IBM Systems & Technology Group

Introduction to Virtualization: z/VM Basic Concepts and Terminology
Session 9102

Bill Bitner
VM Performance Evaluation
bitnerb@us.ibm.com

Trademarks

Trademarks

The following are trademarks of the International Business Machines Corporation in the United States and/or other countries. For a complete list of IBM Trademarks, see www.ibm.com/legal/copytrade.shtml: AS/400, DBE, e-business logo, ESCO, eServer, FICON, IBM, IBM Logo, iSeries, MVS, OS/390, pSeries, RS/6000, S/30, VM/ESA, VSE/ESA, Websphere, xSeries, z/OS, zSeries, z/VM

The following are trademarks or registered trademarks of other companies

Lotus, Notes, and Domino are trademarks or registered trademarks of Lotus Development Corporation
Java and all Java-related trademarks and logos are trademarks of Sun Microsystems, Inc., in the United States and other countries
LINUX is a registered trademark of Linus Torvalds
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.
SET and Secure Electronic Transaction are trademarks owned by SET Secure Electronic Transaction LLC.
Intel is a registered trademark of Intel Corporation
* All other products may be trademarks or registered trademarks of their respective companies.

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.

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.

Prices subject to change without notice. Contact your IBM representative or Business Partner for the most current pricing in your geography.

References in this document to IBM products or services do not imply that IBM intends to make them available in every country.

Any proposed use of claims in this presentation outside of the United States must be reviewed by local IBM country counsel prior to such use.

The information could include technical inaccuracies or typographical errors. Changes are periodically made to the information herein; these changes will be incorporated in new editions of the publication. IBM may make improvements and/or changes in the product(s) and/or the program(s) described in this publication at any time without notice.

Any references in this information to non-IBM Web sites are provided for convenience only and do not in any manner serve as an endorsement of those Web sites. The materials at those Web sites are not part of the materials for this IBM product and use of those Web sites is at your own risk.

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

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

Why z/VM?

Infrastructure Simplification

- Consolidate distributed, discrete servers and their networks
- IBM Mainframe qualities of service
- Exploit built-in z/VM system management

Speed to Market

- Deploy servers, networks, and solutions **fast**
- React quickly to challenges and opportunities
- Allocate server capacity when needed

Technology Exploitation

- Linux with z/VM offers more function than Linux alone
- Linux exploits unique z/VM technology features
- Build innovative on demand solutions

Terminology & Background

System z Architecture

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 System z architecture

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

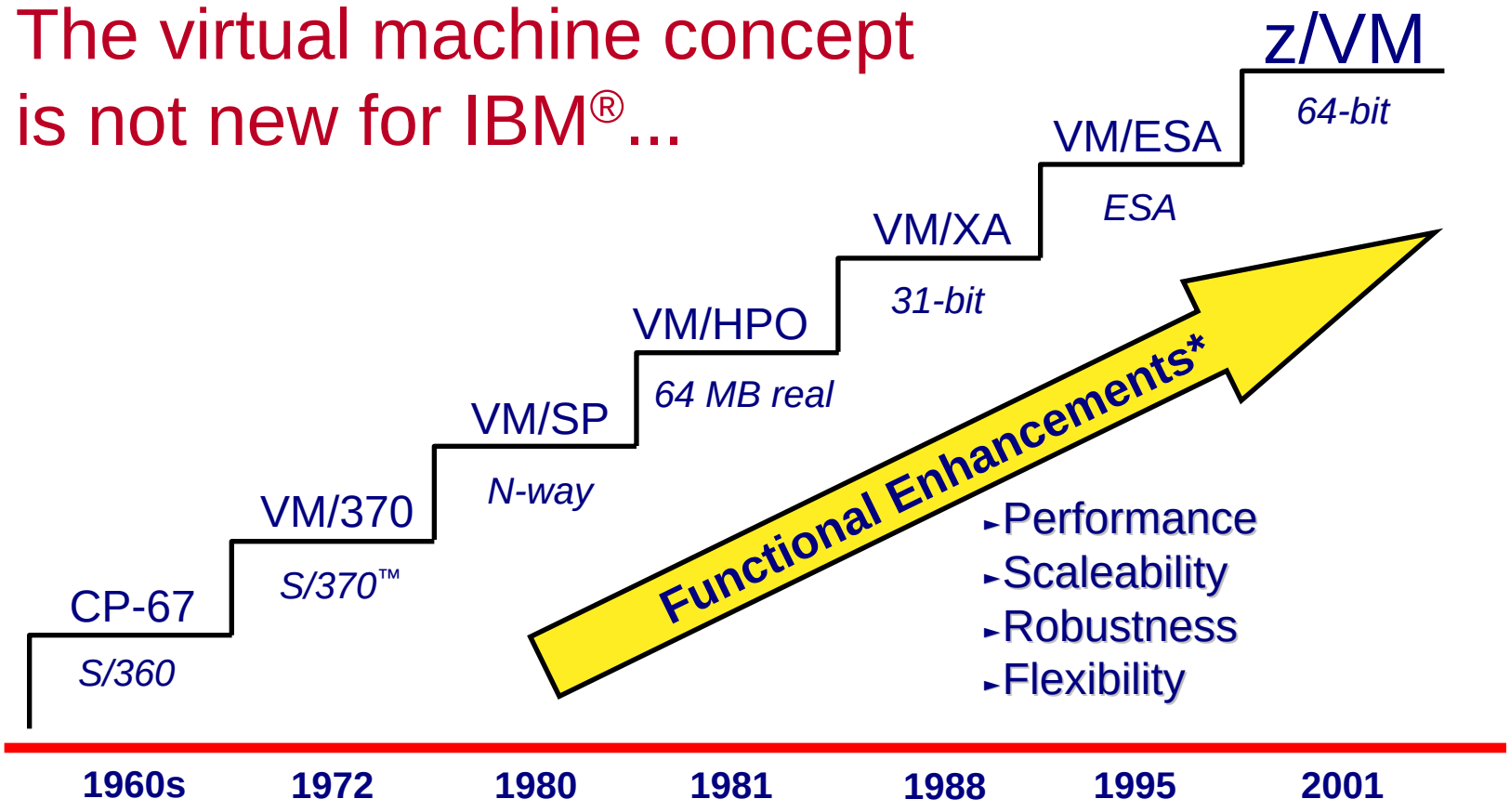
Different *models* implement the architecture in different ways.

