B31

Virtualization Basics

Brian K. Wade, Ph.D.

zSeries® EXPO
FEATURING Z/OS, Z/VM, Z/VSE AND LINUX ON ZSERIES

September 19 - 23, 2005
San Francisco, CA

RETURN TO INDEX

Trademarks

IBM eserver 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.

Credits

People who contributed ideas and charts:

- Alan Altmark
- Bill Bitner
- John Franciscovich
- Reed Mullen
- Brian Wade
- Romney White

Thanks to everyone who contributed!

Introduction

IBM eserver zSeries

We'll explain basic concepts of zSeries:

- Terminology
- Processors
- Memory
- I/O
- Networking

We'll see that z/VM *virtualizes* a zSeries machine:

- Virtual processors
- Virtual memory
- ... and so on

Where appropriate, we'll compare or contrast:

- PR/SM or LPAR
- z/OS
- Linux

Terminology

zSeries Architecture

IBM eserver zSeries

Every computer system has an *architecture.*

- Formal definition of how the hardware operates
- It's the hardware's functional specification
- What the software can expect from the hardware
- *What it does*, not how it does it

IBM's book z/Architecture Principles of Operation defines zSeries architecture

- Instruction set
- Processor features (registers, timers, interruption management)
- Arrangement of memory
- How I/O is to be done

Different *models* implement the architecture in different ways.

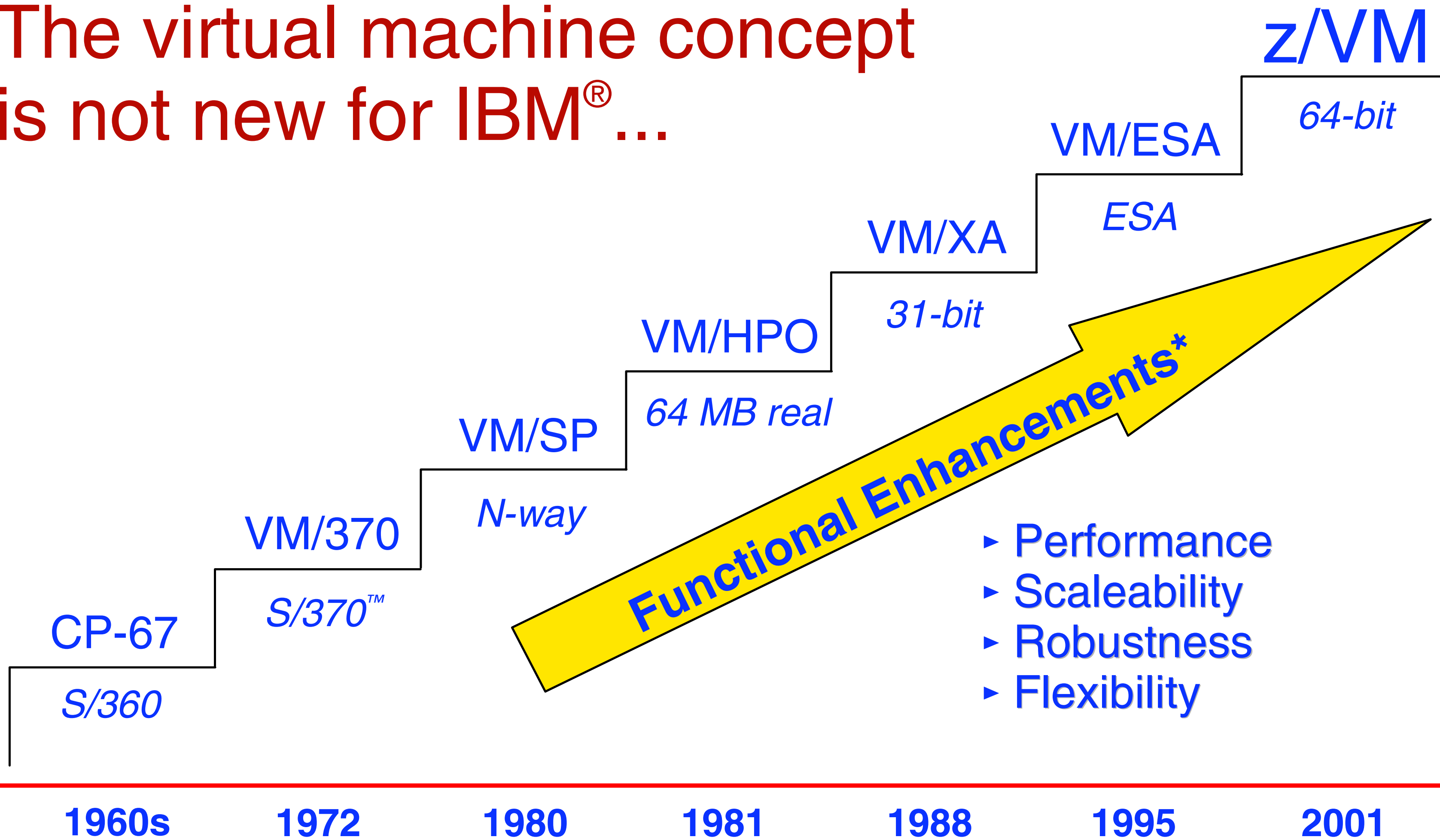
- How many processors there are
- How the processors connect to the memory bus
- How the cache is arranged
- How much physical memory there is
- How much I/O capability there is

z900, z990, and z890 are all *models* implementing z/Architecture.

IBM Virtualization Technology Evolution

IBM eserver zSeries

The virtual machine concept is not new for IBM®...



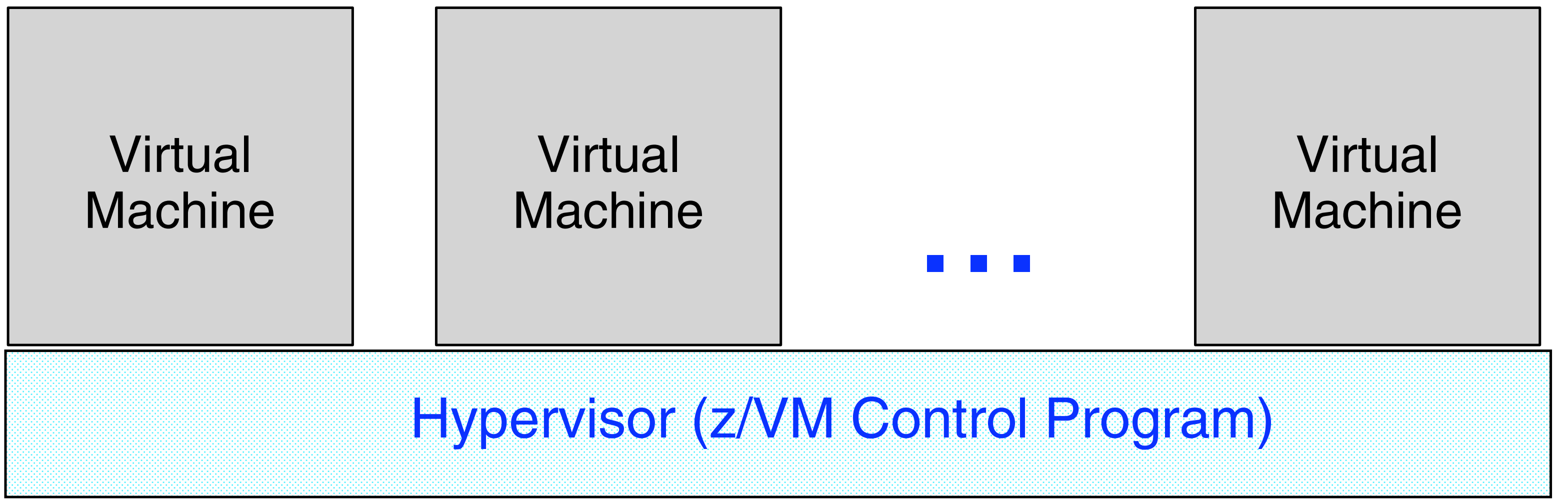
* Investments made in hardware, architecture, microcode, software

zSeries Parts Nomenclature

Intel, pSeries, etc.	zSeries
Memory	Storage (though we are moving toward "memory")
Disk, storage	DASD- Direct Access Storage Device
Processor	Processor, CPU (central processing unit), engine, IFL (Integrated Facility for Linux), IOP (I/O processor), SAP (system assist processor), CP (central processor), PU (processing unit), zAAP (zSeries Application Assist Processor)
Computer	CEC (central electronics complex)

Virtual Machines

What: Virtual Machines



A **virtual machine** is an execution context that obeys the architecture.

The purpose of z/VM is to **virtualize** the real hardware:

- Faithfully replicate the z/Architecture Principles of Operation
- Permit any virtual configuration that could legitimately exist in real hardware
- Let many virtual machines operate simultaneously
- Allow overcommittment of the real hardware (processors, for example)
- Designed for many thousands of virtual machines per z/VM image (I have seen 40,000)
- Your limits will depend on the size of your physical zSeries computer

What: A Virtual Machine

Virtual machine

z/Architecture

512 MB of memory

2 processors

Basic I/O devices:
- A console
- A card reader
- A card punch
- A printer

Some read-only disks

Some read-write disks

Some networking devices

We permit any configuration that a real zSeries machine could have.

In other words, we completely implement the z/Architecture Principles of Operation.

There is no "standard virtual machine configuration".

How: VM User Directory

Definitions of:

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

```
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 TWOBIT MR
```

How: 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

Changing the virtual configuration after logon is considered normal.
Usually the guest operating system detects and responds to the change.

