

Virtualization Basics

Jacob Gagnon
CP development
z/VM Development Lab
Endicott, NY

```
                / VV                VVV MM                MM
                / VV                VVV  MMM                MMM
ZZZZZZ        / VV                VVV  MMMM               MMMM
      ZZ      / VV                VVV  MM MM MM MM
      ZZ      / VV  VVV          MM  MMM  MM
      ZZ      / VVVVV          MM  M  MM
      ZZ      / VVV          MM          MM
ZZZZZZ / V          MM          MM
```

built on IBM Virtualization Technology



Virtualization Basics

session ID: 25883

August 6th, 2019
Pittsburgh, PA

Jacob Gagnon
CP Development
z/VM Development Lab
Endicott, NY



Agenda:

Virtualization (As told from the view of z/VM)

- What is it
- A brief history
- How it's used today

What is a virtual machine?

Introduction to basic concepts

- Processor, Memory, I/O, etc...

Overcommitment of resources

How do you build z/VM?

Dynamic resource management

Monitoring your system

What happens when things go wrong?

```
          / VV          VVV MM          MM
         /  VV          VVV  MMM          MMM
ZZZZZZ /  VV          VVV   MMMM        MMMM
      ZZ /  VV          VVV   MM MM MM MM
      ZZ /  VV  VVV          MM  MMM  MM
      ZZ /  VVVVV          MM  M  MM
      ZZ /  VVV          MM          MM
ZZZZZZ /  V          MM          MM
```

built on IBM Virtualization Technology

Z Nomenclature

Level Set



Intel, System p, etc.	IBM Z
Memory	Storage (central storage)
Disk Storage	DASD (Direct Access Storage Device) FCP (Fibre Channel Protocol)
Computer Host System	CEC (Central Electronics Complex) CPC (Central Processor Complex) Server
Socket Slot	Book, drawer or node
Chip (e.g., multiple cores on a chip)	Core (e.g., multiple processors on a core)
CPU GPU Processor Core HT Core	Processor or Engine CPU or PU CP (Central Processor) IFL (Integrated Facility for Linux) zIIP (zSeries Integrated Information Processor) ICF (Integrated Coupling Feature) SAP (System Assist Processor) IOP (I/O Processor)
Boot	IPL (initial program load)

IBM Z and LinuxONE Servers currently in support for z/VM

IBM Z	IBM LinuxONE	z/VM Releases
zEnterprise 196 (2817) zEnterprise 114 (2818)		z/VM 6.4
zEnterprise EC12 (2827) zEnterprise BC12 (2828)		z/VM 6.4 & z/VM 7.1
z Systems z13 (2964) z Systems z13s (2965)	Emperor Rockhopper	z/VM 6.4 & z/VM 7.1
IBM z14 (3906) IBM z14 ZR1 (3907)	Emperor™ II Rockhopper™ II	z/VM 6.4 & z/VM 7.1

Why Virtualize?



Why Virtualize?

Virtualization is the creation of substitutes for real resources.



Benefits:

- **Consolidation** of physical resources reduces hardware cost
- **Optimization** of workloads that run near each other
- **IT flexibility** and responsiveness – make changes without a hardware outage!



Are we there yet?

1964

CP-40 development begun, this will be the predecessor of today's z/VM.



1972

In 1972, VM/370 is launched, the first in the VM product line from IBM.

1984

In 1984, Amdahl introduces MDF (multiple domain feature) to partition the real device into several separate machines. IBM releases LPAR technology in 1988

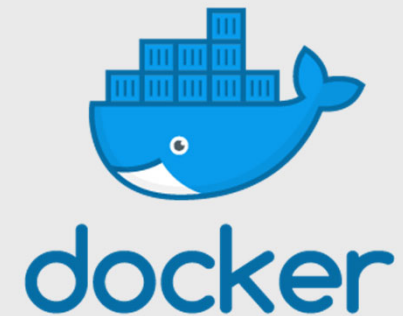


2001

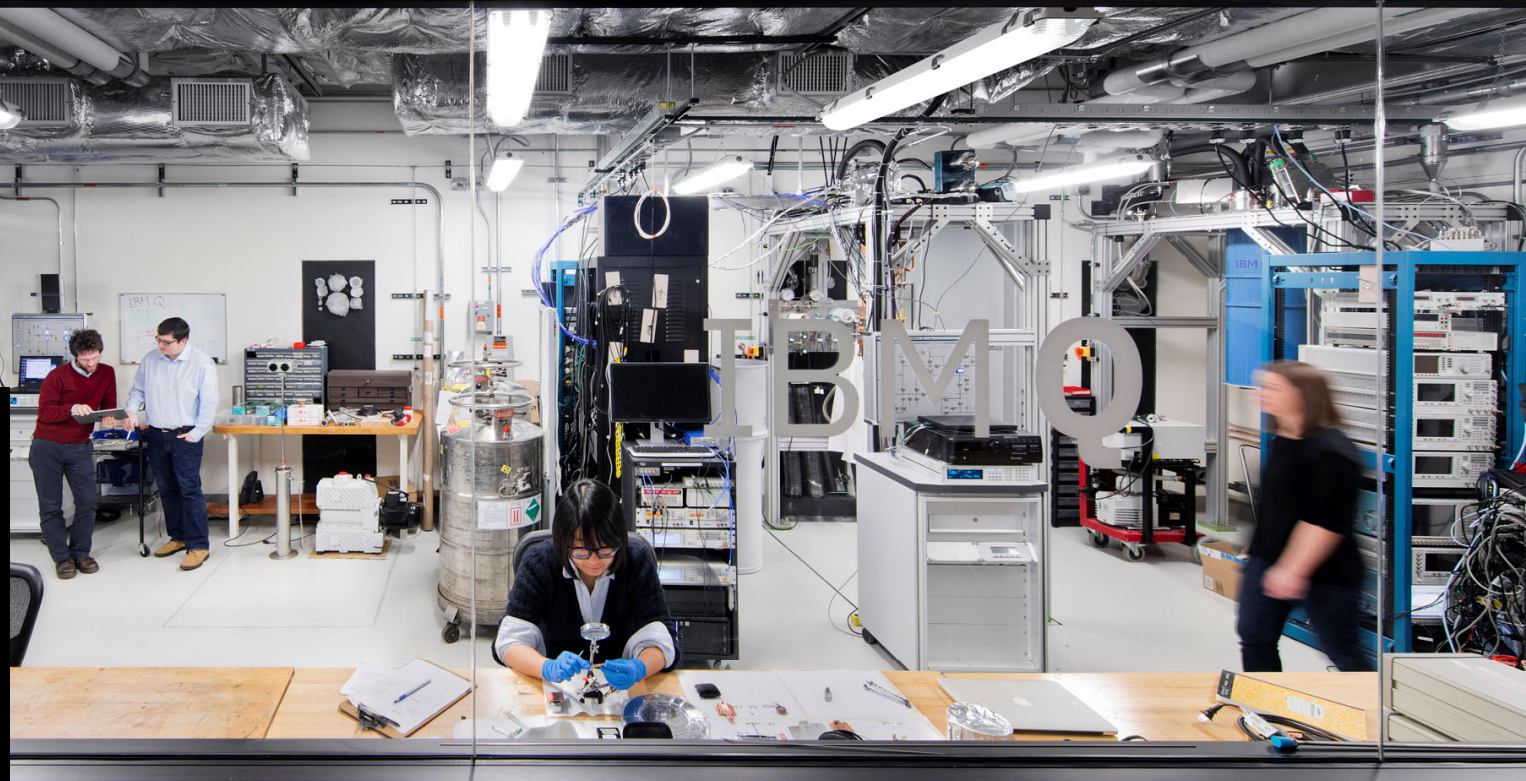
VMware ESX Server is launched, marking the start of a new wave of virtualization. z/VM, the latest version of the VM operating system, debuted in 2000.

2013

The first release of Docker, a containerization technology, released to open source.