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

z800, z900, z890, z990, z9, and z10 are all *models* implementing z/Architecture.

IBM Virtualization Technology Evolution

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



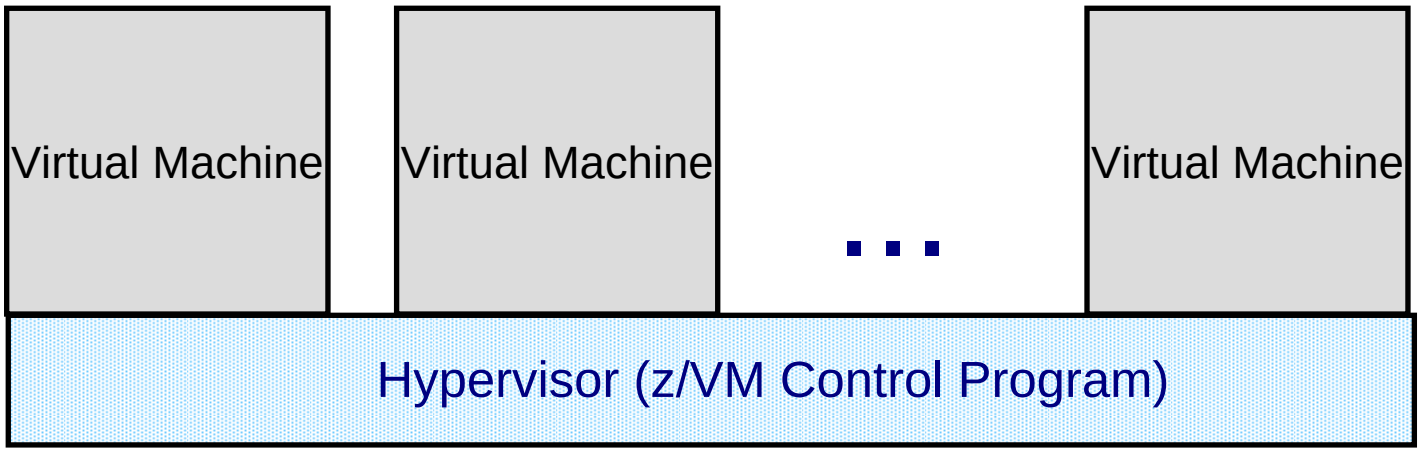
* Investments made in hardware, architecture, microcode, software

System z 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), zIIP (zSeries Integrated Information Processor)
Computer	CEC (central electronics complex) Server

Virtual Machines

What: Virtual Machines



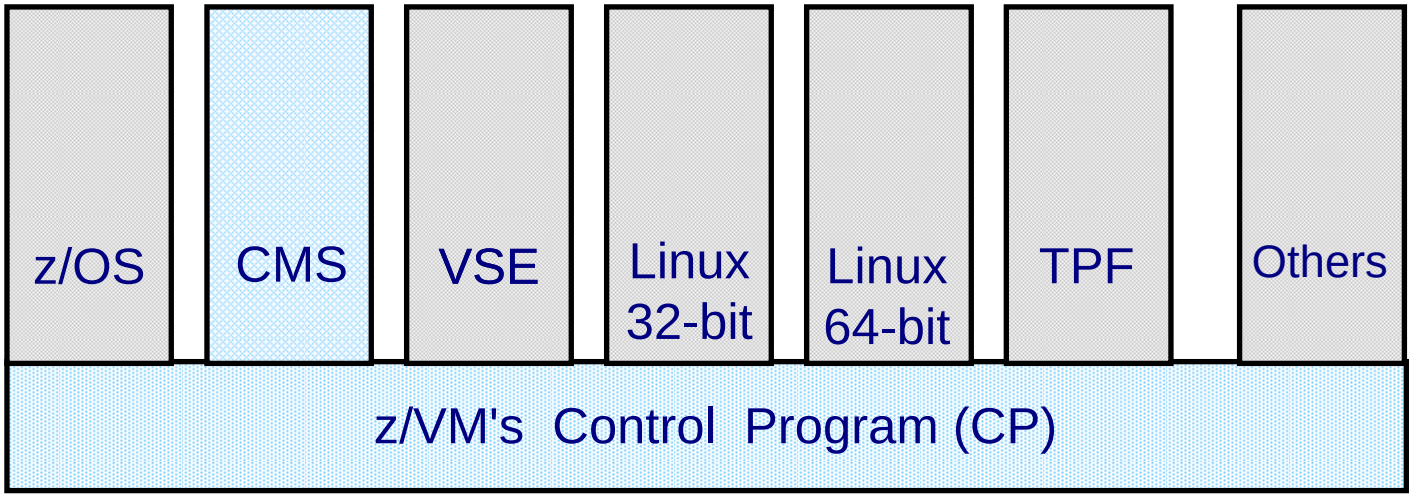
A **virtual machine** is an execution context that obeys the architecture.

