z/VM Virtualization Basics

WAVV 2004

Bill Bitner
bitnerb@us.ibm.com

Last Updated: April 24, 2004

Trademarks

IBM @server zSeries

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

CICS*
DB2
DB2 Connect
DB2 Universal Database
e-business logo*
FICON
HiperSockets
IBM*
IBM logo
MQSeries*
Multiprise*
OS/390
RISC
S/390
S/390 Parallel Enterprise Server*
Virtual Image Facility
VM/ESA*
VSE/ESA
WebSphere
z/OS
z/VM
zSeries

* Registered trademarks of the IBM Corporation

The following are trademarks or registered trademarks of other companies.

Lotus, Notes, and Domino are trademarks or registered trademarks of Lotus Development Corporation.
Tivoli is a trademark of Tivoli Systems Inc.
Linux is a registered trademark of Linus Torvalds.
Java and all Java-related trademarks and logos are trademarks of Sun Microsystems, Inc., in the United States and other countries
UNIX is a registered trademark of The Open Group in the United States and other countries.
Microsoft, Windows and Windows NT are registered trademarks of Microsoft Corporation.

Notes:

Performance is in 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 from 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.

This publication was produced in the United States. IBM may not offer the products, services or features discussed in this document in other countries, and the information may be subject to change without notice. Consult your local IBM business contact for information on the product or services available in your area.

IBM considers a product “Year 2000 ready” if the product, when used in accordance with its associated documentation, is capable of correctly processing, providing and/or receiving date data within and between the 20th and 21st centuries, provided that all products (for example, hardware, software and firmware) used with the product properly exchange accurate date data with it. Any statements concerning the Year 2000 readiness of any IBM products contained in this presentation are Year 2000 Readiness Disclosures, subject to the Year 2000 Information and Readiness Disclosure Act of 1998.

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 from 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.

Introduction

IBM @server zSeries

Basic Concepts of VM explained

- Virtual Machine
- Guests
- etc.

Comparison to the following:

- LPAR
- z/OS
- Linux

zSeries Dialects

IBM @server zSeries

Architecture

- Strict and formal language

VM

- The original virtualization language
- Fair amount of "slang"

z/OS

- Evolved from MVS
- Some cross over from VM and LPAR

LPAR

- Origins related to VM, though adopted a unique language

Marketing

- Contains no negative phrases
- Spoken very quickly at times

System Resources

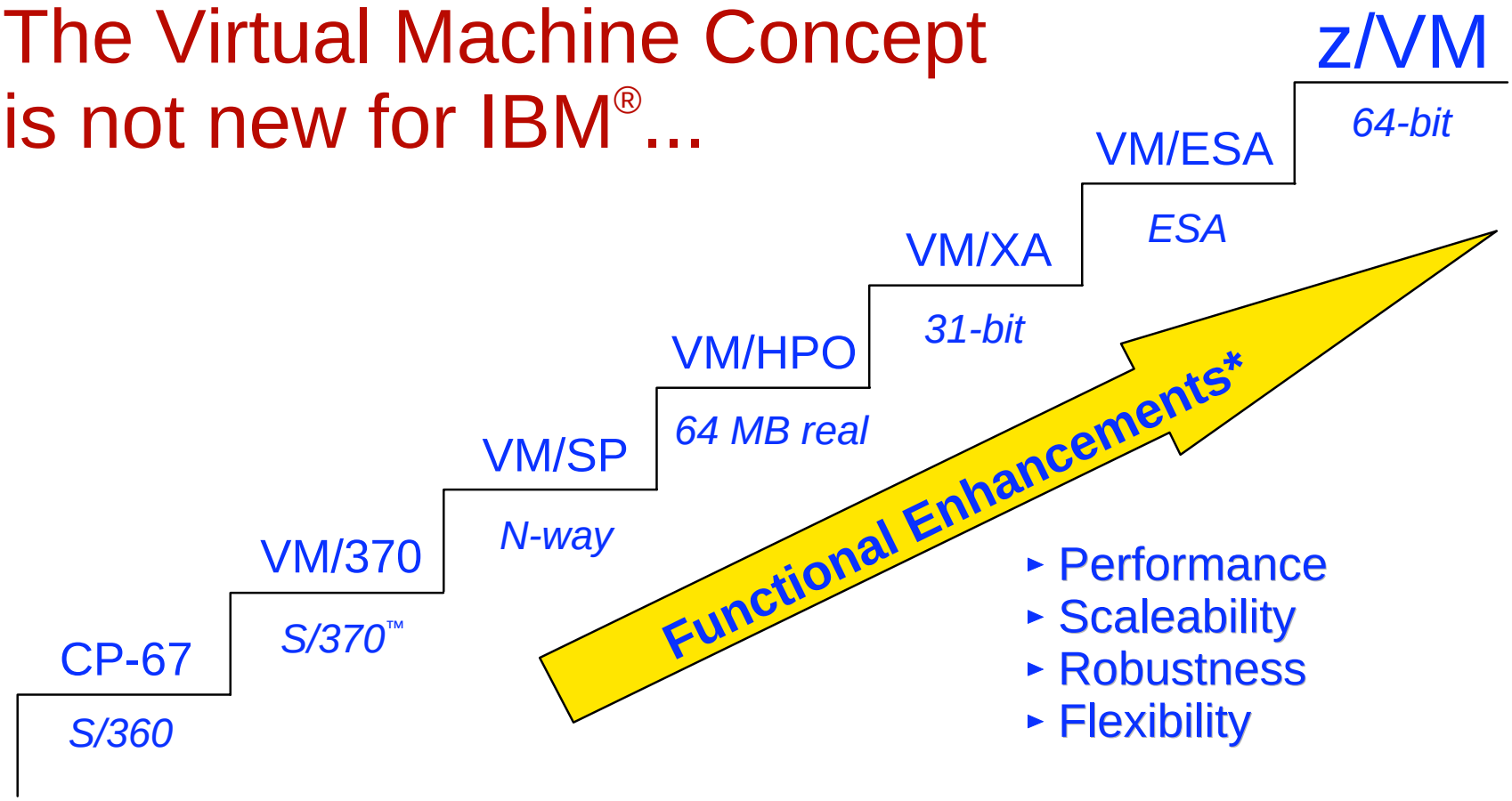
IBM @server zSeries

Linux	zSeries
Memory	Storage
Disk, Storage	DASD- Direct Access Storage Device
Processor	Processor, CPU, PU, Engine, CP, IFL
Computer	CEC, System