Processors

What: Processors

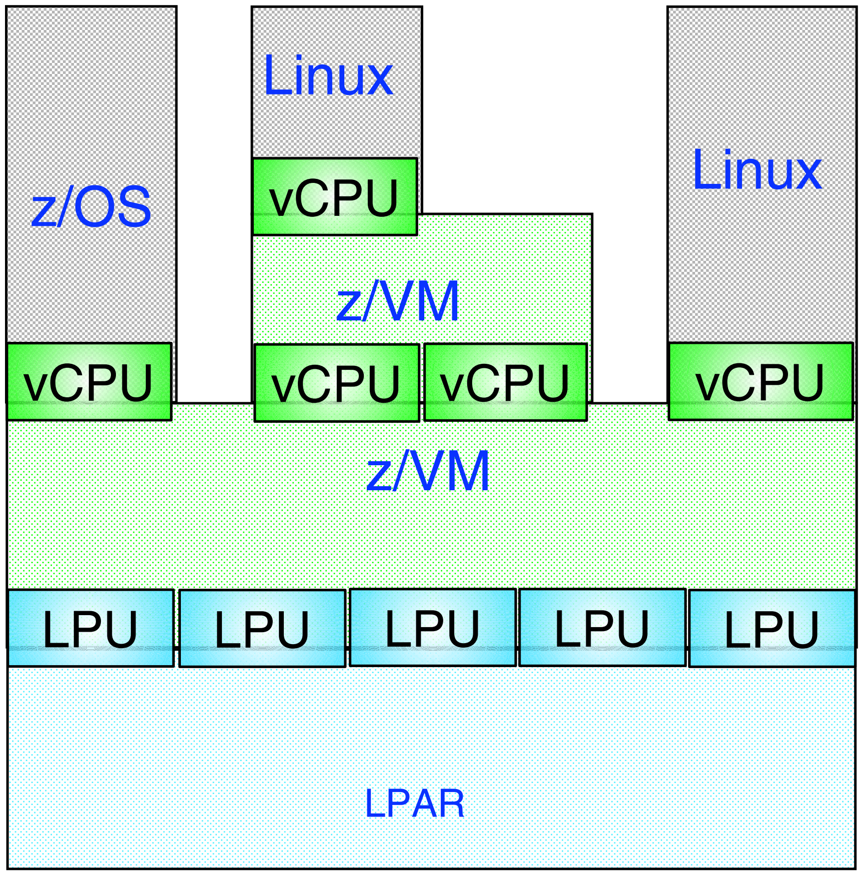
Configuration

- Virtual 1- to 64-way
  - Defined in user directory, or
  - Defined by CP command
- A real processor can be dedicated to a virtual machine

Control and Limits

- Scheduler selects virtual processors according to apparent CPU need
- "Share" setting - prioritizes real CPU consumption
  - Absolute or relative
  - Target minimum and maximum values
  - Maximum values (limit shares) either hard or soft
- "Share" for virtual machine is divided among its virtual processors

What: Logical and Virtual Processors



How: Start Interpretive Execution (SIE)

IBM eserver zSeries

- SIE = "Start Interpretive Execution", an instruction
- z/VM (like the LPAR hypervisor) uses the SIE instruction to "run" virtual processors for a given virtual machine.
- Our processor chips contain special hardware (registers, etc.) to make SIE fast
- SIE has access to:
  - A control block that describes the virtual processor state (registers, etc.)
  - The Dynamic Address Translation (DAT) tables for the virtual machine
- 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)

How: Scheduling and Dispatching

VM

- *Scheduler* determines priorities based on *share* setting and other factors
- *Dispatcher* runs a virtual processor on a real processor
- Virtual processor runs for (up to) a *minor time slice*
- Virtual processor keeps competing for (up to) an *elapsed time slice*

LPAR hypervisor

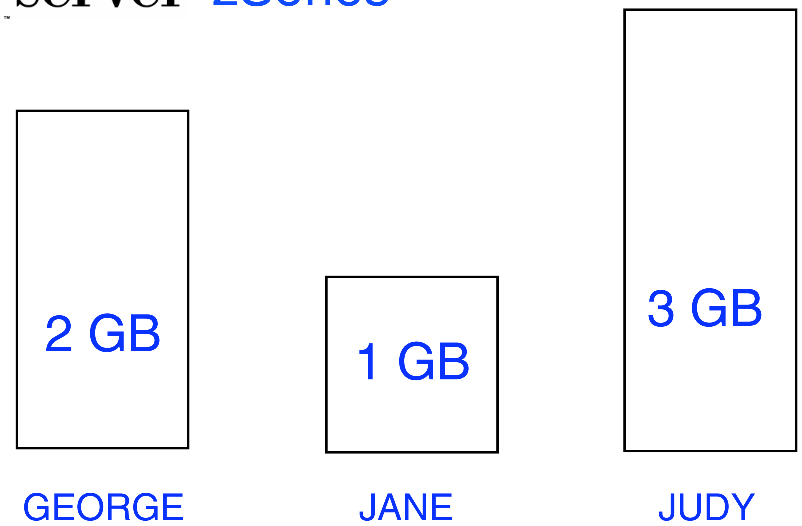
- Uses *weight* settings for partitions, similar to share settings for virtual machines
- Dispatches logical processors on real engines

Linux

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

Memory

What: Virtual Memory



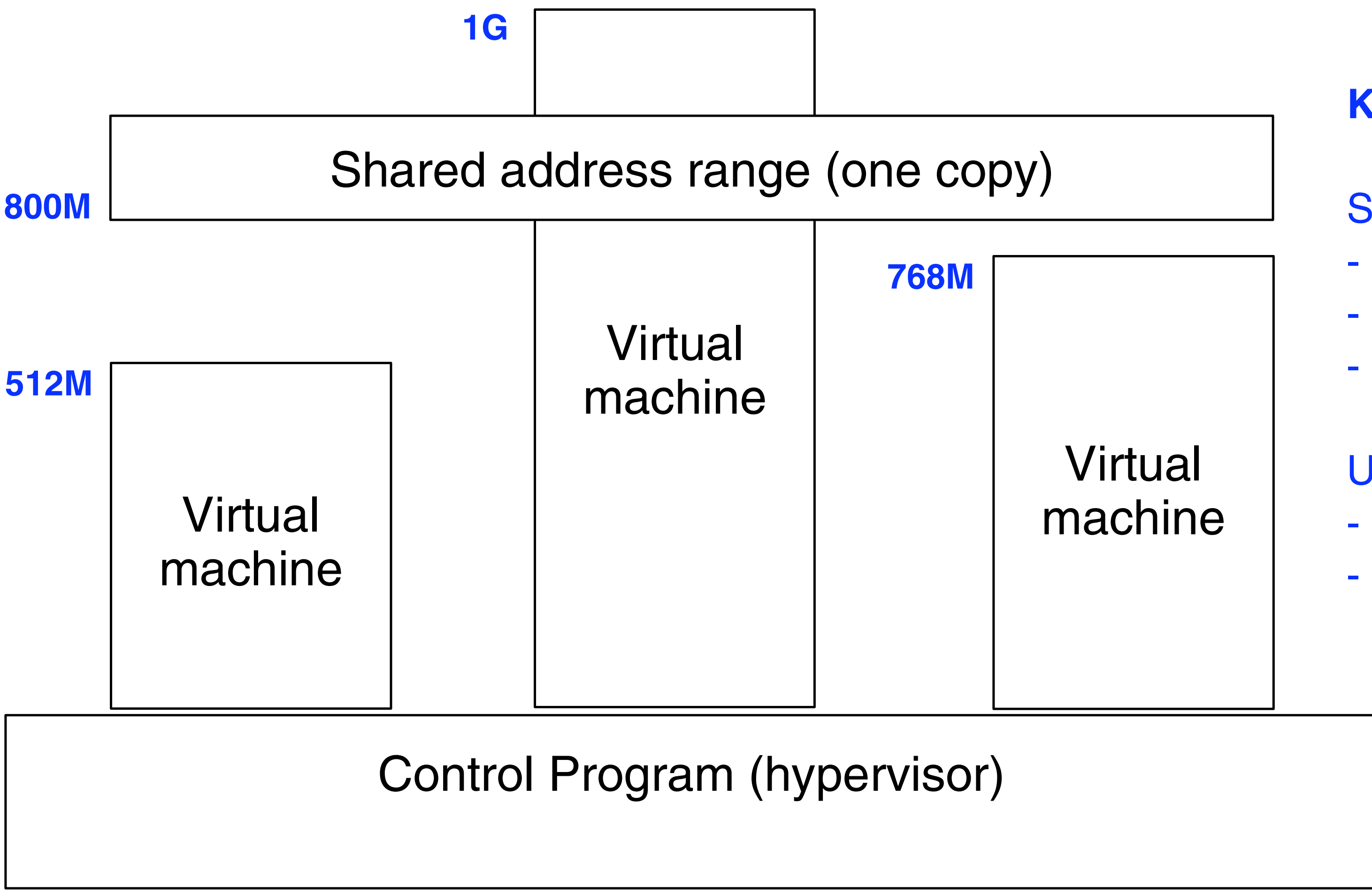
Configuration

- Defined in CP directory entry or via CP command
- Can define storage with gaps (useful for testing)
- Can attach expanded storage to virtual machine

Control and Limits

- Scheduler selects virtual machines according to apparent need for storage and paging capacity
- Virtual machines that do not fit criteria are placed in the *eligible list*
- Can reserve an amount of real storage for a guest's pages
- Can lock certain specific guest pages into real storage

