z Systems

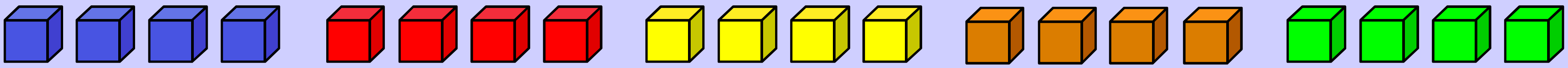
Introduction to Virtualization

SHARE Seattle
Linux & VM Program

Romney White, IBM

romneyw@us.ibm.com
z Systems Architecture and Technology

- **Introduction to Virtualization**
  - Concept
  - Server Virtualization Approaches
  - Hypervisor Implementation Methods
  - Why Virtualization Matters
- **Virtualization on z Systems**
  - Logical Partitions
  - Virtual Machines



Virtual Resources

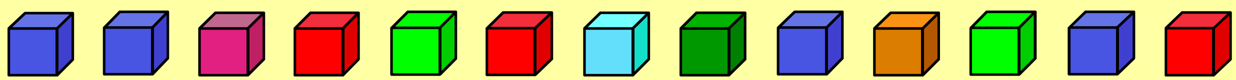
- Proxies for real resources: same interfaces/functions, different attributes
- May be part of a physical resource or multiple physical resources

Virtualization

- Creates virtual resources and "maps" them to real resources
- Primarily accomplished with software or firmware

Resources

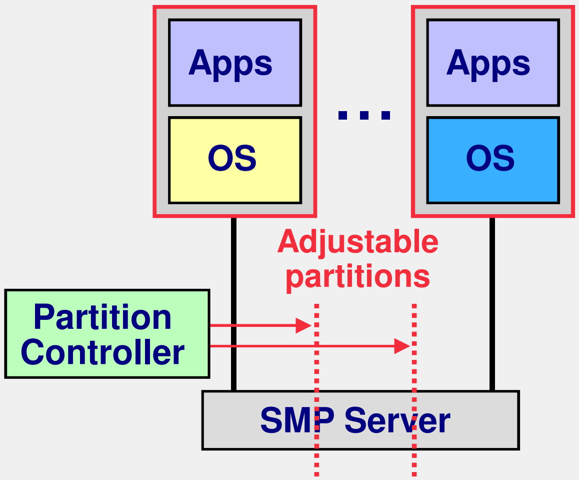
- Components with architecturally-defined interfaces/functions
- May be centralized or distributed - usually physical
- Examples: memory, disk drives, networks, servers



- **Separates presentation of resources to users from actual resources**
- **Aggregates pools of resources for allocation to users as virtual resources**

Server Virtualization Approaches

Hardware Partitioning

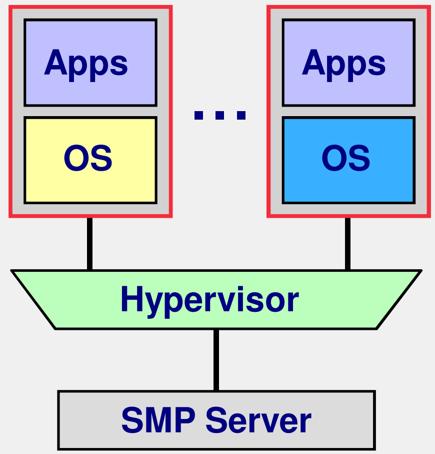


Server is subdivided into fractions each of which can run an OS

Physical partitioning
- S/370™ SI-to-PP and PP-to-SI
- Sun Domains, HP nPartitions

Logical partitioning
- IBM eServer™ pSeries® LPAR
- HP vPartitions

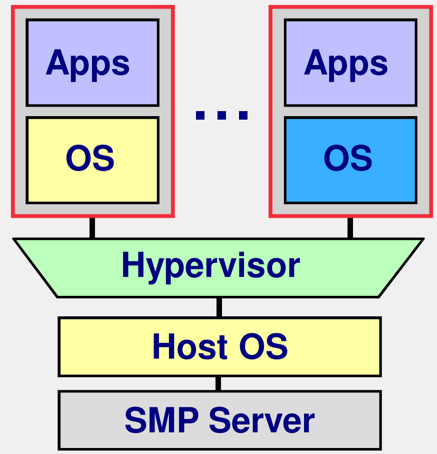
Bare-metal Hypervisor



Hypervisor provides fine-grained timesharing of all resources

Hypervisor software/firmware runs directly on server
- z Systems LPAR and z/VM®
- POWER™ Hypervisor
- VMware ESX Server
- Xen Hypervisor

Hosted Hypervisor

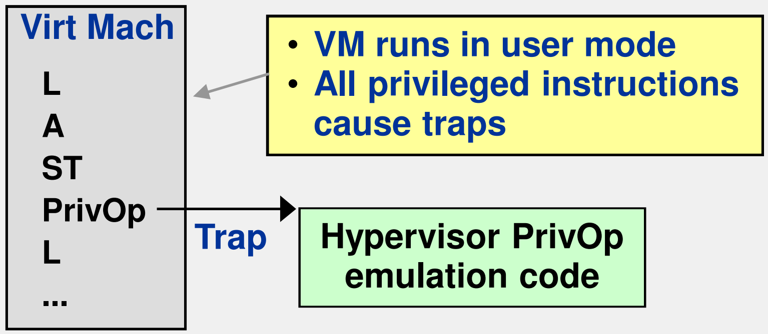


Hypervisor uses OS services to do timesharing of all resources

Hypervisor software runs on a host operating system
- VMware GSX
- Microsoft® Virtual Server
- HP Integrity VM
- KVM

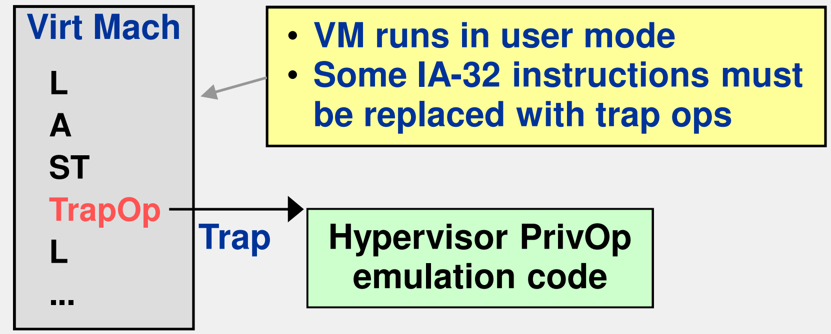
Characteristics:
- Bare-metal hypervisors offer high efficiency and availability
- Hosted hypervisors are useful for clients where host OS integration is important
- Hardware partitioning is less flexible than hypervisor-based solutions