IBM Virtualization Technology Evolution

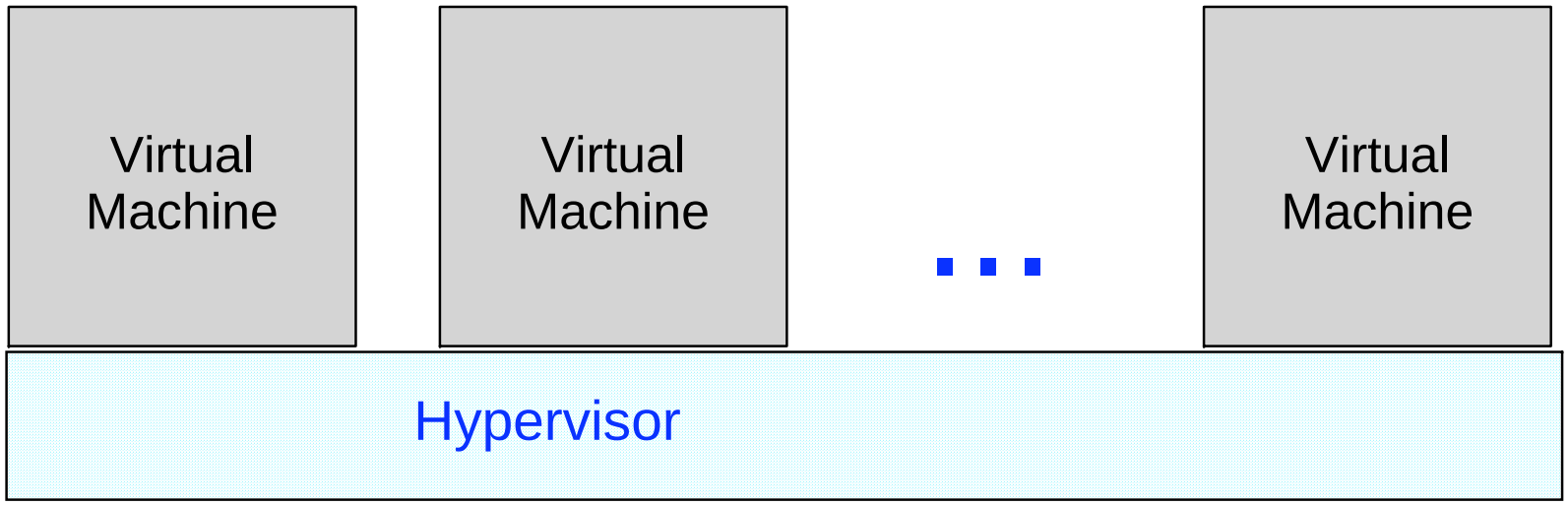
IBM @server zSeries

The Virtual Machine Concept is not new for IBM®...



* Investments made in hardware, architecture, microcode, software

Virtual Machine Basics in Theory

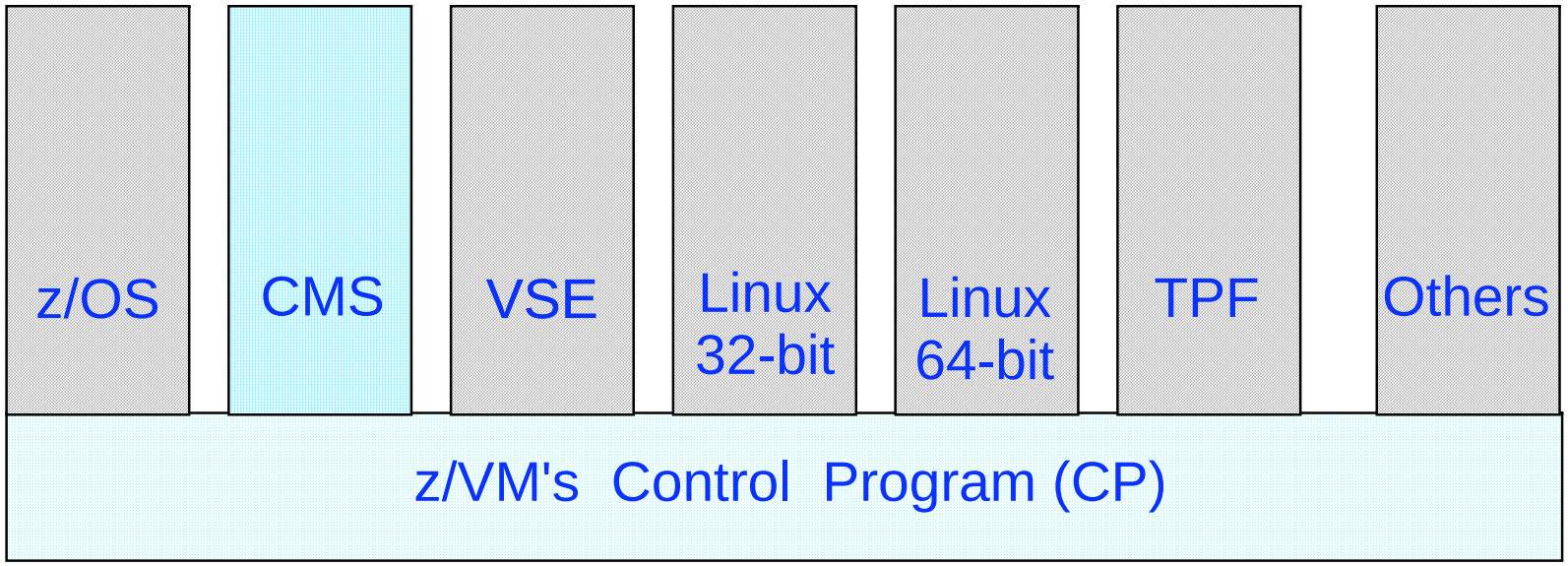


Take real hardware and virtualize it

- Provide same features to virtual hardware as real hardware
- Allow multiple virtual machines to exist at same time
- Allow more virtual objects than there are real objects