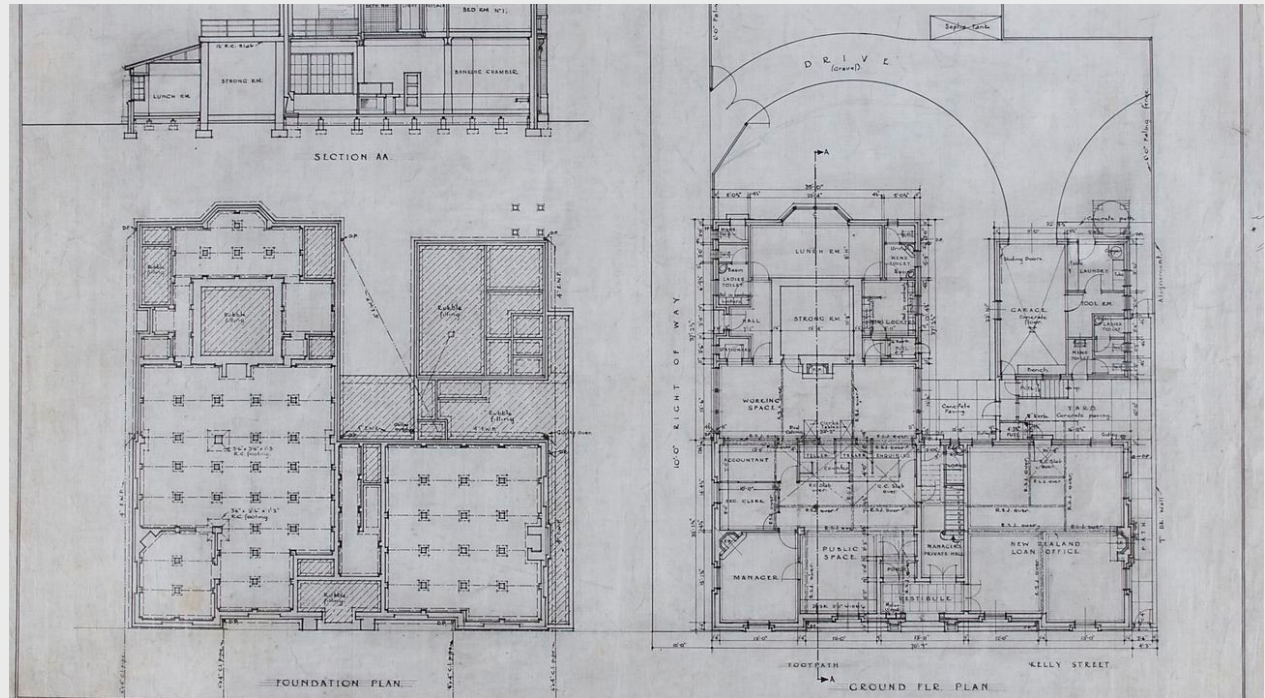


Businesses continue to prize flexibility, portability and openness as they develop new applications and delve into piles of data. The future is bright for virtualization.



What are we virtualizing? Z Architecture

- Every computer system has an **architecture**
 - It's the specs for how to build software on the system
 - Defined in z/Architecture Principles of Operation
 - z/VM must both **follow** and **implement** z/Architecture



What does virtualization mean on IBM Z?

z/VM allows the **sharing** of the mainframe's physical resources such as

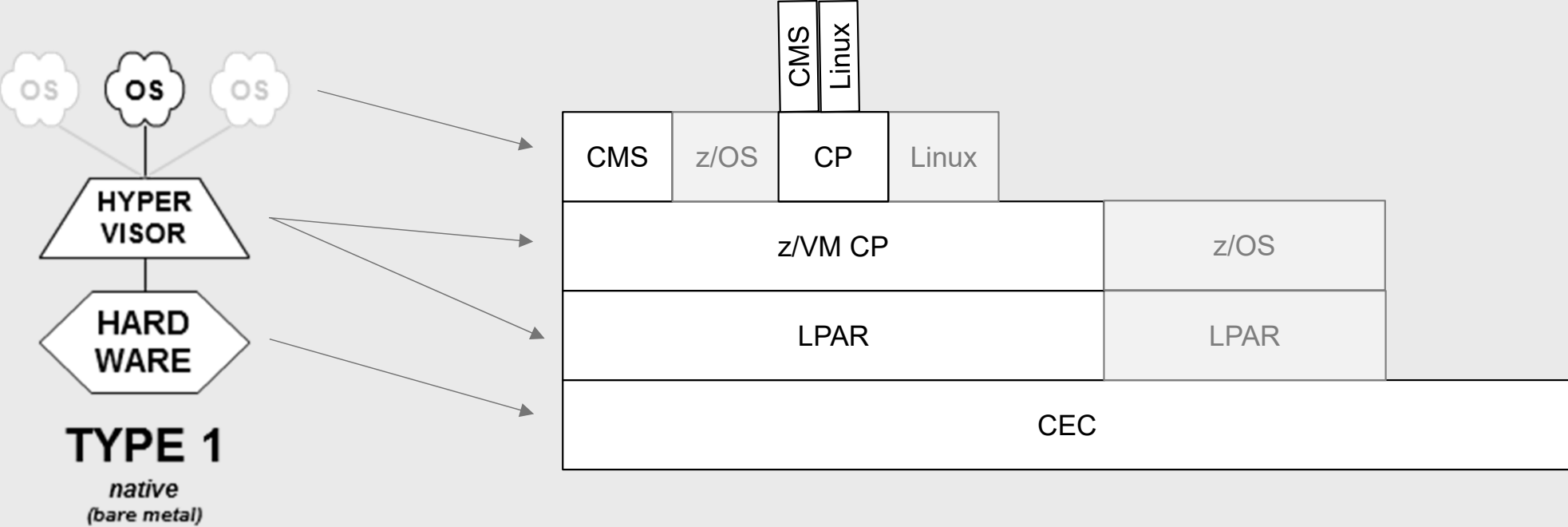
- Disk
- Memory
- Network adapters
- CPUs

These resources are managed by a **hypervisor**.

Virtualization enables four fundamental capabilities:

- resource sharing
- resource aggregation
- emulation of function
- insulation

What does Virtualization look like on IBM Z?



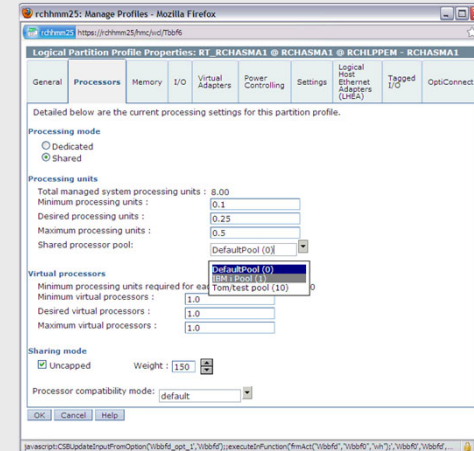
Real, Logical, & Virtual

It's important that everyone understands which view YOU are talking about!



What the accounting team sees
(REAL)

What the system programmer sees
(LOGICAL)



What the application developer sees
(VIRTUAL)

Why z/VM?



z/VM has an awesome **community**.

World-class **clustering** and **mobility**, allowing virtual machines to move to new hardware without an outage.

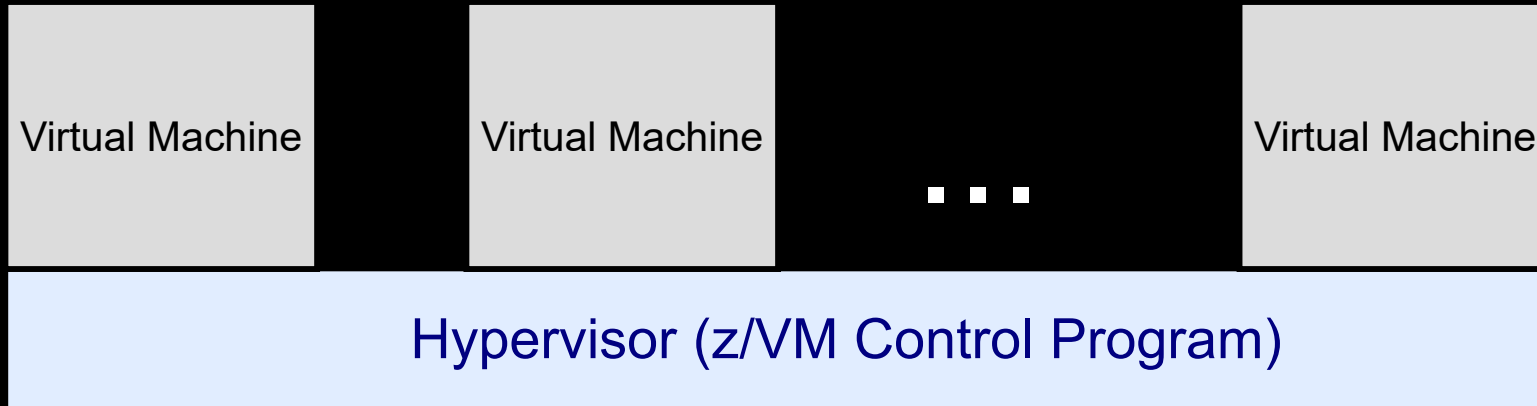
Scale – not just large machines, but fully utilized large machines

Resource **management** facilities for large and diverse workloads.

IBM has the **entire Z stack**, with special assists that bring the hardware and software closer together.

Virtual Machines

What is a virtual machine?