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)
- Your limits will depend on the size of your physical zSeries computer

Virtual machine aka VM user ID, VM logon, VM Guest, Virtual Server

What: Virtual Machines in Practice



- 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 interaction via console device

Phrases associated with Virtual Machines

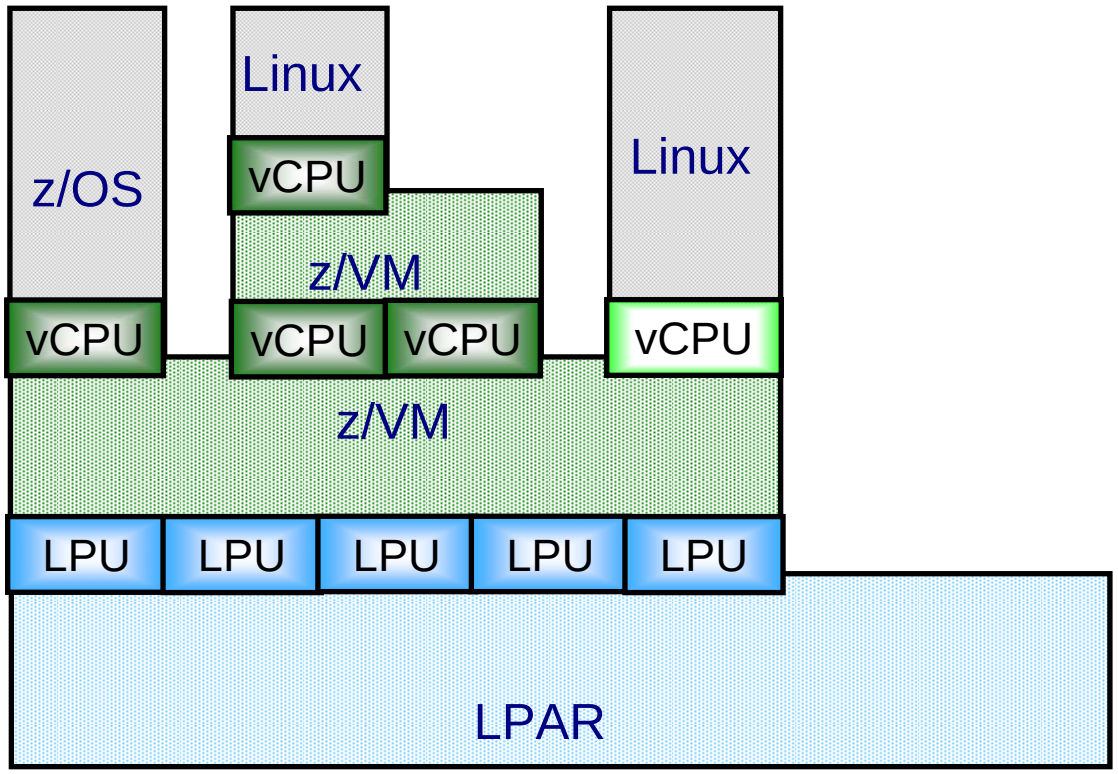
In VM...

- *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*
- Sharing is very important.

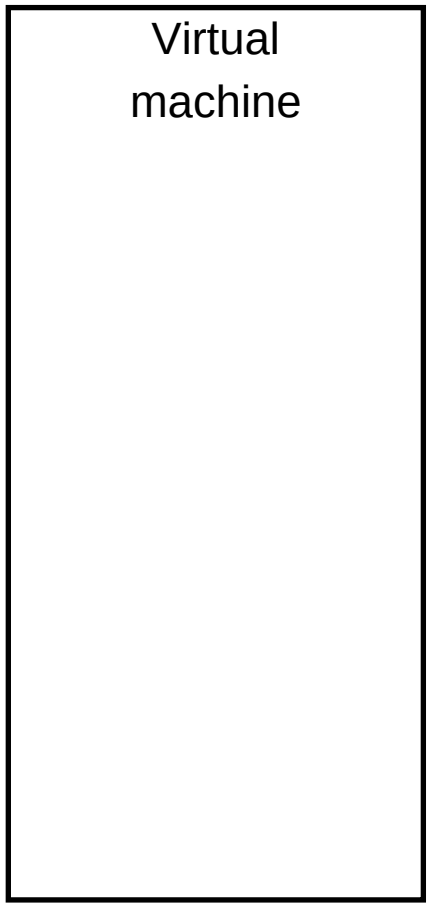
In relationship to LPAR (partitioning)...

- *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
- Isolation is very important.

Phrases Associated with Virtual Machines



What: A 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

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.

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

Processors

What: Processors

Configuration

- Virtual 1- to 64-way
  - Defined in user directory, or
  - Defined by CP command
  - Specialty or General Purpose
- 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

How: Start Interpretive Execution (SIE)

- 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.
- 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