Trap and Emulate



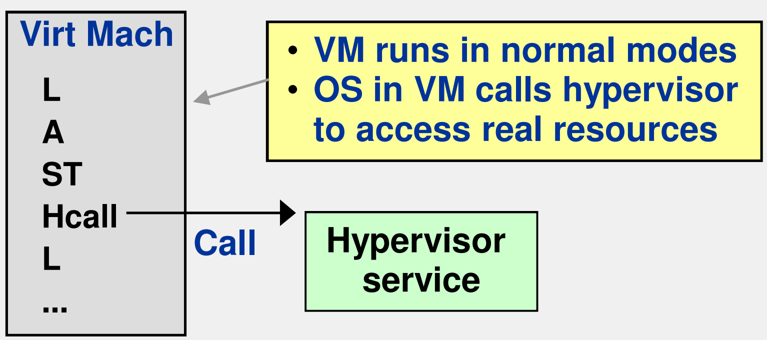
Examples	CP-67, VM/370
Benefits	Runs unmodified OS
Issues	Substantial overhead

Translate, Trap, and Emulate



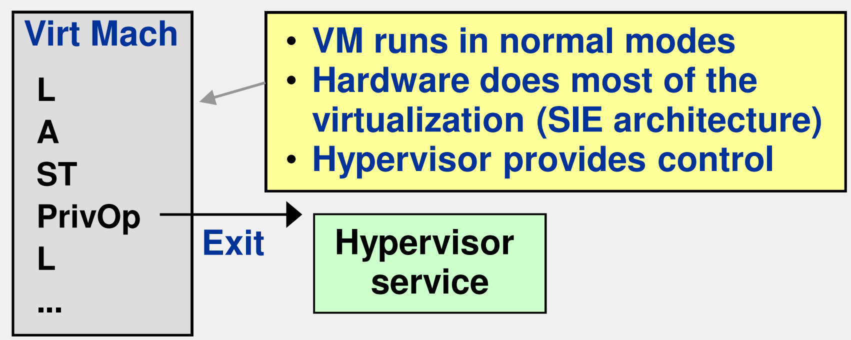
Examples	VMware, Microsoft VS
Benefits	Runs unmodified, translated OS
Issues	Substantial overhead

Hypervisor Calls (“Paravirtualization”)



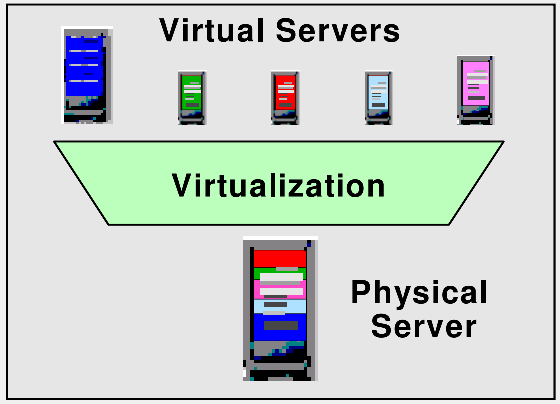
Examples	POWER Hypervisor, Xen
Benefits	High efficiency
Issues	OS must be modified to issue Hcalls

Direct Hardware Virtualization



Examples	z Systems LPAR, z/VM, Xen
Benefits	High efficiency, runs unmodified OS
Issues	Requires underlying hardware support

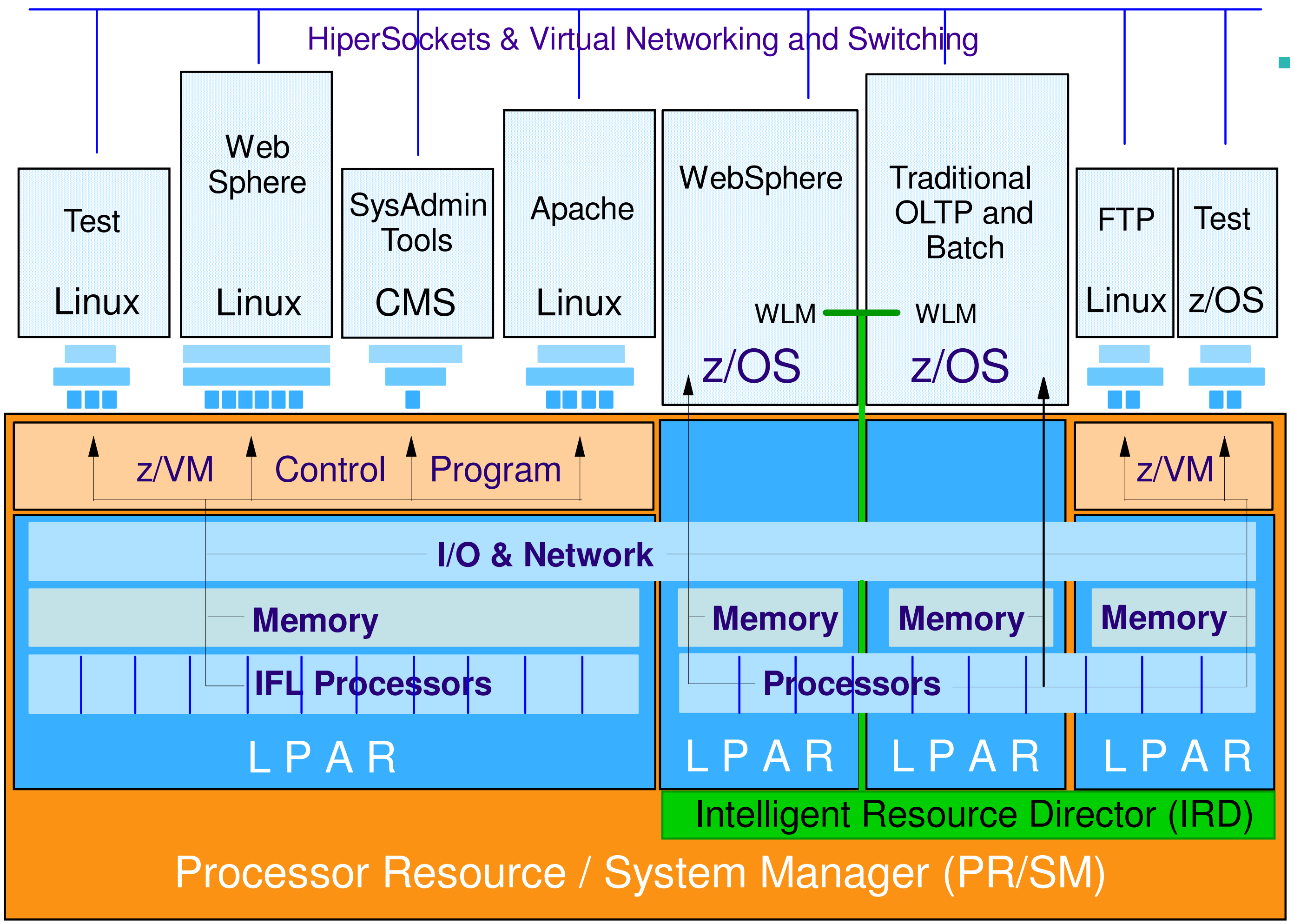
Roles:
- Consolidations
- Dynamic provisioning / hosting
- Workload management
- Workload isolation
- Software release migration
- Mixed production and test
- Mixed OS types/releases
- Reconfigurable clusters
- Low-cost backup servers