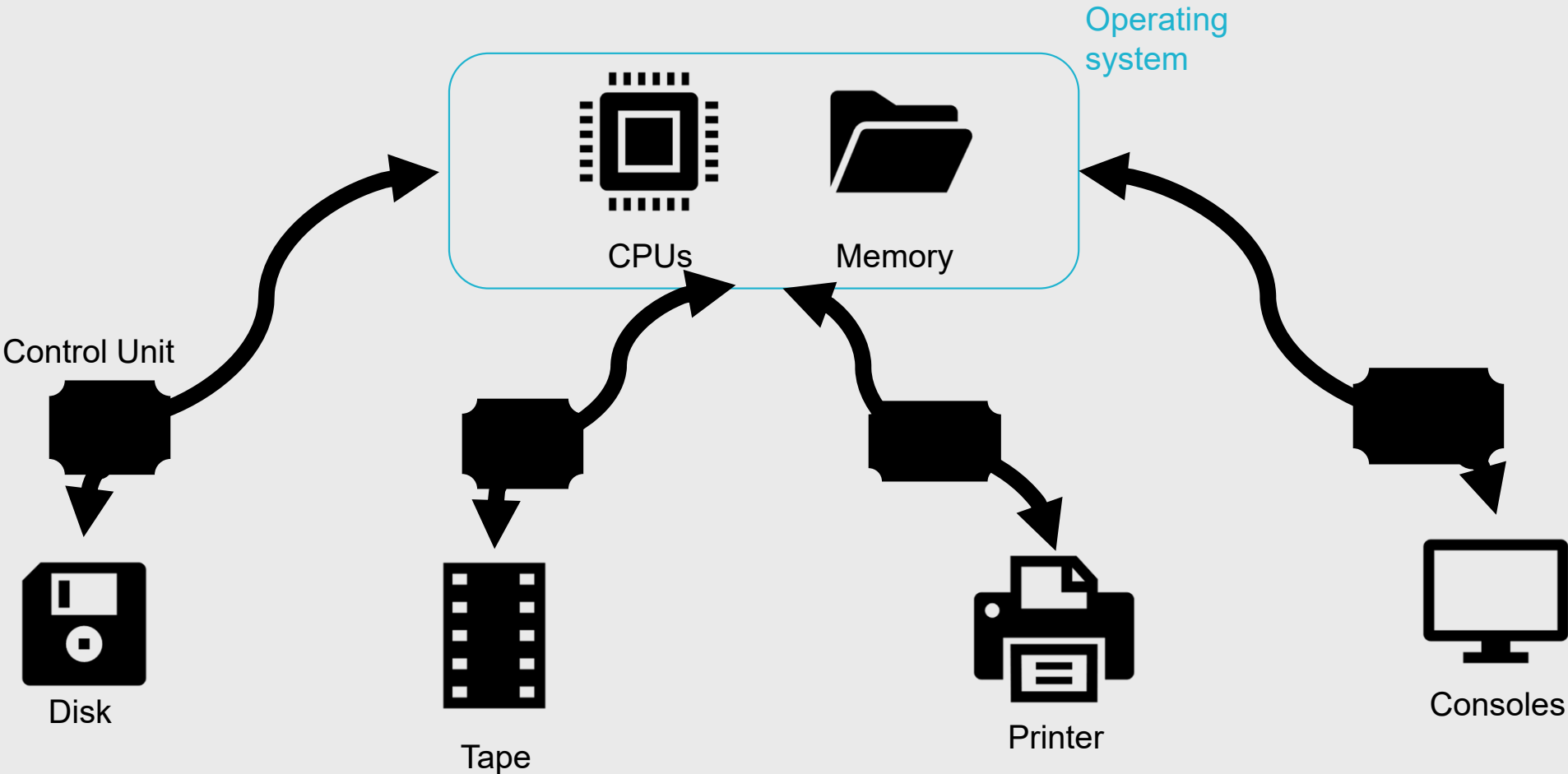
An execution context that obeys the architecture.

A faithful replication of the IBM Z architecture

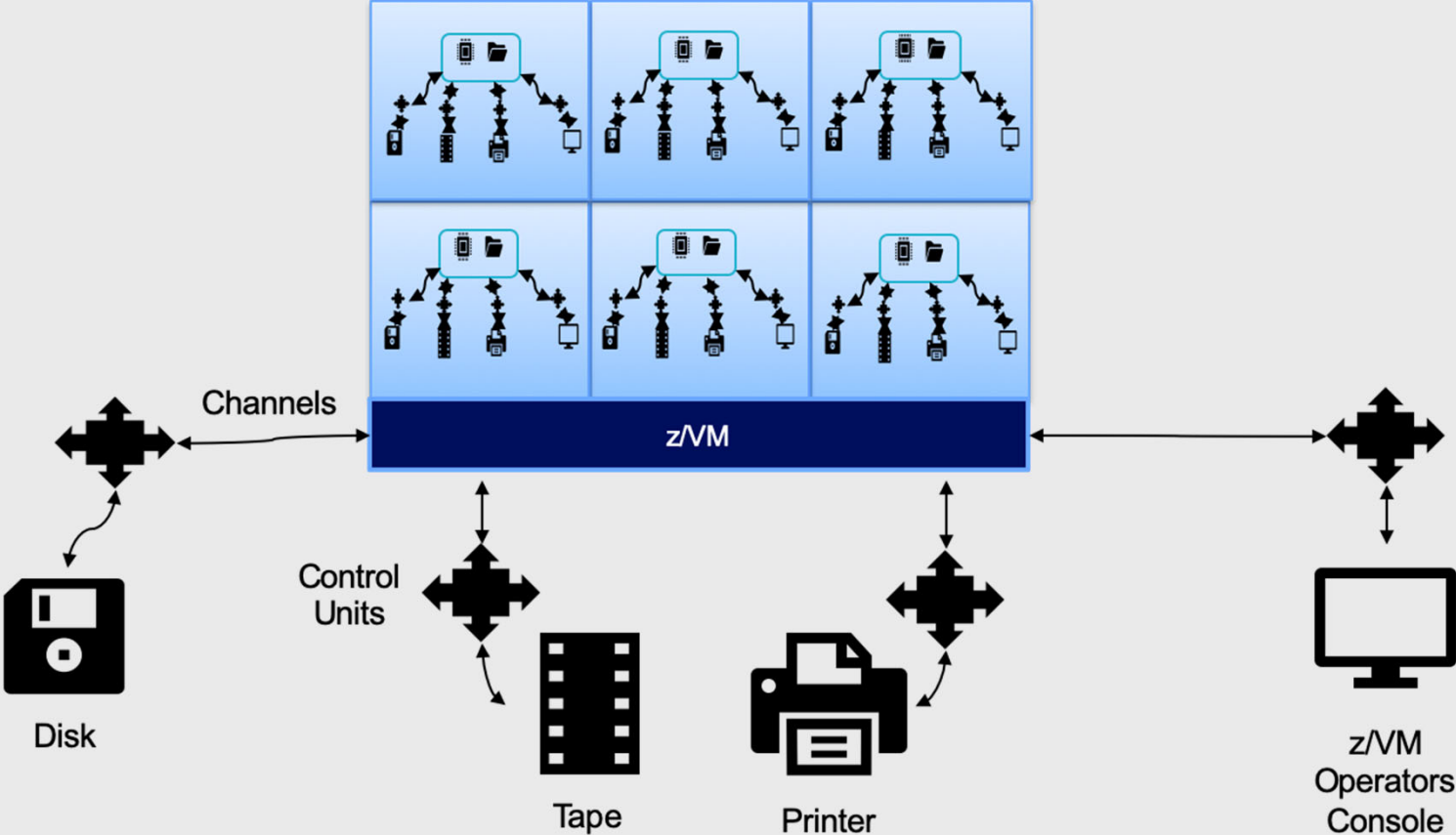
A virtual mainframe, with CPU, memory, disks and devices

Something just like the underlying OS, or nothing like it!

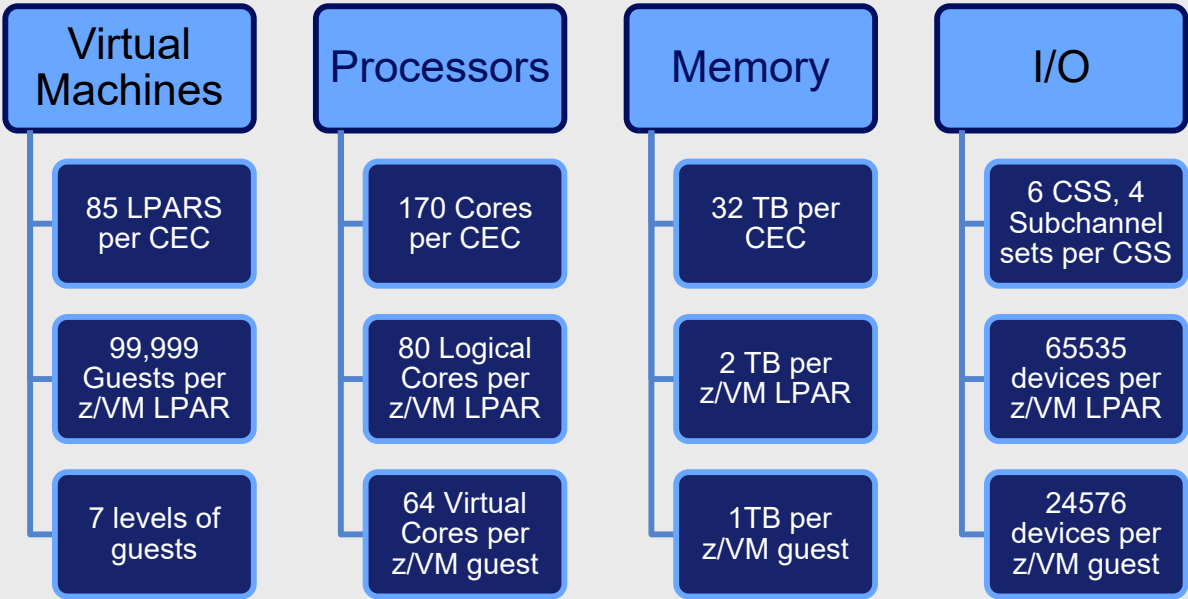
Just like the real thing, we have



Just lots of them!