What: Shared Memory

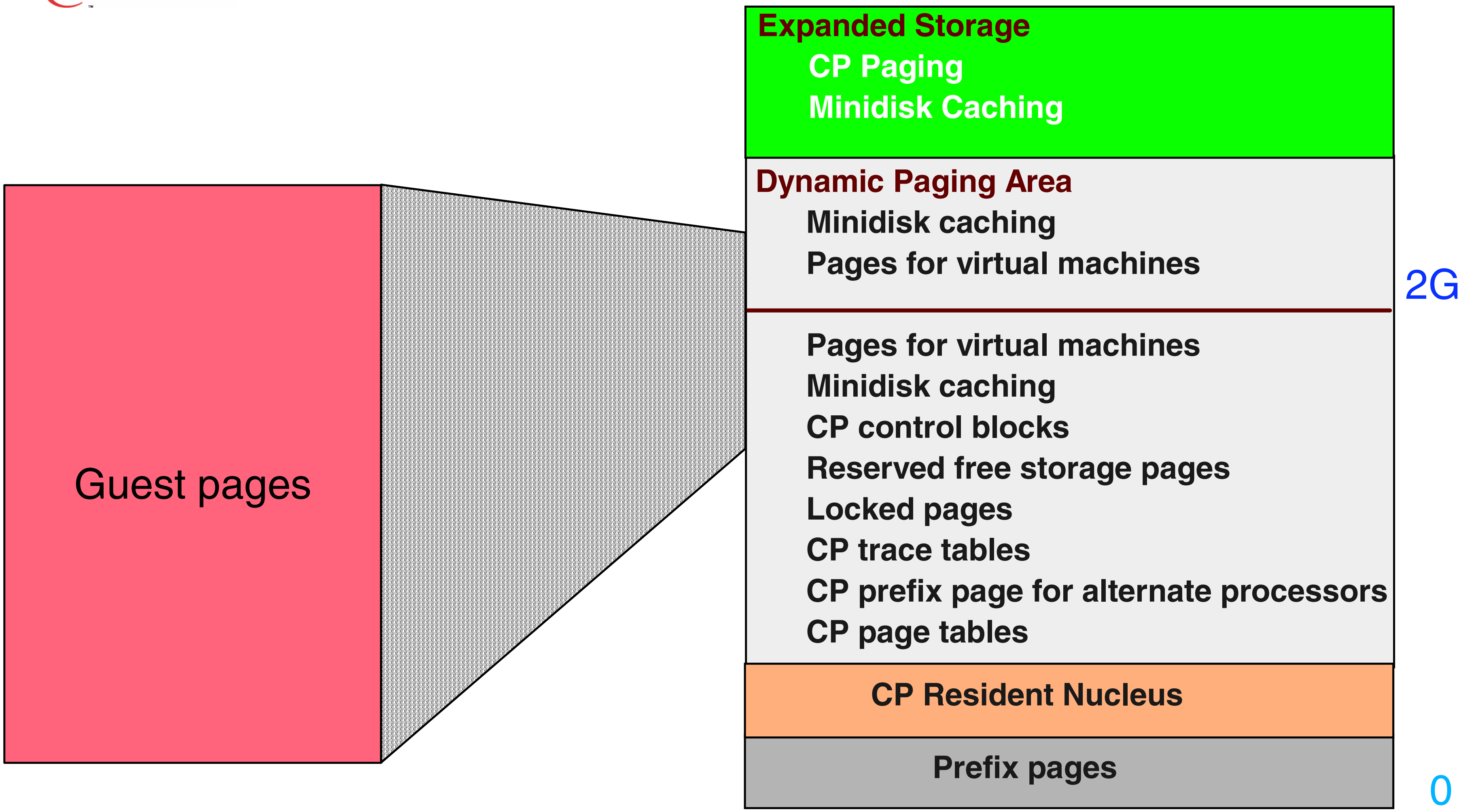


Key Points:

Sharing:
- Read-only
- Read-write
- Security knobs

Uses:
- Common kernel
- Shared programs

How: Layout of Real Storage



How: Memory Management

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

LPAR

- Dedicated storage, no paging

Linux

- Paging on per-page basis to swap disks
- No longer swaps entire processes
- Traditionally considered bad

I/O Resources

What: Device Management Concepts

- *Dedicated* or *attached*
  - The guest has exclusive use of the entire real device.

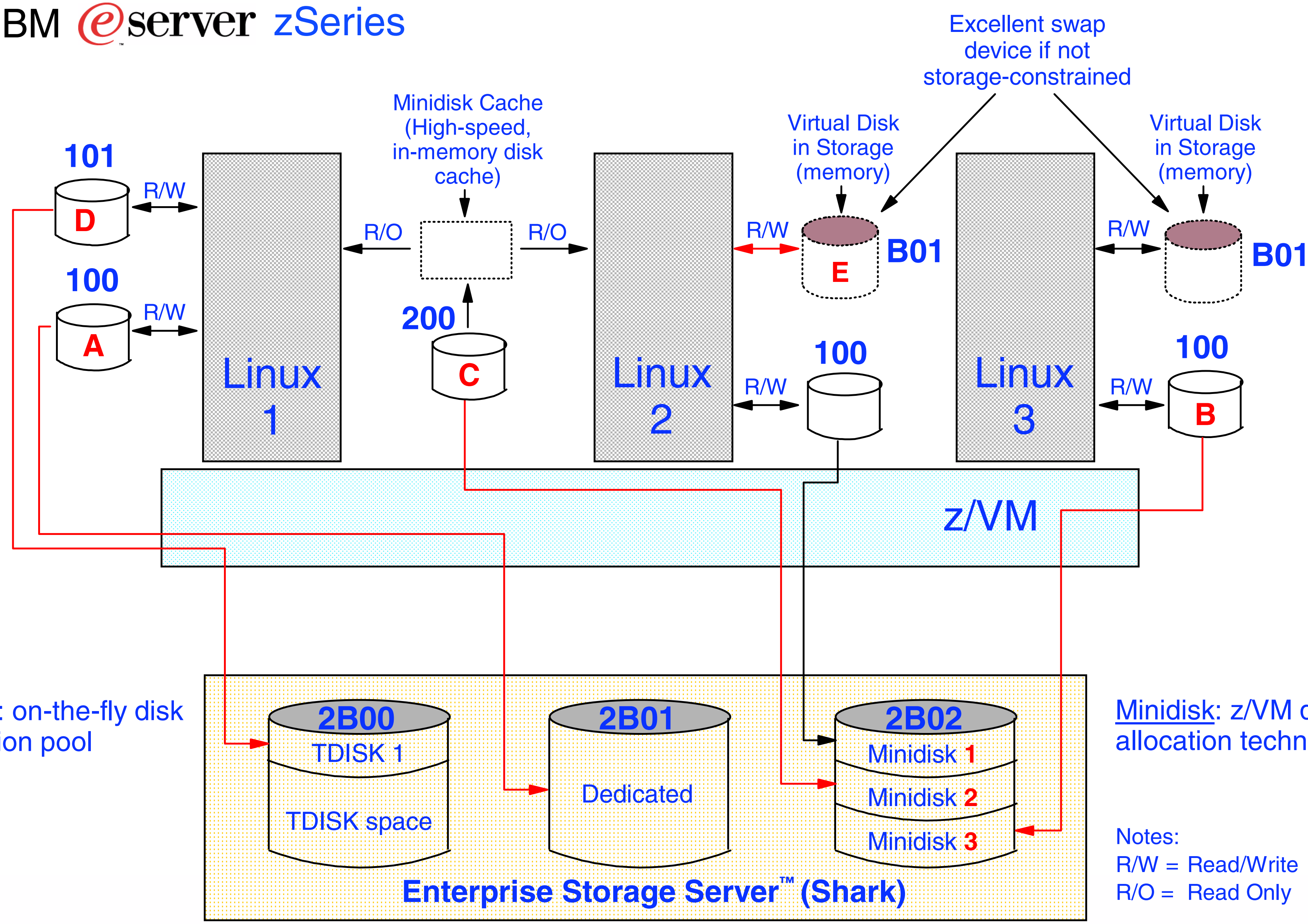
- Virtualized
  - Present a slice of a real device to multiple virtual machines
  - Slice in time or slice in space
  - E.g., DASD, crypto devices

- Simulated
  - Provide a device to a virtual machine without the help of real hardware
  - Virtual CTCAs, virtual disks, guest LANs, spool devices

- Control and Limits
  - Indirect control through "share" setting
  - Real devices can be "throttled" at device level
  - Channel priority can be set for virtual machine
  - MDC fair share limits (can be overridden)

