

# Mainframe Topics

## Kapitel 4

# Virtualisierung auf dem Mainframe

## Introduction of IBM z/VM

**Frank Heimes**

Senior IT Architect & Technical Lead for Ubuntu on z  
Canonical Ltd.

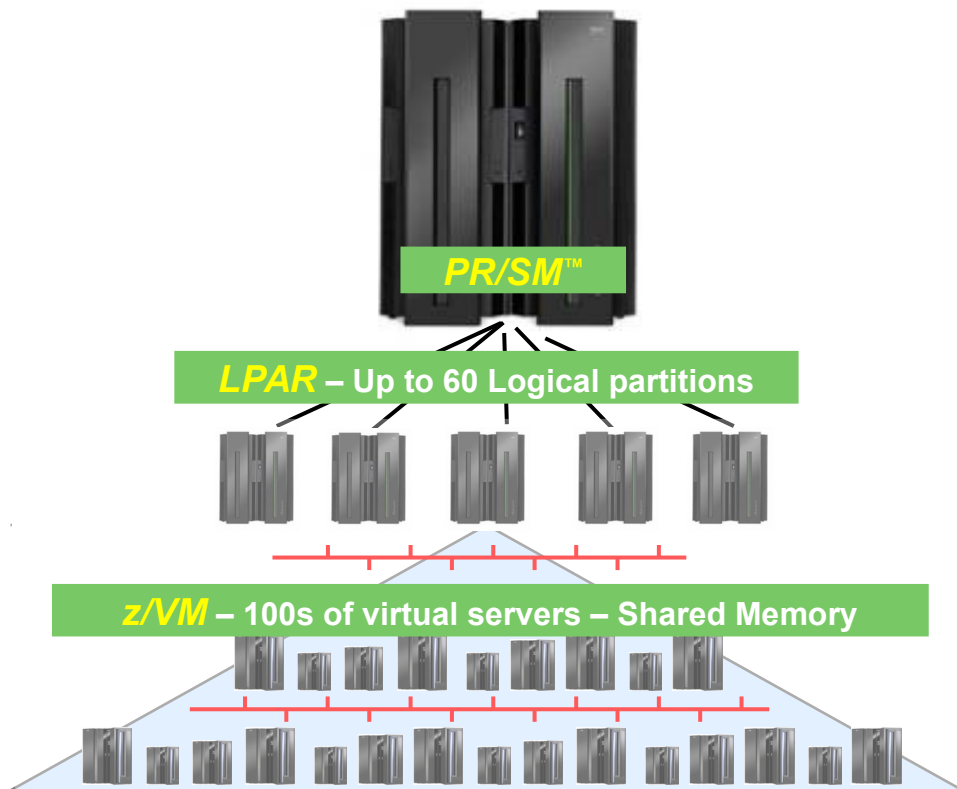
# Agenda

---

- **Was ist z/VM?**
- **Warum z/VM?**
- **Komponenten von z/VM**
- **Warum und wie setzt man z/VM ein?**
- **System z Terminologie**
- **Was ist CMS?**
- **Aufgaben und Merkmale ausgewählter Komponenten**
- **Was ist eine virtuelle Maschine?**
- **Memory Management**
- **VSWITCH und VLAN**

# System z – Extreme Virtualization

Built into the architecture not an “add on” feature



## IBM System z

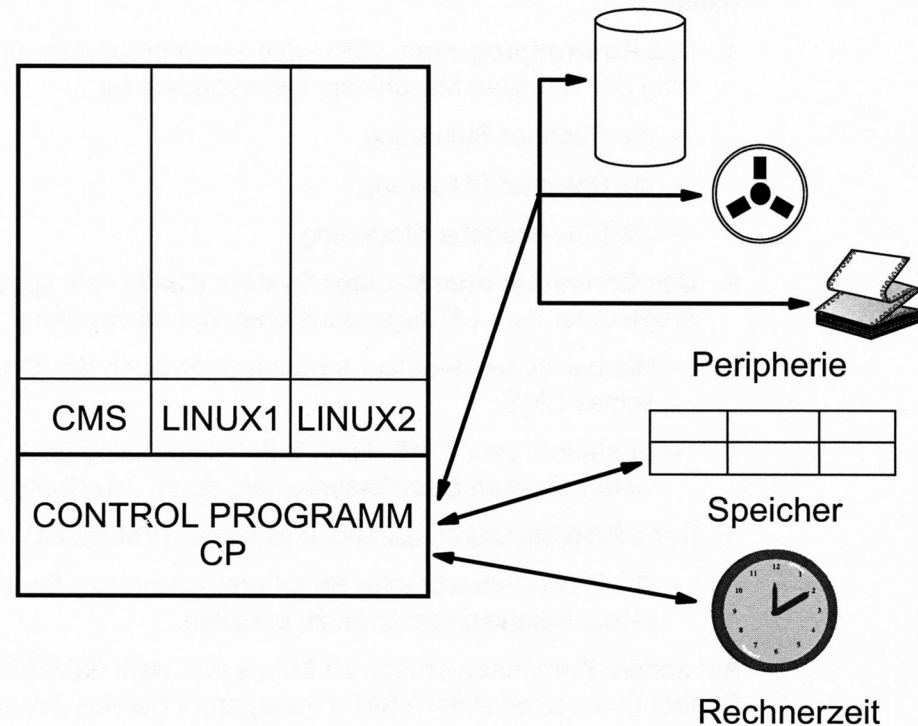
- **Deploy** virtual servers in **seconds**
- Highly **granular** resource sharing (<1%)
- Add physical resources without taking system down, scale out to **1000s** of virtual servers
- **Do more with less:** More virtual servers per core, Share more physical resources across servers
- Extensive virtual server **life-cycle management**
- **Hardware-enforced isolation**

## Distributed Platforms

- Limited per-core virtual server scalability
- Physical server sprawl is needed to scale
- Operational complexity increases as virtual server images grow
- VMware, Xen focus only on x86, no cross platform hardware management.

# Was ist z/VM?

z/VM (früher VM/CMS) ist ein Betriebssystem der Firma IBM für Grossrechnersysteme. Es dient dazu, eine (fast) beliebige Anzahl von virtuellen Maschinen (VMs oder auch Gäste/guests) auf einem Grossrechnersystem zu erzeugen, zu verwalten und parallel zu betreiben. z/VM ist mit seinen Wurzeln aus den 1960er Jahren der Urahn der PC-basierten Virtualisierungssysteme.



VM hat bereits in den frühen 1970er Jahren vom Konzept her die heutigen lokalen Netzwerke vollständig vorweggenommen. Die virtuellen Maschinen entsprechen einem Personalcomputer oder einem Server und können miteinander in Verbindung treten bzw. sich gegenseitig Dienste bereitstellen. VM ist ein Akronym für Virtual Machine. *Quelle: de.wikipedia.org*

# System z Interpretive Execution

---

## Advanced Technology for Virtual Server Hosting

- **Start Interpretive Execution (SIE) instruction**
  - Operand is a state descriptor for an LPAR or virtual machine
  - Accommodates fixed-storage and pageable guests
  - Interception controls allow hypervisor intervention
  - Reduces context switch time
- **System z implements two levels of SIE**
  - No performance penalty for running z/VM in an LPAR
  - No shadow page tables required for DAT-on guests
  - Considerable architectural and hardware investment required
    - Potential instruction behavioral differences at each level
    - Multiple control register sets

# Haupt-Komponenten von z/VM

---

## 1. Das Kontrollprogramm (CP)

Das Steuerprogramm für die Steuerung der virtuellen Maschinen, insbesondere für die Rechner-Steuerung, die Speicher-Steuerung, die Ein/Ausgabe-Steuerung.

## 2. Das Conversational Monitor System (CMS)

Ein speziell auf die Arbeit unter dem CP zugeschnittenes Betriebssystem. Hierbei ist der Benutzer am Bildschirm auch der 'Operator'seines CMS. Er steuert 'sein' CMS durch Befehle im Dialog.

## 3. Der VM/REstructured eXtended eXecutor (VM/REXX) Interpreter

Die Prozedurensprache REXX ermöglicht dem Benutzer, selbst Befehlsprozeduren zu erstellen.

4. VMS ES/E (Virtual Machine Serviceability Enhancements Staged/Extended) ein Service Tool für Installation und Service.

5. GCS (Group Control System) ein virtueller Maschinen Supervisor, ein Hybrid Betriebssystem.

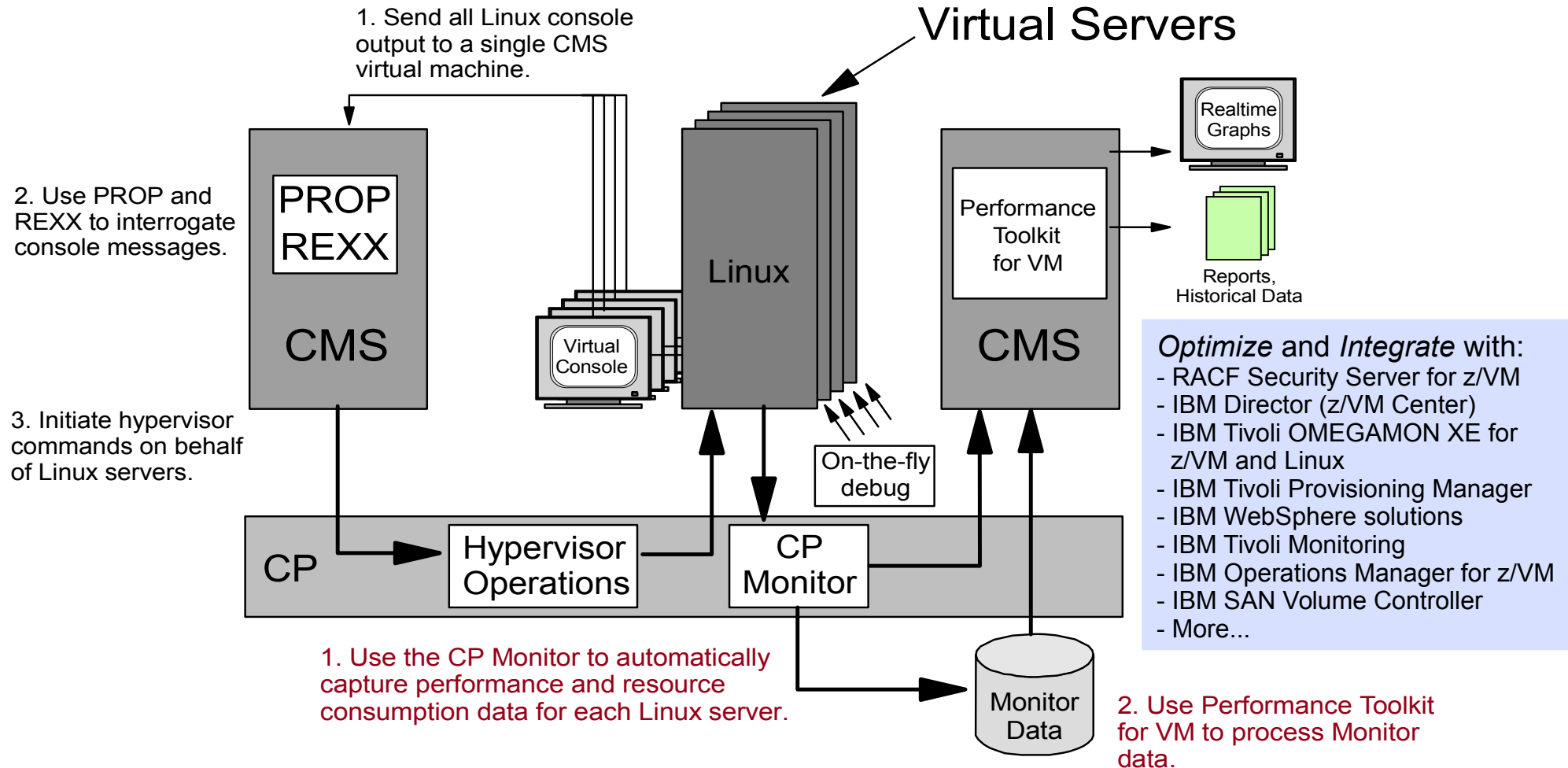
6. TSAF (Transparent Service Access Facility)