Possible Benefits:
- High resource utilization
- Great usage flexibility
- Enhanced workload QoS
- High availability / security
- Low cost of availability
- Low management costs
- Enhanced interoperability
- Legacy compatibility
- Investment protection

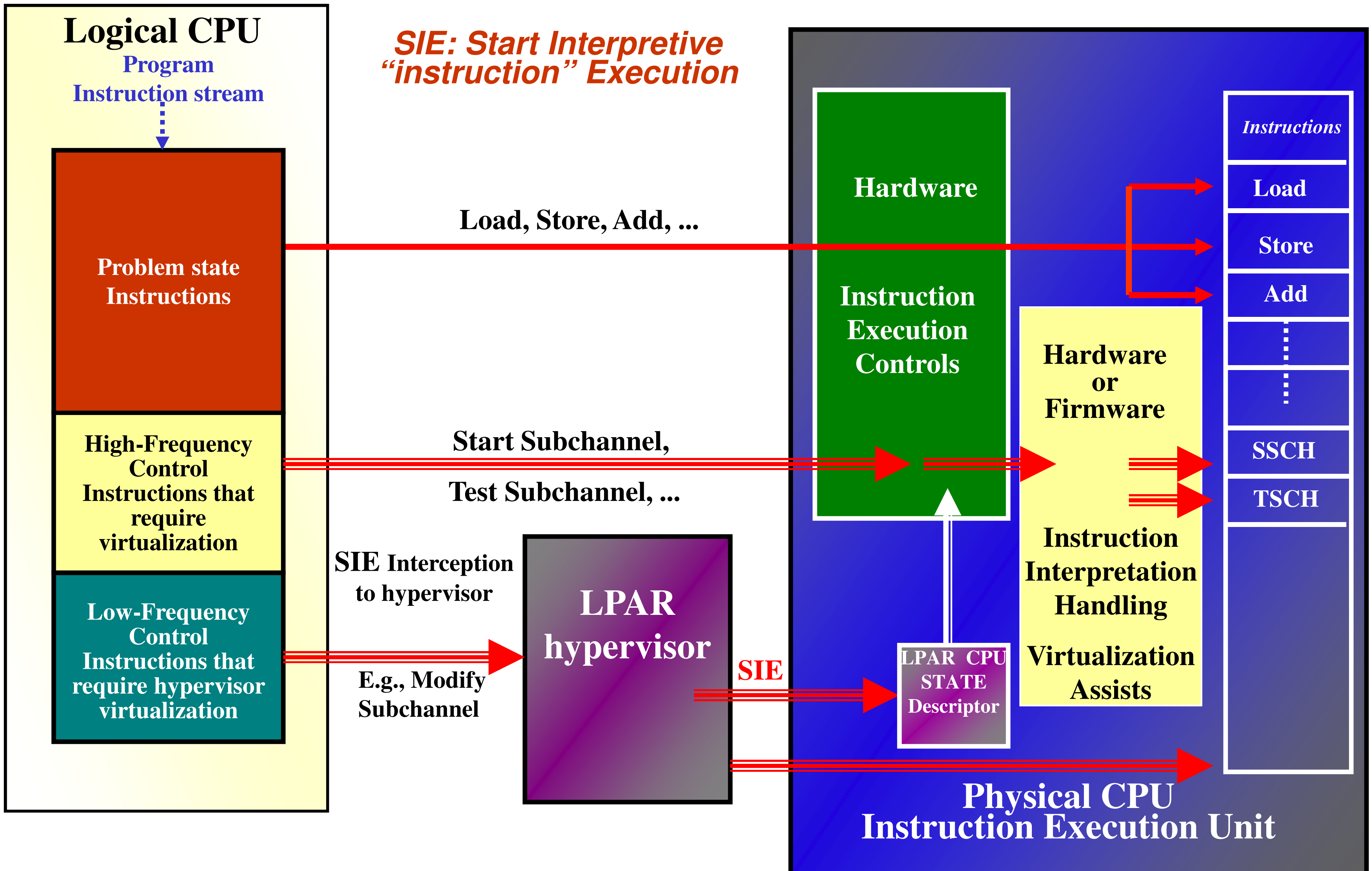
Virtualization can fill many roles and provide many benefits
In the final analysis, its potential benefits take three forms:

- **Help reduce hardware costs**
  - Help increase physical resource utilization
  - Small footprints
- **Can improve flexibility and responsiveness**
  - Virtual resources can be adjusted dynamically to meet new or changing needs and to optimize service level achievement
  - Virtualization is a key enabler of on demand operating environments such as cloud
- **Can reduce management costs**
  - Fewer physical servers to manage
  - Many common management tasks become much easier