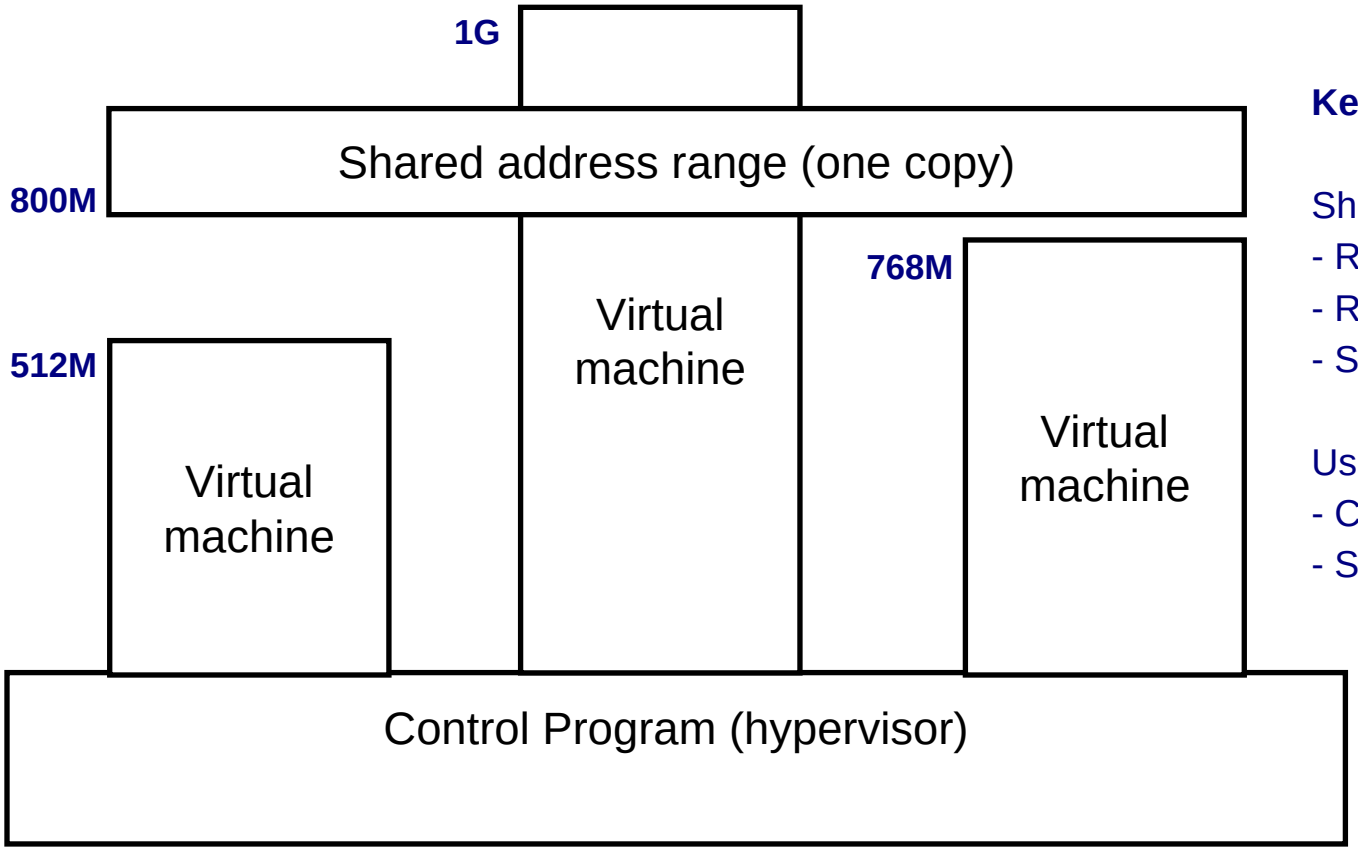
Configuration

- 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: 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