7. AVS (Appc/VM VTAM – Virtual Telecommunication Access Method - Support)

8. DVF (Dump Viewing Facility)



# z/VM Technology – Command and Control Infrastructure



# Vorinstallierte Features und Produkte

---

ICKDSF (Device Support Facilities)

RACF (Resource Access Control Facility)

OSA (Open Systems Adapter)

ECKD (Extended Count Key Data)

HCD (Hardware Configuration Definition)

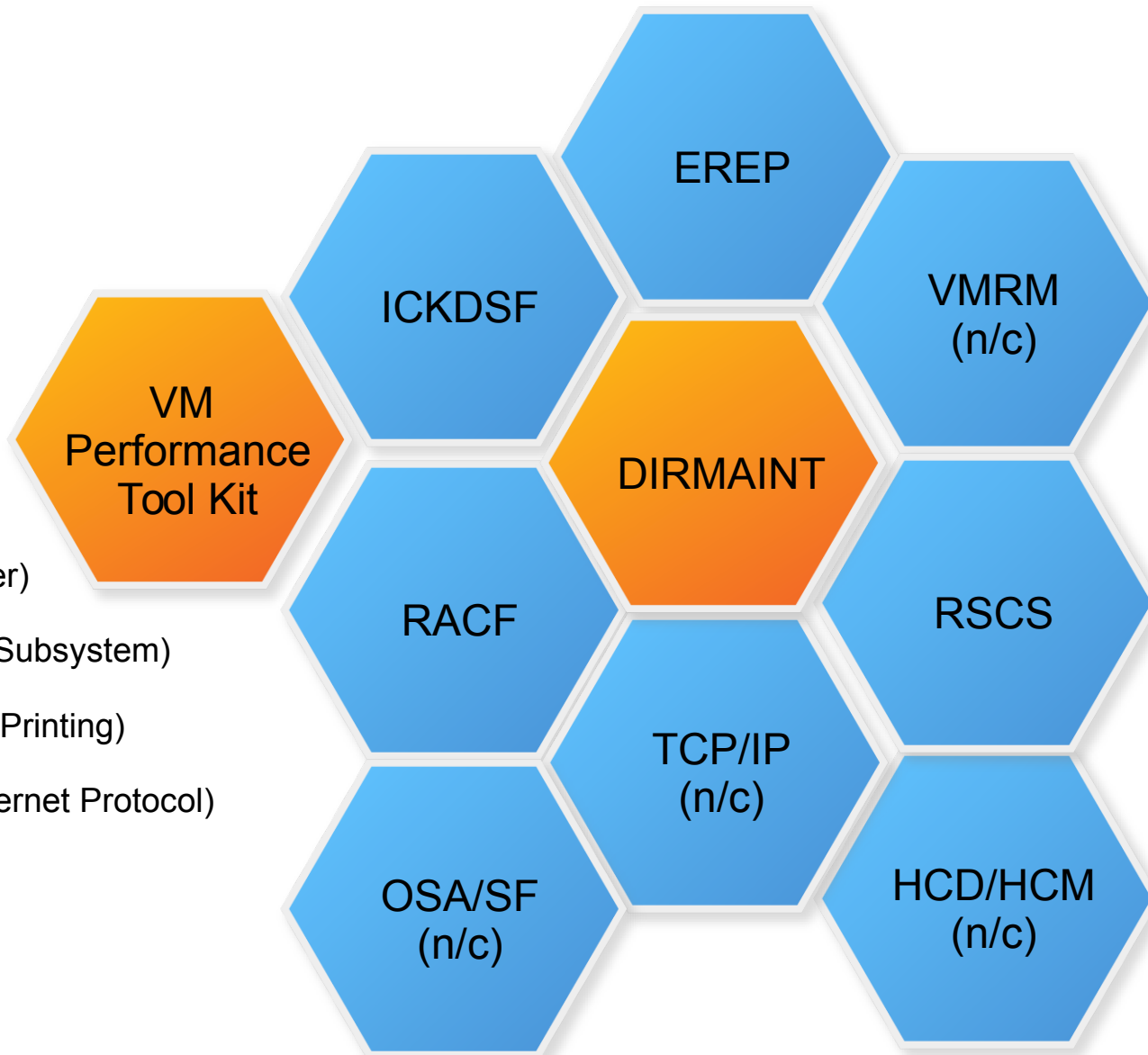
HCM (Hardware Configuration Manager)

VMRM (Virtual Machine Resource Manager)

RSCS (Remote Spooling Communication Subsystem)

EREP (Environmental Record Editing and Printing)

TCP/IP (Transmission Control Protocol/Internet Protocol)



# Warum und wie setzt man z/VM ein?

---

## **z/VM für den Betrieb von VSE oder z/OS**

- ◆ Möglichkeit von Testmaschinen (z. B. Für Umstellung)
- ◆ Trennung von online- und Batch-Verarbeitung
- ◆ Möglichkeit der Nutzung anderer Betriebssysteme, die ein zusätzliches Systemsteuerprogramm erfordern
- ◆ Erhöhung der Verfügbarkeit einzelner Anwendungen
- ◆ Erhöhung der Sicherheit
- ◆ Adressraumerweiterung
- ◆ Development und Debugging

## **Einsatz von CMS (Conversational Monitor System)**

- ◆ Automation – grösste Flexibilität mit CMS und REXX (**RE**structured **eX**ended **eX**ecutor Interpreter = Scriptsprache)
- ◆ Dialogprogrammierung
- ◆ Revisionsgerechte Programmänderung
- ◆ Arbeitsvorbereitung
- ◆ Anwendungen in den Fachabteilungen (z. B. Technische Berechnungen, Textverarbeitung, Abfragesprachen)
- ◆ CMS als Schulungssystem

## **z/VM für den Betrieb von Linux**

- ◆ Einrichten eines Linux-Systems in Minuten!
- ◆ Virtuelle Server Farmen auf einer Maschine
- ◆ Bessere Ressourcen Nutzung
- ◆ Flexiblere Server Wartung
- ◆ Hochverfügbarkeit mit Hot Stand-by

# PR/SM LPAR vs. z/VM Guests

---

## PR/SM **LPAR**

- ◆ Hardware & Firmware function
- ◆ Dedicated or shared PUs
- ◆ Dynamic PU add
  - ◆ Pre-planning required, OS support required
- ◆ Dynamic memory add
  - ◆ Pre-planning required, OS support required
- ◆ **No memory sharing**
  - ◆ Each LPAR has its own dedicated memory
- ◆ **Highest Isolation level: CC EAL 5+ \***
- ◆ Dedicated or shared devices
- ◆ Hardware devices must be physically addressed

## **z/VM Guest**

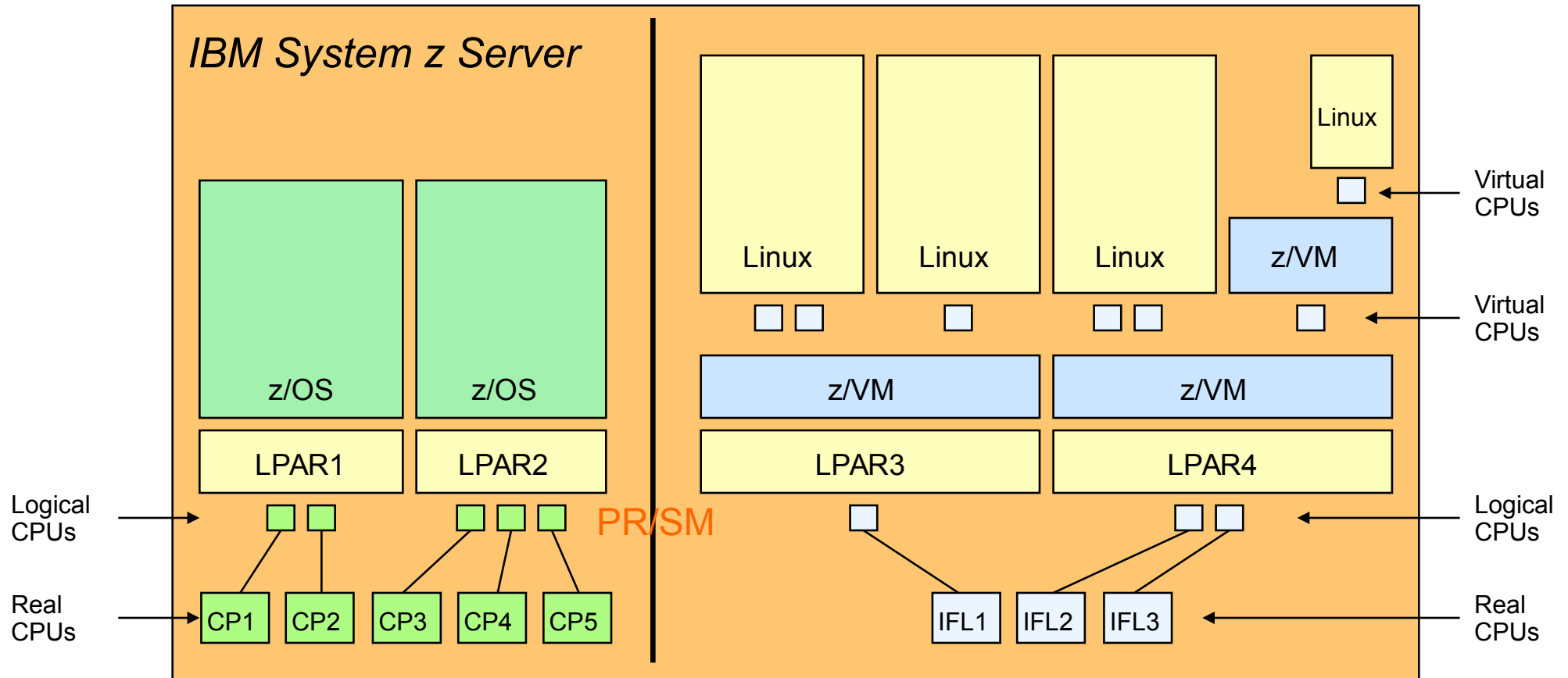
- ◆ Hardware & Firmware **& SW function**
- ◆ Dedicated or shared PUs
- ◆ Dynamic PU add
  - ◆ Pre-planning required, OS support required
- ◆ Dynamic memory add
  - ◆ Pre-planning required, OS support required
- ◆ **Memory sharing**
  - ◆ Memory over-commitment is possible
- ◆ **High Isolation level: CC EAL 4+ \***
- ◆ Dedicated or shared devices
- ◆ Hardware devices can be addressed physically, **but virtual addressing is also possible**
- ◆ **Additional Virtual devices**
  - ◆ Printer/Reader/Puncher
  - ◆ VDisk
  - ◆ Network (VSwitch, etc.)
  - ◆ Tapes
- ◆ **Additional Options, like:**
  - ◆ Single System Image (SSI) with
  - ◆ Live Guest Relocation (LGR)