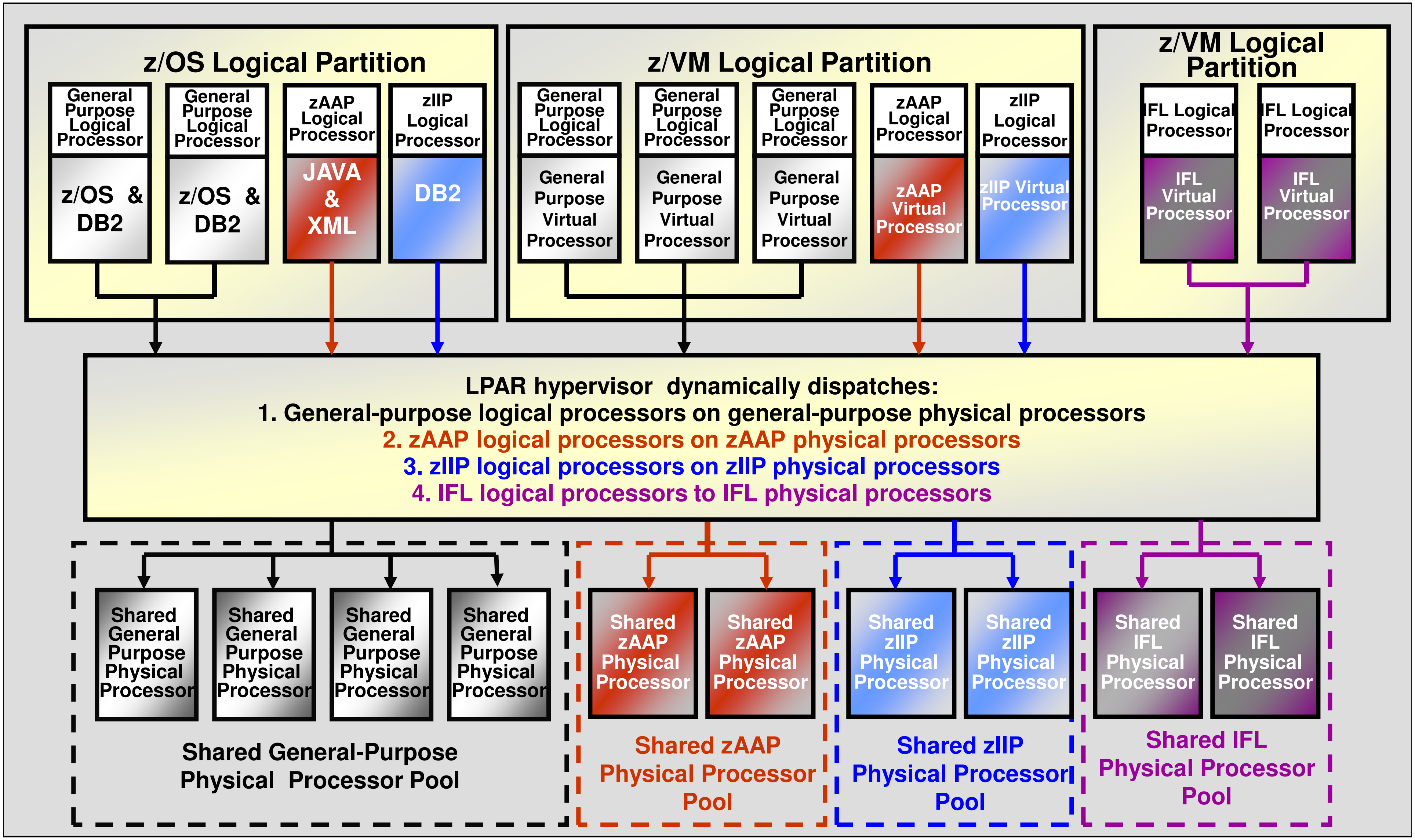


- **Multi-dimensional virtualization technology**
  - z Systems provides logical (LPAR) and software (z/VM) partitioning
  - PR/SM enables highly scalable virtual server hosting for LPAR *and* z/VM virtual machine environments
  - IRD coordinates allocation of CPU and I/O resources among z/OS and non-z/OS® LPARs*

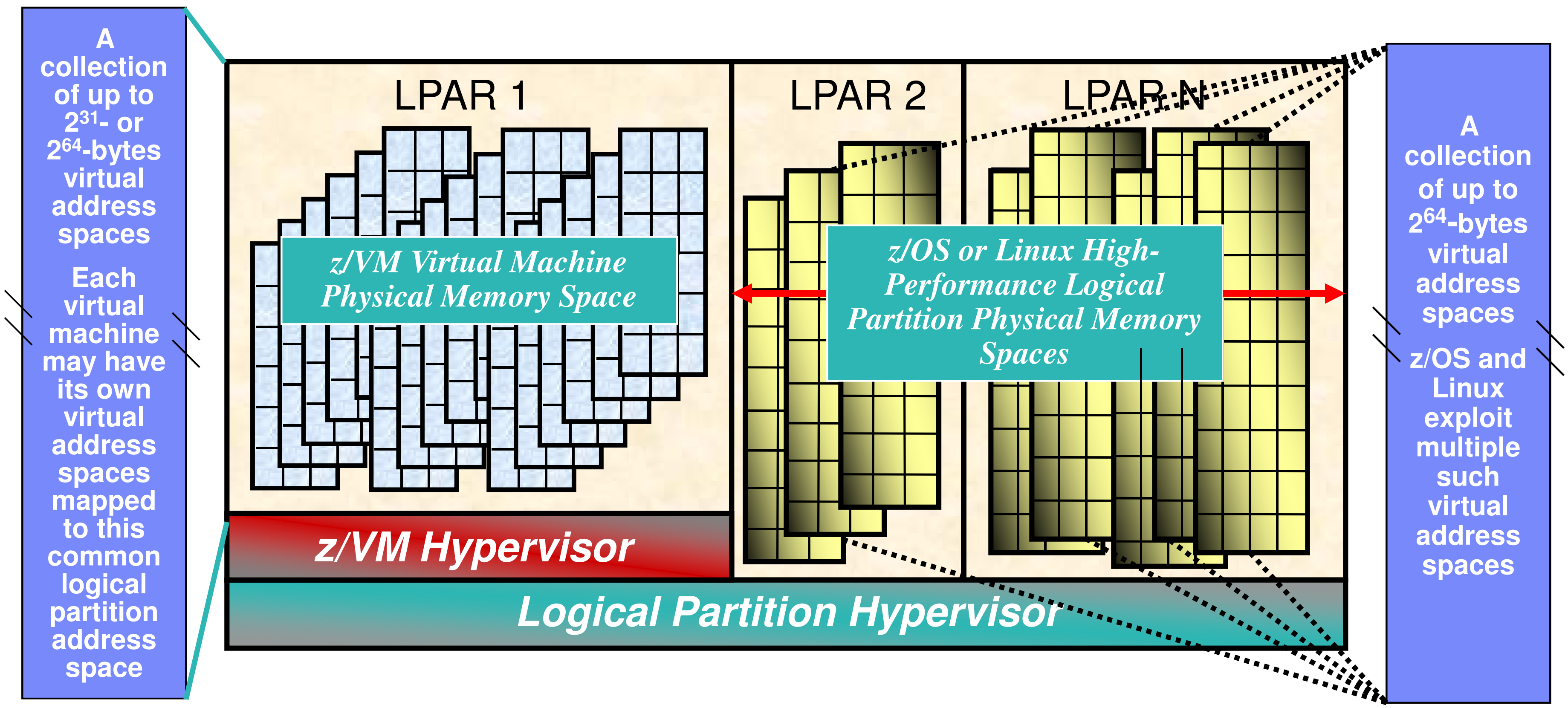
* Excluding non-shared resources like Integrated Facility for Linux processors

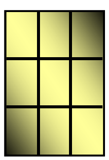


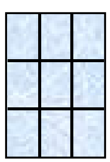
Logical CPU
Program
Instruction stream
Problem state Instructions
High-Frequency Control Instructions that require virtualization
Low-Frequency Control Instructions that require hypervisor virtualization
SIE: Start Interpretive "instruction" Execution
Load, Store, Add, ...
Start Subchannel,
Test Subchannel, ...
SIE Interception to hypervisor
E.g., Modify Subchannel
LPAR hypervisor
SIE
Hardware
Instruction Execution Controls
Hardware or Firmware
Instruction Interpretation Handling
Virtualization Assists
LPAR CPU STATE Descriptor
Instructions
Load
Store
Add
SSCH
TSCH
Physical CPU
Instruction Execution Unit