Virtual Machine Basics in Practice

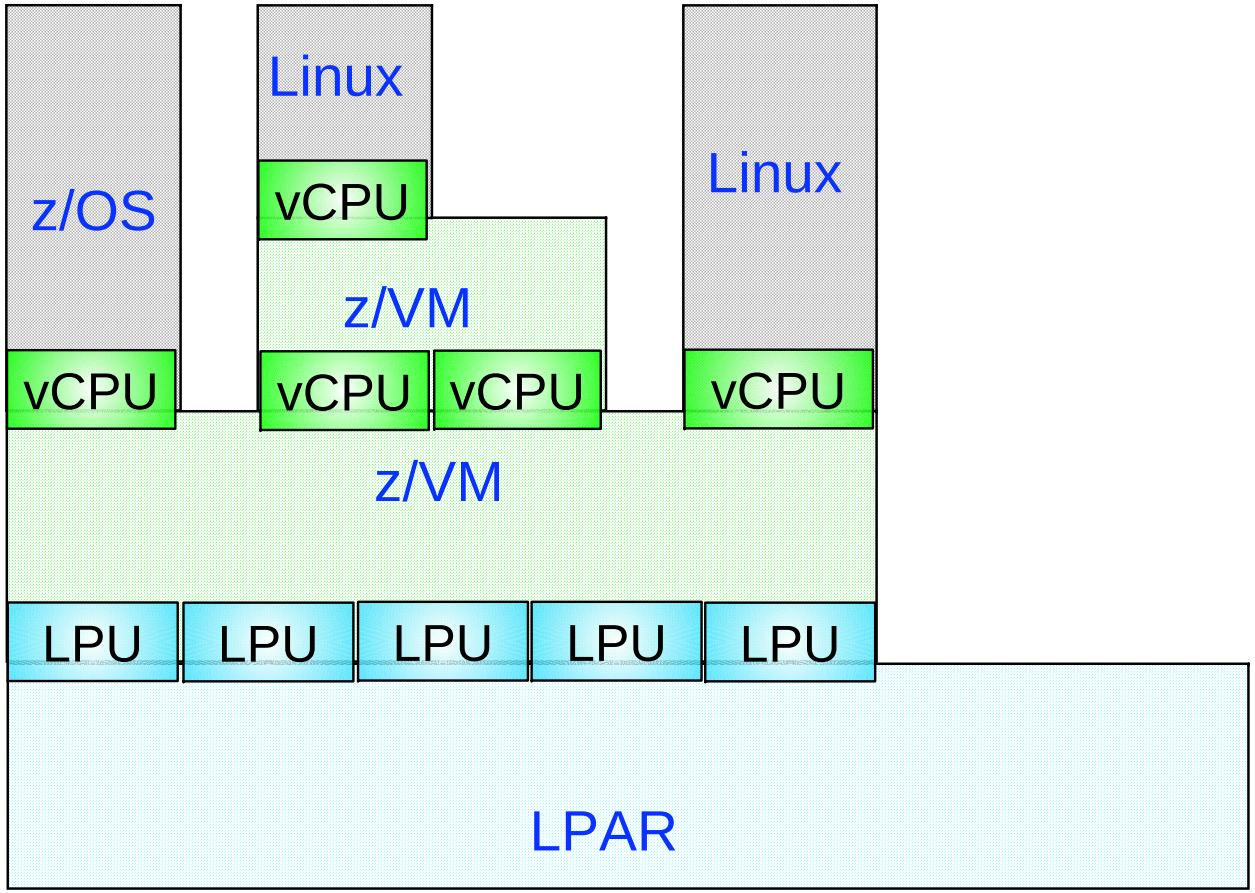
IBM @server zSeries



- Control Program Component - manages virtual machines that adhere to 390- and z-architecture
- Extensions available through CP system services and features
- CMS is special single user system and part of z/VM
- Control Program can via interactive via console device

Phrases Associated with Virtual Machines

IBM @server zSeries



Phrases associated with Virtual Machines

IBM @server zSeries

In VM...

- *Guest*: a system that is operating in a virtual machine, also known as user or userid.
- *Running under VM*: running a system as a guest of VM
- *Running on VM*: running a system as a guest of VM
- *Running second level:* running a system as a guest of VM which is itself a guest of another VM
- A virtual machine may have multiple *virtual processors*

In relationship to LPAR...

- *Logical Partition:* LPAR equivalent of a virtual machine
- *Logical Processor:* LPAR equivalent of a virtual processor
- *Running native:* running without LPAR
- *Running in basic mode:* running without LPAR

VM User Directory

IBM @server zSeries

```
USER LINUX01       MYPASS 128M 128M  G
MACHINE ESA
IPL 190 PARM AUTOCR
CONSOLE 01F 3270 A
SPOOL   00C 2540 READER *
SPOOL   00D 2540 PUNCH  A
SPOOL   00E 1403 A
MDISK   191 3390 012 001 ONEBIT MW
MDISK   200 3390 050 100 TWOBIT MR
LINK    MAINT 190 190 RR
LINK    MAINT 19D 19D RR
LINK    MAINT 19E 19E RR
```

Getting Started

IBM @server zSeries

IML

- Initial Machine Load or Initial Microcode Load
- Power on and configure processor complex prior to running any operating system
- VM equivalents
  - **Logon**
  - **SET MACHINE** command
- In LPAR is image profile activation

IPL

- Initial Program Load
- Like *booting* a Linux system
- zSeries hardware allows you to IPL a system
- z/VM allows you to *IPL* a system in a virtual machine via the **IPL** command
- Linux *kernel* is like VM *nucleus*
- LPAR Load Function

Memory (Storage) Management

IBM @server zSeries

VM

- demand paging between central and expanded
- block paging with DASD (disk)
- steal from central based on LRU with reference bits
- steal from expanded based on LRU with timestamps
- paging activity is traditionally considered normal
- transparent to the guest
- virtual storage can be greater than real