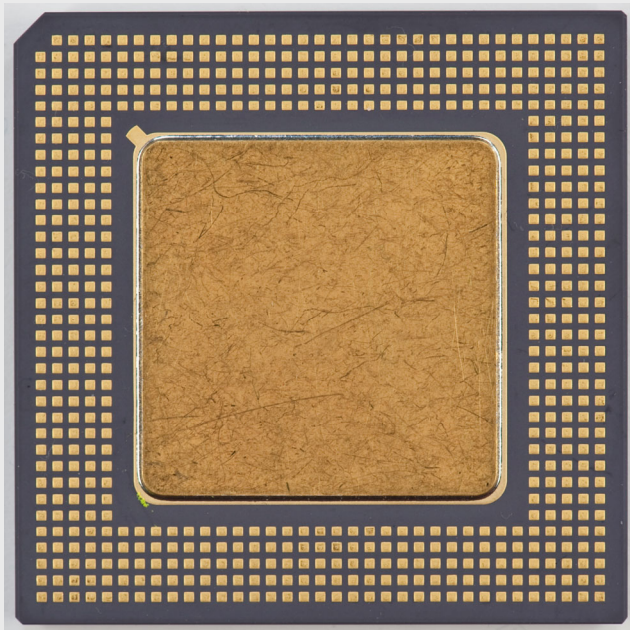


Testing the Limits (z14)



Intro to Processors and z/VM

Running CPUs 90% busy doesn't scare us.
We want you to get what you pay for!

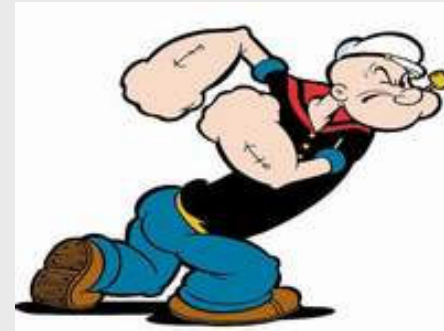


z/VM supports **logical** processors

Z/VM faithfully replicates **virtual** processors

z/VM as a Hypervisor

z/VM as a guest



Did someone say more power?

Intro to Memory and z/VM

Fully utilizing your memory is not a bad thing
- z/VM expects to have to move some memory to paging media.



Flexibility to grow or shrink your memory



Strength to handle many VMs



Intro to I/O and z/VM

z/VM can accommodate an extremely large variety of I/O devices, allowing for incredible flexibility



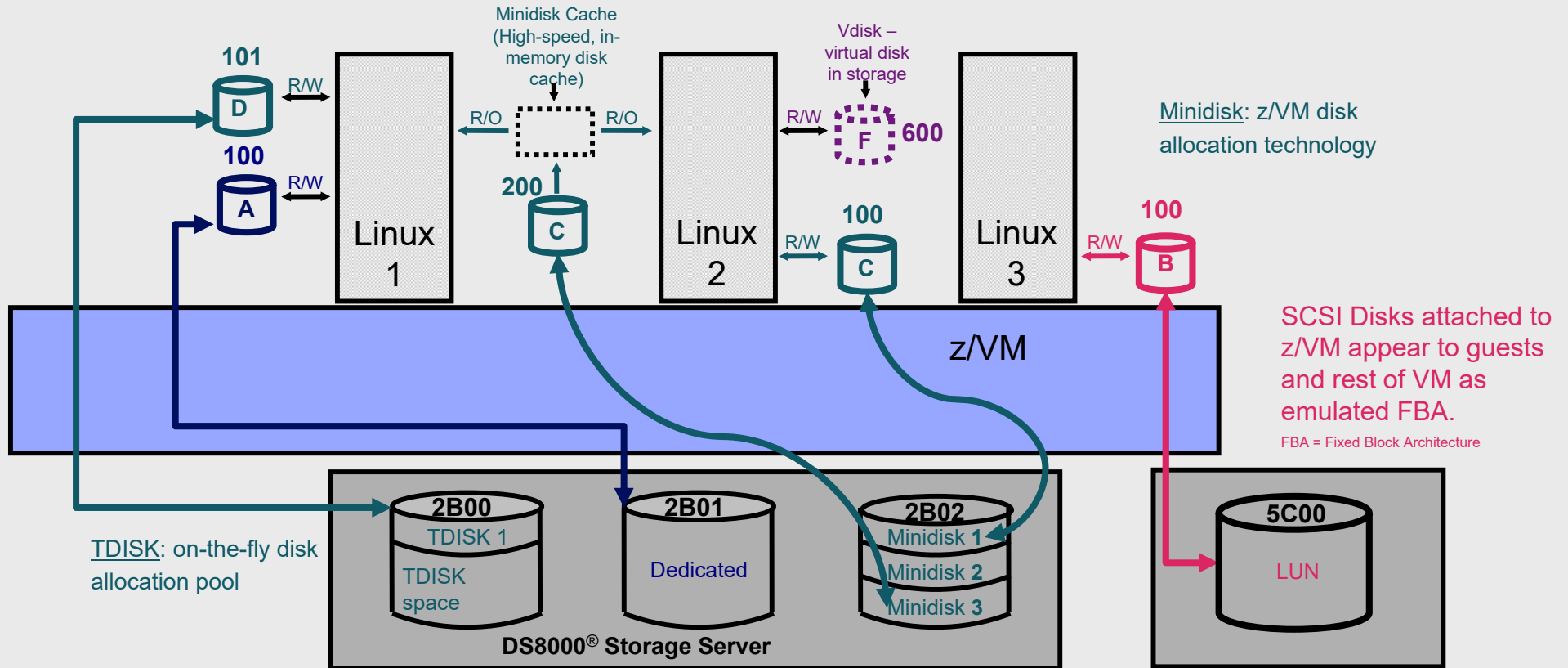


Device Management Concepts

How can I get from my house to the mall?

- **Dedicated or attached**
 - My parents give me my own car to go to the mall
- **Virtualized**
 - I have to share the car keys with my whole family, so I can go to the mall sometimes
- **Simulated**
 - I shop online so I don't have to go anywhere physically
- **Emulated**
 - I take the bus to the mall

Device Management Concepts

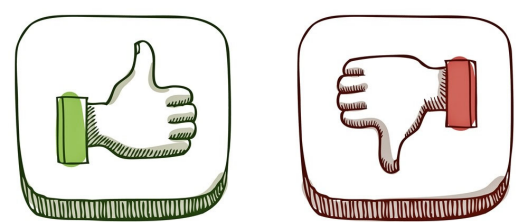


Examples of devices that z/VM supports:

	Examples
Direct-Access Storage Devices (DASD)	3380, 3390
Tapes	3480, 3490, 3590
Printers	3268, 3287
Unit Record Devices	3505 reader, 3525 punch
Displays	3270
Communication Controllers	3745
OSA Devices	OSA-2, OSA-Express, OSA-Express2

FICON vs FCP vs PCIe

What's right for
you?



Fiber Connection (FICON)

- Fiber-optic channel technology
- Well established on IBM Z
- Command driven

Fiber Channel Protocol (FCP)

- Small Computer System Interface (SCSI) protocol
- Utilizes Fiber channel connection

Peripheral Component Interconnect Express (PCIe)

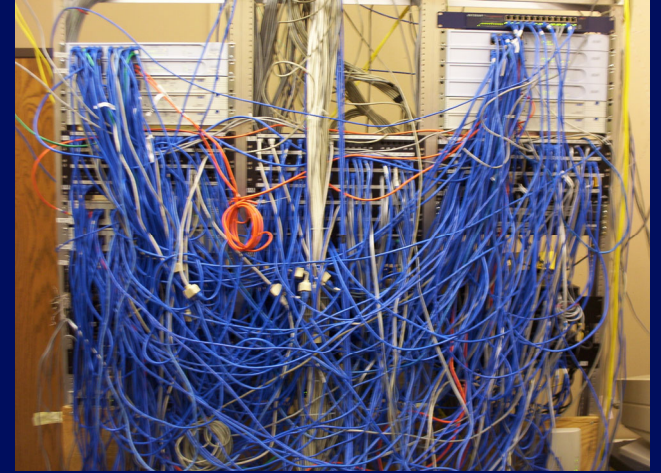
- PCI standard used to connect high-speed devices
- New kid on the block

Intro to Networking and z/VM

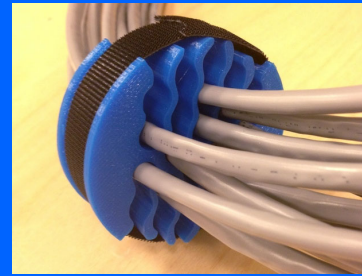
Users have many options for networking connections on z/VM

Virtualizing network infrastructure saves complexity and time

From this



To virtual!