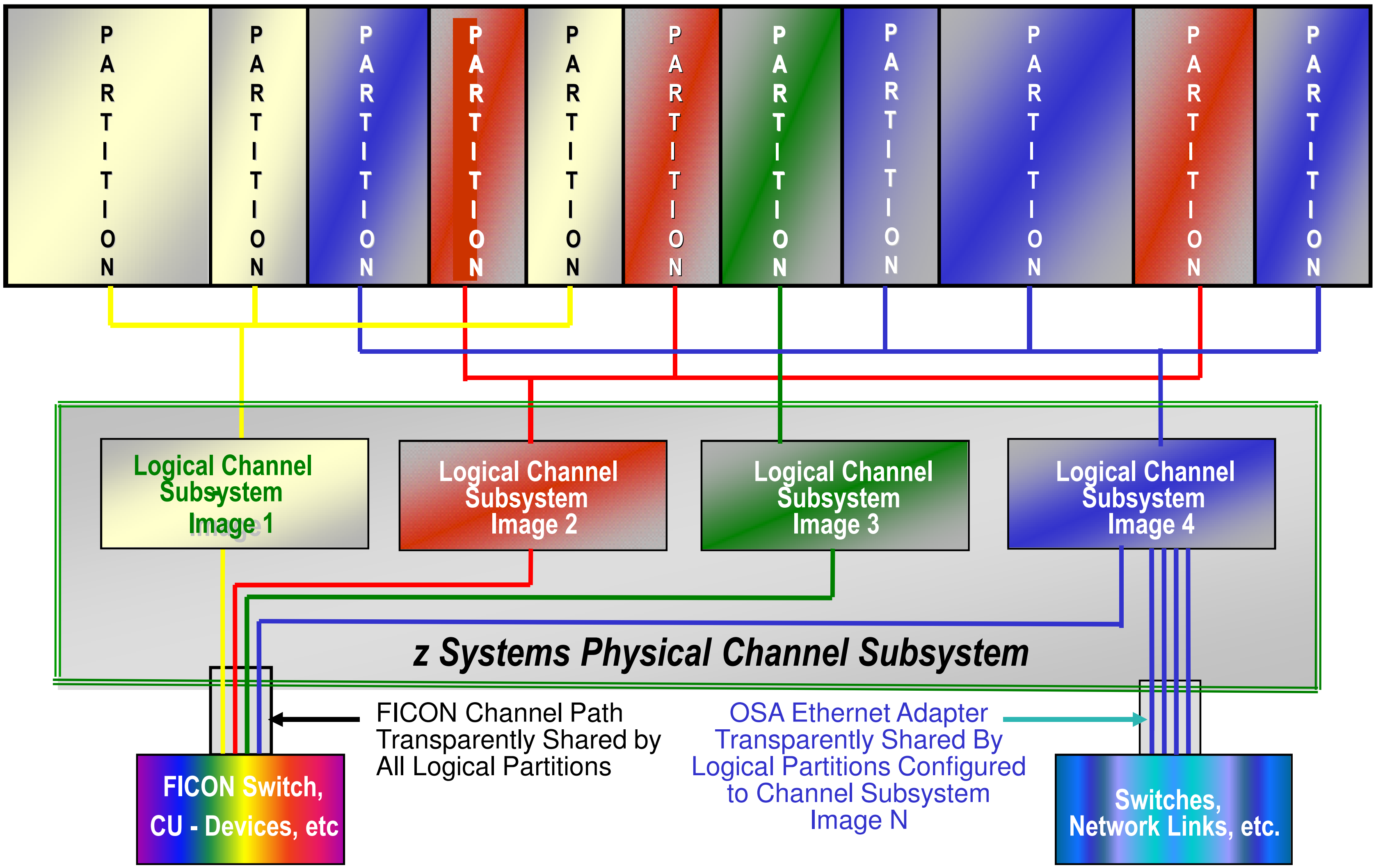
z/OS Logical Partition
General Purpose Logical Processor
z/OS & DB2
General Purpose Logical Processor
z/OS & DB2
zAAP Logical Processor
JAVA & XML
zIIP Logical Processor
DB2
z/VM Logical Partition
General Purpose Logical Processor
General Purpose Virtual Processor
General Purpose Logical Processor
General Purpose Virtual Processor
General Purpose Logical Processor
General Purpose Virtual Processor
zAAP Logical Processor
zAAP Virtual Processor
zIIP Logical Processor
zIIP Virtual Processor
z/VM Logical Partition
IFL Logical Processor
IFL Virtual Processor
IFL Logical Processor
IFL Virtual Processor
LPAR hypervisor dynamically dispatches:
1. General-purpose logical processors on general-purpose physical processors
2. zAAP logical processors on zAAP physical processors
3. zIIP logical processors on zIIP physical processors
4. IFL logical processors to IFL physical processors
Shared General Purpose Physical Processor
Shared General Purpose Physical Processor
Shared General Purpose Physical Processor
Shared General Purpose Physical Processor
Shared General-Purpose Physical Processor Pool
Shared zAAP Physical Processor
Shared zAAP Physical Processor
Shared zAAP Physical Processor Pool
Shared zIIP Physical Processor
Shared zIIP Physical Processor
Shared zIIP Physical Processor Pool
Shared IFL Physical Processor
Shared IFL Physical Processor
Shared IFL Physical Processor Pool