What: Virtualization of Disks

IBM eserver zSeries



TDISK: on-the-fly disk allocation pool

Minidisk: z/VM disk allocation technology

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

What: Data-in-Memory

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
- Plenty of knobs here too

Networking

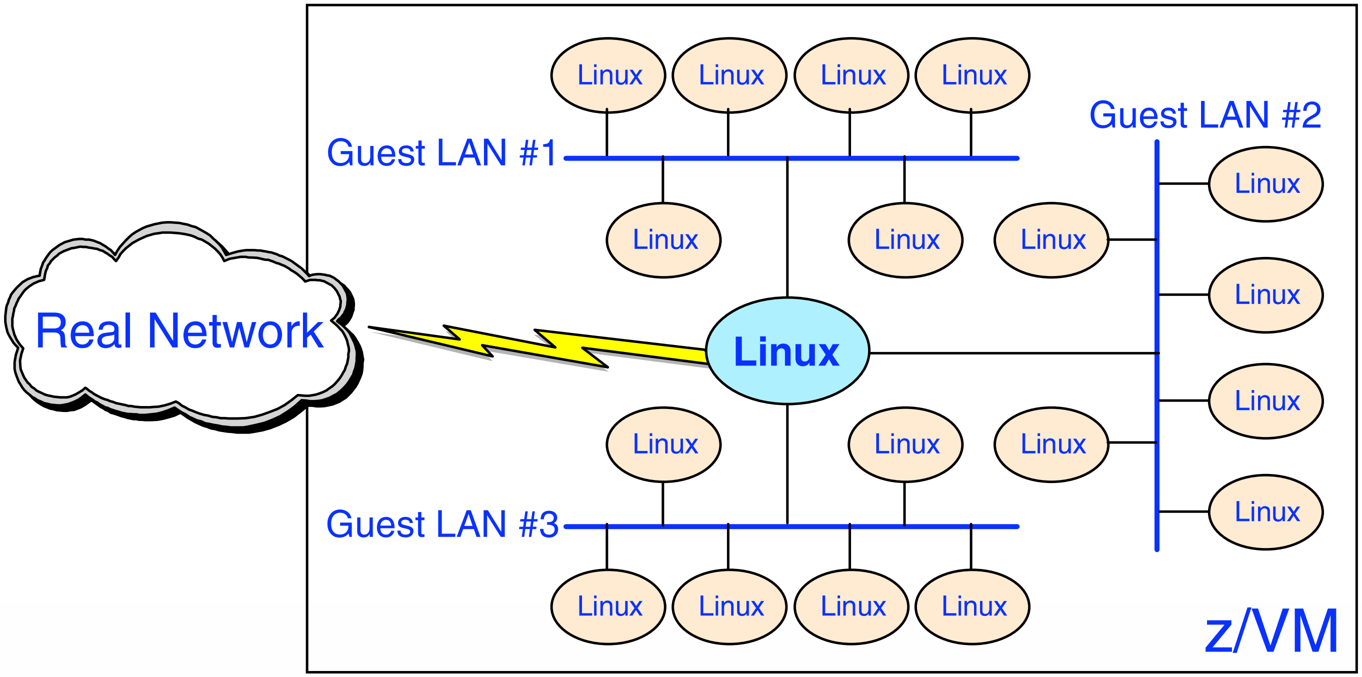
What: Virtual Networks

IBM eserver zSeries

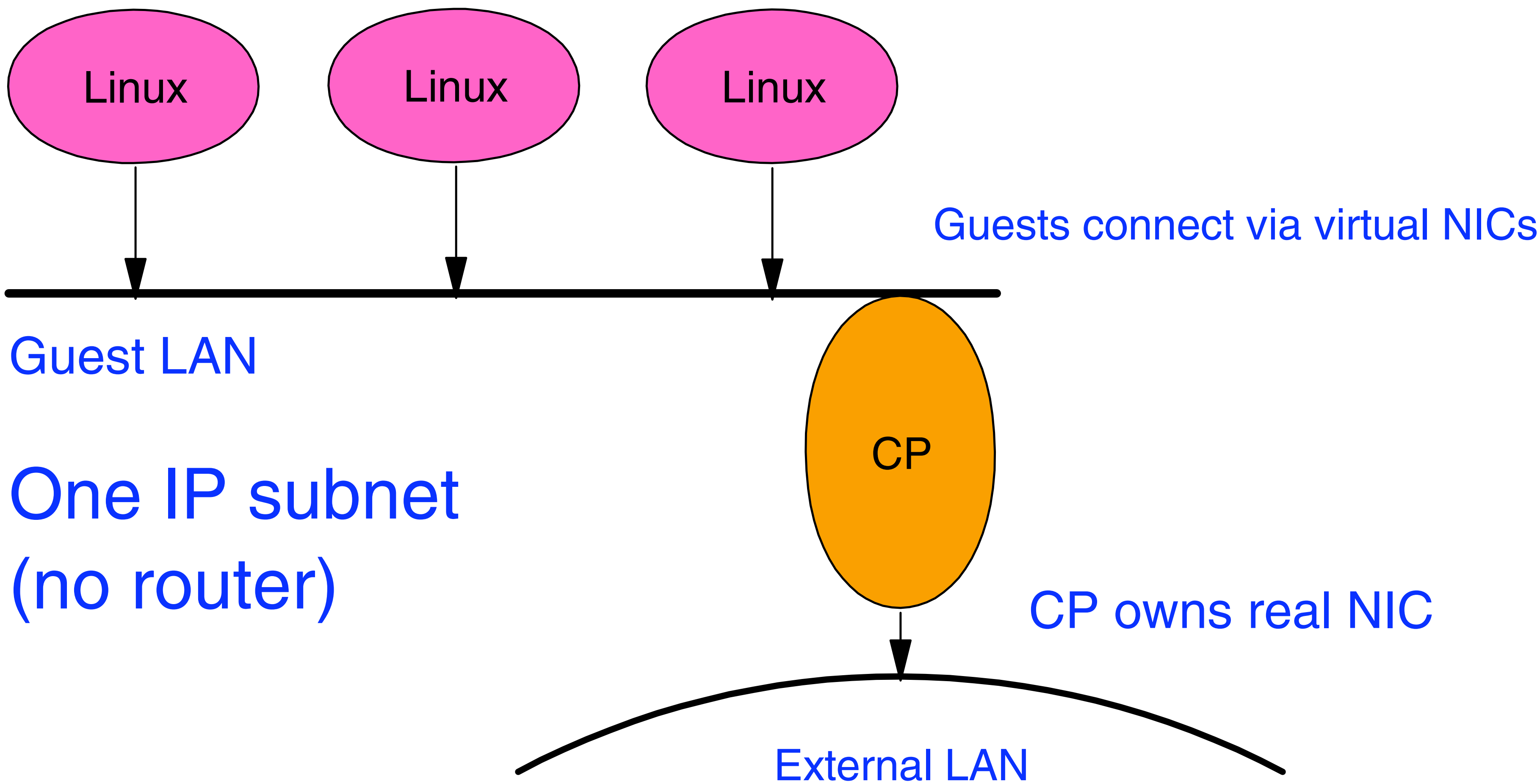
One Linux guest (or z/VM TCP/IP stack) connects to the external network
- Owns the physical OSA (to real LAN) or HiperSockets device (to another LPAR)
- Also connected to multiple guest LANs (each guest LAN is a distinct IP subnet)
- Provides routing services for guests

Another choice is the z/VM Virtual Switch
- z/VM CP itself owns the physical OSA
- Guests' virtual network adapters seem to be on the external IP subnet

Other Linux guests connect to individual guest LAN(s)
- Virtual HiperSockets and OSA Express connections supported
- Point-to-point, Multicast, and Broadcast (QDIO) supported



What: Virtual Switch



One IP subnet
(no router)

Beyond Virtualization

What: Other Control Program (CP) 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

- Connectionless or connection-oriented protocols
- Most pre-date TCP/IP

System Services

- Enduring connection to hypervisor via a connection-oriented program-to-program API
- Various services: Monitor (performance data), Accounting, Security

Diagnose Instructions

- These are really programming APIs (semantically, procedure calls)
- Operands communicate with hardware (or in this case the virtual hardware) in various ways