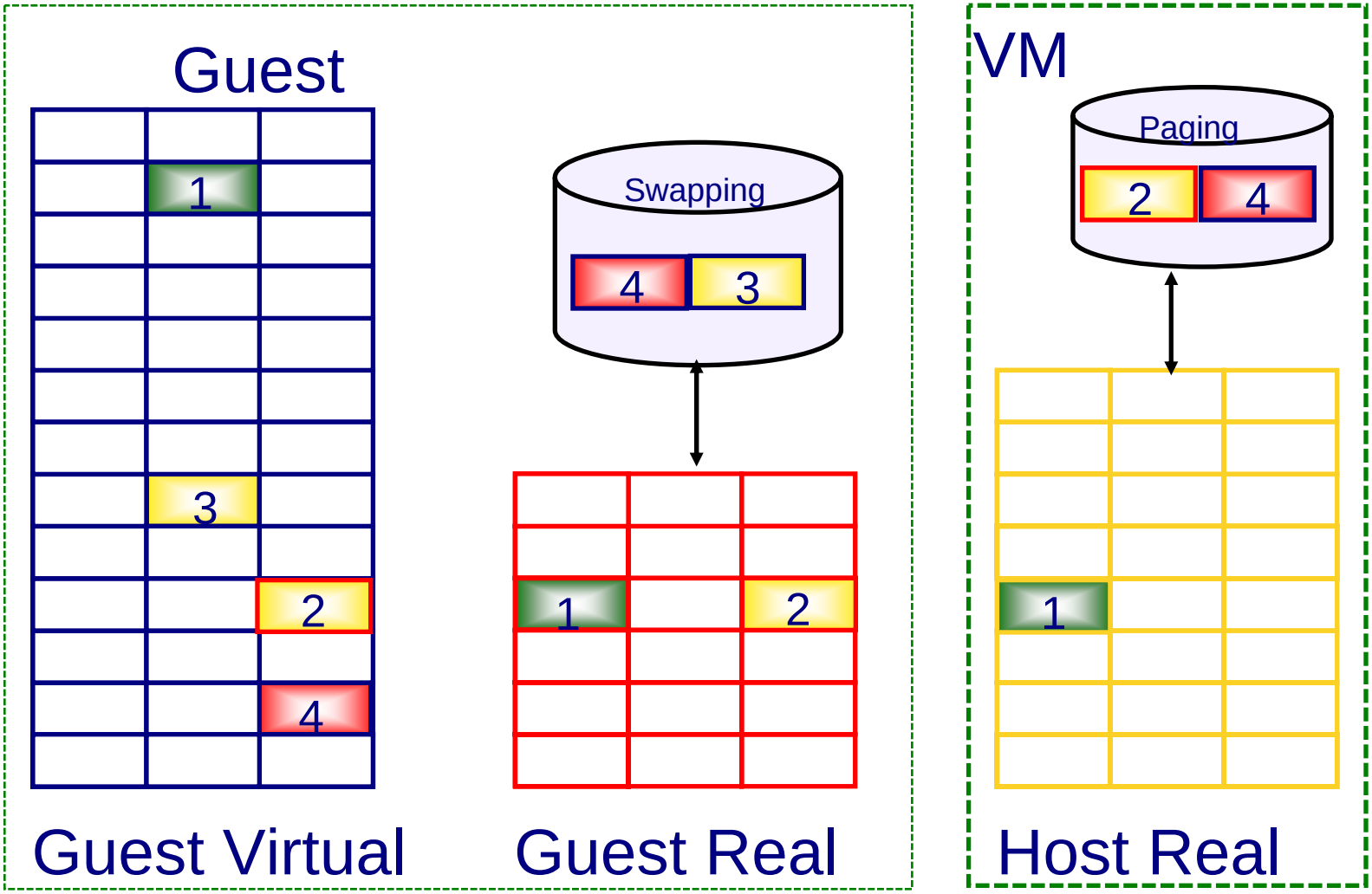
LPAR

- Dedicated storage, no paging

Linux

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

VM Memory Virtualization



I/O Resources

What: Device Management Concepts

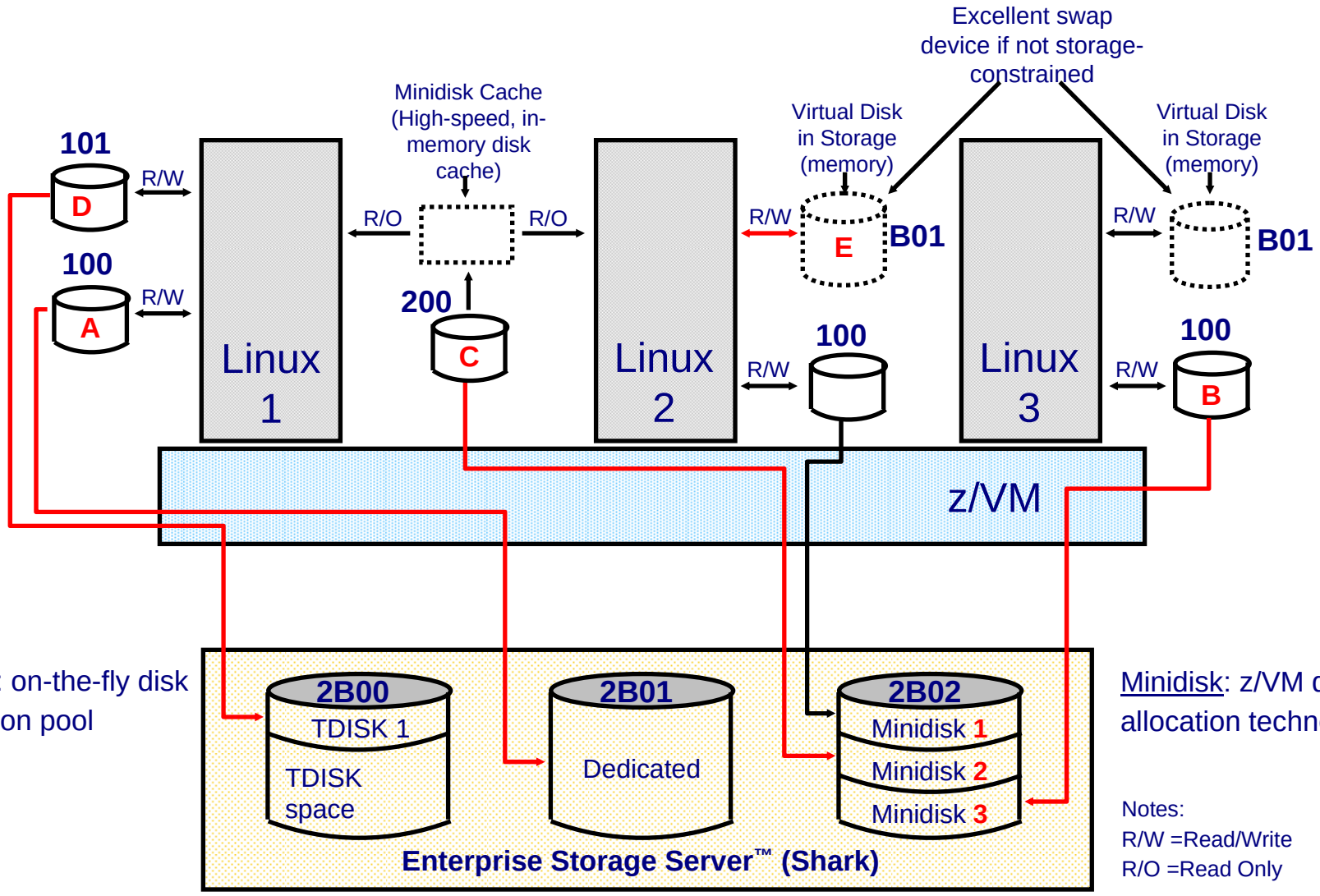
- Dedicated or Attached
  - The guest has exclusive use of the entire real device.
- Virtualized
  - Present a slice of a real device to multiple virtual machines
  - Slice in time or slice in space
  - E.g., DASD, crypto devices
- Simulated
  - Provide a device to a virtual machine without the help of real hardware
  - Virtual CTCAs, virtual disks, guest LANs, spool devices
- Emulated
  - Provide a device of one type on top of a device of a different type
  - FBA emulated on FCP SCSI