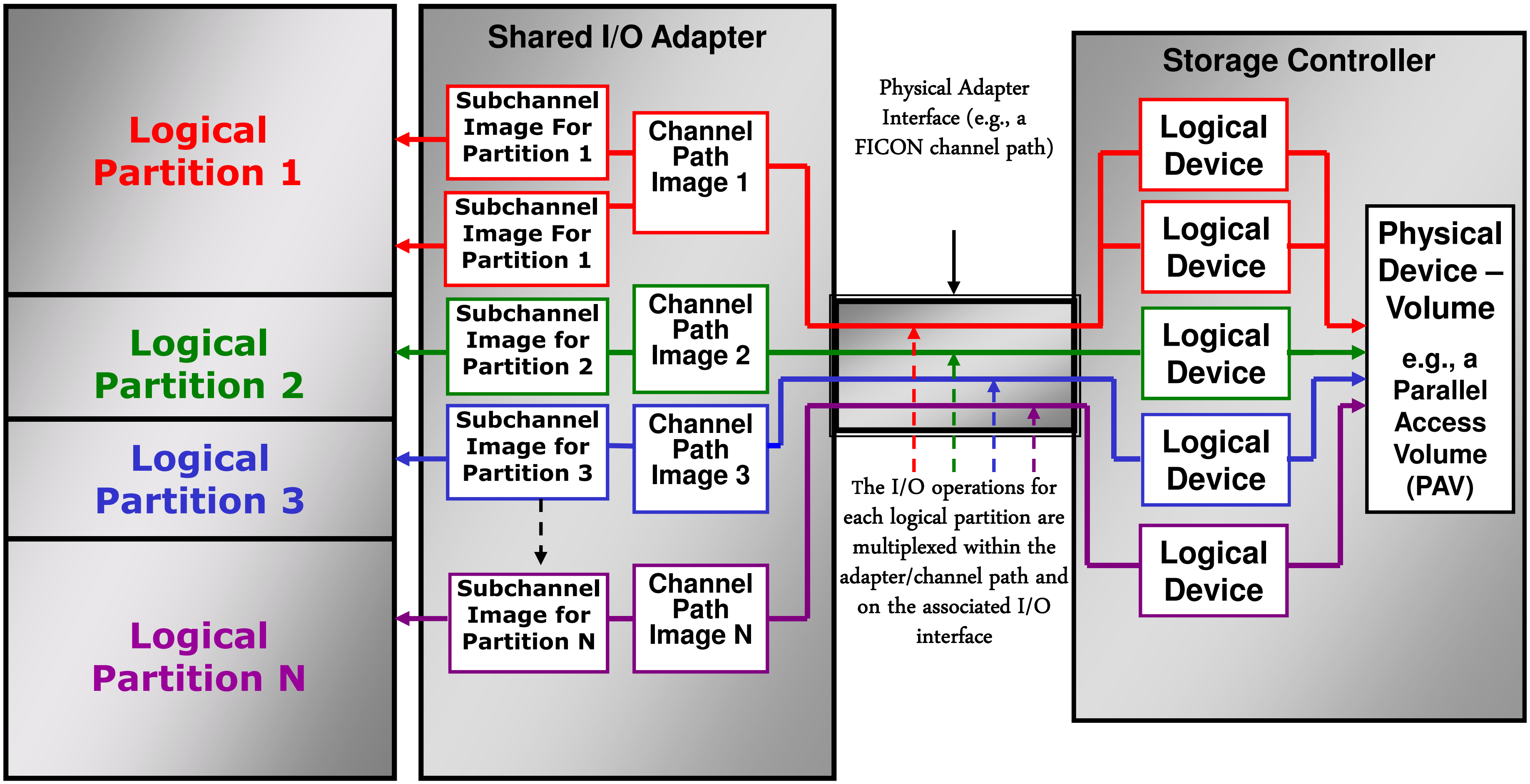



= the real partition memory pages associated with a virtual address space; that is, the sets of dynamically-allocated physical memory pages necessary to run a z/OS task or a Linux process

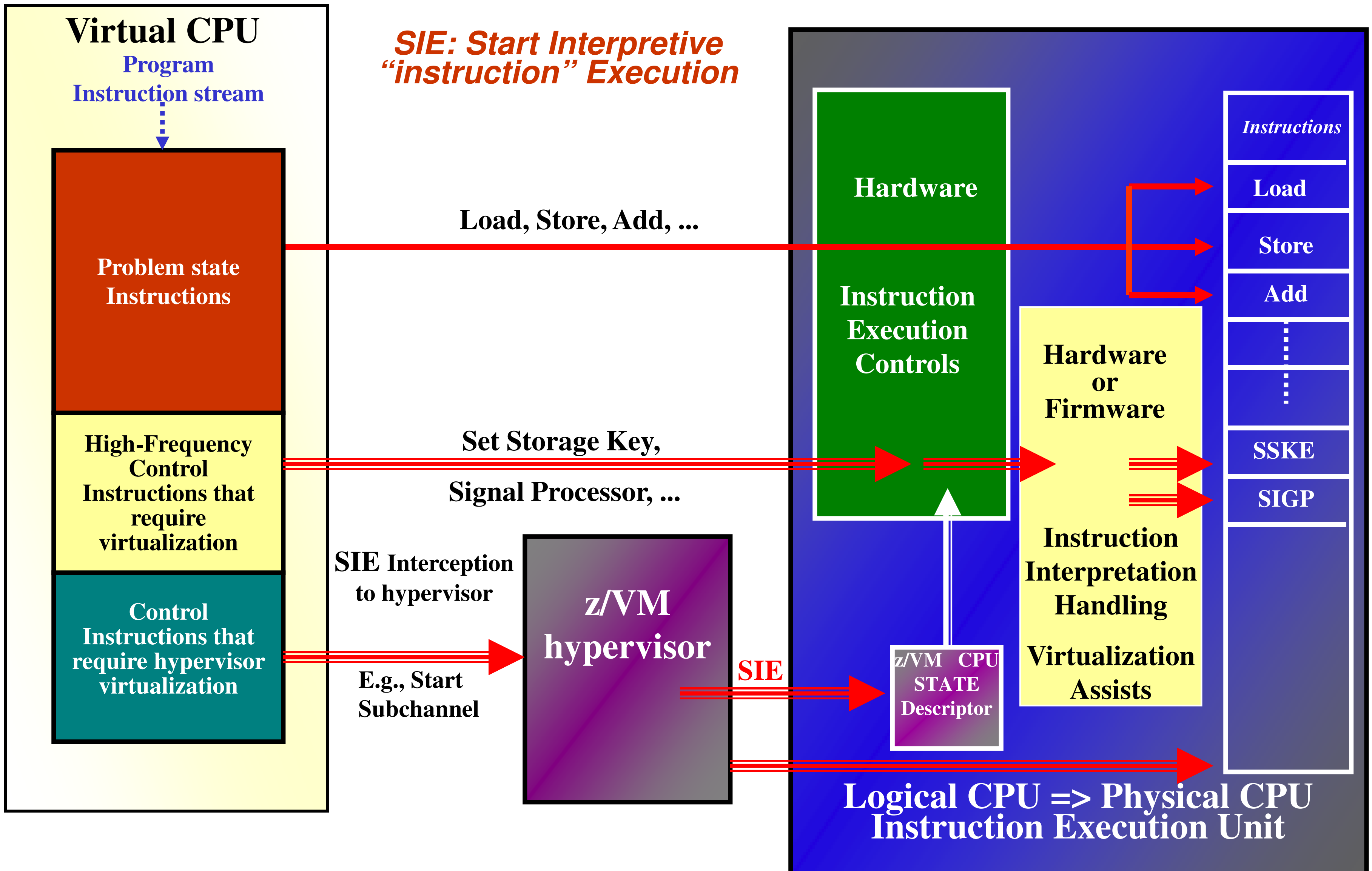

= the real partition memory pages associated with a virtual machine; that is, the sets of dynamically-allocated physical memory pages necessary to run a guest operating system in a virtual machine