What: Debugging a Virtual Machine

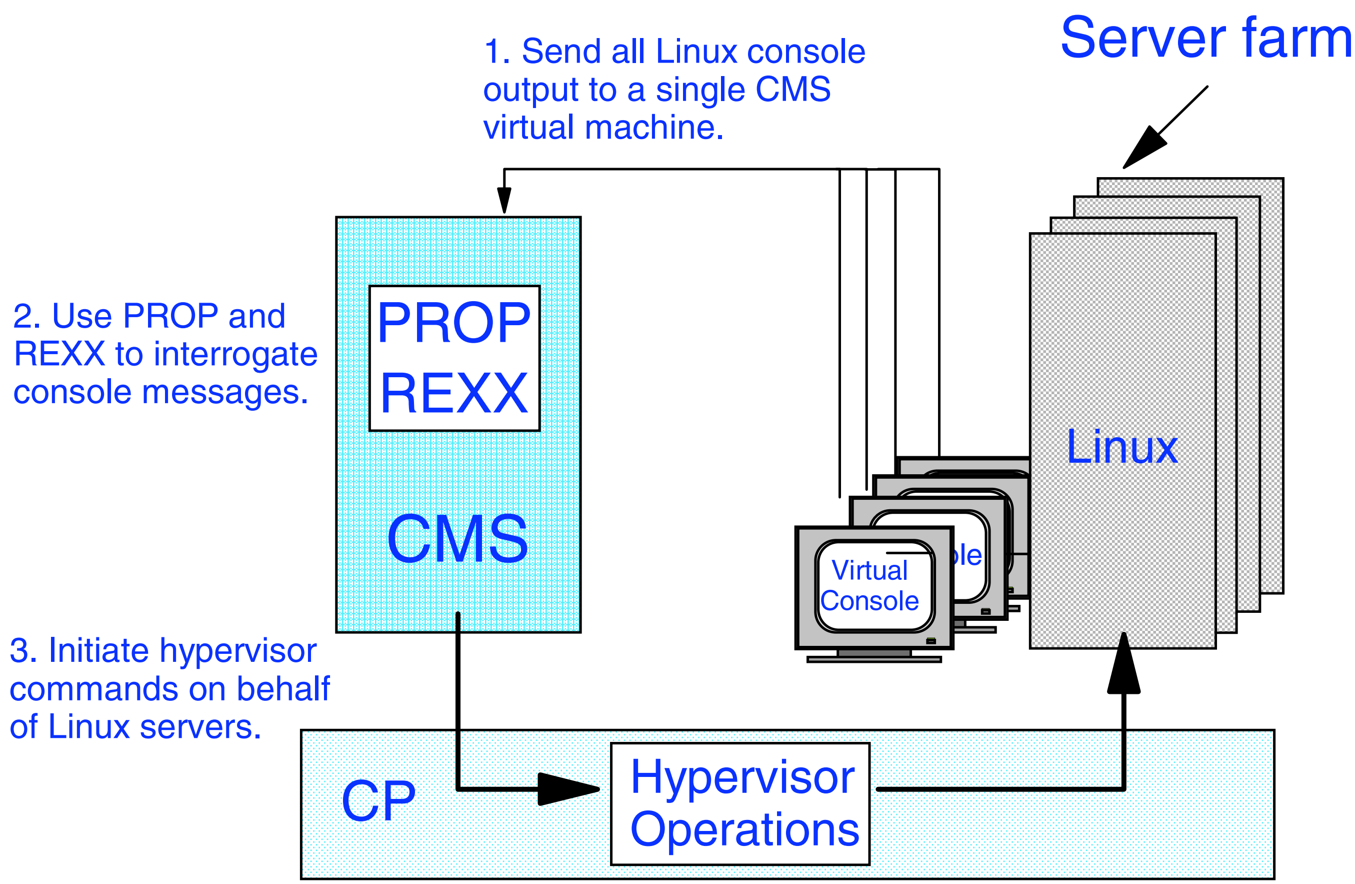
Tracing of virtual machine

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

Display or store into virtual memory

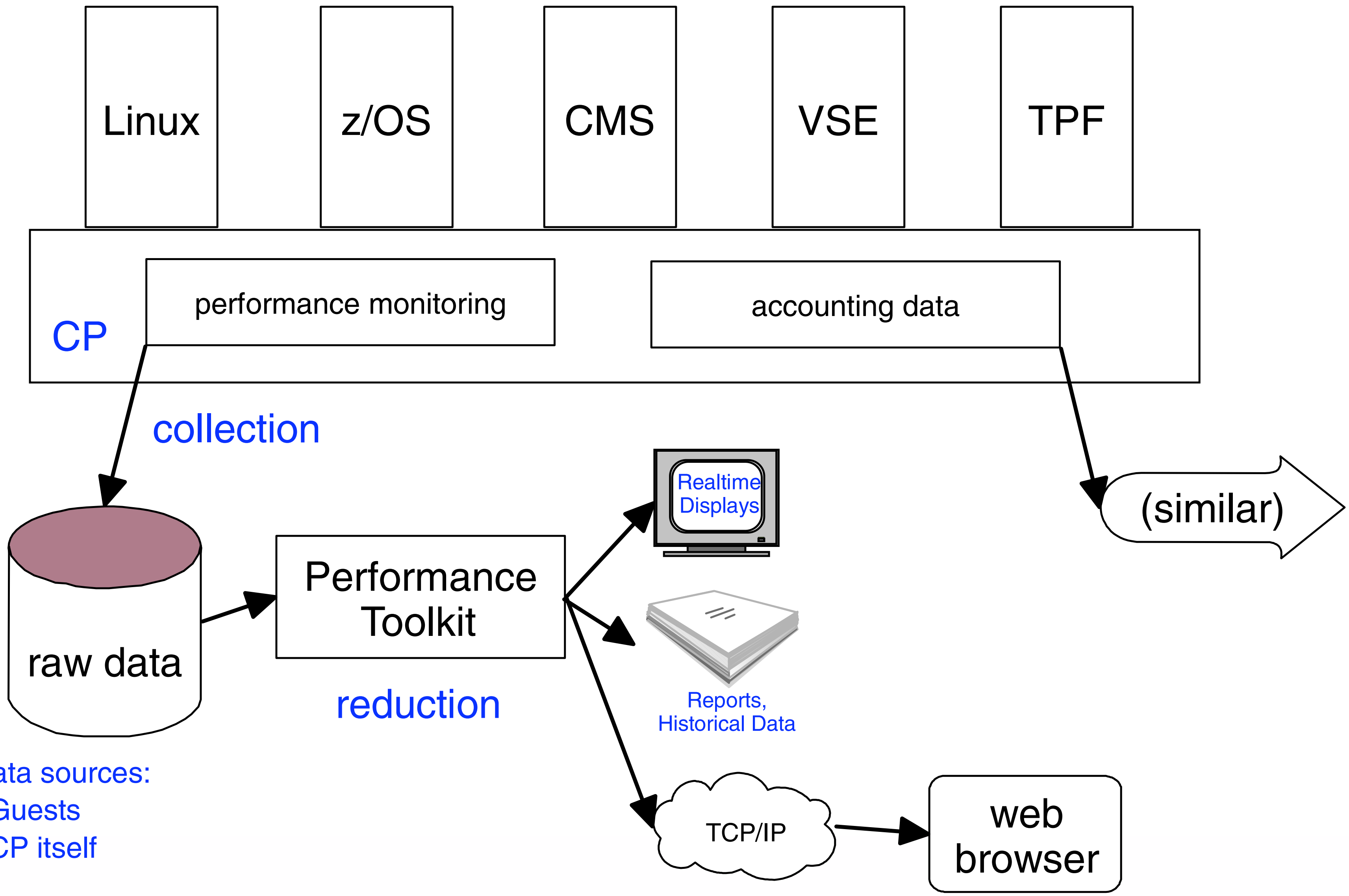
- Helpful, especially when used with tracing
- Valid for various virtual address spaces
- Options for translation as EBCDIC, ASCII, or 390 opcode
- Locate strings in storage
- Store into virtual memory (code, data, etc.)

What: Programmable Operator



What: Performance and Accounting Data

IBM eserver zSeries



Data sources:
- Guests
- CP itself

References

VM web site: www.vm.ibm.com

Publications on VM Web Site

- http://www.vm.ibm.com/pubs/
- Follow the links to the latest z/VM library
- Of particular interest:
  - z/VM CP Command and Utility Reference
  - z/VM CP Planning and Administration
  - z/VM CP Programming Services
  - z/VM Performance

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

- Good article on SIE

End of Presentation

Question and Answer Time

Bonus Material

Some obsolete, some too detailed.
Only the speaker knows.

Phrases Associated with Virtual Machines

In VM...

- *Guest*: any system operating in a virtual machine
- *Running first level:* running as close to the hardware as it permits
- *Running under VM*: running a system as a guest of VM
- *Running on (top of) VM*: same as *running under VM*
- *Running second level:* running VM as a guest of VM
- A virtual machine may have multiple *virtual processors*

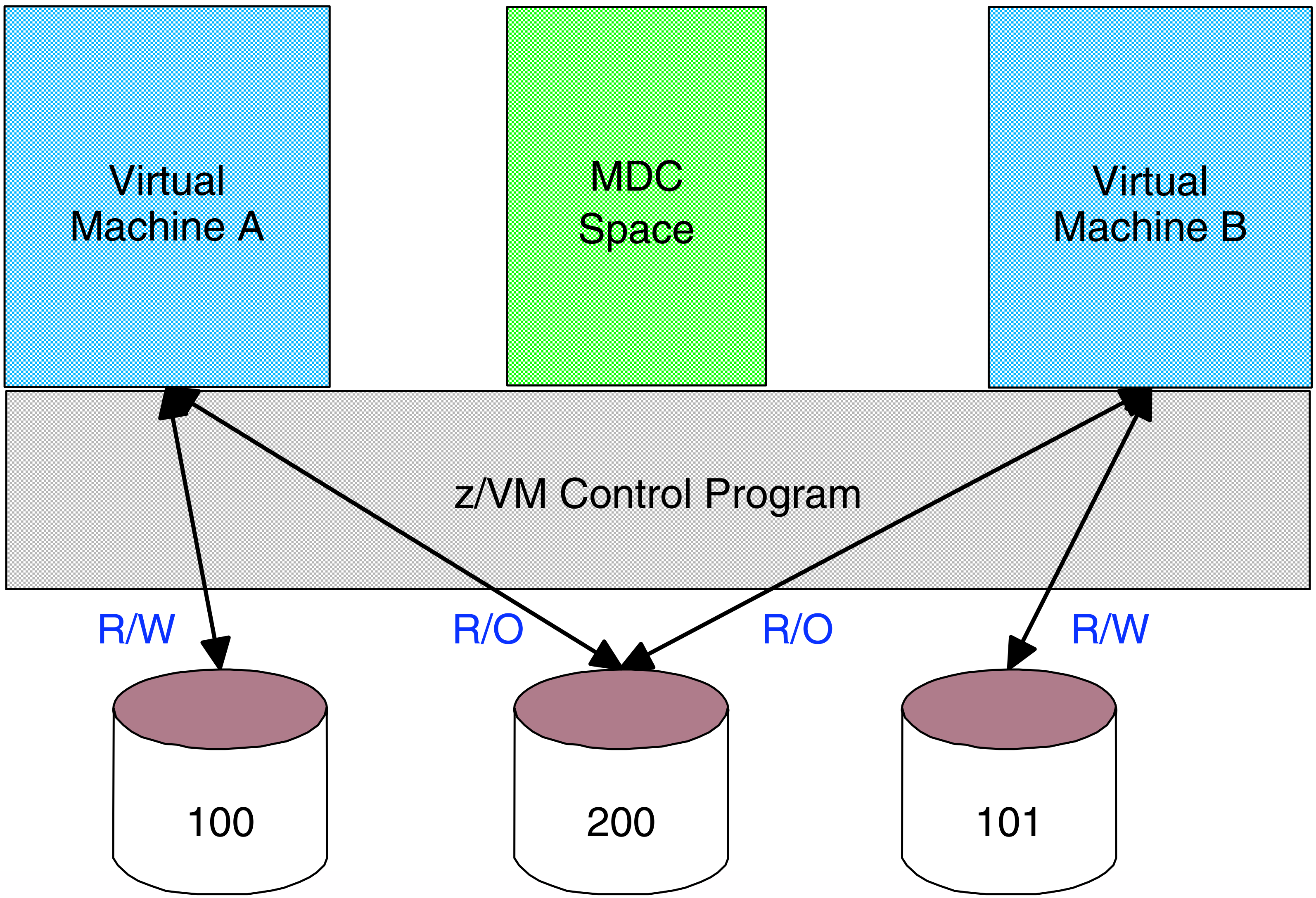
In relationship to partitioning...