Machines can be setup in a way that they can be started in an LPAR or z/VM environment.

\* <https://www.bsi.bund.de/ContentBSI/EN/Topics/Certification/newcertificates.html>

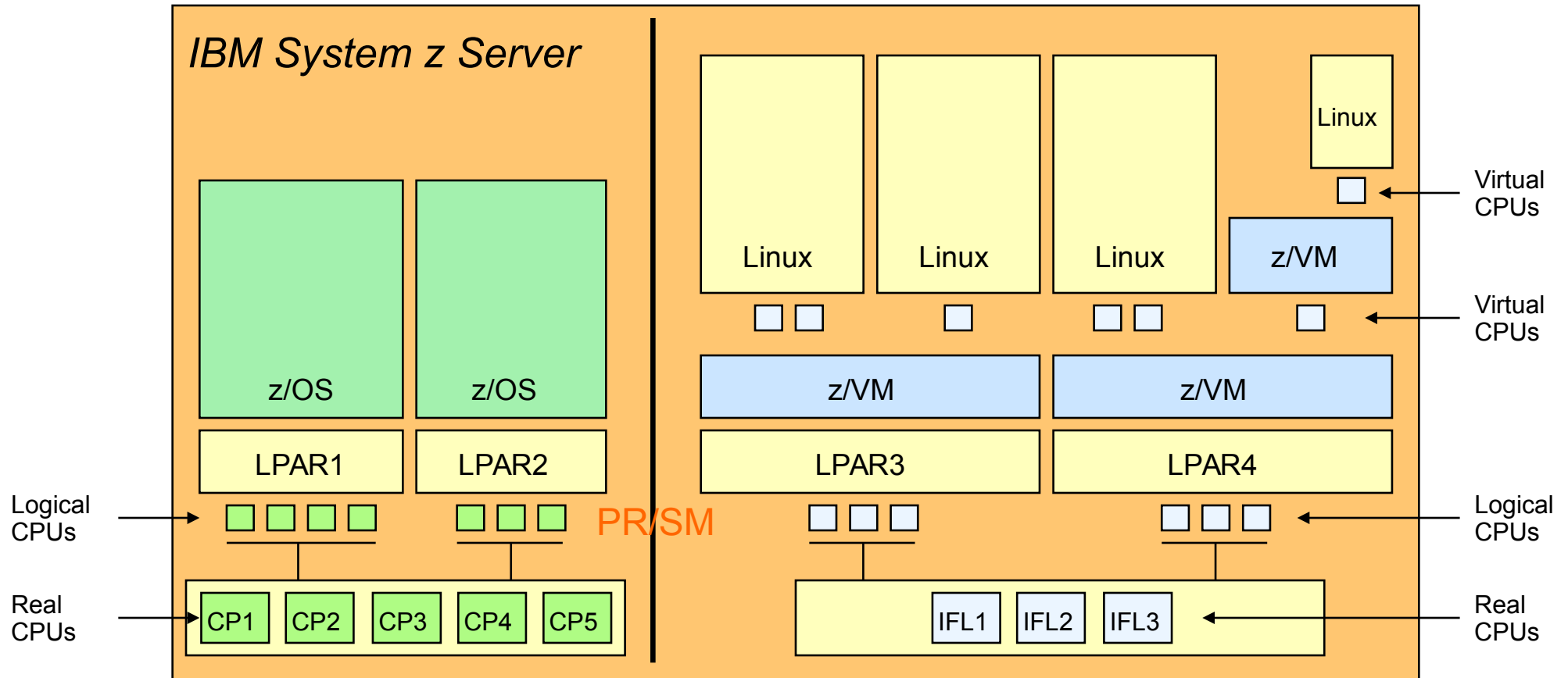
# Real, Logical, and Virtual Processors

## Dedicated Real CPU Example



# Real, Logical, and Virtual Processors

## Shared Real CPU Example



PR/SM: Processor Resource/System Manager

# z/VM CPU Resource Controls

## High Granular Sharing of System Resources

- Allocate system resources per guest image using SHARE command

- This is a highly flexible and self-managed function of the z/VM Control Program

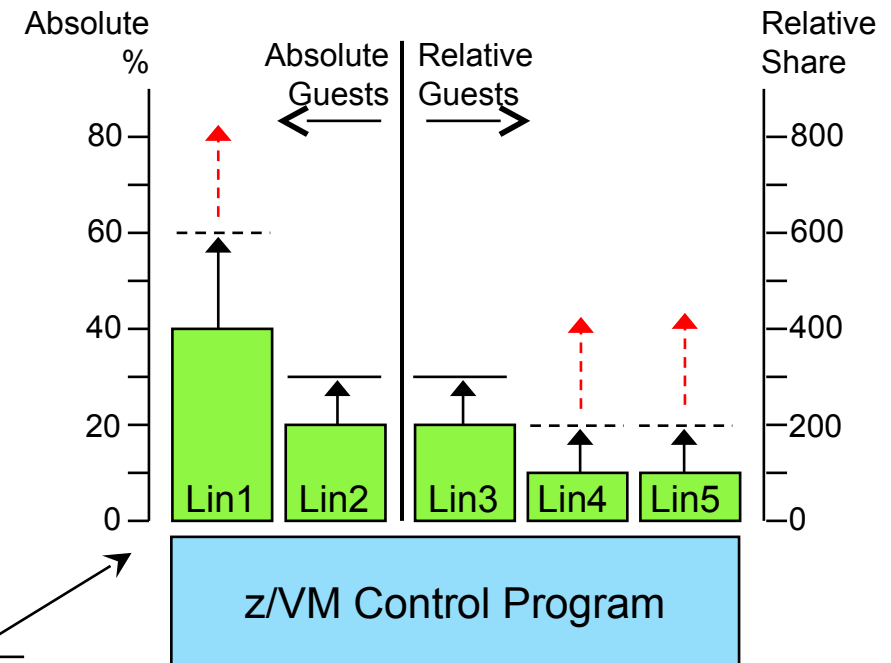
- Reserve CPU capacity for peak usage

- Use it when needed
- Relinquish the processor cycles for other servers when not needed

- "Absolute guests" receive top priority

- The **Virtual Machine Resource Manager** can be used to monitor and adjust remaining capacity allocated to "Relative guests"

- Also use VMRM to prioritize I/O operations among guest images via "I/O Priority Queuing"



```
SHARE Lin1 ABSOLUTE 40% ABSOLUTE 60% LIMITSOFT
SHARE Lin2 ABSOLUTE 20% ABSOLUTE 30% LIMITHARD
SHARE Lin3 RELATIVE 200 RELATIVE 300 LIMITHARD
SHARE Lin4 RELATIVE 100 RELATIVE 200 LIMITSOFT
SHARE Lin5 RELATIVE 100 RELATIVE 200 LIMITSOFT
```

### Notes:

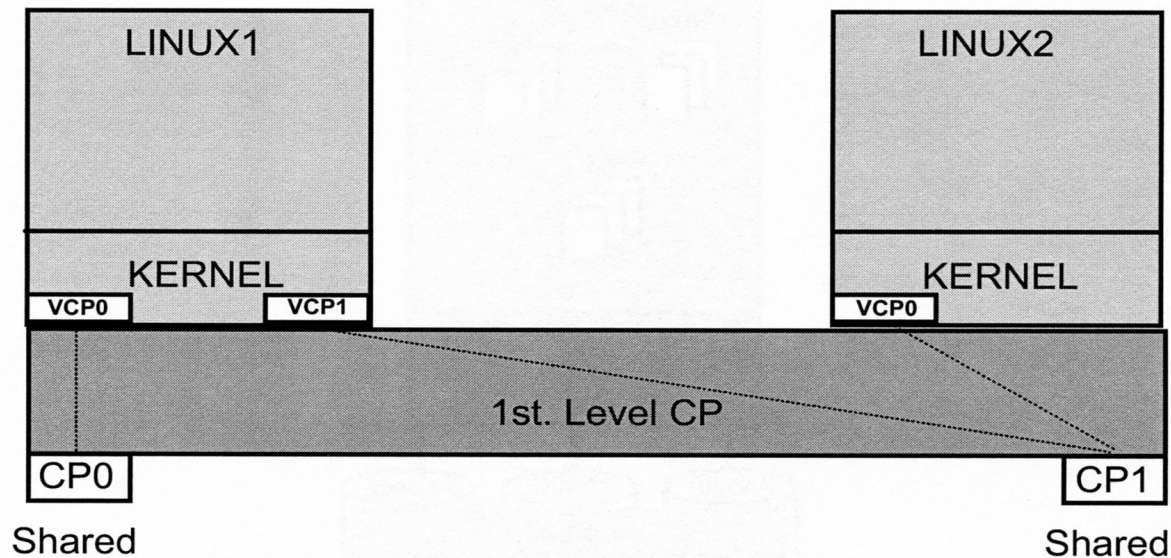
----- = limit can be exceeded if unused capacity is available (LIMITSOFT)

———— = limit will not be exceeded (LIMITHARD)

# Aufgaben des CP (Control Program)

---

Die grundlegende Rechnersteuerung erfolgt nach einem Zeitscheibenverfahren. Dabei werden nach bestimmten Spielregeln den einzelnen virtuellen Maschinen Rechnerzeiten und Prioritäten entsprechend ihrem Bedarf (Fair Share Algorithm) zugeteilt.