What: Device Management Concepts

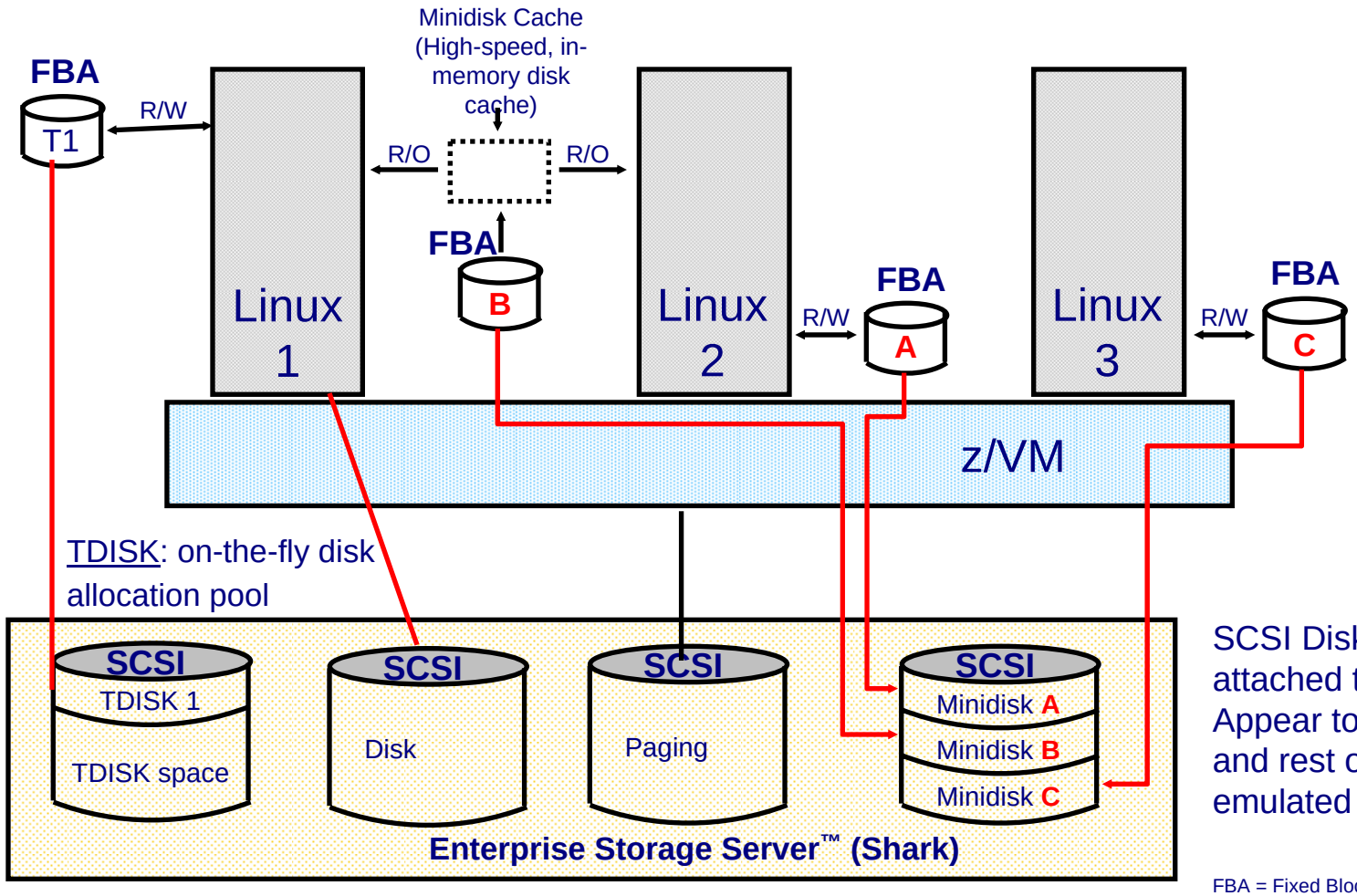
- Terminology
  - RDEV is Real Device
    - can refer to the device address or the control block
  - VDEV is Virtual Device
    - can refer to the device address or the control block
  - UCB is Unit Control Block
    - used in hardware definitions
  - RDEV=UCB=subchannel=device=adapter

- 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



z/VM Disk Technology - SCSI



SCSI Disks attached to z/VM. Appear to guests and rest of VM as emulated FBA.

FBA = Fixed Block Architecture

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

Connecting virtual machines to one another

- Guest LAN
  - QDIO or HiperSockets
- Virtual Switch Guest LAN
  - Layer 2 or Layer 3

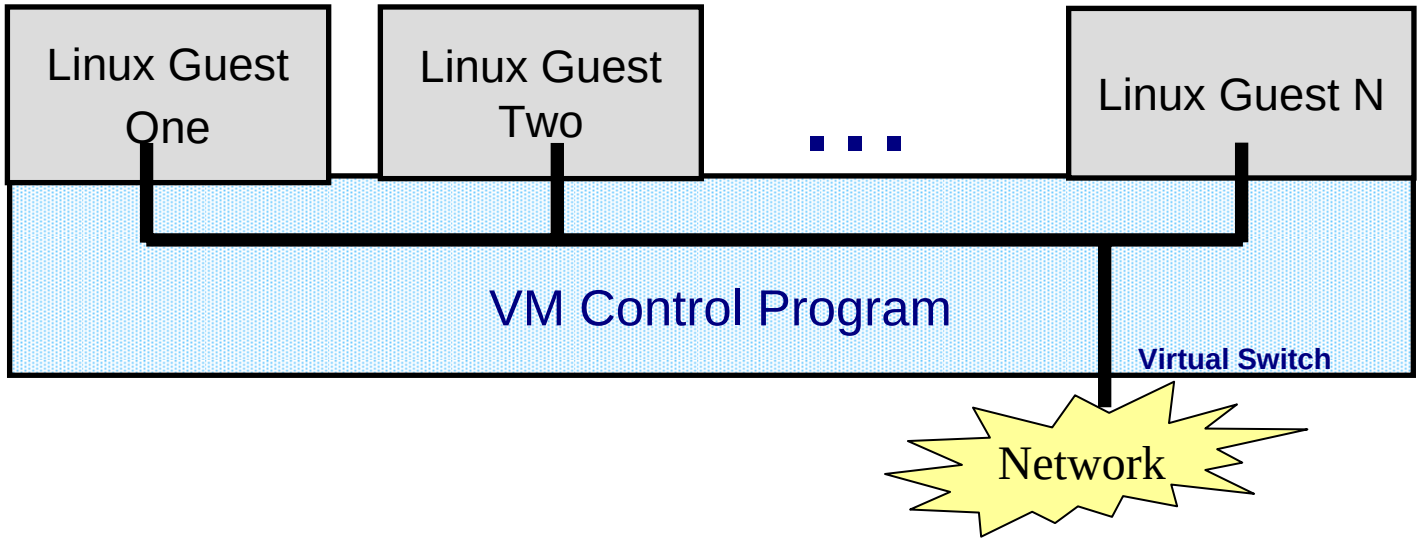
Connecting virtual machines to another LPAR