- *LPAR hypervisor:* the code that creates and manages partitions
- *Logical partition (LPAR):* Similar to a virtual machine, but instantiated by LPAR hypervisor
- *Logical processor:* LPAR equivalent of a virtual processor
- *Running native(ly):* running without LPAR
- *Running in basic mode:* same as *running native(ly)*

VM: lots of contexts, sharing very important.

LPARs: few contexts, isolation very important.

VM Data in Memory Features - MDC



zSeries Dialects

Architecture

- Strict and formal language

VM

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

z/OS

- Evolved from MVS
- Not really a virtualizer (does virtualize memory)

LPAR hypervisor (creator of logical partitions or LPARs)

- Origins related to VM, though adopted a unique language
- Makes use of Processor Resource/System Manager (PR/SM) hardware and firmware so as to manage partitions

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 zSeries 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

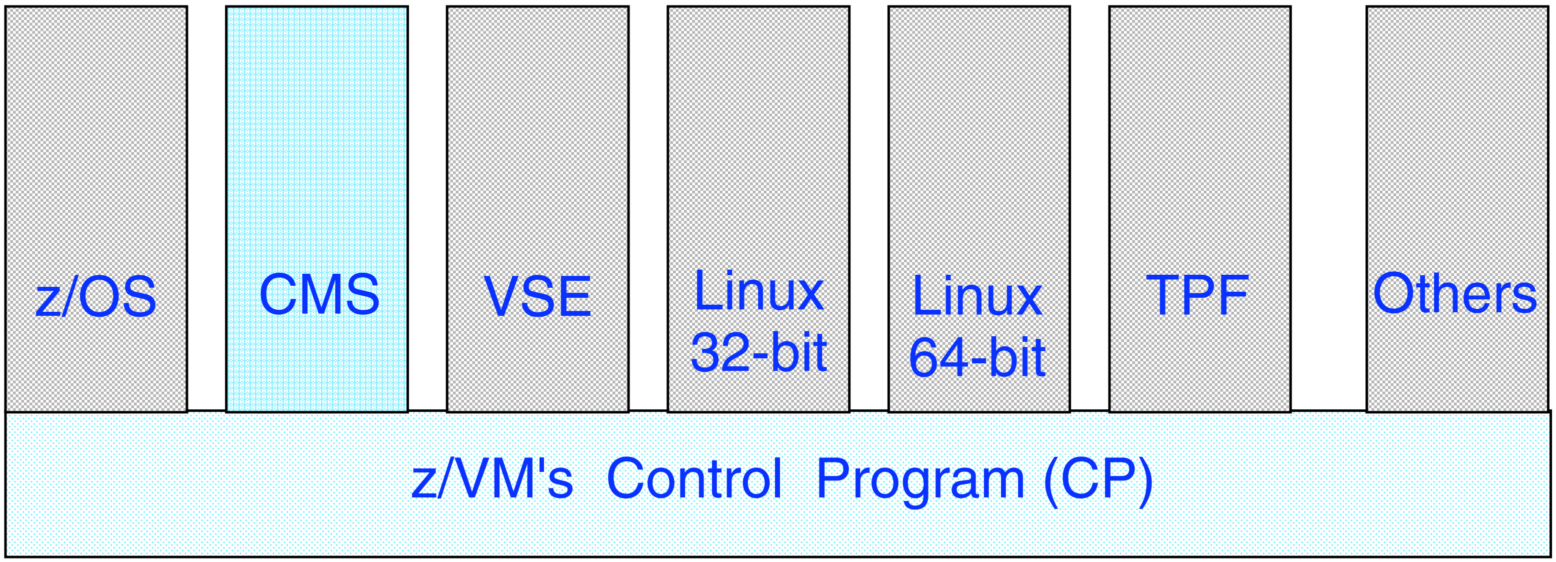
- Processes the same as ESA mode (compatibility with older VM releases)

370

- No longer supported as a virtual machine mode
- Processes according to ESA/370 architecture
- CP and CMS provide 370 Accomodation features to help run 370 applications in ESA, XA, and XC modes (DAT off)

Virtual Machine Basics in Practice

IBM eserver zSeries



- Control Program - manages virtual machines that adhere to 390 or zSeries architecture
- Extensions available through CP system services and features
- CMS is special single user system and is part of z/VM

Getting Started

IML

- Initial Machine Load or Initial Microcode Load
- 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

- 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 **CP IPL** command
- Linux *kernel* is like VM *nucleus*
- Analogous to the LPAR *LOAD* function

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

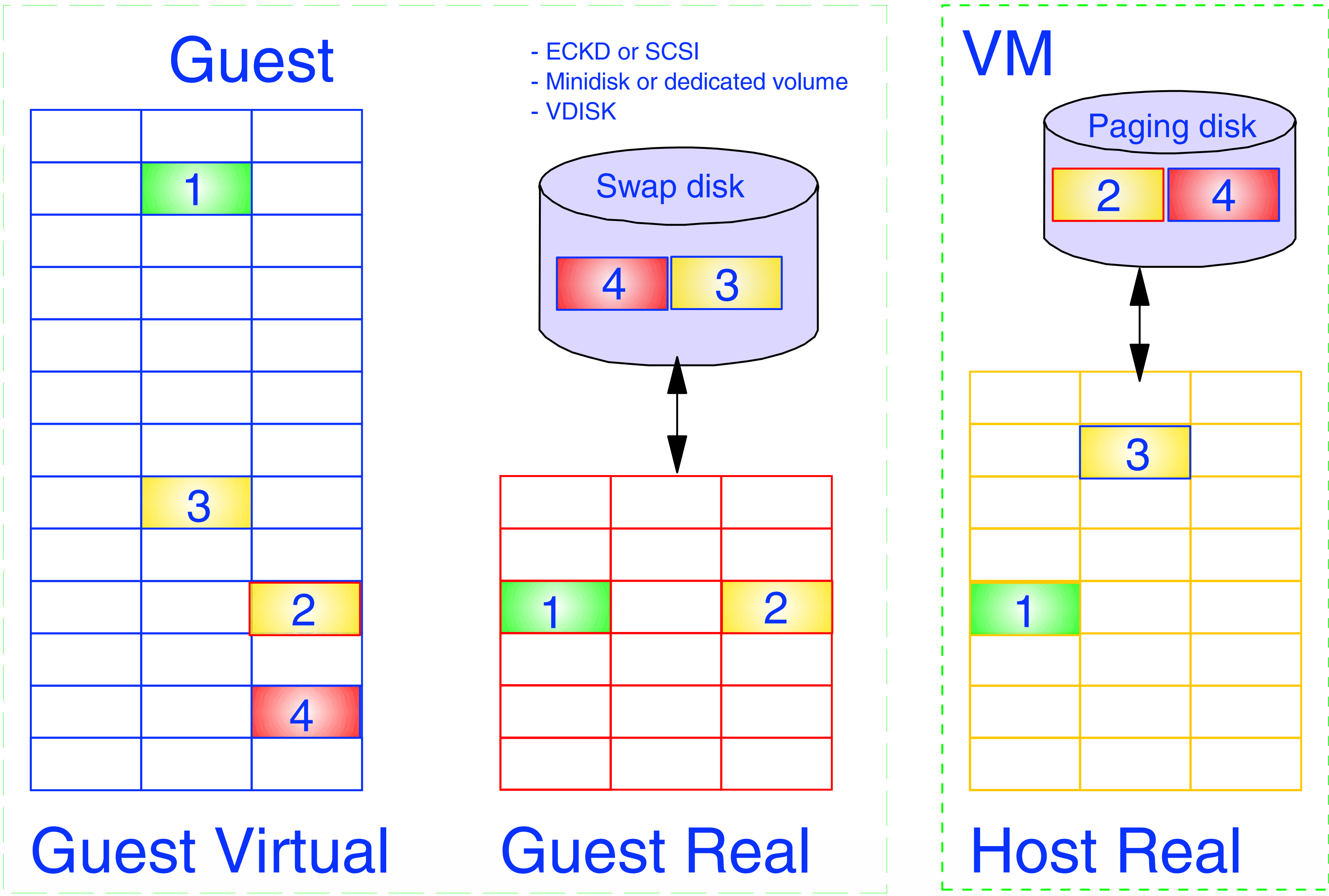
- Writable by guest, or by CP
- Restricted (sensitive data)
- Can have both exclusive and shared ranges