Intro to SPOOL in z/VM

- z/VM simulates unit-record devices for virtual machines
 - Simulated reader, punch, printer, console for every virtual machine
- Simulated UR devices are still widely used on z/VM; upcycled to
 - Exchange application data
 - Share data files among multiple virtual machines
 - Collect console output
 - Store data collected by various system functions
- "Intermediary" between information in memory and information that you might devote long term storage to



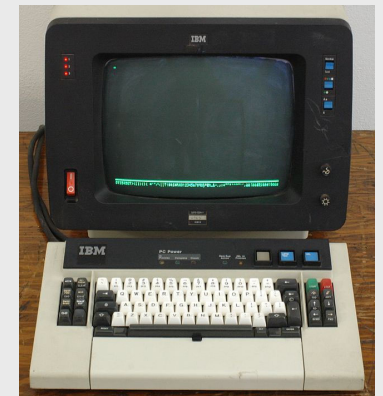
Card Reader



Card Punch



Printer



Console

Intro to Security and z/VM

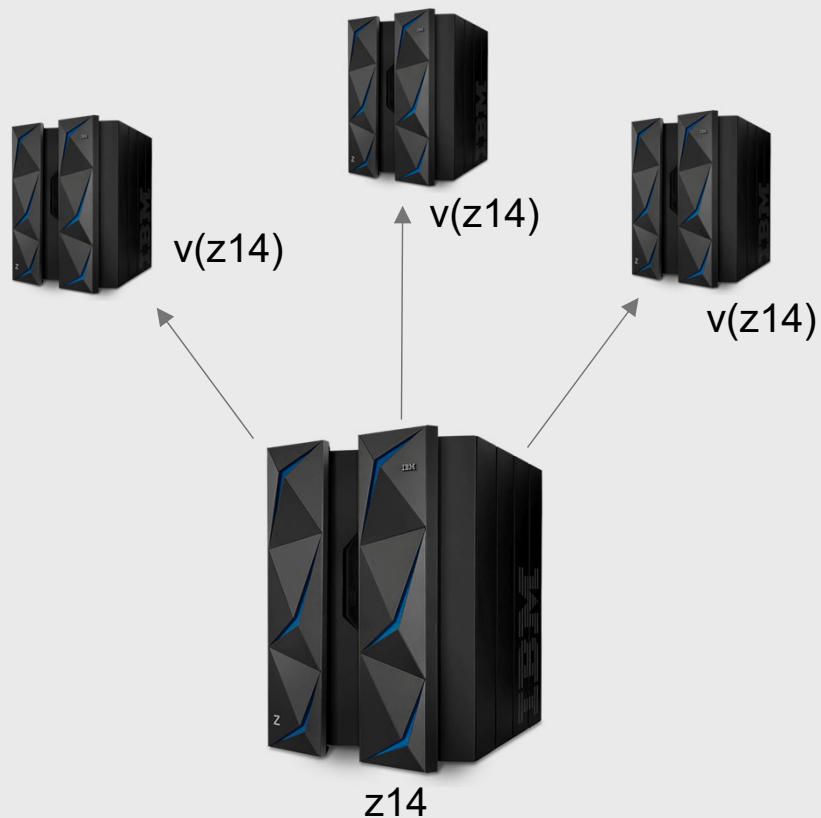
Crypto accelerators offload the hard math of cryptography, making security faster and easier.



Overcommitment

Doing more with less

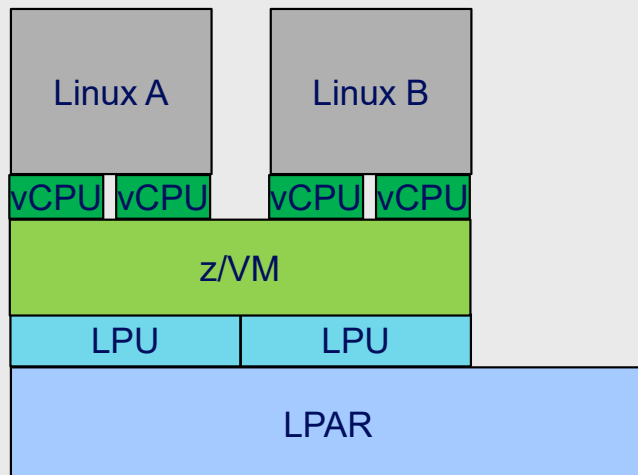
Resource Overcommitment



What is it?

- When the virtual resources defined to an OS (Or multiple OS's) exceed the physical resources of the machine
- **Fundamental** to the idea of virtualization
- Possible because a virtual resource will not need 100% of the physical resource 100% of the time
- What things can be overcommitted? CPU, Memory, I/O, Networking, Crypto

Overcommitting CPU



How's it work?

- When vCPUs need to run they must be **dispatched** onto the logical processor
- Hypervisor must manage vCPU access to the LPU

Tools of CPU management

- Entitlement – The amount of real CPU time a logical partition is entitled based on LPAR configuration.
- Processor Topology – The amount of real CPU a logical processor is entitled
 - Vertical High – entitled to 100% of a real CPU
 - Vertical Medium – entitled to 50-100% of a real CPU
 - Vertical Low – entitled to 0% of a real CPU
- Consumption/Utilization - How much of a real CPU is being used (for LPAR or for entire CEC)
- Share Setting – How much entitlement is due after comparison to other users
- Processor Affinity – A vCPU that must run on a logical processor of the same type

Determining "Who's on first?!"



Gather Partition Information:

- Processor **Topology**
- LPAR weights
- Processor Utilization



Gather Hypervisor Information:

- Share Settings
 - Absolute/Relative
 - Hard/Soft
- Resource Pool Information
- Processor Affinity



Schedule vCPU based on relative priority:

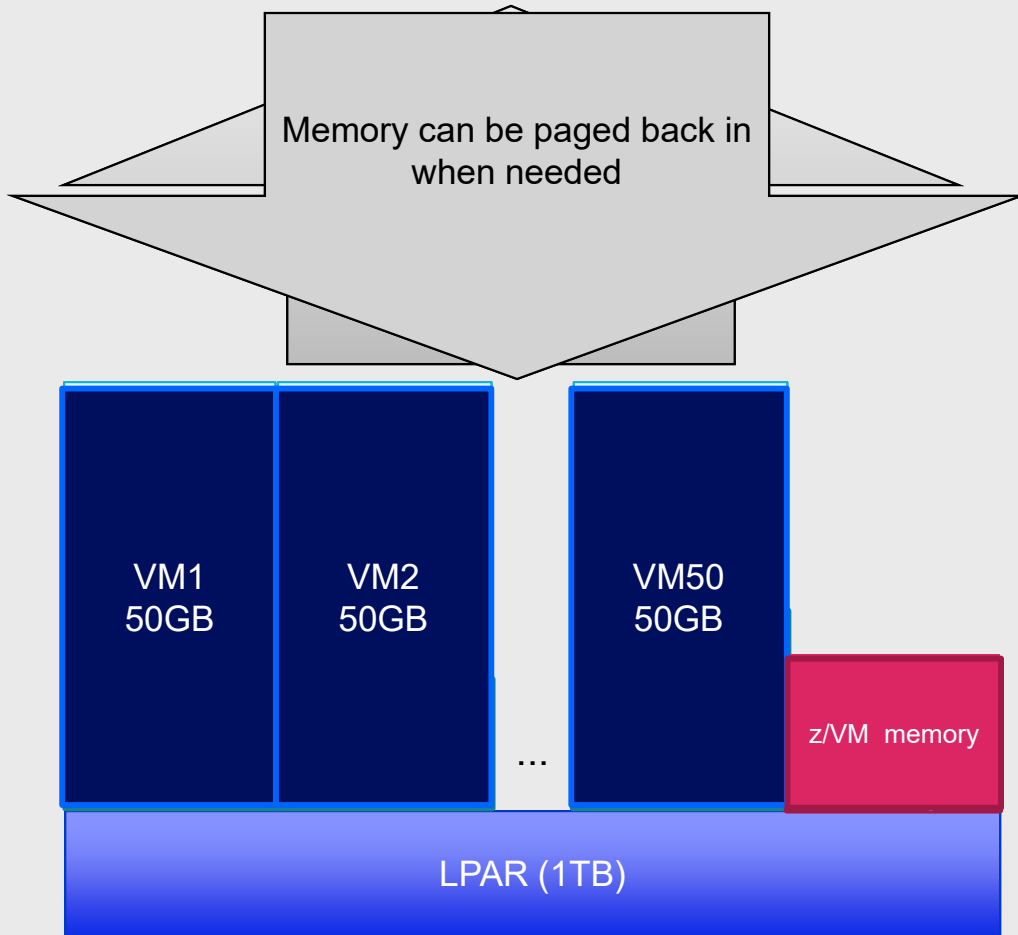
- Use partition and Hypervisor information to choose the best logical processor to run on.



Dispatch vCPU onto Logical Processor:

- Run CPU under **SIE**
- Let vCPU run for **AT LEAST** a **Minor Time Slice**.
- Compete with other vCPUs for unused or **until Elapsed Time Slice**.

Overcommitting Memory



How's it work?

- We can only keep so much in memory
 - The 1TB must have z/VM's own structures and memory as well as guest memory
- So we choose some memory to page out to disk

Memory overcommit – how does it work?

- The magic of Dynamic Address Translation!
- Allows storage to be non-contiguous and allow us to do things with the real page (like page it out, reuse it) without affecting the content.
- Each address is broken up into groups representing its place in tables.