LPAR

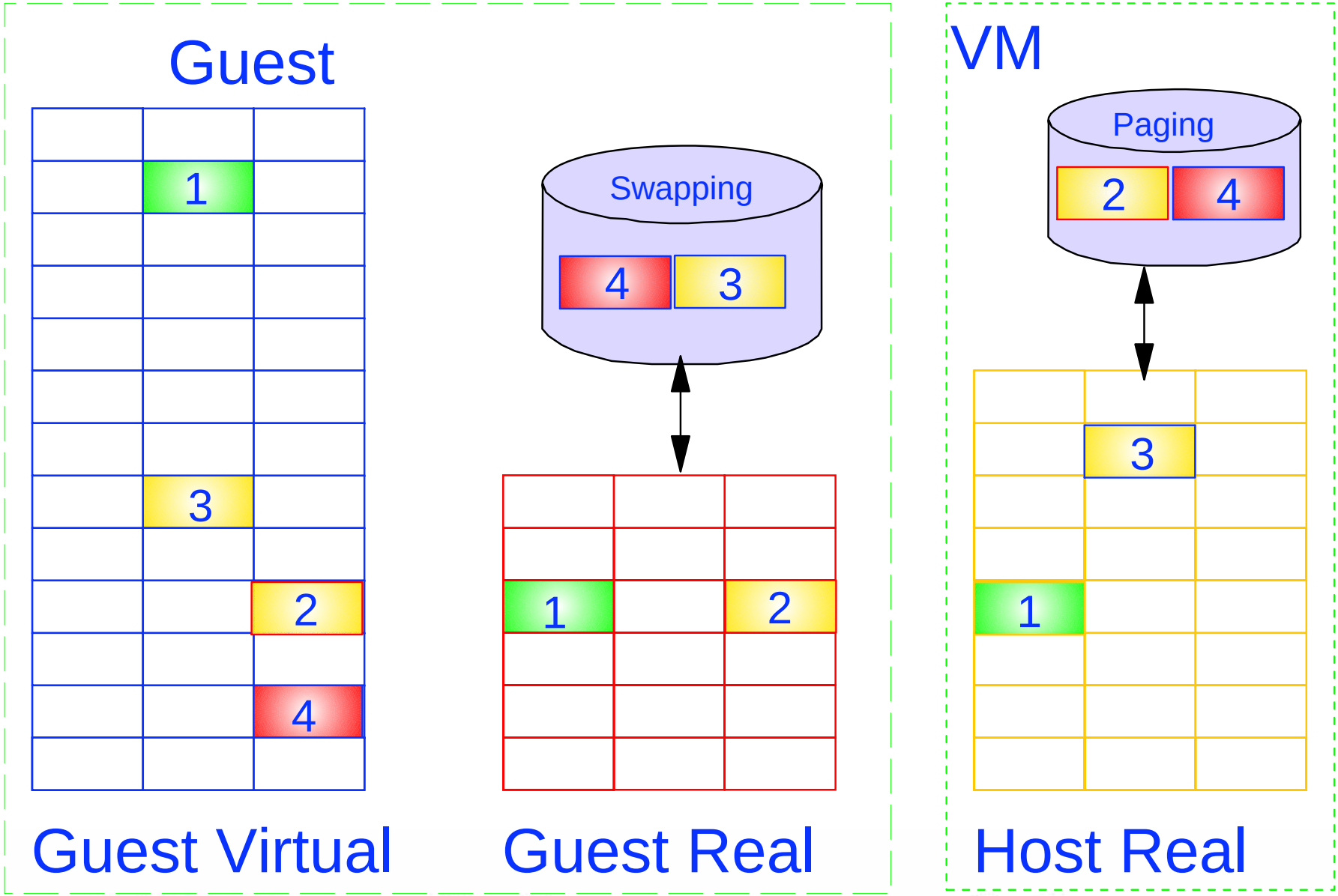
- dedicated storage, no paging

Linux

- paging on page basis
- swapping activity is traditionally considered bad

VM Memory Virtualization

IBM @server zSeries



Processor Mangement

IBM @server zSeries

VM

- *Scheduler*
  - determines priorities based on *Share* setting, etc.
  - factors in other resource usage and workload characteristics
- *Dispatcher* runs a virtual processor on a real processor for (up to) a *minor time slice*
- Can dedicate processors to virtual processor

LPAR

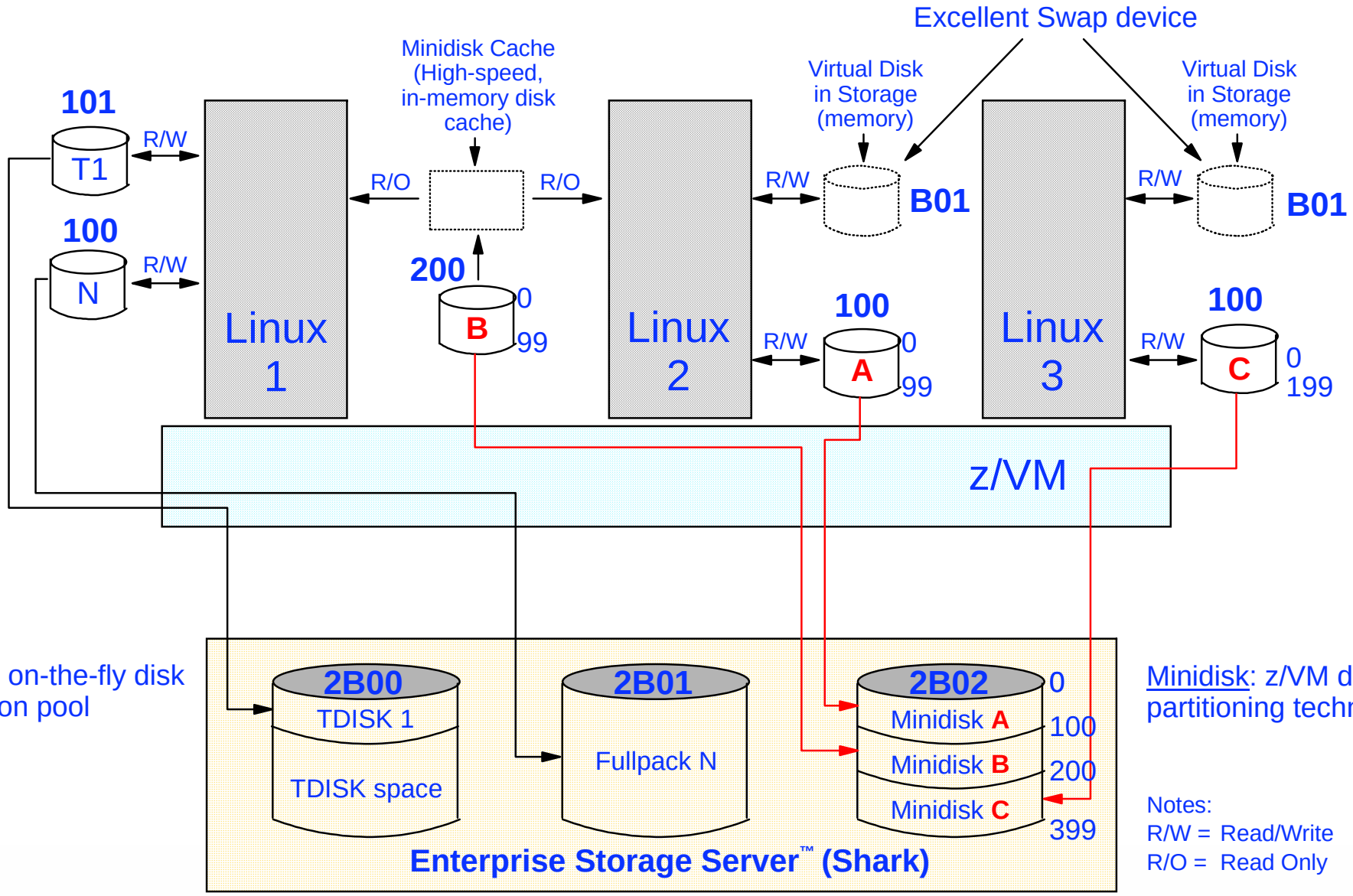
- uses *Weight* setting like Share setting
- dispatches LPs on CPs
- partitions can have dedicated processors