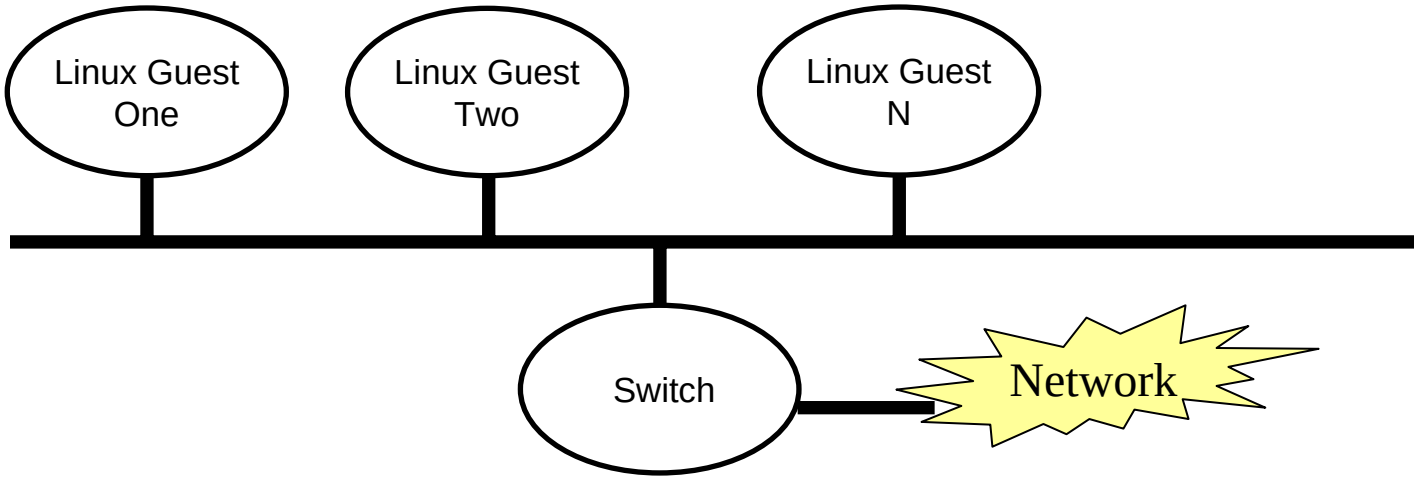
- HiperSockets
- Shared OSA

Connecting virtual machines to the physical network

- Dedicated OSA device
- Virtual Switch
  - IP or MAC oriented

What: Virtual Switch Guest LAN





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

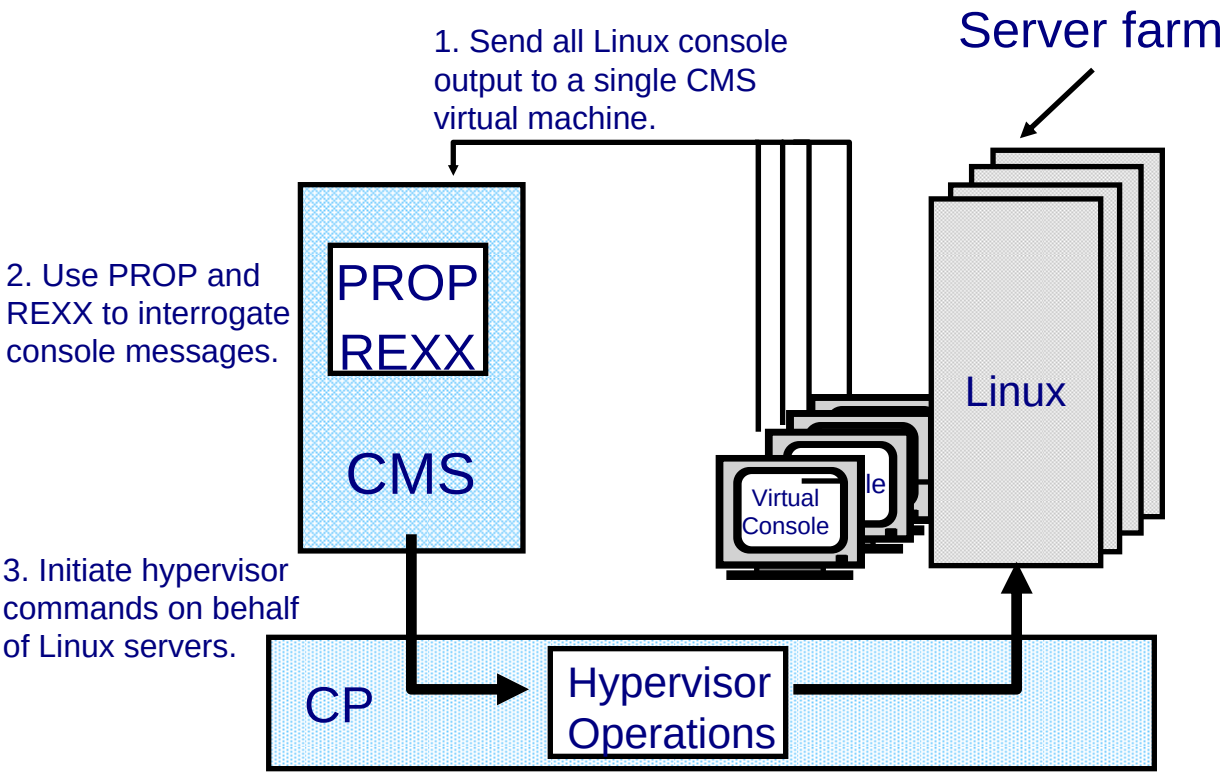
Tracing of virtual machine

- 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

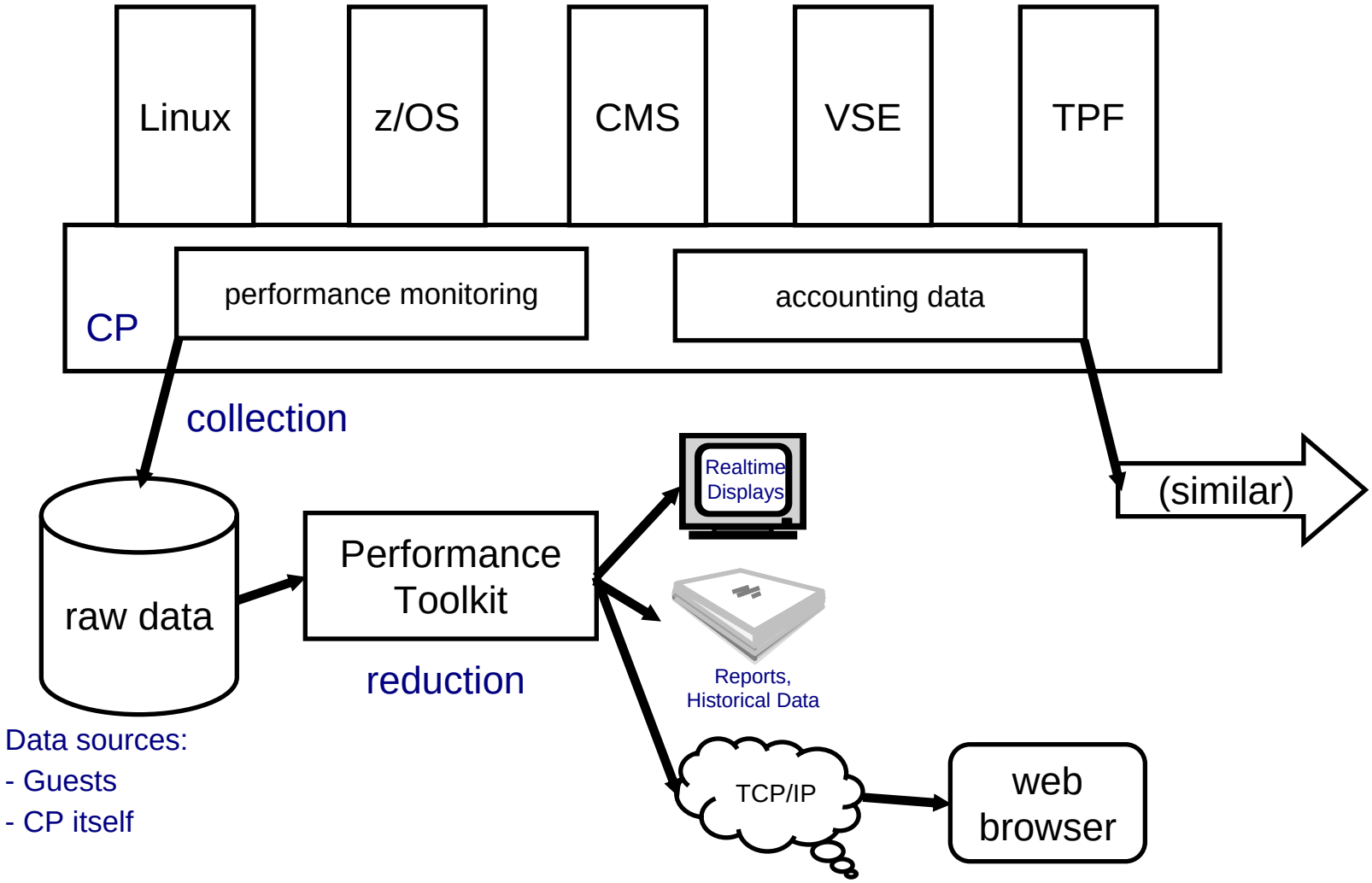
Display or store into virtual memory

- 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



References

- VM web site: www.vm.ibm.com
  - www.vm.ibm.com/events/ for various conferences
  - www.vm.ibm.com/education/ for classes
  - www.vm.ibm.com/techinfo/ for good stuff, plus links to listservs
- 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
- z/Journal article based on this presentation
  - http://zjournal.com/index.cfm?section=article&aid=946
- IBM Systems Journal Vol. 30, No. 1, 1991
  - Good article on SIE
  - http://www.research.ibm.com/journal/sj/301/ibmsj3001E.pdf

End of Presentation

Question and Answer Time