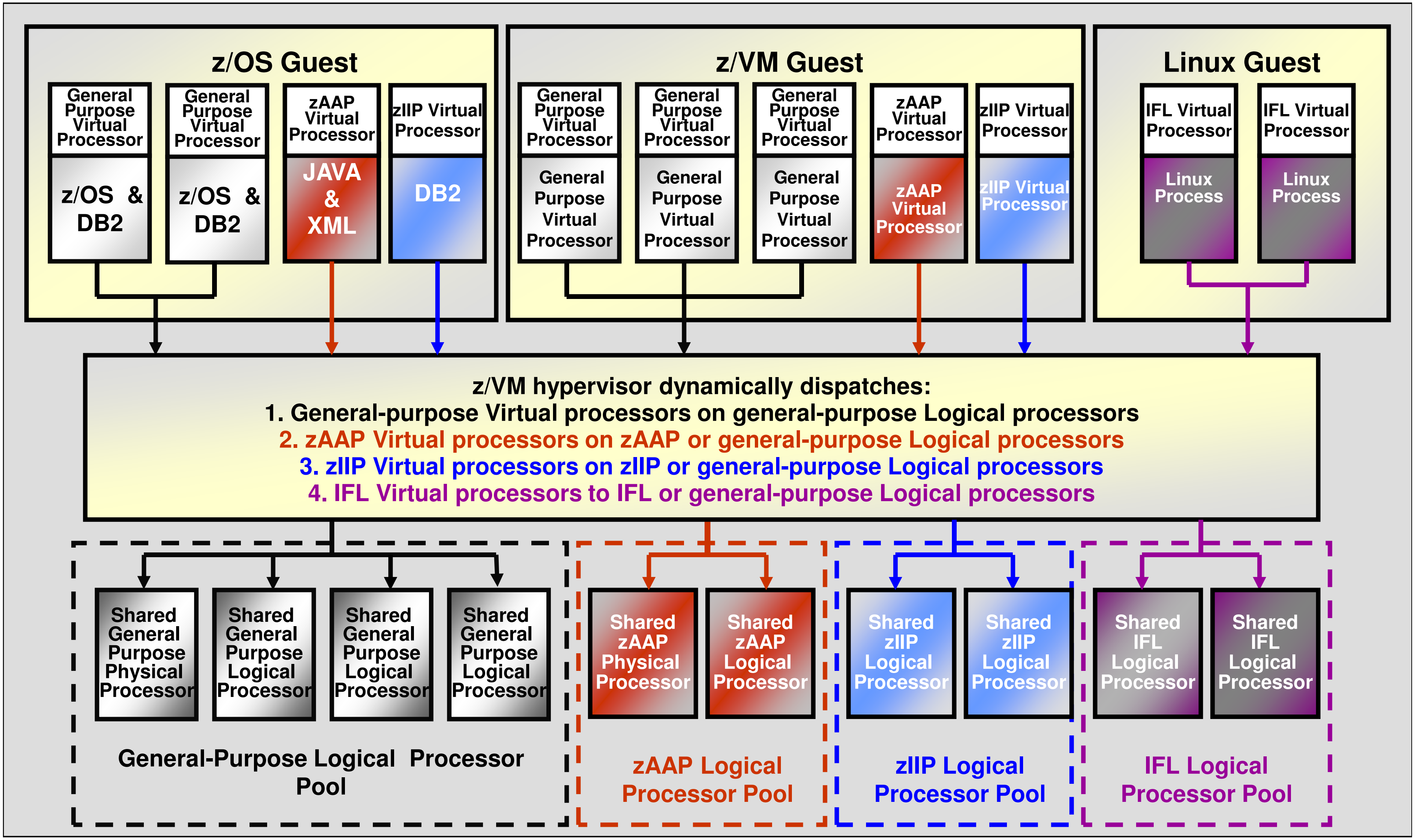
LPAR Multiple Logical Channel Subsystems

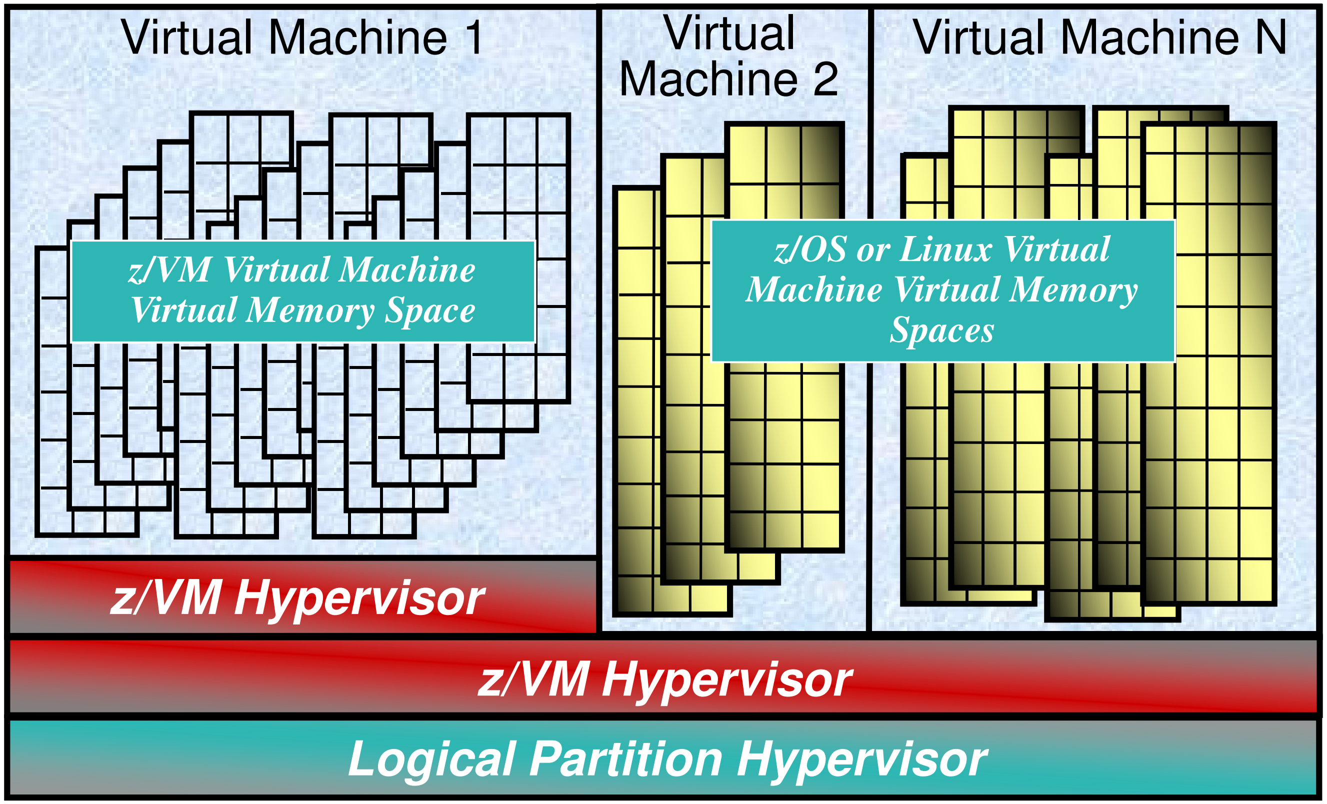


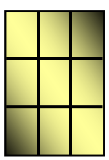


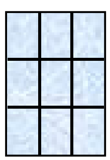
- The I/O infrastructure (adapters/channels, their transmission links, and attached I/O resources are shared by LPARs at native speeds (without hypervisor involvement)
  - I/O requests, their associated data transfers , and I/O interruptions flow between each OS instance and the shared I/O components, just as if the I/O components were physically dedicated to a single OS instance