Guest address

Rx			Sx		Px			Bx			
0		32	33		43	44		51	52		63

Region Table
Region 1
Region 2

Region 5
Region 6

Segment Table
Segment 1

Segment 3

Segment 6
Segment 7

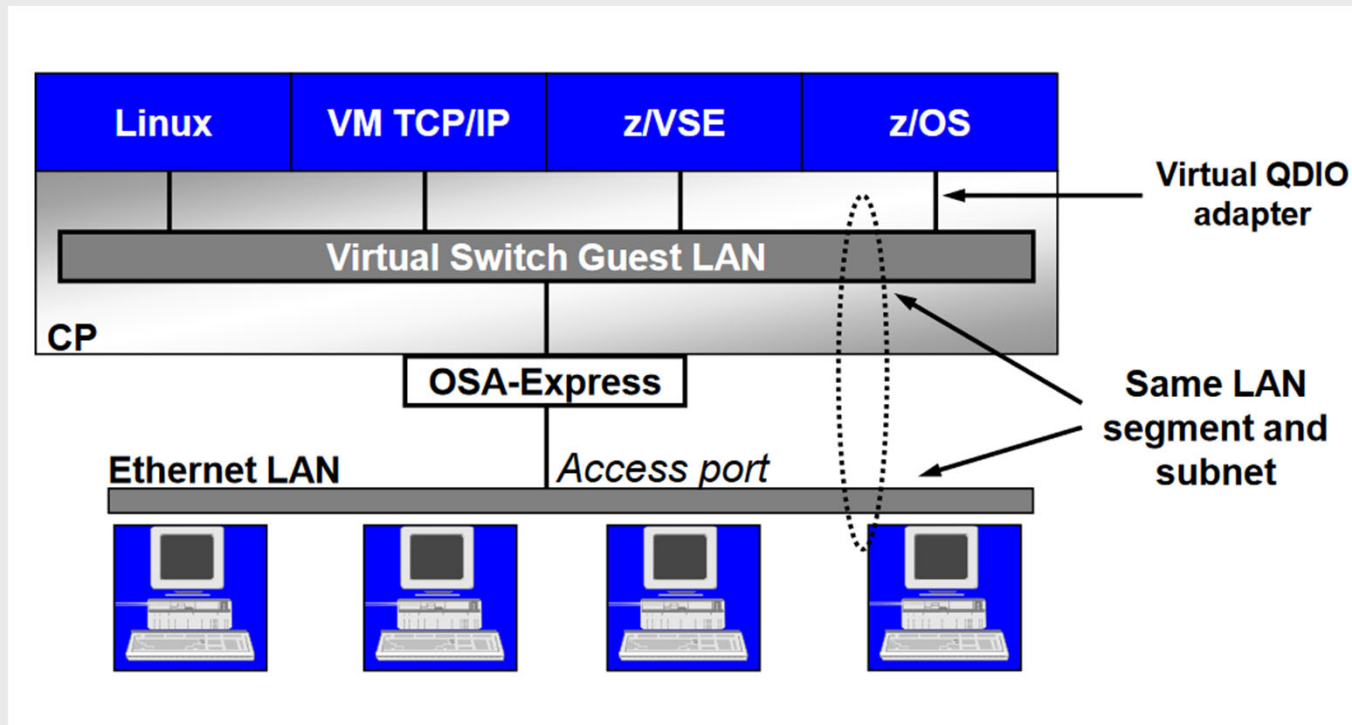
Page Table

Page 3
Page 4

Page 7



Overcommitting network with a VSWITCH



How do you Build z/VM?

What do you need?

- LPAR definition - HMC/SE, what PR/SM controls
- IOCDS - I/O definition – or use DPM to give you both IOCDS & PR/SM
- System Config – What's defined to the z/VM system
- User Directory – What's defined to the z/VM users

z/VM User Directory

- Where all virtual machines are defined
 - Configuration (initial)
 - Operating characteristics
- Tools are available to help with directory updates
 - e.g. DirMaint

What does the directory look like?

- Source directory
 - Text file or files on disk
- Object directory
 - Created from source directory file(s)
 - DIRECTXA utility
 - Stored on disk that z/VM system

System Config: What makes up my system?

What is defined in the system configuration file?

- Operator_Consoles statement to find a console
- vSWITCH statements to define our virtual network
- CP_Owned statements to define the residence volume
- And so much more!!!

```
/* Set up Console */
```

```
operator_consoles 01F 009 3E0 500
```

```
/* Set up VSWITCHES */
```

```
DEFINE VSWITCH VSWITCH1 RDEV 7080 CONNECT CONTR DTCVSW1 ETH  
MODIFY VSWITCH VSWITCH1 GRANT TCPIP  
MODIFY VSWITCH VSWITCH1 GRANT LINUX01  
DEFINE VSWITCH VSWITCH2 RDEV 6240 CONNECT CONTR DTCVSW1 ETH
```

```
/* Set up CP Volumes */
```

```
CP_owned slot 001 2ND600  
CP_owned slot 002 2ND601  
CP_owned slot 003 2ND603  
CP_owned slot 004 2ND604
```

User Directory: What Does a Virtual Machine Definition Look Like?

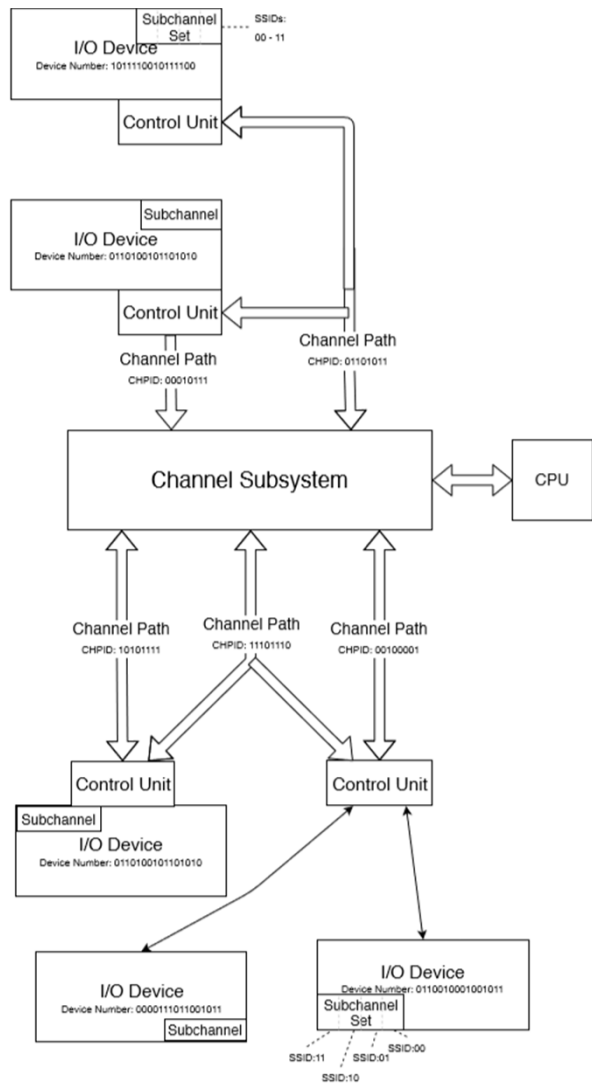
What is defined in a directory entry?

- Login credentials
- Authorizations
 - Access to system resources
 - Command privileges
- Virtual machine characteristics

```
USER LINUX01 MYPASS 512M 1024M      G
MACHINE ESA 2
IPL 190 PARM AUTOCR
CONSOLE 01F 3270 A
SPOOL 00C 2540 READER *
SPOOL 00D 2540 PUNCH A
SPOOL 00E 1403 A
SPECIAL 500 QDIO 3 SYSTEM MYLAN
LINK MAINT 190 190 RR
LINK MAINT 19D 19D RR
LINK MAINT 19E 19E RR
MDISK 191 3390 012 001 ONEBIT MW
MDISK 200 3390 050 100 TSOBIT MR
```

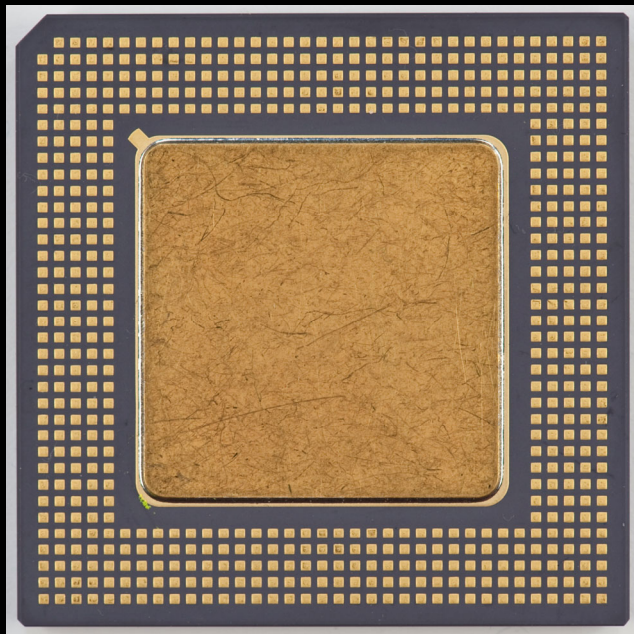
Tools to help you

- DIRECTXA – creates and compiles your user directory
- Directory manager – Does this work for you!



Dynamic Resource Management

Dynamic Processor Management



VARY PROC/CORE

SMT Enabled – CORE
SMT Disabled – PROC

DYNAMIC SMT

HIPERDISPATCH (Park/Unpark)

Adjustable via:
CP SET SRM UNPARKING

Dynamic Memory Management



chmem

DEFINE STORAGE

Used to change the memory on a guest.