Linux

- *Scheduler* handles prioritization and dispatching processes for a time slice or *quantum*

z/VM Technology - Disk

IBM @server zSeries



TDISK: on-the-fly disk allocation pool

Minidisk: z/VM disk partitioning technology

Notes:
R/W = Read/Write
R/O = Read Only

Other VM Device Management Concepts

IBM @server zSeries

- *RDEV*
  - real device address or the control block associated with it
  - attached or dedicated to a guest for its exclusive use
  - attached to the VM system to be virtualized or partitioned
- *VDEV*
  - virtual device address or the control block associated with it
  - may be a partitioned, virtualized, or simulated device
  - to the guest virtual machine it appears to be a real device
- Virtualized
  - present an image of a real device to multiple virtual machines
  - e.g., crypto devices
- Simulated
  - provide a device to a virtual machine without real hardware
  - virtual CTCAs, virtual disks, Guest LANs

Other Resources

IBM @server zSeries

Timers

- CPU timer and clock comparator
- Virtualized TOD clock
  - SET VTOD command to set vTOD to specific value or that of another virtual machine

Registers

- General Purpose, Control, Access, and Floating Point
  - CP saves/restores between invocations of SIE
  - Manipulation of control registers sometimes requires CP's involvement (SIE exit)

Storage Keys

PSW, Interrupts, Prefixing, and other Architecture structures

Anomalies of Time

VM virtualizes various timers or clocks

- CPU Timer - runs as processor time consumed
- Time of Day (TOD)
- Clock Comparator

Anomaly

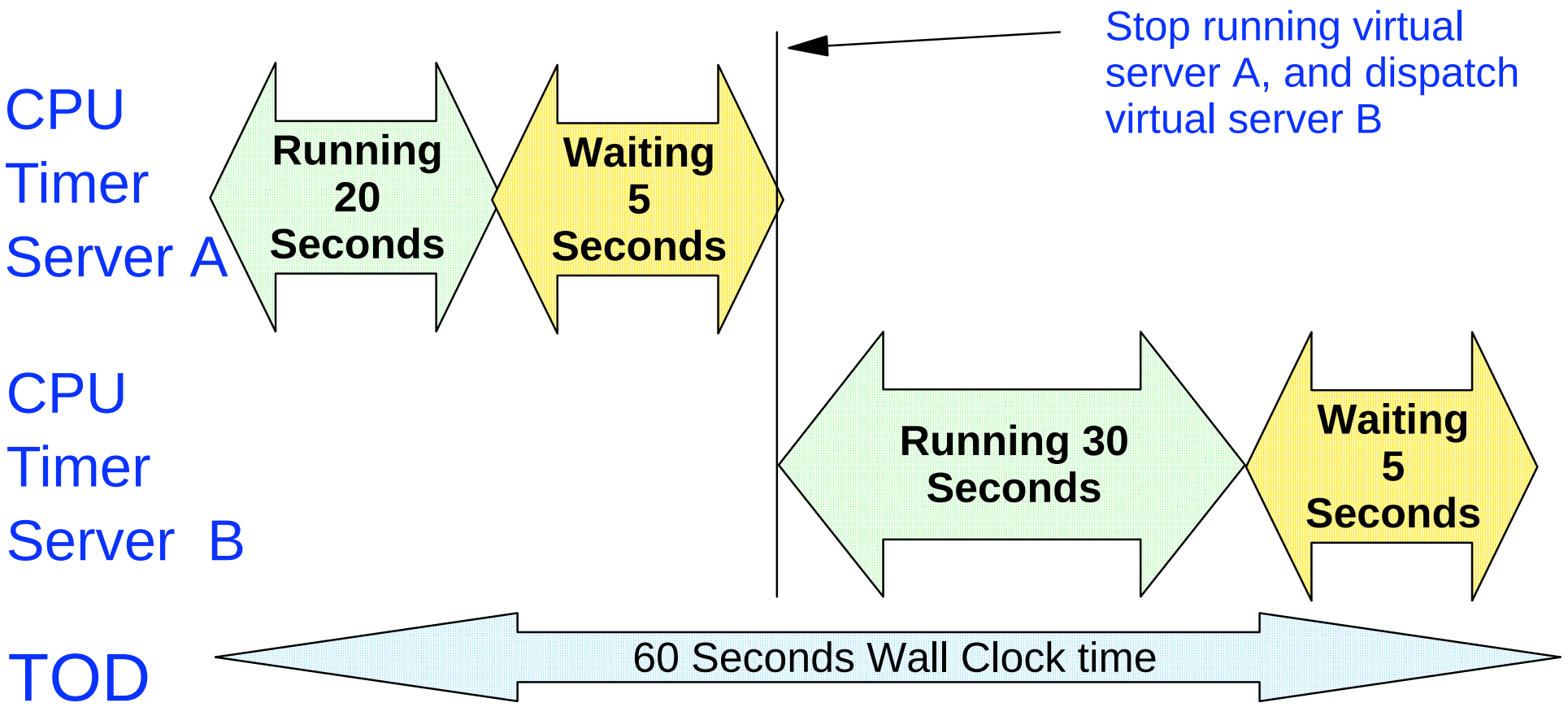
- TOD always moves at wall clock speed
- virtual CPU Timer "moves" slower as the sharing of the real processor increases
- Problem when calculations assume CPU Timer is moving at TOD speed