Virtual Machine Address Translation

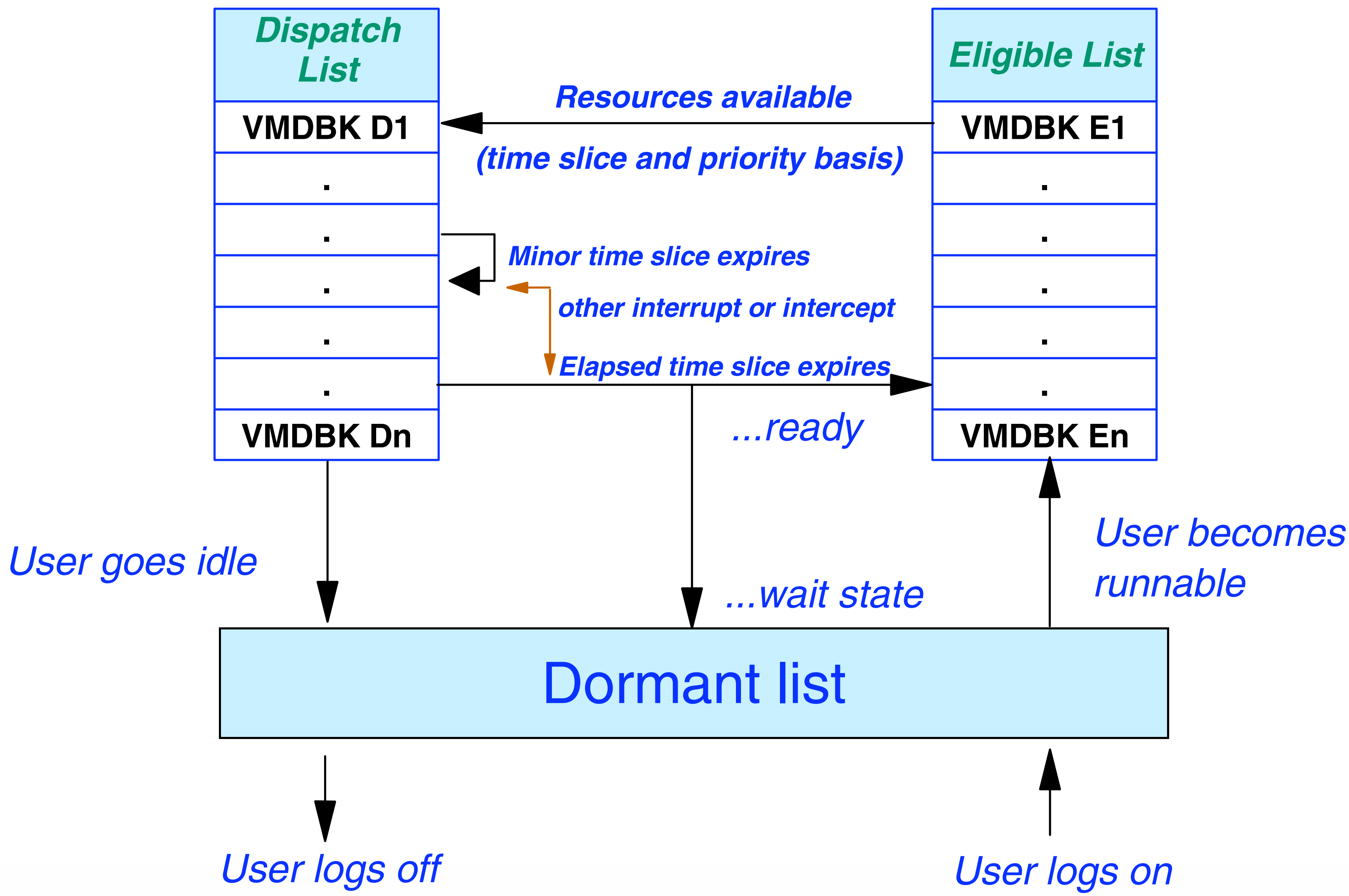
V=R (Virtual=Real)	V=F (Virtual=Fixed)	V=V (Virtual=Virtual)
Fixed contiguous area of host real storage	Fixed contiguous area of host real storage	Does not map permanently to host real storage
Absolute page zero (low end of V=R area) - no address translation	High end of V=R area- never absolute page 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	No 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 Version 5	Not supported in z/VM Version 5	Available in z/VM Version 5

VM Memory Virtualization

IBM eserver zSeries



Classic Scheduler / Dispatcher Picture



I/O Resources

Configuration

- Dedicated devices
- Virtualized (minidisks, crypto, tape, network)
- Simulated (Guest LAN, vCTCA, HiperSockets, virtual disk in storage)

Control and Limits

- Indirect control through "share" setting
- Real devices can be "throttled" at device level
- Priority can be set for virtual machine
  - CP uses to affect queue placement for DASD devices
  - HW uses to affect priority in channel usage
- Minidisk Cache (MDC) fair share limits can be turned off for virtual machine

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

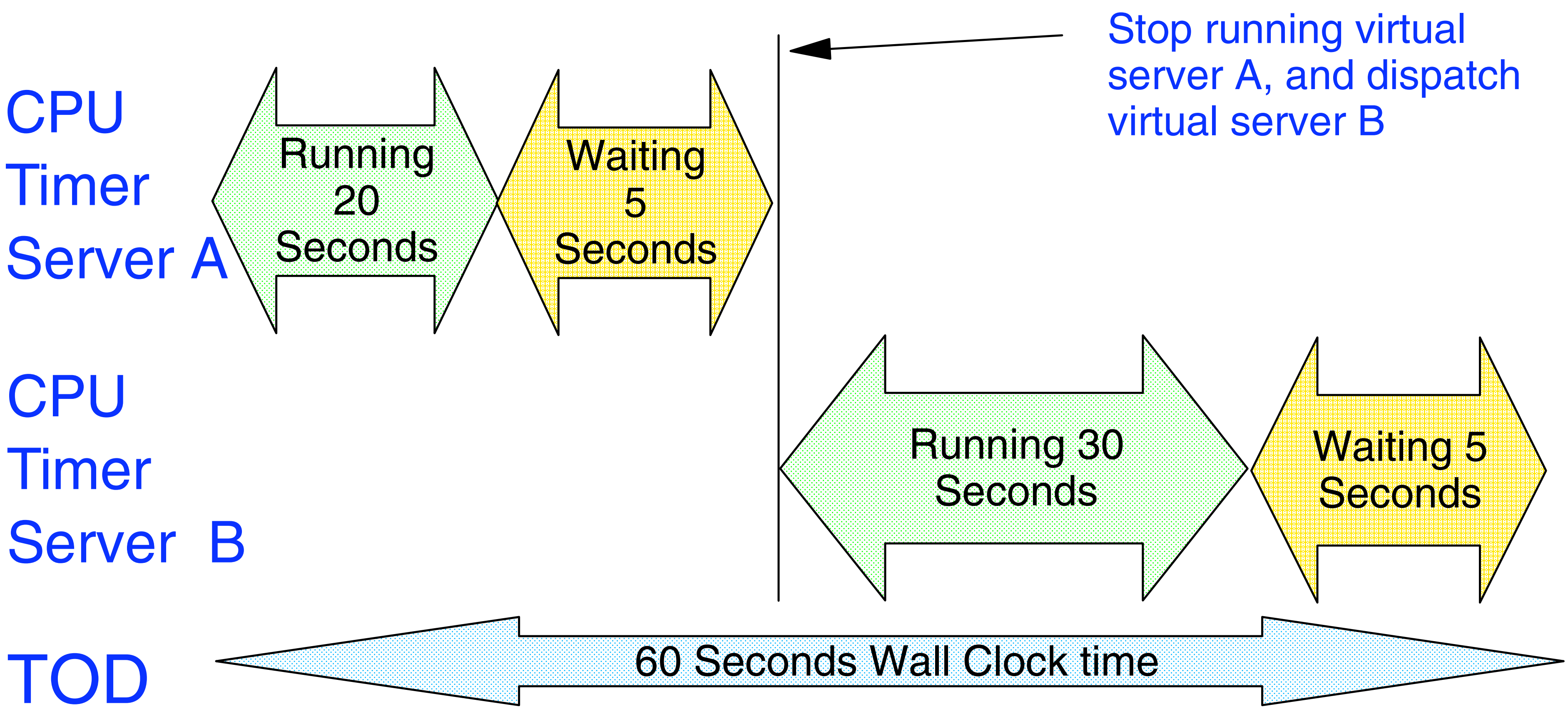
Anomaly

- 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

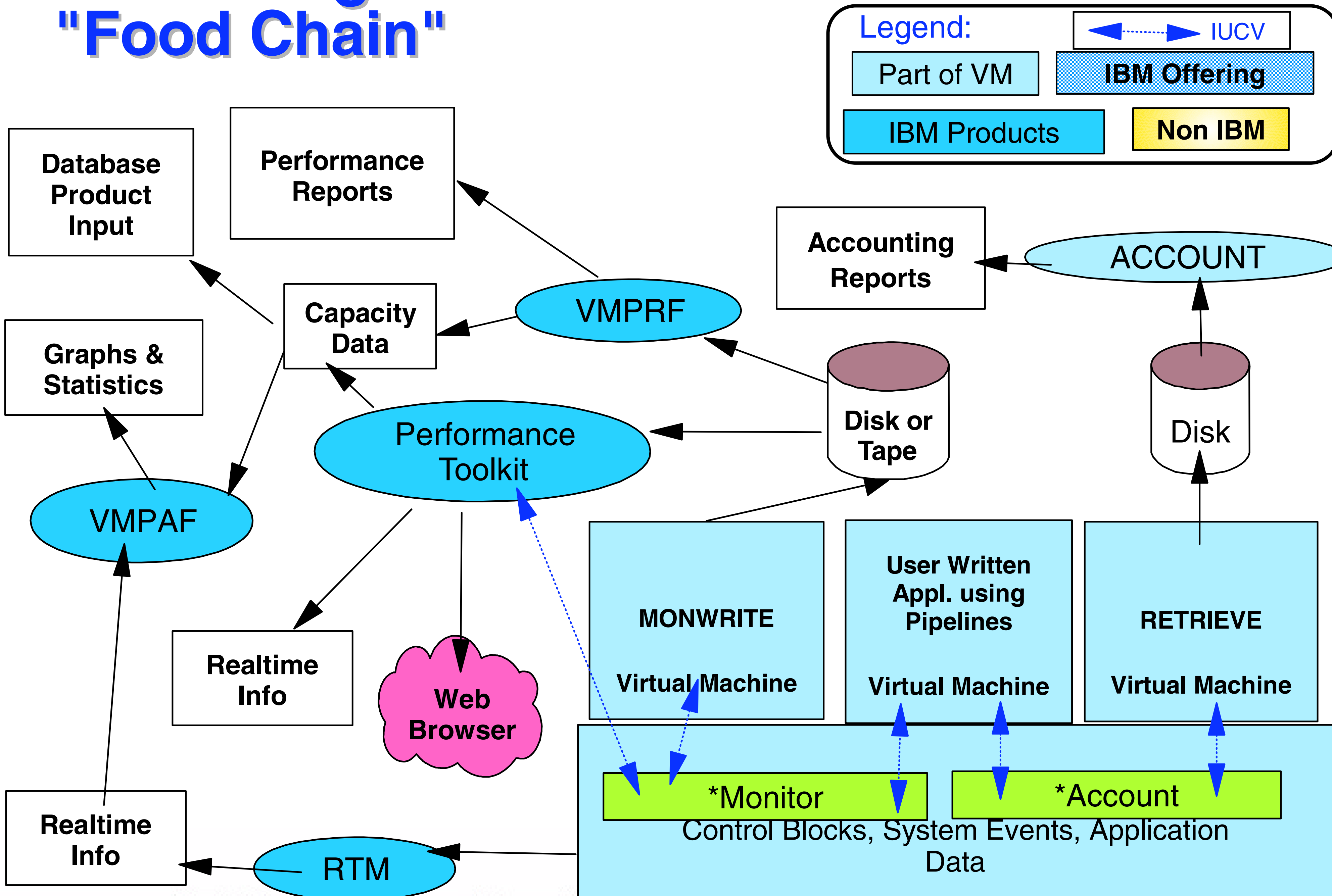
- Same potential, but seldom shares processors to high enough degree to create drastic anomalies