Aktiver Prozess wird vom Prozessor verdrängt zugunsten eines bereiten Prozesses:

Typischerweise nach Ablauf einer bestimmten, dem Prozess zugeteilten Zeit, genannt Zeitscheibe (dispatch time slice max. 5ms), deren Ablauf gemeldet wird durch Zeitgeberunterbrechung (timerinterrupt).

# Was ist CMS?

```
Ready; T=0.01/0.01 14:12:56
q stor
STORAGE = 16G CONFIGURED = 16G INC = 256M STANDBY = 0 RESERVED = 0
Ready; T=0.01/0.01 14:13:02
q disk
LABEL  VDEV M  STAT   CYL TYPE BLKSZ   FILES  BLKS USED-(%) BLKS LEFT  BLK TOTAL
MNT191 191  A   R/W   175 3390 4096     92    353-01    31147    31500
MNT5E5 5E5  B   R/W    9 3390 4096    129   1209-75     411    1620
MNT2CC 2CC  C   R/W    5 3390 4096    60    600-67     300     900
MNT51D 51D  D   R/W   26 3390 4096   303   1379-29    3301    4680
MNT190 190  S   R/D   100 3390 4096   694   15028-83   2972   18000
MNT19E 19E  Y/S R/D   250 3390 4096  1021  28254-63  16746  45000
Ready; T=0.01/0.01 14:13:07
q cpu
CPUID = FF026A1C20978000
Ready; T=0.01/0.01 14:13:10
q processor
PROCESSOR 00 MASTER CP
PROCESSOR 01 ALTERNATE IFL
PROCESSOR 02 ALTERNATE IFL
PROCESSOR 03 ALTERNATE IFL
PROCESSOR 04 ALTERNATE IFL
PROCESSOR 05 ALTERNATE IFL
PROCESSOR 06 ALTERNATE IFL
PROCESSOR 07 ALTERNATE IFL
Ready; T=0.01/0.01 14:13:17
q vswitch
VSWITCH SYSTEM VSW1      Type: VSWITCH Connected: 1      Maxconn: INFINITE
PERSISTENT RESTRICTED   NONROUTER      Accounting: OFF
VLAN Unaware
MAC address: 02-00-01-00-00-01
State: Ready
IPTimeout: 5             QueueStorage: 8
Isolation Status: OFF
RDEV: 4004.P00 VDEV: 4004 Controller: DTCVSW2
RDEV: 4200.P00 VDEV: 4200 Controller: DTCVSW1 BACKUP
Ready; T=0.01/0.01 14:13:57
```

RUNNING ZVM01

042/001

## CMS (Conversational Monitor System)

ist ein speziell für die Arbeit unter dem CP zugeschnittenes **Betriebssystem**.

Es wird im Dialog durch Befehle gesteuert, die der Benutzer am Bildschirm eingibt.

Entwickelt wurde es in einem Labor von IBM, dem Cambridge Scientific Center, daher stand die Abkürzung CMS anfangs auch für Cambridge Monitor System. Ab 1972 wurde es von IBM vermarktet.

Der Zugriff auf das System erfolgt mittels Terminals, wie dem IBM 3270.

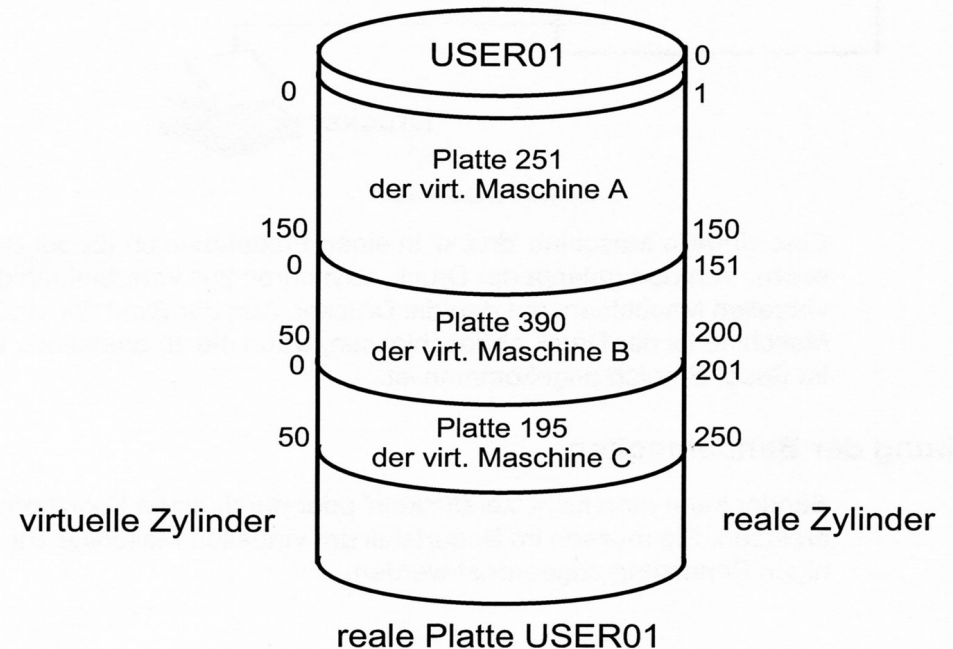
Es wird hauptsächlich zur Softwareentwicklung verwendet, dazu stehen Programmiersprachen wie Fortran, PL/I oder APL zur Verfügung. Ebenso die Batch-Sprachen EXEC und REXX, sowie der Editor XEDIT.

# Das Minidisk-Konzept

Wenn jede virtuelle Maschine mit einer dedizierten Platte arbeiten würde, könnten gleichzeitig nur so viele Benutzer aktiv sein, wie reale Platten im System vorhanden sind. Dies würde eine starke und unnötige Beschränkung der Anzahl von aktiven virtuellen Maschinen bedeuten. Die meisten virtuellen Maschinen benötigen gar keine ganze Platte; sie sind bereits mit einigen Zylindern zufrieden.

Unter VM kann ein realer Plattenstapel in eine Anzahl von Bereichen aufgeteilt werden, von denen jeder Bereich eine Virtuelle Platte darstellt, genannt 'Minidisk'.

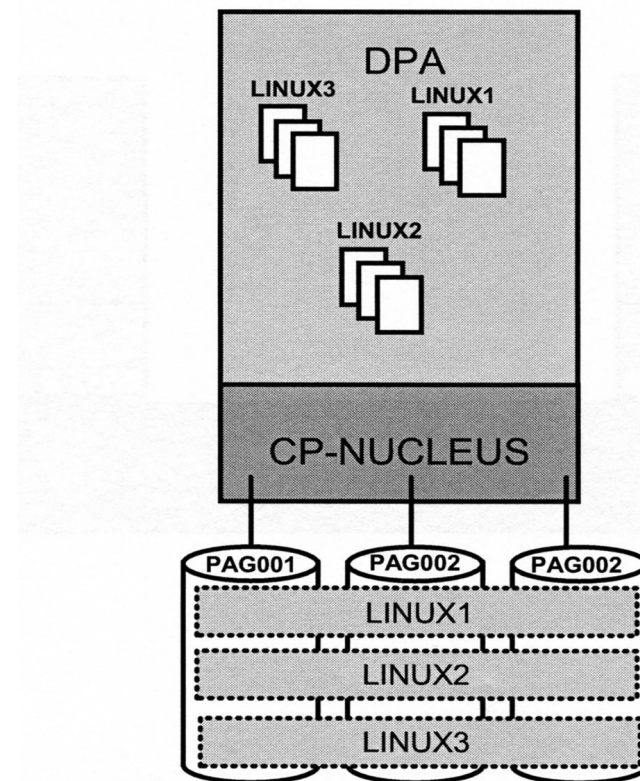
Die Minidisks eines Plattenstapels können zu einer oder zu mehreren virtuellen Maschinen gehören.



# Paging

Übersteigt die Summe des Speichers der einzelnen virtuellen Maschinen unter dem CP die Grösse des Realspeichers, werden momentan inaktive Hauptspeicherseiten auf einen externen Seitenspeicher ausgelagert und bei Bedarf wieder geladen.

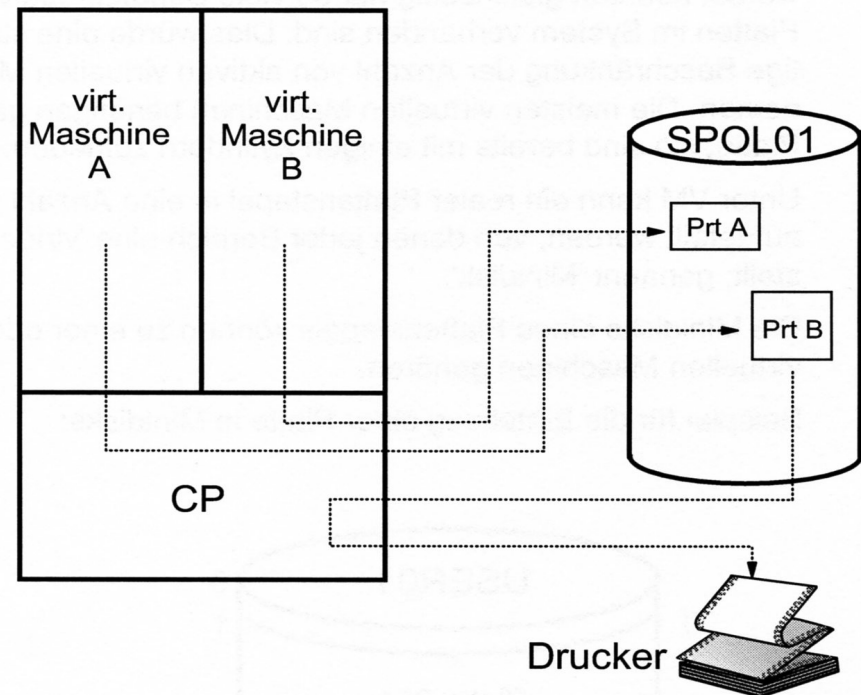
Dieser Vorgang wird mit **Paging** bezeichnet.