LPAR

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

Anomalies of Time

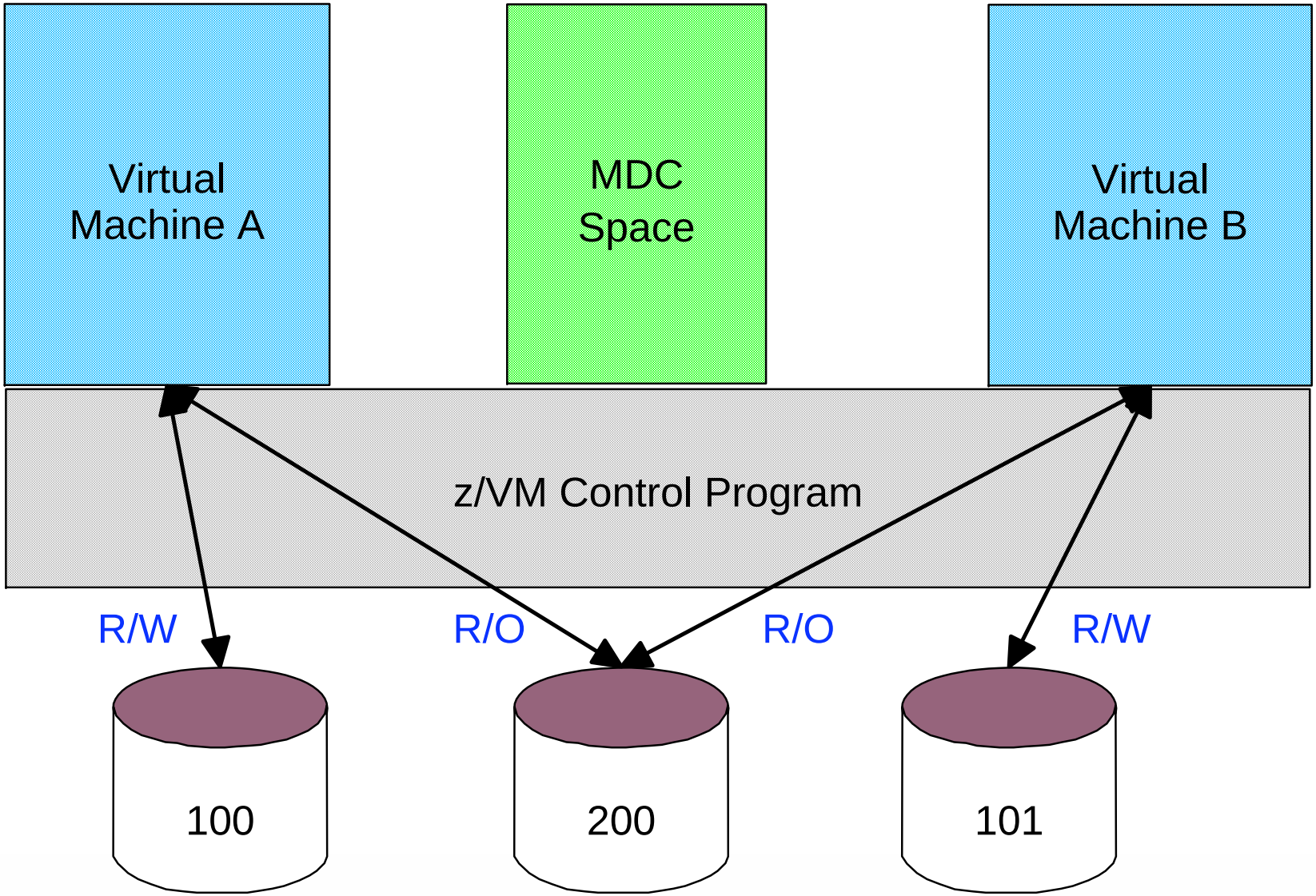
IBM @server zSeries



Virtual Server	Total Time	CPU Timer 'busy'	Incorrect Utilization	Correct Utilization
A	25	20	20/25 = 80%	20/60 = 33%
B	35	30	30/35 = 86%	30/60 = 50%

VM Data in Memory Features - MDC

IBM @server zSeries



VM Data in Memory Features

Minidisk Cache

- Write-through cache for non-dedicated disks
- Cached in central or expanded storage
- Psuedo-track cache
- Great performance - exploits access registers
- Lots of tuning knobs

Virtual Disk in Storage

- Like a RAM disk that is pageable
- Volatile
- Appears like an FBA disk
- Can be shared with other virtual machines

Virtual Networking: Using z/VM Guest LANs

IBM @server zSeries

One Linux guest connects to external network(s)

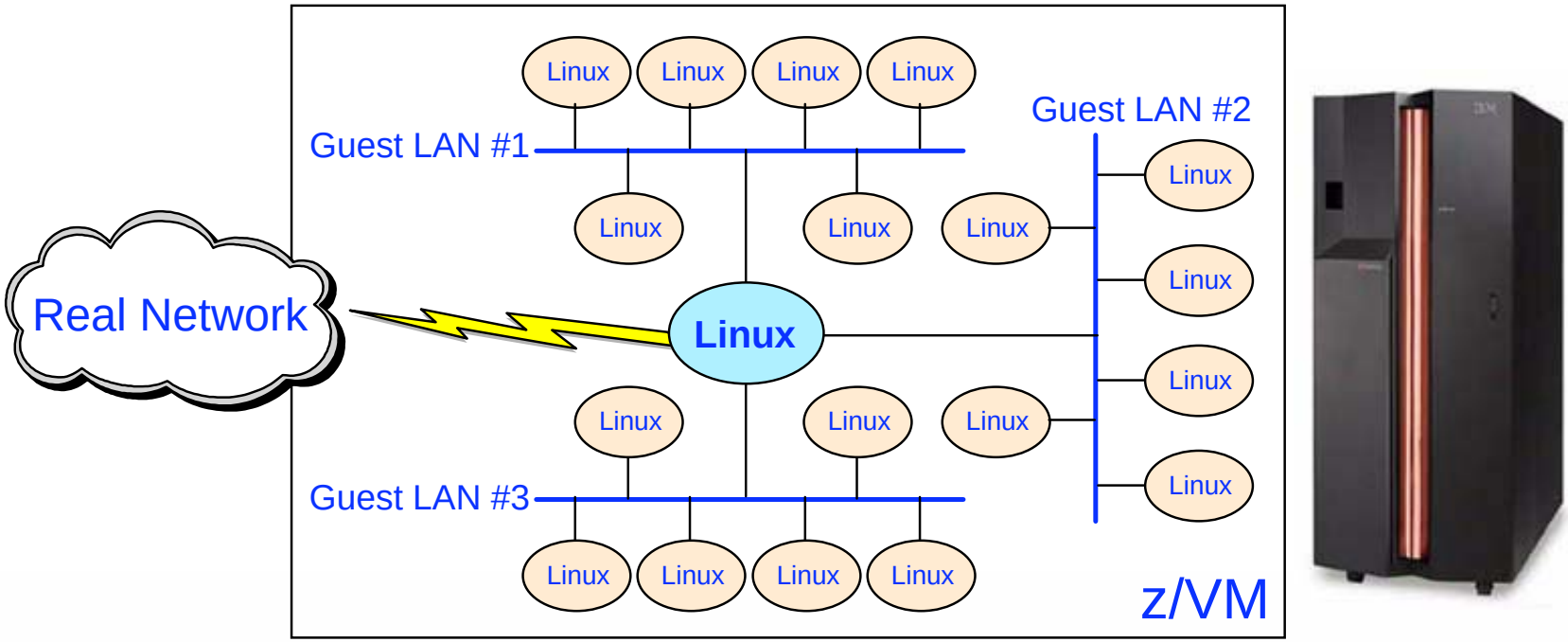
- Also connected to multiple Guest LANs
- Provides external routing and firewall services for guests