Anomalies of Time

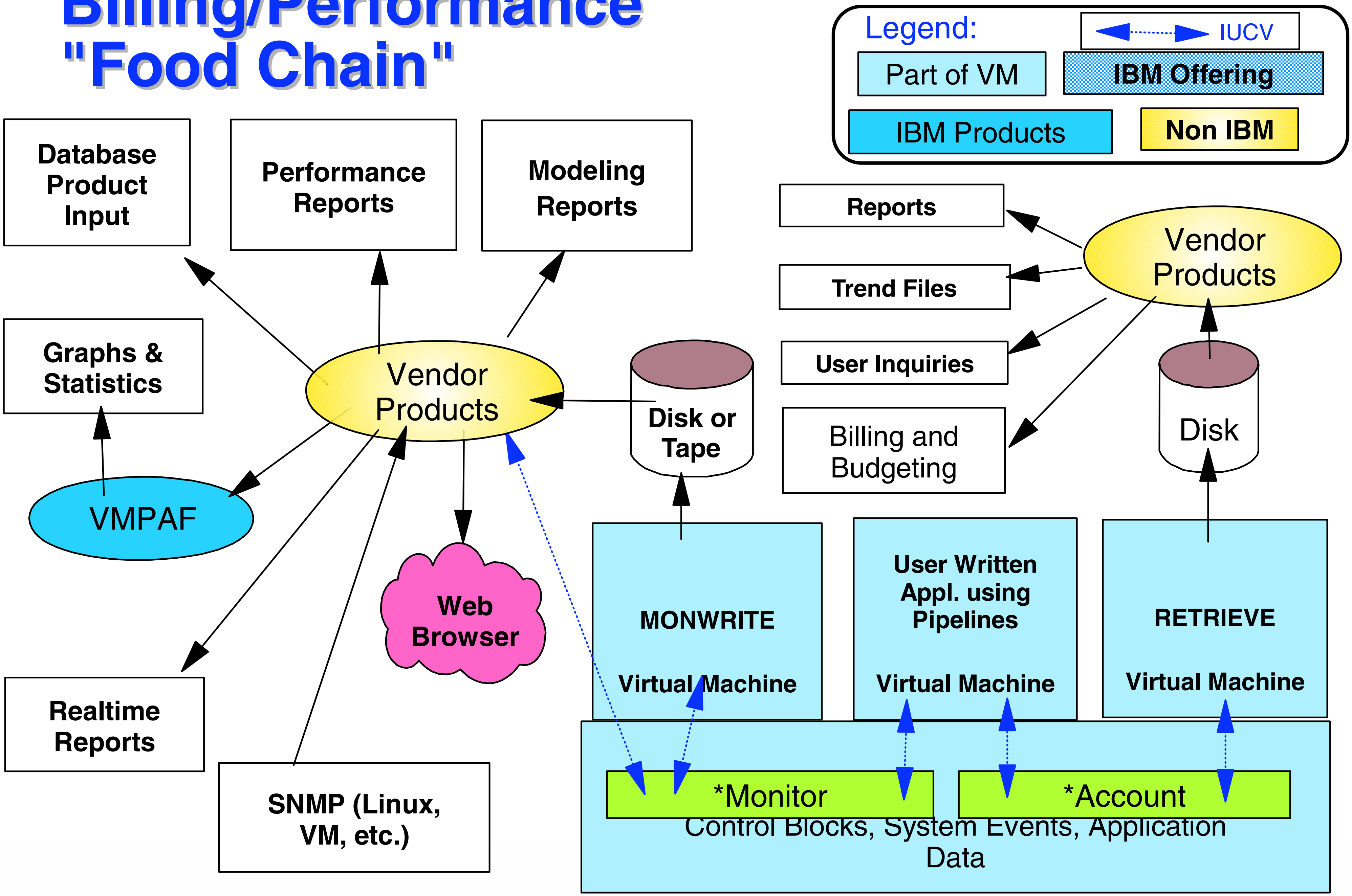


Virtual Server	Total CPU Timer	CPU Timer 'busy'	Incorrect Utilization	Correct Utilization
A	25	20	80%	33%
B	35	30	86%	50%

IBM Billing/Performance "Food Chain"



ISV Billing/Performance "Food Chain"



Other CP Features of Interest

Various CP Commands

- Get quick performance view - INDICATE USER, INDICATE LOAD, ...
- Manage virtual devices - DEFINE, ATTACH, DETACH, GIVE, ...

Communication

- Special APIs (IUCV, VMCF)
- Virtualized network devices
- MSG and WARNING

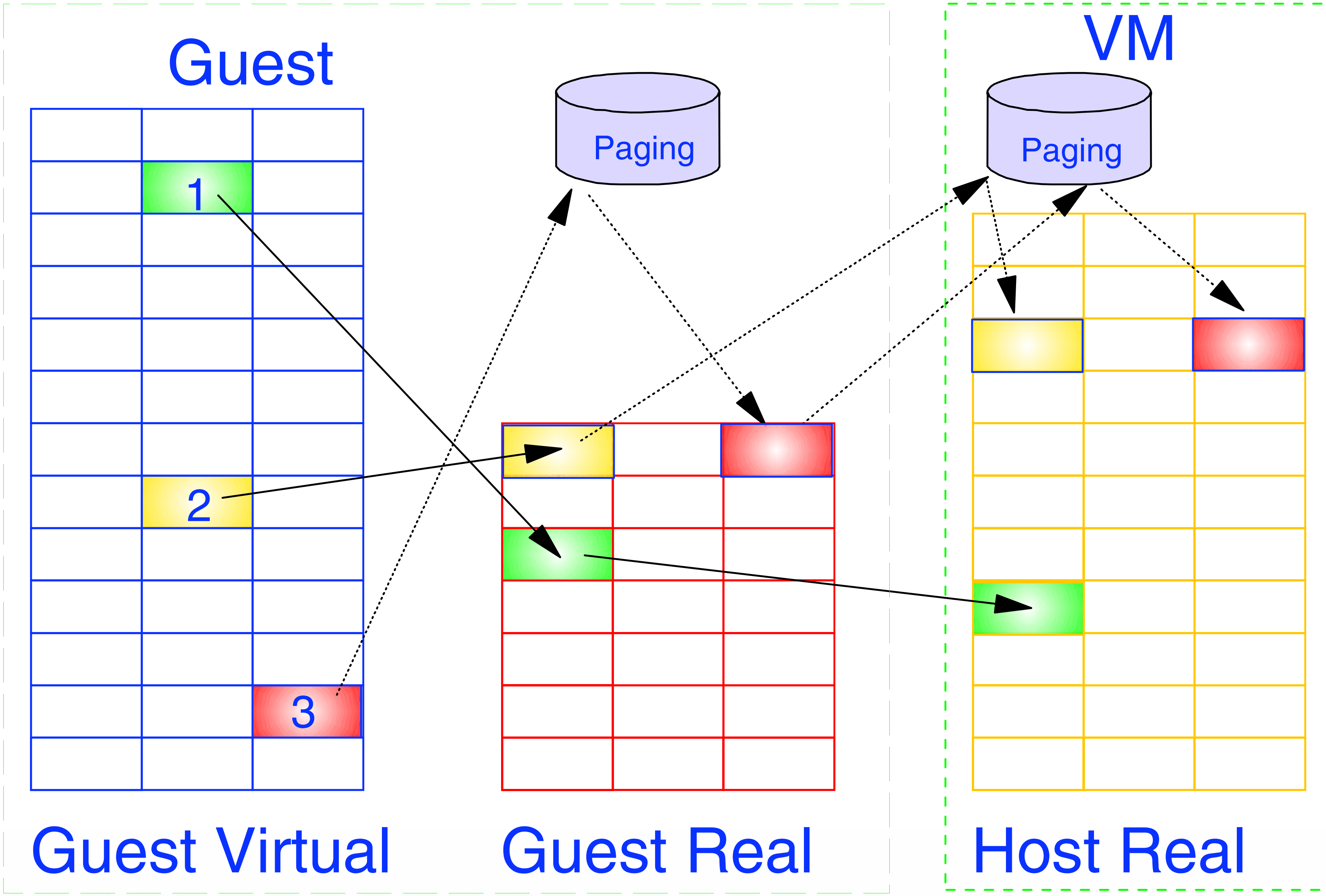
"Star" System Services

- Use IUCV to communicate with special functions in CP
- *MONITOR, *ACCOUNT, *BLOCKIO, *RPI, ...

Programming APIs

- VM Data Space macros
  - mapping minidisks
  - page reference pattern
- Asynchronous Page Fault macro
- IUCV
- Diagnoses

Paging Considerations



How: Layout of Real Storage

IBM eserver zSeries