# Spooling

Eine virtuelle Maschine 'druckt' in einen Plattenbereich (Spool-Bereich). Von dort gelangt der Druck, asynchron zur Verarbeitung der virtuellen Maschinen, auf den/die Drucker. Aus Sicht der virtuellen Maschine ist der Druck abgeschlossen, wenn die zu druckende Datei im Spool-Bereich angekommen ist.

Bänder kann man nicht 'zerstückeln' oder durch einen Spoolbereich ersetzen. Sie müssen im Bedarfsfall der virtuellen Maschine zur alleinigen Benutzung zugeordnet werden.



# Howto: Das VM User Directory

---

## Definitions of:

- memory
- architecture
- processors
- spool devices
- network device
- disk devices
- other attributes

```
USER CCM01 MYPASS 512M 1024M G
MACHINE ESA
IPL 100 PARM AUTOOCR
CONSOLE 009 3270 A
SPOOL 00C 2540 READER *
SPOOL 00D 2540 PUNCH A
SPOOL 00E 1403 A
NICDEF 600 TYPE QDIO LAN SYSTEM VSW1
LINK MAINT 190 190 RR
LINK MAINT 19D 19D RR
LINK MAINT 19E 19E RR
MDISK 0191 3390 2371 50 TM111A
```

# Startprozedur

---

## POR

**P**ower **o**n **R**eset is the initial start of the System z hardware itself.

+

## IML

**I**nitial **M**achine **L**oad or **I**nitial **M**icrocode **L**oad

Power on and configure processor complex

VM equivalents are:

- **LOGON** uses the **MACHINE** statement in the CP directory entry
- The **CP SET MACHINE** command

Analogous to LPAR *image activation*



## IPL

**I**nitial **P**rogram **L**oad

Like *booting* a operating system (CMS, Linux, etc.)

System z hardware allows you to *IPL* a system

z/VM allows you to *IPL* a system in a virtual machine via the CP IPL command

Linux *kernel* is like VM *nucleus*

Analogous to the LPAR *LOAD* function

# Howto: CP Commands

---

## CP DEFINE

*Adds to the virtual configuration somehow*

CP DEFINE STORAGE

CP DEFINE PROC

CP DEFINE {device} {device\_specific\_attributes}

## CP ATTACH

*Gives an entire real device to a virtual machine*

## CP DETACH

*Removes a device from the virtual configuration*

## CP LINK

*Lets one machine's disk device also belong to another's configuration*

## CP SET

*Change various characteristics of virtual machine*

*Changing the virtual configuration after logon is considered normal*

*Usually the guest operating system detects and responds to the change*

# VMRM-CMM

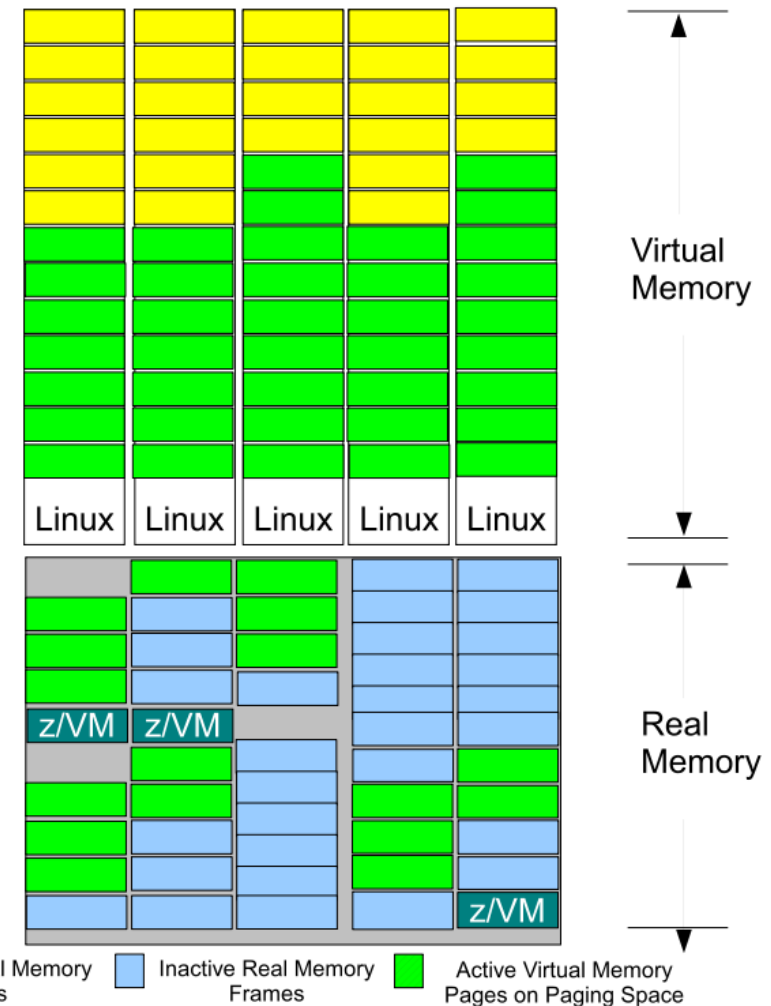
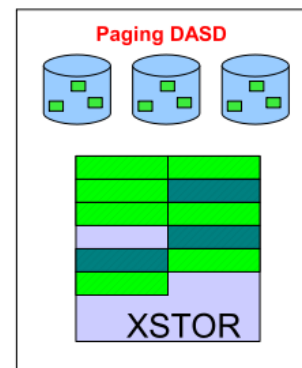
## VM Resource Manager Cooperative Memory Management

The z/VM system maps the guests' virtual memory into the real memory of the System z machine per page frame.

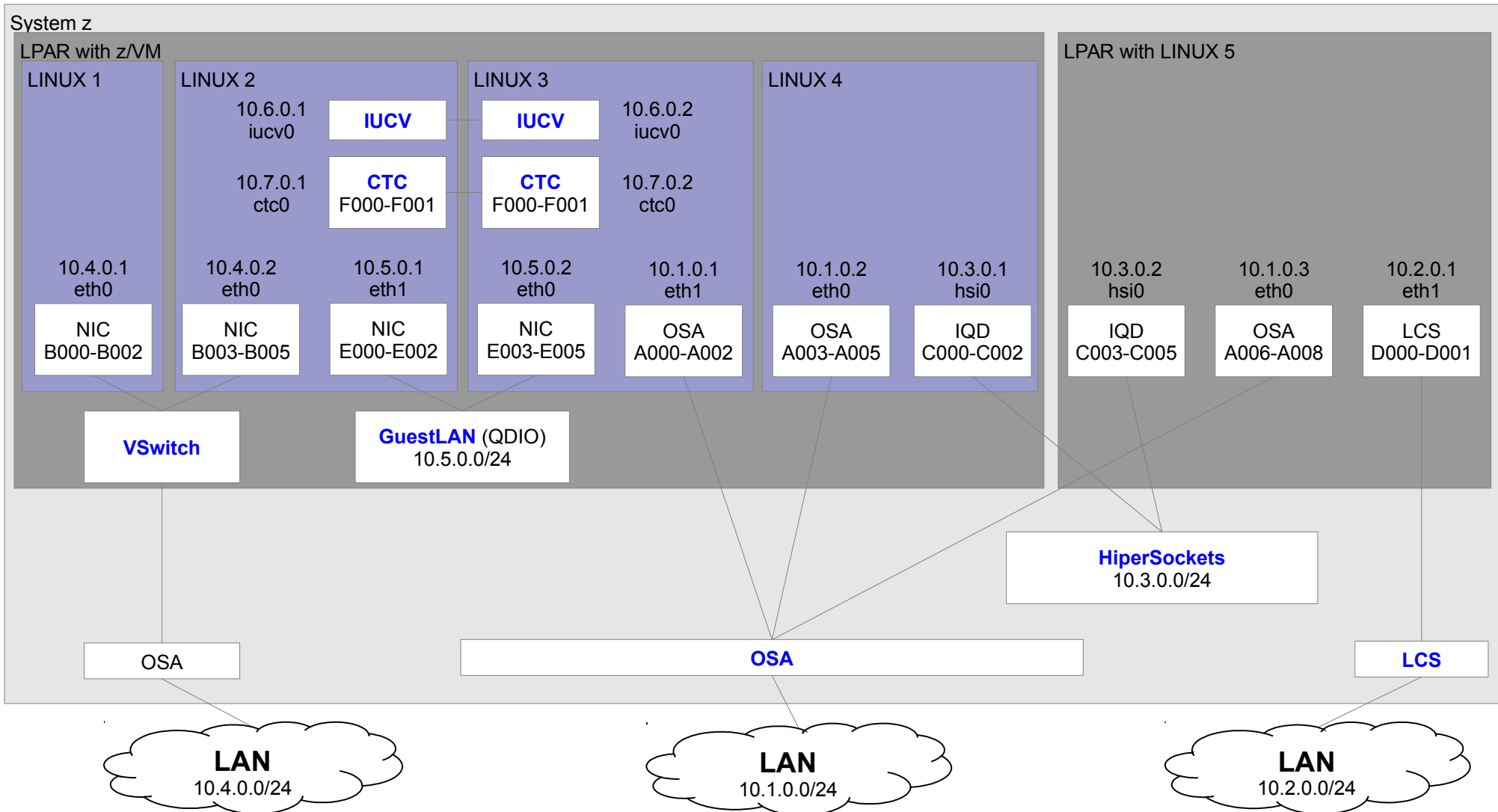
If there are not enough real memory frames to contain all the required active guests' virtual memory pages, the active guests' virtual pages are moved to expanded storage (xstor).