Other Linux guests connect to individual Guest LAN(s)

- Virtual HiperSockets and OSA Express connections supported
- Point-to-point, Multicast, and Broadcast (QDIO) supported

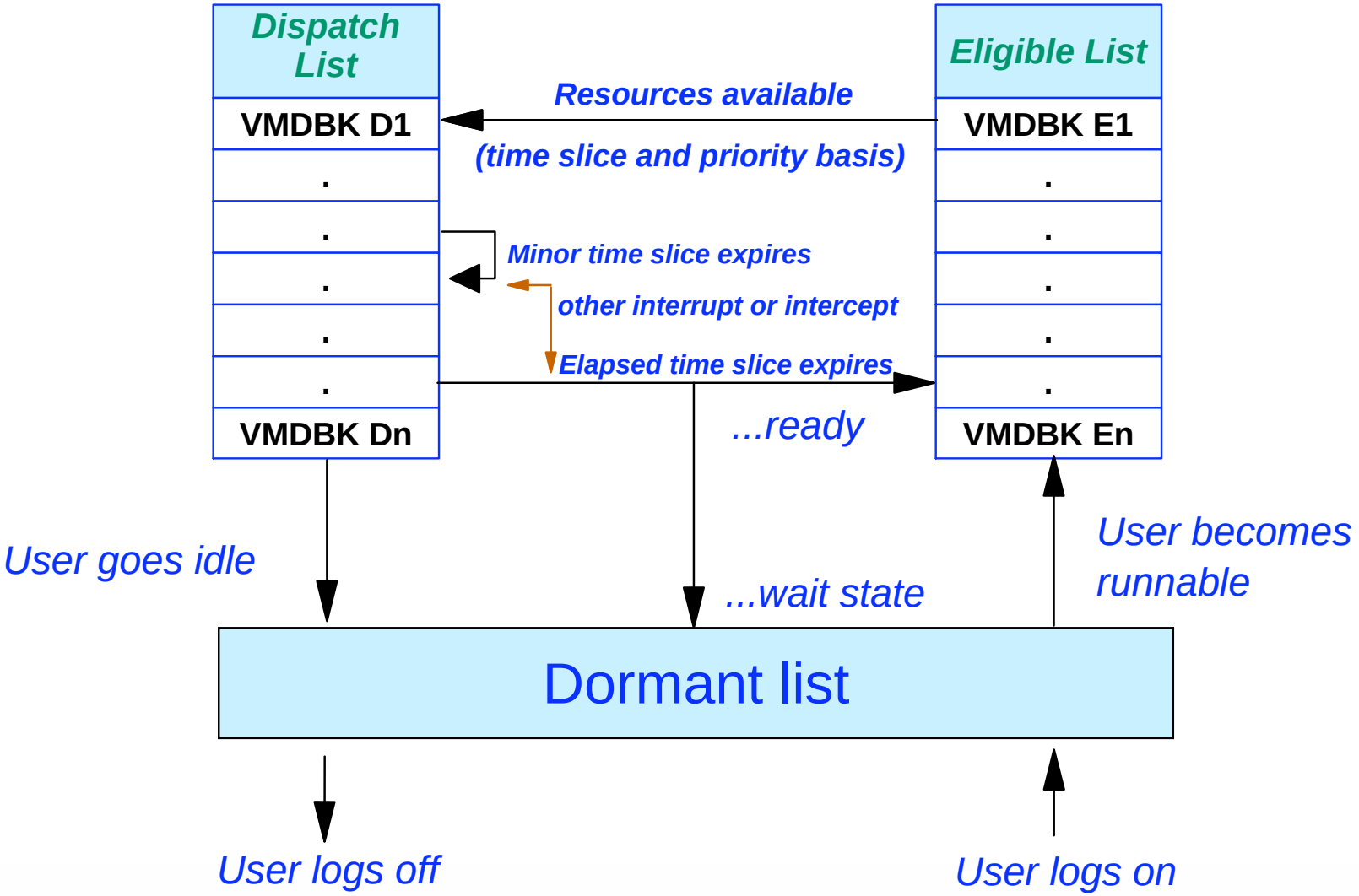
An ideal way to connect a server farm to z/OS

- Using real HiperSockets



Classic Scheduler / Dispatcher Picture

IBM @server zSeries



Scheduling and Dispatching

IBM @server zSeries

Scheduler

- Deadline scheduler (not consumption scheduler)
- Determines whether a guest can be placed in dispatch list
- 4 Classes of users (special, short running, medium running, and long running)
- Elapsed time slice is dynamic based on system feedback

Dispatcher

- Minor time slice (dispatch slice) can be adjusted 1 to 100 milliseconds
  - Default value determined during initialization (5 ms on most machines today, higher on slower processors)
  - 500 milliseconds for a dedicated processor
- Each processor has own dispatch vector, but if idle will steal from a neighbor

Use of SIE

IBM @server zSeries

- SIE = "Start Interpretive Execution", an instruction
- z/VM (like LPAR) uses the SIE instruction to "run" virtual processors for a given virtual machine.
- SIE has access to the region, segment, and page tables used for Dynamic Address Translation (DAT)
- z/VM gets control back from SIE for various reasons:
  - Page faults
  - I/O channel program translation
  - Privileged instructions (including CP system service calls)
  - CPU timer expiration (dispatch slice)
  - Other, including CP asking to get control for special cases
- CP can also shoulder tap SIE from another processor to remove virtual processor from SIE (perhaps to reflect an interrupt)