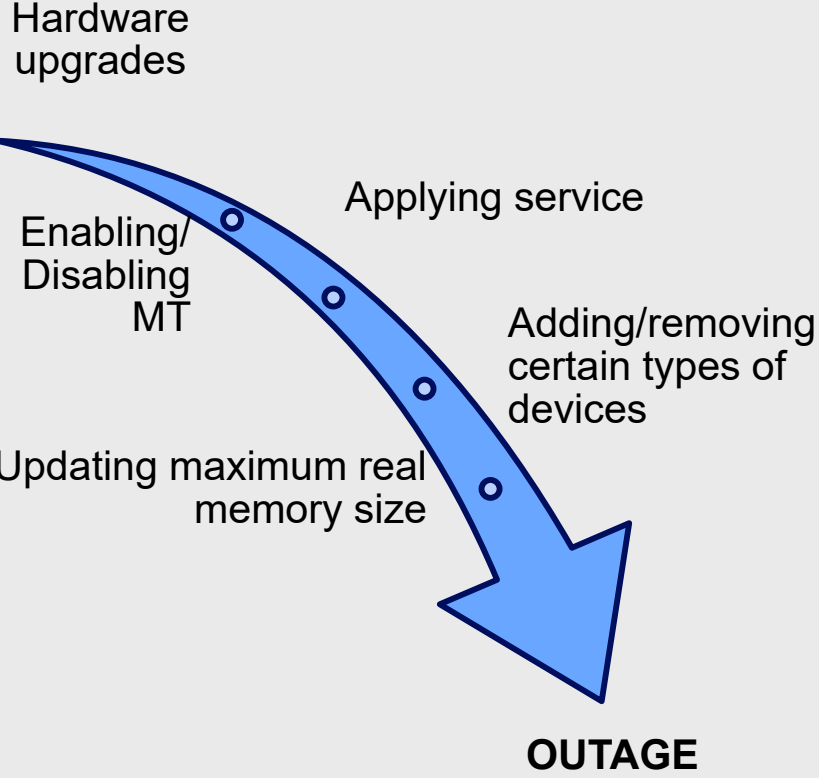
Linux commands are provided to bring more memory online, or take memory offline to the Linux OS.

SET STORAGE

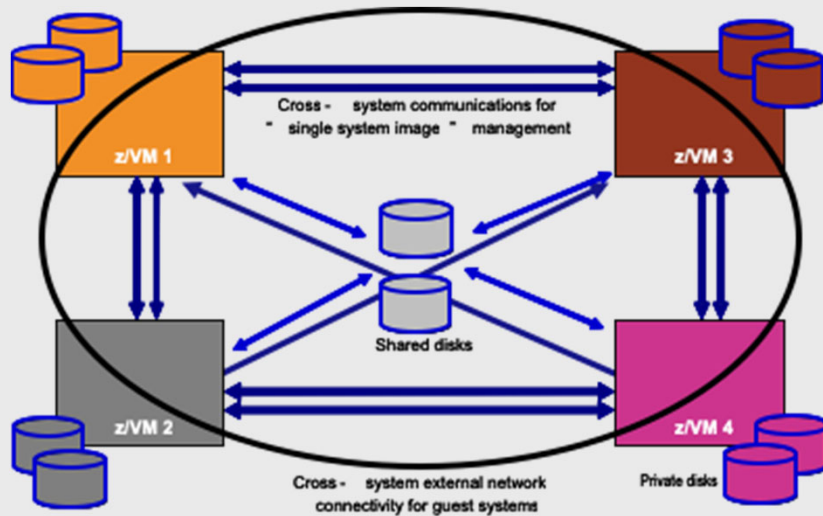
Use to add more memory to your z/VM system.

On the z14 and later machines, use this command to remove memory.

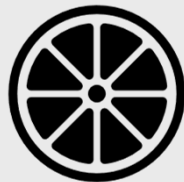
What needs a planned outage?



Single Systems Image (SSI) Feature



z/VM SSI



z/OS
Parallel
Sysplex

1. Manage a **single** user directory
2. Cluster management from any member
 - Apply maintenance to **all members** in the cluster from **one location!**
 - Issue commands from one member to operate another!
3. Built-in **cross-member capabilities**
4. Resource coordination and protection of network and disks
5. Relocate a **running** Linux guest from one system to another

FREE in z/VM 7.1

CP/CMS: A Symbiotic Relationship

CP (Control Program)

CP solved the problem of multiple use by providing separate computing environments at the machine instruction level for each user (virtual machines).

CMS (Conversational Monitoring System)

CMS then provided a single user service that did not have to worry about the problems of sharing, allocation, and protections of resources such as memory and disk storage.

CMS Pipelines

- Similar to piped commands in other scripting languages
- Allows the output of the first command, or stage, to be given to the second as input
- These are treated as one command
- One of the most powerful features of CMS

```
PIPE (end ?)
    < input text
    | locate /result/
    | xlate upper
    | > resultfile xedit a
    ?
```

```
DO WHILE DATATYPE(last) \= 'NUM'
    SAY 'Please enter the lunch amount again.'
    SAY 'The amount you entered was not a number without a',
        'dollar sign.'
    PARSE PULL last
END
```

- REstructured eXtended eXecutor language
- Interpreted, not compiled
- Used for scripting and macros for CP and CMS
- Often utilizes pipelines

REXX

Monitoring z/VM Performance

Performance monitoring is the periodic collection of performance data and serves two major purposes:

1. Early Detection

An unfavorable trend can be detected early so that corrective action can be taken before it develops into a serious performance problem. Early detection is done by tracking key performance indicators over time and comparing them to established limits of acceptable performance. These indicators are often various measures of response time.

2. Basis for Performance Problem Determination

After a problem has been identified, the monitor data serves as the basis for determining the likely cause of the problem. This is done by comparing the current data that reflects the problem to past data collected when performance was adequate.

CP monitor data is the key source of information normally used to monitor the performance of a z/VM system.

Performance Toolkit for VM assists operators and system programmers with system console operation in full screen mode, and with performance monitoring on z/VM systems.

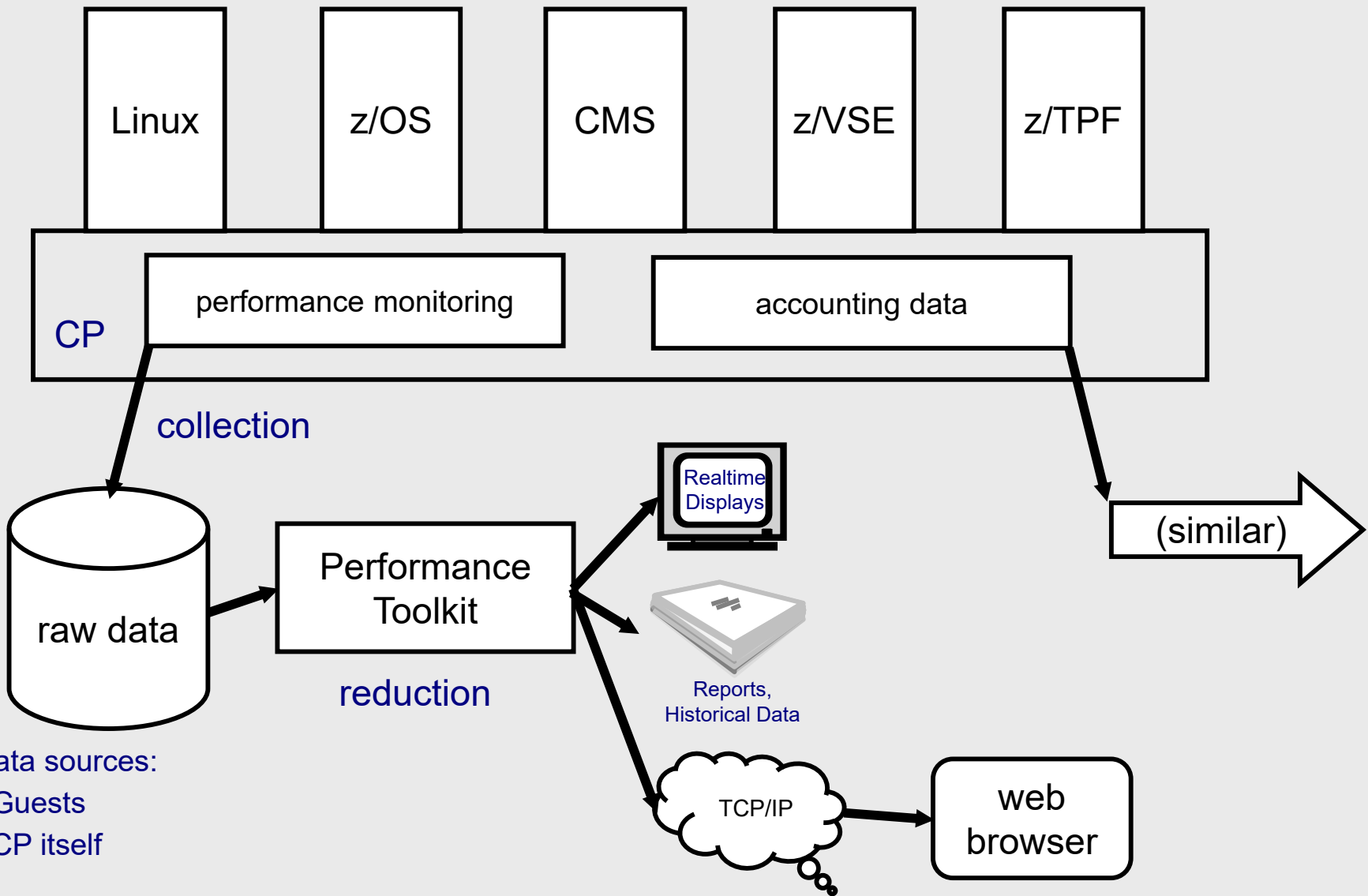
The **INDICATE** and **MONITOR** commands provide a system performance measurement facility for z/VM. These commands provide a method of:

- Obtaining system resource usage data while z/VM is operating so that steps can be taken then or later to improve performance
- Collecting measurement data using a z/VM monitor for later analysis by system analysts.