Once xstor becomes full, the guests' pages are migrated from xstor to DASD paging space. As the number of servers increases in a z/VM system, memory management overhead increases due to increased paging.

z/VM Page Management



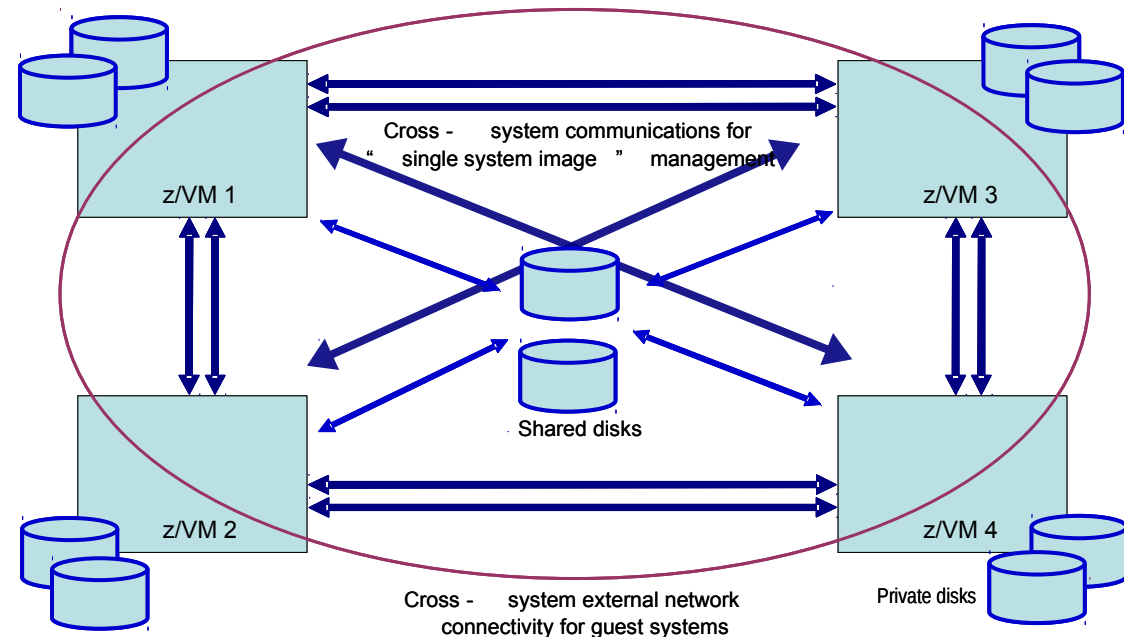
# System z Network Connectivity & Virtualization



# Single System Image Feature

## Clustered Hypervisor with Live Guest Relocation

- Connect up to four z/VM systems as members of a Single System Image (SSI) cluster
- Provides a set of shared resources for member systems and their hosted virtual machines
- Cluster members can be run on the same or different System z servers
- Simplifies systems management of a multi-z/VM environment
  - Single user directory
  - Cluster management from any member
    - Apply maintenance to all members in the cluster from one location
    - Issue commands from one member to operate on another
  - Built-in cross-member capabilities
  - Resource coordination and protection of network and disks
- Provided as an optional priced feature.



# 3270 Terminals

The screenshot displays a 3270 terminal session titled "Vista Session A". The main window shows a "VM Read" screen with the text "PM vm.marist.edu" and "VM Read" in green. A keyboard overlay is visible on the right side of the screen, featuring various function keys (PF1-PF12), arrow keys, and other controls. The terminal window also shows a status bar at the bottom with "RUNNING ZVM01" and "039/017".

The terminal window title is "Vista Session A". The menu bar includes File, Edit, Font, Transfer, Macro, Options, Window, and Help. The toolbar contains various icons for file operations and editing. The status bar at the top shows "Ready; T=0.08/0.09 14:12:05".

The terminal window title is "x3270-4 localhost:10023". The menu bar includes File, Options, and Macros. The status bar at the top shows "z/VM ONLINE".

The terminal window displays a screen with the following text:

```
ZZZZZZ / VV VVV MM MM
ZZ / VV VVV MMM MMM
ZZ / VV VVV MMM MMM
ZZ / VV VVV MM MM MM
ZZ / VVVV MM M MM
ZZ / VVV MM MM
ZZZZZZ / V MM MM
```

built on IBM Virtualization Technology

Fill in your USERID and PASSWORD and press ENTER  
(Your password will not appear when you type it)  
USERID ==>  
PASSWORD ==>  
COMMAND ==>

RUNNING ZVM01  
039/017

The keyboard overlay includes the following keys:

PF1	PF2	PF3
PF4	PF5	PF6
PF7	PF8	PF9
PF10	PF11	PF12
↑		
←	↶	→
⌂	↓	⌂
PA1	PA2	PA3
← →		
Clear	Reset	
Erase EOF	Erase Input	
Dup	Field Mark	
Sys Req	Cursor Select	
Attn	Compose	
↶	Enter	

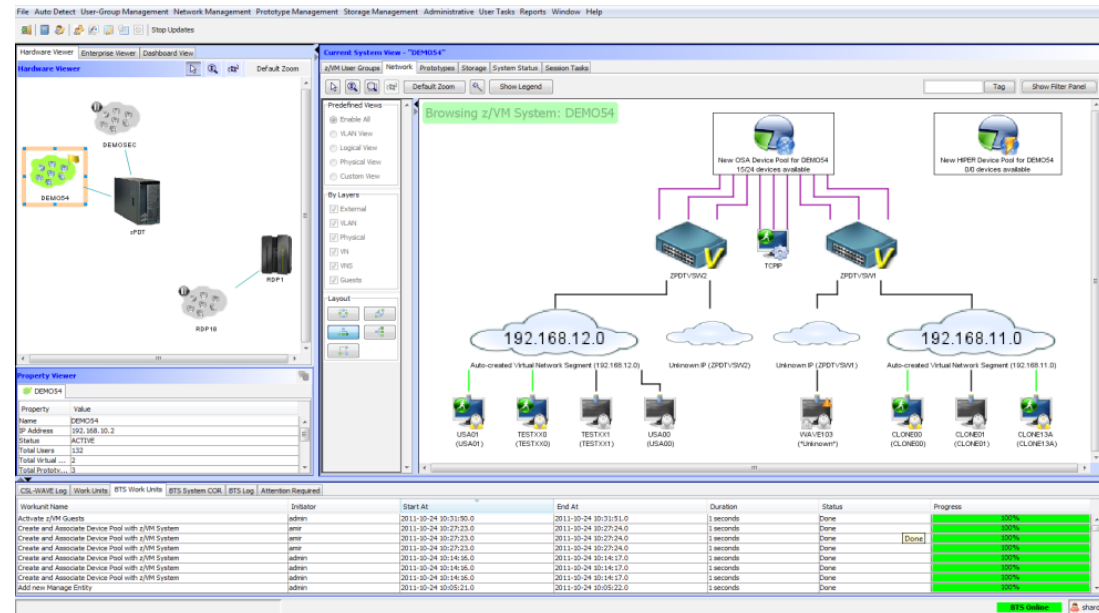
The "ShowKeyPad" button is highlighted in yellow. The "Vista Session A" window also shows a "ShowKeyPad" button.



# IBM Wave for z/VM (Web Administration for VM Environments)

IBM Wave provides a graphical interface that simplifies and helps to automate the management of z/VM and Linux on System z virtual servers.

- **Monitors and manages virtual servers and resources** from a single graphical interface
- **Simplifies and Automates** tasks
- **Provisions virtual resources** (Guests, Network, Storage)
- **Supports advanced z/VM capabilities** such as  
Single System Image (SSI)  
and Live Guest Relocation (LGR)
- **Allows delegation of administrative capabilities** to the appropriate teams
- **Competitive and comparable** to other virtualization **center** solutions

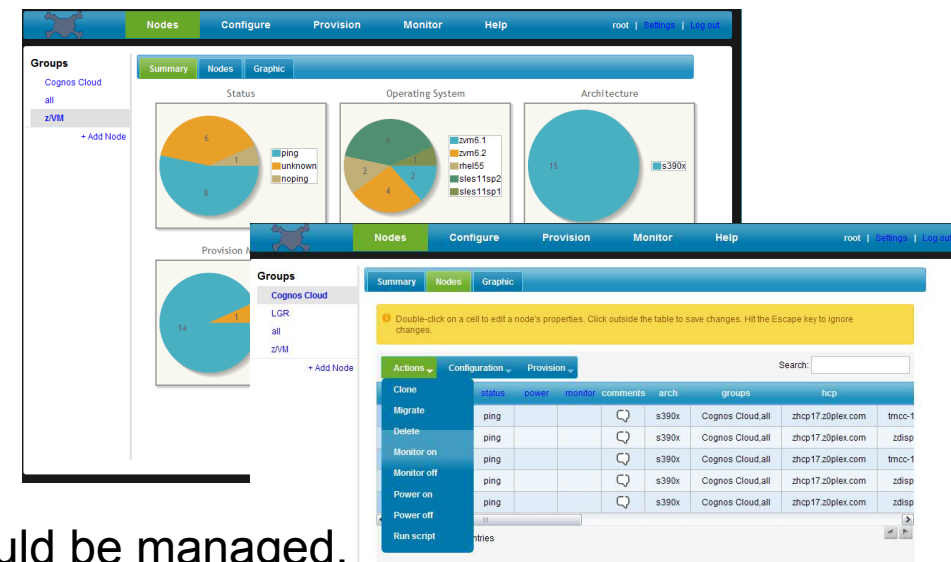


*A simple, intuitive graphical tool providing management, provisioning, and automation for a z/VM environment, supporting Linux virtual servers.*


# xCAT (extreme Cloud Administration Toolkit)



- Tool to manage, provision, and monitor physical and virtual machines (x, p and z)
- Open sourced in 2007 by IBM and licensed as EPL (Eclipse Public License)
- Shipped as an optional part of z/VM 6.3
- It communicates with z/VM over VMCP and SMAPI
- Allows customers to set up and administer a rudimentary cloud environment on z/VM only, without purchasing any other cloud management software
- Consists of two or more components: zHCP (one per HV) and one MN
- Scalable open source toolkit that includes:
  - Provisioning and de-provisioning of guests
  - Monitor physical and virtual resources
  - Provide network, storage and image management
- No upgrade path to the IBM Cloud suite
- xCAT has four different interfaces
  - REST (Representational State Transfer) APIs
  - Browser based Graphical User Interface
  - Command Line Interface (CLI)
  - XML
- In case other environments besides z/VM should be managed, an full open source version of xCAT is available here:  
[http://sourceforge.net/apps/mediawiki/xcat/index.php?title=Main\\_Page](http://sourceforge.net/apps/mediawiki/xcat/index.php?title=Main_Page)