Multiple Virtualization Layers

IBM @server zSeries

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)

VM Saved Segment and NSS Support

IBM @server zSeries

DCSS (Discontiguous Saved Segments)

- Can define an address range (MB boundary) to the system
- A single copy will exist and is shared among all users
- Virtual Machine loads dynamically
  - Can be located outside virtual machine's defined storage
- DAT architecture allows this to work with minimal CP involvement
- Used to contain
  - Data (e.g., file system control blocks)
  - Code (e.g., CMS code libraries)

NSS (Named Saved Systems)

- Place kernel code in a segment
- Able to IPL the NSS (boot the NSS)
  - 1 shared copy on system for N virtual machines instead of N copies
  - Faster boot

Special Cases

- Writable by guest, or by CP
- Restricted
- Shared between CP and guests
- Can have both exclusive and shared ranges

VM Control Program Interfaces

Commands

- Query or change virtual machine configuration
- Debug and tracing
- Commands fall into different privilege classes
- Some commands affect entire system

Inter-virtual machine communication

- IUCV - Inter User Communication Vehicle
- VMCF - Virtual Machine Communication Facility

System Services

- Communicate with CP via IUCV
- Various services: Monitor, Accounting, Security

Diagnose Instruction

- Operands used communicate with hardware (or in this case the virtual hardware) in various ways

CP Debug Features

Tracing of virtual machine

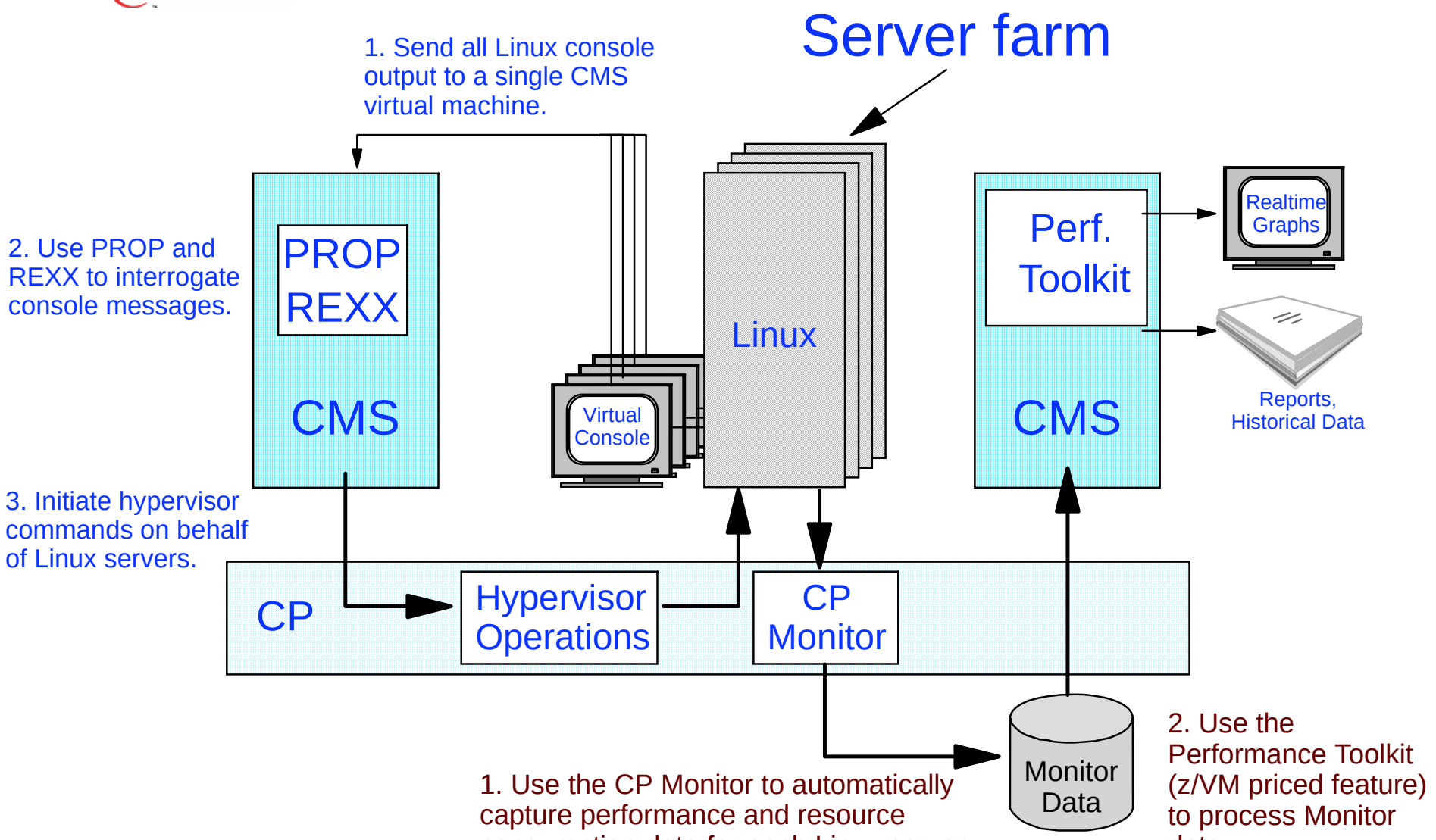
- CP TRACE command has >40 pages of documentation:
  - instructions
  - storage references
  - some specific opcodes or privileged instructions
  - branches
  - various address space usage
  - registers
- Step through execution or run and collect information to spool
- Trace points can trigger other commands

Display or store into virtual memory

- Helpful, especially when used with tracing
- Valid for various virtual address spaces
- Options for translation as EBCDIC, ASCII, or 390 opcode
- Locate strings in storage
- Store into virtual memory (code, data, etc.)

z/VM Technology - Command and Control

IBM @server zSeries



References

IBM @server zSeries

VM Home Page http://www.vm.ibm.com

Pubs on VM Home Page

- Of particular interest
  - z/VM V4R4.0 CP Command and Utility Reference
  - z/VM V4R4.0 CP Planning and Administration
  - z/VM V4R4.0 CP Programming Services
  - z/VM V4R4.0 Performance

IBM Systems Journal Vol. 30, No. 1, 1991

- Good article on SIE