System services



There are ways to connect into the system to monitor what's going on at a deeper level, or to issue commands to the hardware.



Data sources:
 - Guests
 - CP itself

**Houston...we have
a problem**

What happened?

Soft ABEND

- Something went wrong in the program, but CP is ok.
- System stays up, Virtual Machine may terminate.

Hard ABEND

- Something went wrong with the control program.
- The entire system terminates.

Hung User

- A specific user is stuck in CP and cannot continue.
- The rest of the system continues operation.

Hung System

- The entire system is stuck in CP and cannot continue.

Determining Why



CP will dump system storage to aid debug for a hard or soft abend.

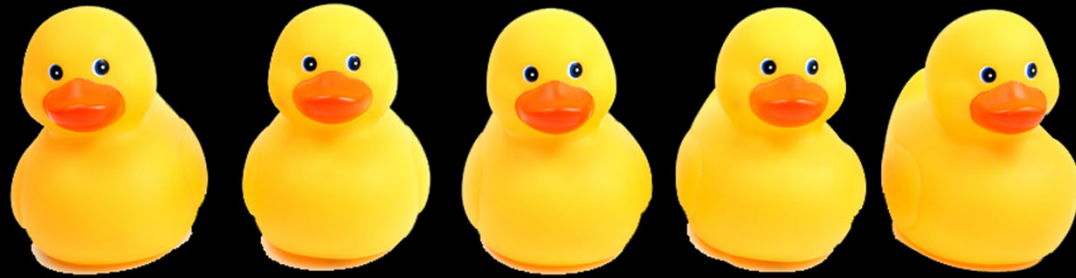
SNAPDUMP or **RESTART** dumps help gather information for system or user hangs.

IMPORTANT: configuring your system with adequate dump space will reduce headaches should you need to use it!

Dumping z/VM is not instantaneous and can take on the order of minutes depending on the size of your system. The longer you wait, the more information you'll have.

**If you stay ready,
you don't have to
get ready**

Installing z/VM



Installing z/VM

Have your device information ready before you start installation: Disk, Networking for the z/VM TCPIP stack, CTCs if you're using SSI

Pictures help! As much as you can, make layouts of your device configurations, networking layouts, and SSIs.

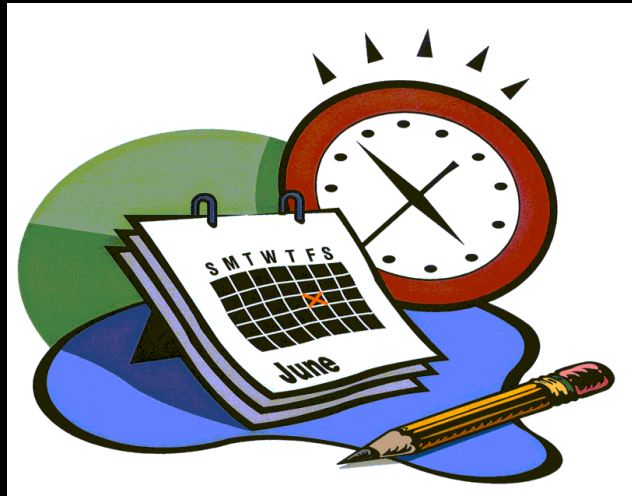
- z/VM install guide has worksheets to help you gather the information you need, fill these out first!
- Many shops install z/VM second level to try out the process and test new levels

Optional Features

If you've purchased DirMaint, Perfkit, RACF or any additional paid features, have their Program Directories available when you install.

There are special codes necessary to enable these.

Servicing z/VM



Servicing z/VM

- Every 2 years, in the 3rd quarter, z/VM releases a new level of the product.
 - We'll always have one "stable" and one "new function" release in service
 - Releases will be out of service 6 months after the N+2 release comes out (e.g., z/VM 6.4 would end service 6 months after z/VM 7.2 was available)
- z/VM 6.4 is our current stable release
- z/VM 7.1 is our current new function release
- In between releases, new function and fixes are shipped via APARs.

PMR

Problem Management Report
what you open when you find a problem,
e.g., 65118,442,000

APAR

Authorized Program Analysis Report
for new function and problem fixes between releases,
e.g., VM66209

PTF

Program Temporary Fix
actual code for the APAR,
e.g., UM35371

RSU

Recommended Service Upgrade
group of PTFs we think all customers should apply,
e.g., z/VM 7.1 RSU 1901

Where to go when you're stuck

Community Forums

- IBMVM Listserv (z/VM community)
<http://listserv.uark.edu/cgi-bin/wa?A0=IBMVM>
- LINUX 390 Listserv (Linux on Z)
<http://www.cavmen.org/list390.html>
- Additional Listserv can be found here
<https://www.vm.ibm.com/techinfo/listserv.html>

Technical Publications

- VM Library
<https://www.vm.ibm.com/library/>
- IBM Knowledge Center
<https://www.ibm.com/support/knowledgecenter/en/>

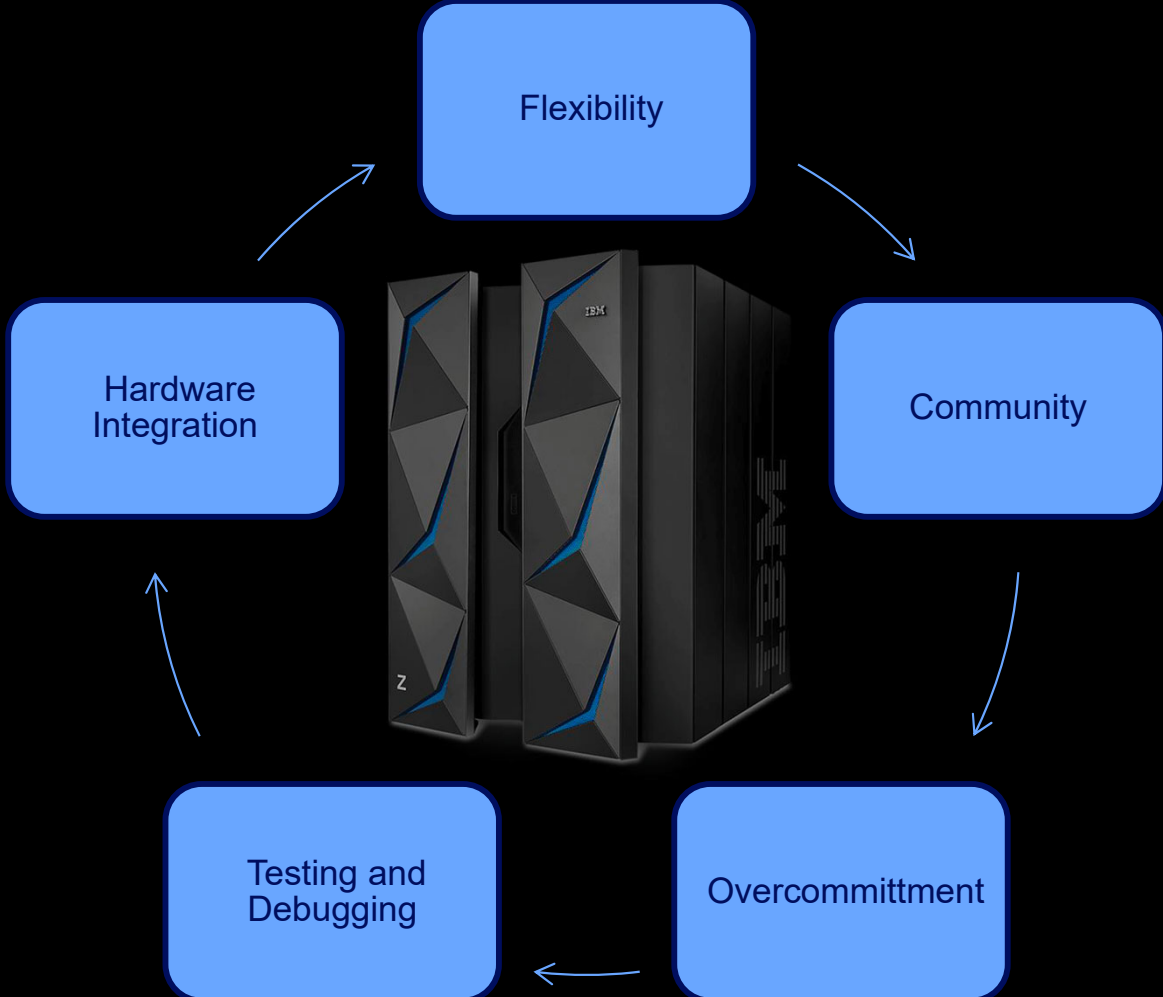
Redbooks

- System Z Redbooks
<http://www.redbooks.ibm.com/redbooks.nsf/domains/zsystems>

Ask an IBMer

- Attend Conferences
- Engage User Groups
- Contact the z/VM experts!
<https://www.vm.ibm.com/forms/>

Conclusion



Complete your session evaluations for a chance at daily prizes!

To complete, visit
www.share.org/evaluation
and see your progress on the
leaderboard!



Conclusion

z/VM is a powerful hypervisor for IBM Z (or LinuxONE) that can host a variety of operating systems.

It is one very important part of the Linux on Z ecosystem.

Hardware, LPAR, z/VM, Linux, containers all working together for you!

- An Overcommitment to helping our customers get the most out of their hardware
- A unique set of features for dynamic management and ease of use
- Debugging and monitoring tools to help find and solve problems
- A robust community for support

BACKUP