# Monitoring – Performance Toolkit



**General User Resource Utilization (TMCCXDR)**

Select a user for user details or [IDLEUSER](#) for a list of idle users

Command
Refresh
Systems
Menu
Help
☐ Auto-Refresh

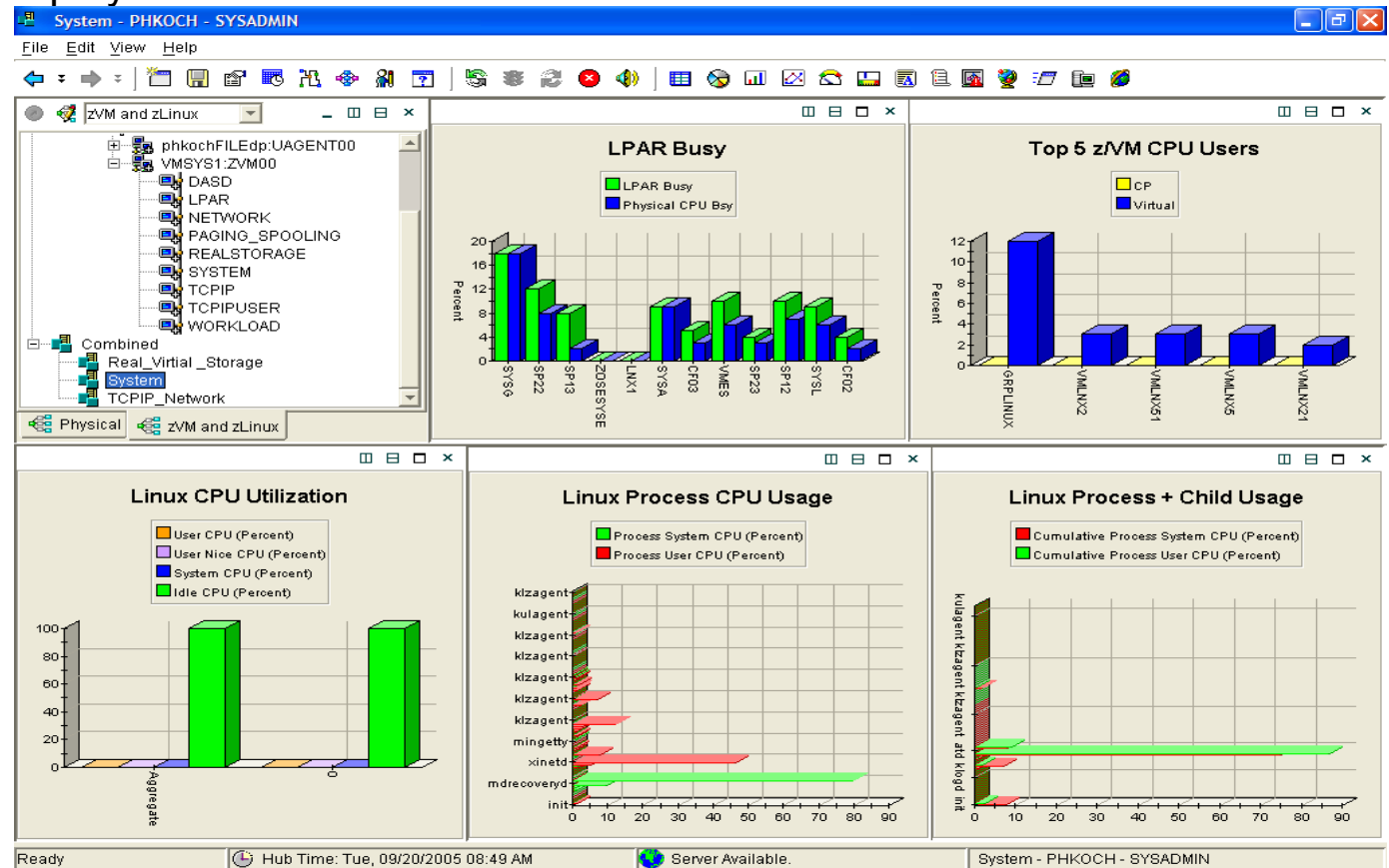
Interval 13:32:54-13:33:54, on 2005/06/27 (CURRENT interval, select [interim](#) or [average](#) data)

Userid	<----- CPU Load ----->	<----- Virtual IO/s ----->	<----- User Time ----->
	<--Seconds--> T/V		<--Minutes-->
	%CPU TCPU VCPU Ratio	Total DASD Avoid Diag98 UR Pg/s	User Status Logged Active
>System<	6.25 3.747 3.723 1.0	2.5 2.5 .0 .0 .0 .0	---,---,--- 1.0 .4
User Class Data:			
GENERAL	6.25 3.747 3.723 1.0	2.5 2.5 .0 .0 .0 .0	---,---,--- 1.0 .4
User Data:			
DATAMOVE	0 0 0 ...	0 0 0 0 0 0	ESA,---,DORM 1 0
DIRMAINT	0 0 0 ...	0 0 0 0 0 0	ESA,---,DORM 1 0
DISKACNT	0 0 0 ...	0 0 0 0 0 0	ESA,---,DORM 1 0
EREP	0 0 0 ...	0 0 0 0 0 0	ESA,---,DORM 1 0
MAINT	0 0 0 ...	0 0 0 0 0 0	ESA,---,DORM 1 0
OPERATOR	0 0 0 ...	0 0 0 0 0 0	ESA,---,DORM 1 0
OPERSYMP	0 0 0 ...	0 0 0 0 0 0	ESA,---,DORM 1 0
PERFSVM	.04 .022 .016 1.4	.8 .8 .5 .0 .0 .0	ESA,---,DORM 1 1
PORTMAP	0 0 0 ...	0 0 0 0 0 0	ESA,---,DORM 1 0
TCPCTRL1	.00 .000 .000 ...	.0 .0 .0 .0 .0 .0	ESA,---,DORM 1 1
TCPCTRL2	.02 .009 .000 ...	.0 .0 .0 .0 .0 .0	ESA,CL0,DISP 1 1
TCPIP	.05 .027 .017 1.6	.1 .0 .0 .0 .0 .0	ESA,CL0,DISP 1 1
TMCCXDR0	2.86 1.715 1.601 1.1	5.7 5.7 .0 .0 .0 .0	ESA,CL3,DISP 1 1
TMCCXDR1	95.0 56.99 56.77 1.0	31.0 31.0 .0 .0 .0 .0	ESA,CL3,DISP 1 1
TMCCXDR2	1.98 1.190 1.157 1.0	2.4 2.4 .0 .0 .0 .0	EME,CL3,DISP 1 1
VMSERVE	0 0 0 ...	0 0 0 0 0 0	ESA,---,DORM 1 0

# Monitoring

## IBM Tivoli OMEGAMON XE on z/VM and Linux

- Combined product offering that monitors z/VM and Linux for System z
- Provides work spaces that display:
  - Overall system health
  - Workload metrics for logged-in users
  - Individual device metrics
  - LPAR data
- Provides composite views of Linux running on z/VM
- New function in V4.1.2:
  - Additional monitoring to help identify bottlenecks in the I/O subsystem
  - Processor spin lock wait statistics



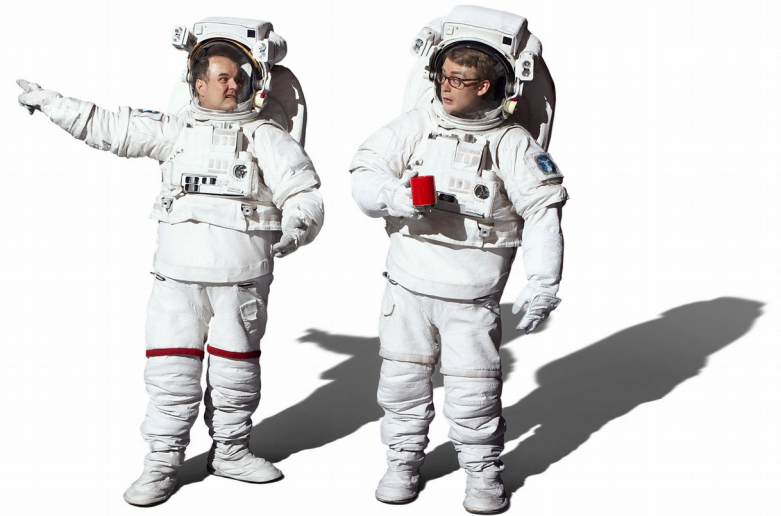
Learn more at: [ibm.com/software/tivoli/products/omegamon-xe-zvm-linux](http://ibm.com/software/tivoli/products/omegamon-xe-zvm-linux)



„Wie lautet die kürzeste EDV-Lüge? Es funktioniert...“



„Mouse not found, please click to continue!“



I ♥  
Linux

---

## Bonus Material

# System z Terminologie...

---

x86, UNIX, etc.	System z
Memory	Storage (though we are moving toward "memory")
Disk, Storage	DASD – Direct Access Storage Device
Processor	Central processor) SAP (system assist processor) Specialty engines <ul style="list-style-type: none"><li>–IFL (Integrated Facility for Linux)</li><li>–zAAP (System z Application Assist Processor)</li><li>–zIIP (System z9 Integrated Information Processor)</li><li>–ICF (Internal coupling Facility)</li></ul>
Computer	CEC (central electronics complex)Server

# Virtual Machine Modes (Architectures)

---

An *architecture* is a formal set of rules for how a computer operates  
VM has kept pace with the evolution of IBM mainframe architecture

ESA

ESA/390 or z/Architecture if running on System z processor  
*SIGP Set Architecture* order must be issued for z/Architecture  
ESA/390 when running on ESA/390 processor

XC

ESA/XC is unique to z/VM virtual machines (DAT-off use of AR mode)

XA

Processes the same as ESA mode (compatibility with older VM releases)

370

No longer supported as a virtual machine mode

Processes according to ESA/370 architecture

CP and CMS provide 370 Accommodation features to help run 370 applications  
in ESA, XA, and XC modes (DAT off)

# Other Processor Resources

---

## Registers

General purpose, control, access, and floating point

- CP saves and restores between invocations of SIE
- Manipulation of control registers sometimes requires CP's involvement (SIE exit)

## Timers

CPU timer

Clock comparator

## Virtualized TOD clock

- SET VTOD command to set virtual machine TOD clock to a specific value or to that of another virtual machine

## Storage Keys

PSW, interrupts, prefixing, and other architected structures

# Saved Segment and NSS Support

---

DCSS (Discontiguous Saved Segments)

Defines an address range (MB boundary) to the system

A single copy is shared among all guests

Guest "loads" the DCSS (maps DCSS into its address space)

- Can be located outside guest's defined storage

DAT lets this work with minimal CP involvement

Contains:

- Data (e.g. file system control blocks)
- Code (e.g. CMS code libraries)

NSS (Named Saved Systems)

