Memory is expensive, disk is cheap. But the difference in speed between memory and disk is huge (difference between cache and RAM is large too).

1GHz is 1 CPU cycle per nanosecond (assuming all data is in cache and level 1 registers. Getting from main memory is very expensive. Disk is about 1ms).

- When supply of real storage becomes low, z/OS uses page stealing to

## **Swapping**

Swapping has the effect of moving an entire address space into, or out of, real storage.

## **Storage Protection Keys**

Secures memory without Page tables.

- Only allows address spaces to access certain memory based on these keys.

### **Brief History of zOS addressability**

64-bit is current standard, however zOS has good backwards compatability.

Originally z/OS defined storage addresses as 24-bits (16mb of virtual storage). This is the “line”.

In 1983 this became 31-bits (2gb of virtual storage). This caused a problem as some people used the 8 bits of the 24-bit addresses for flags. The 32nd bit of the address is a flag to say if it's a 24-bit address or a 31-bit address. The “bar”

In 2000 this was upped to 64-bits.

### **64-bit address space**

24-bit kept below the line. 32-bit kept below the bar.

## **0.4.3 Workload Management (WLM)**

WLM manages processing of workloads according to business goals, managing system resources to accomplish these goals.

Balances throughput and turnaround.

WLM has tentacles in all of the OS with coopiration from applications.

## **0.4.4 Serilizing the use of resources**

z/OS shares nicely.

Global Resource Serialization (GRS): Serilizes access to resources to protect thier integrity. Mostly used when resources are shared across systems (DLM (Distributed Lock Manager) on other OSes).

## **0.5 TSO/E, ISPF and UNIX: Interactive facilities of z/OS**

TSO - Time Sharing Option (equivalent of a shell, without scripting, etc.). Single user logon session.

ISPF - Text Menu interface to a lot of convinient commands.

REXX - Restructured Exended Executor.

## **0.6 Data Sets**

Place to store data.

More complicated than a file though.

### **0.6.1 What is a Data Set**

A way to store data on disks (Volumes).

### 0.6.2 Terminology

Data sets can be cataloged.

- e.g. MONKEY1 points to memory 0x001B

- A record is a fixed number of bytes containing data.

- A field is a specific portion of a record. e.g. a record is 8 bytes long, a-h. A field is a or b or c, etc.

- Sequential data is stored sequentially. It is typically a single file.

- A Partitioned Data Set (PDS) consists of a directory and members.

- Virtual Storage Access Method - defines the access method and the data set type.

### 0.6.3 How is data stored on zOS

Data is stored on a Direct Access Storage Device (DASD). It is effectively a HDD.

Records can be stored or retrieved directly (i.e. record 8) or sequentially (iterate through until you find 8).

DASD volumes store data and executable programs. Inc. OS

### 0.6.4 Data management

Allocation, placement, monitoring (ensuring it's intact), migration, backup, recall, recovery and deletion.

- Done manually or through automated processes.

- Data Facility: System-Managed Storage (DFSMS) is used to automate storage management.

### 0.6.5 Access Methods

Defines the technique to store and retrieve data.

- Not as simple as on PCs

### DASD

Disk Drives contain cylinders, cylinders contain tracks, tracks contain records.

### 0.6.6 Data Sets

To use: allocate first (DFSMS handles this), then you access the data using macros for the access method that you have chosen.

### Naming

Must be unique

- Max 44 characters.

- Maximum of 22 name segments:

- Left hand name is the High Level Qualifier (typically the username).



Everything else is the Low Level Qualifier.  
Qualifiers separated by '.'  
1-8 Characters for each level.  
First character must be A-Z,@,#,dollar  
Every other character can be A-Z,0-9,-  
Uppercase only  
UID.JCL.FILE2 (3 qualifiers)  
Member name of partitioned data set - Same rules as a qualifier.

### 0.6.7 Allocating space on DASD

Space is specified either explicitly (SPACE parameter) or implicitly (SMS data class).

Logical records and blocks are the smallest amounts of data that can be processed (grouped in physical records named blocks).

## 0.7 Application Design and Programming on z/OS

Application programs do things in terms of business, sitting on top of system programs, etc.

### 0.7.1 Batch versus Online

**Batch** Set of commands run with no or little user input. Usually less complex.

**Online** Someone at an ATM.

### 0.7.2 Data sources and Access methods

Fast access if the type of data is known.

## 0.8 Programming Language in z/OS

Binder - Update the link editor (linker).

IGYWCLG Compile and link and Go.

Re-entry Program - Loaded once and run across every single processor, concurrently. Different variable data locations.

### 0.8.1 Language Environment (LE)

**Processes** Highest level component of LE. Consists of at least one enclave.

Logically separate from other processes.

**Enclave** Collection of routines that makes up an application.

**Threads** An independent instance of a routine running under an enclave's resources.