An IPL-able saved segment

Great for CMS or for Linux

- 1 shared copy on system for N guests, instead of N copies.
- Faster boot

Special Cases

Writable by guest, or by CP

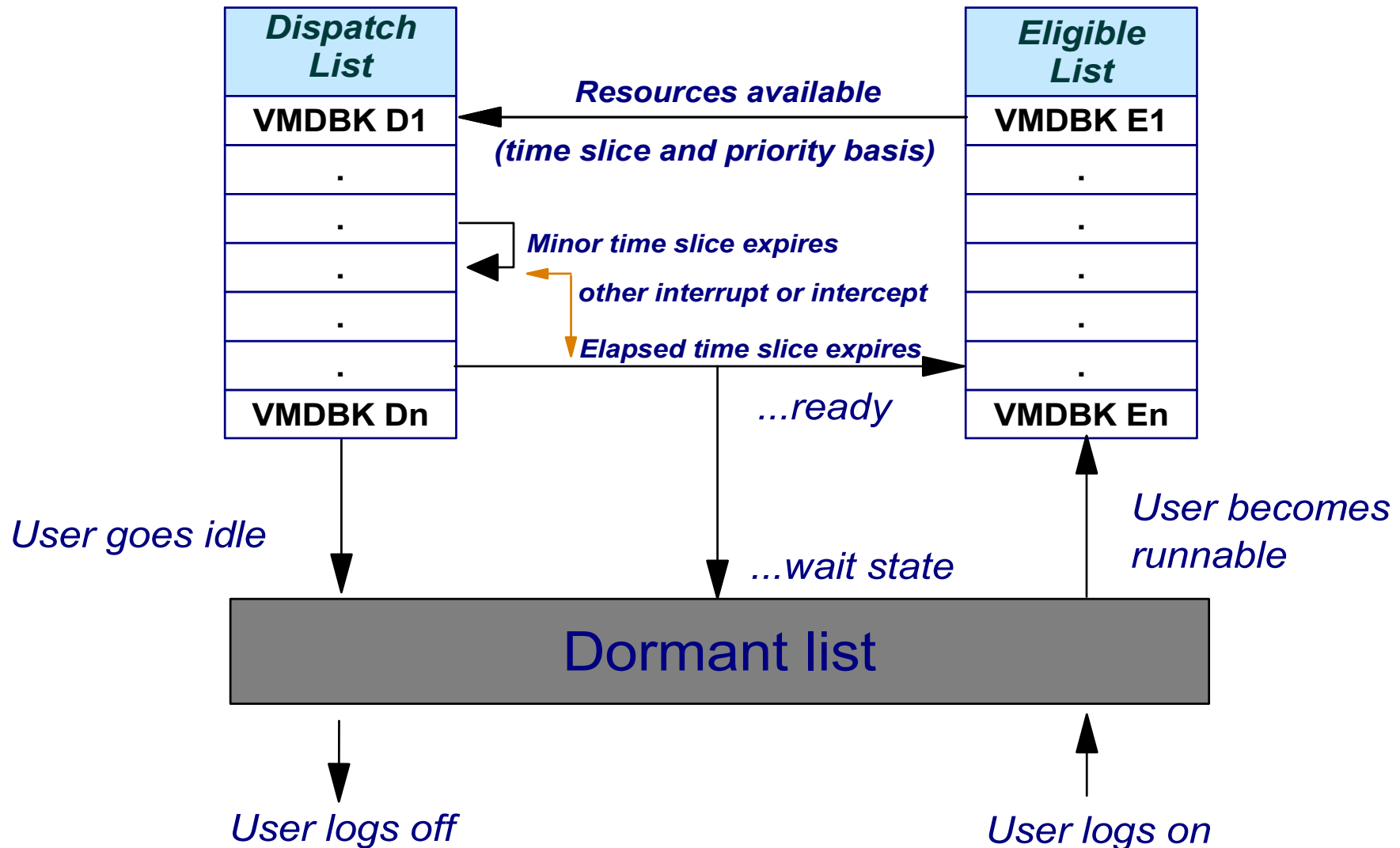
Restricted (sensitive data)

Can have both exclusive and shared ranges

# Virtual Machine Address Translation

<b>V=R</b> <b>(Virtual=Real)</b>	<b>V=F</b> <b>(Virtual=Fixed)</b>	<b>V=V</b> <b>(Virtual=Virtual)</b>
Fixed contiguous area of host real storage	Fixed contiguous area of host real storage	Does not map permanently to host real storage
Absolute page zero (low end of V=R area) – no address translation	High end of V=R area – never absolute page zero	Storage allocated from DPA
Not paged by CP	Not paged by CP	Guest real storage paged in and out of host real storage by CP
Automatic recovery	Automatic recovery	No automatic recovery
Preferred guest – CP provides performance benefits	Preferred guest – CP provides performance benefits	Not preferred
Only 1 may be logged on	Up to 6 may be logged on (or 5 plus 1 V=R)	Limited only by resources, design point of roughly 100,000
Not supported in z/VM V5	Not supported in z/VM V5 Not supported on z890/z990/z9	Available in z/VM V5

# Classic Scheduler / Dispatcher Picture



# Multiple Virtualization Layers

---

## Multiple Levels of SIE

Both z/VM and LPAR use SIE

z/VM running on LPAR = 2 levels of SIE

- No V=F support, and V=R loses I/O Assist
- Rest of SIE features can be *shared* without performance loss

z/VM running on z/VM on LPAR = 3 levels of SIE

- A layer of SIE now has to be virtualized
- Fairly expensive

2nd level (and 3rd level ...) Systems

Often used for testing purposes or disaster recovery

Most levels I ever saw was 9

Performance Data between Levels

LPAR and VM support Diagnose 204 to provide processor utilization to virtual servers supported

VM provides a Diagnose that a guest can use to pass data to the Control Program

VM provides Diagnoses for guest to gather some information

Anomalies in data when guest systems make poor assumptions (i.e. wall clock time = total processor time)

# Anomalies of Time

---

## VM virtualizes various timers or clocks

CPU timer – runs as processor time consumed

Time of day (TOD) clock

Clock comparator

## Anomaly

TOD always moves at wall clock speed

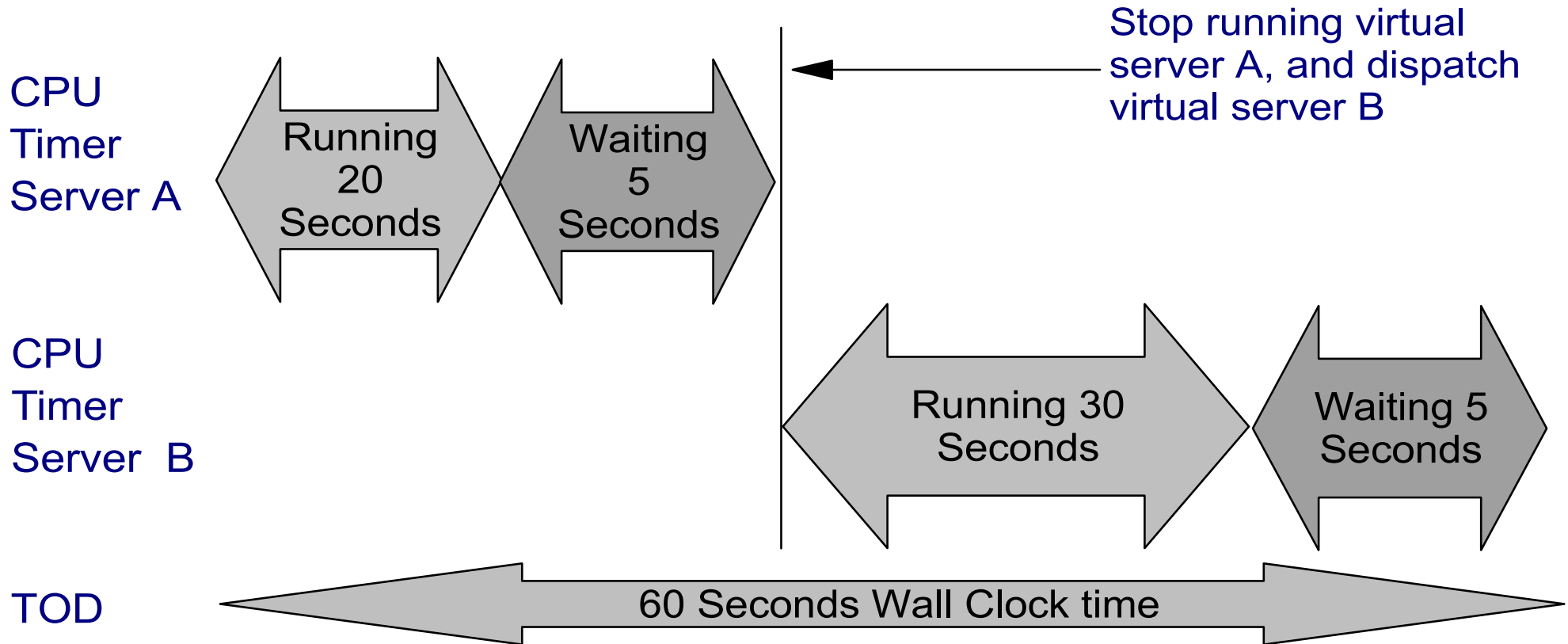
Virtual CPU timer “moves” slower as the sharing of the real processor increases

Problem when calculations assume CPU timer is moving at TOD clock speed

## LPAR

Same potential, but seldom shares processors to high enough degree to create drastic anomalies

# Anomalies of Time



Virtual Server	Total CPU Timer	CPU Timer 'busy'	Incorrect Utilization	Correct Utilization
A	25	20	80%	33%
B	35	30	86%	50%

# z/VM v6.2 News

---

## Single System Image & Live Guest Relocation (LGR) support

- **Improve business resilience for z/VM customers**
  - Addresses vertical capacity constraints with SSI as a cluster of 2, 3 or 4 z/VM systems
  - Make z/VM and hardware maintenance non-disruptive to virtual servers
  - Clustering approach preserves data and network integrity
- **Single image facilitates failover**
  - Definitive system status within cluster
  - Consistent guest definition across systems
  - Cross-system control
- **Facilitate horizontal growth of z/VM workloads**
  - Extend and improve current z/VM clustering technology
  - Remove requirement for additional software
  - Manage multiple z/VM images as a single resource pool
  - Ease deployment and maintenance of multiple images
- **Reinforce System z's leadership position in virtualization**

# Single System Image Feature

## Clustered Hypervisor with Live Guest Relocation

- Provided as an optional priced feature.
- Connect up to four z/VM systems as members of a Single System Image (SSI) cluster
- Provides a set of shared resources for member systems and their hosted virtual machines
- Cluster members can be run on the same or different System z servers
- Simplifies systems management of a multi-z/VM environment
  - Single user directory
  - Cluster management from any member
    - Apply maintenance to all members in the cluster from one location
    - Issue commands from one member to operate on another
  - Built-in cross-member capabilities
  - Resource coordination and protection of network and disks