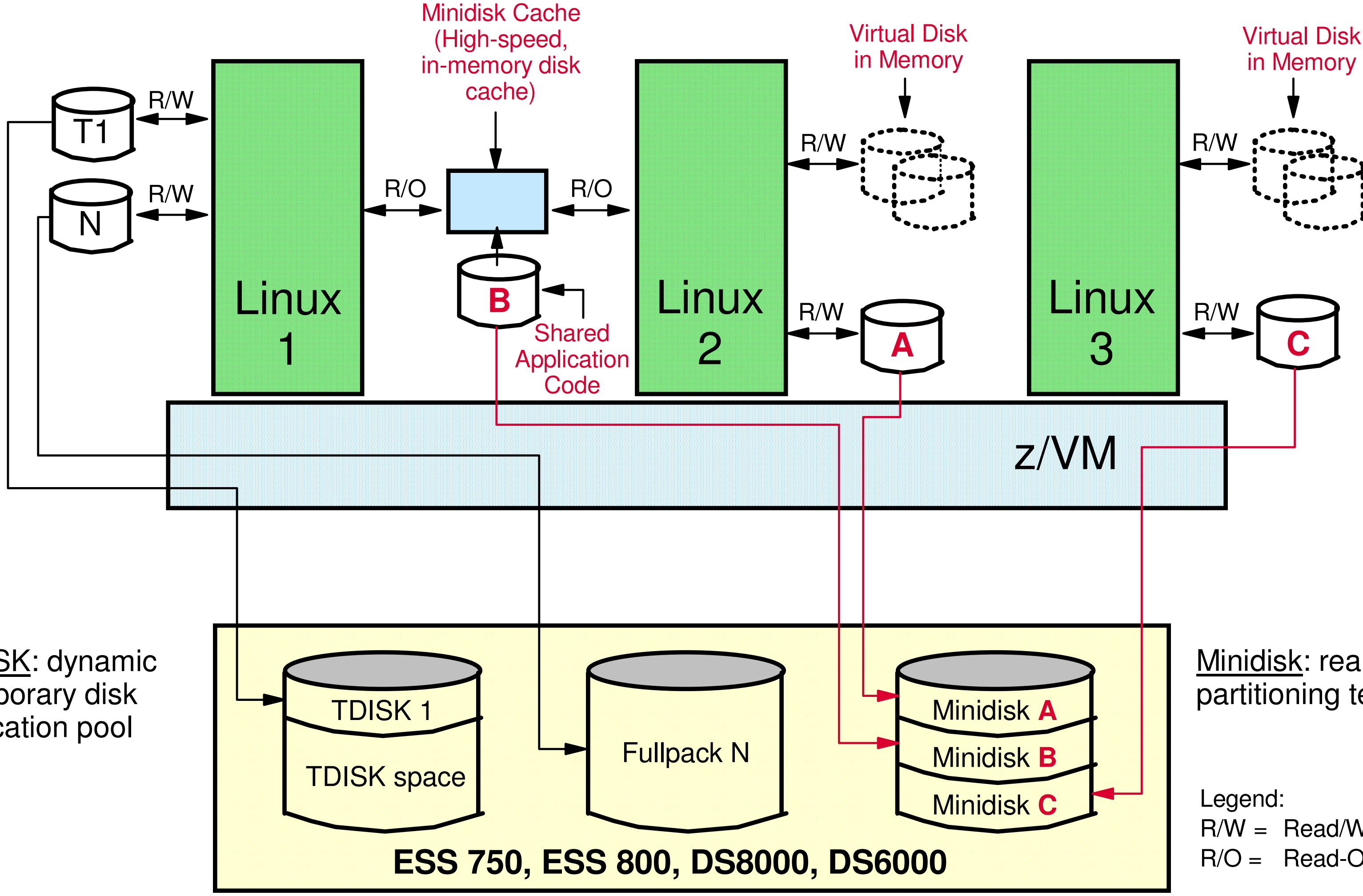

Virtual CPU
Program
Instruction stream
Problem state Instructions
High-Frequency Control Instructions that require virtualization
Control Instructions that require hypervisor virtualization
SIE: Start Interpretive "instruction" Execution
Load, Store, Add, ...
Set Storage Key,
Signal Processor, ...
SIE Interception to hypervisor
E.g., Start Subchannel
z/VM hypervisor
SIE
Hardware
Instruction Execution Controls
z/VM CPU STATE Descriptor
Hardware or Firmware
Instruction Interpretation Handling
Virtualization Assists
Instructions
Load
Store
Add
SSKE
SIGP
Logical CPU => Physical CPU Instruction Execution Unit


z/OS Guest
General Purpose Virtual Processor
z/OS & DB2
General Purpose Virtual Processor
z/OS & DB2
zAAP Virtual Processor
JAVA & XML
zIIP Virtual Processor
DB2
z/VM Guest
General Purpose Virtual Processor
General Purpose Virtual Processor
General Purpose Virtual Processor
General Purpose Virtual Processor
General Purpose Virtual Processor
General Purpose Virtual Processor
zAAP Virtual Processor
zAAP Virtual Processor
zIIP Virtual Processor
zIIP Virtual Processor
Linux Guest
IFL Virtual Processor
Linux Process
IFL Virtual Processor
Linux Process
z/VM hypervisor dynamically dispatches:
1. General-purpose Virtual processors on general-purpose Logical processors
2. zAAP Virtual processors on zAAP or general-purpose Logical processors
3. zIIP Virtual processors on zIIP or general-purpose Logical processors
4. IFL Virtual processors to IFL or general-purpose Logical processors
Shared General Purpose Physical Processor
Shared General Purpose Logical Processor
Shared General Purpose Logical Processor
Shared General Purpose Logical Processor
General-Purpose Logical Processor Pool
Shared zAAP Physical Processor
Shared zAAP Logical Processor
zAAP Logical Processor Pool
Shared zIIP Logical Processor
Shared zIIP Logical Processor
zIIP Logical Processor Pool
Shared IFL Logical Processor
Shared IFL Logical Processor
IFL Logical Processor Pool




= the guest real memory pages associated with a virtual address space; that is, the sets of dynamically-allocated host virtual memory pages necessary to run a z/OS task or a Linux process

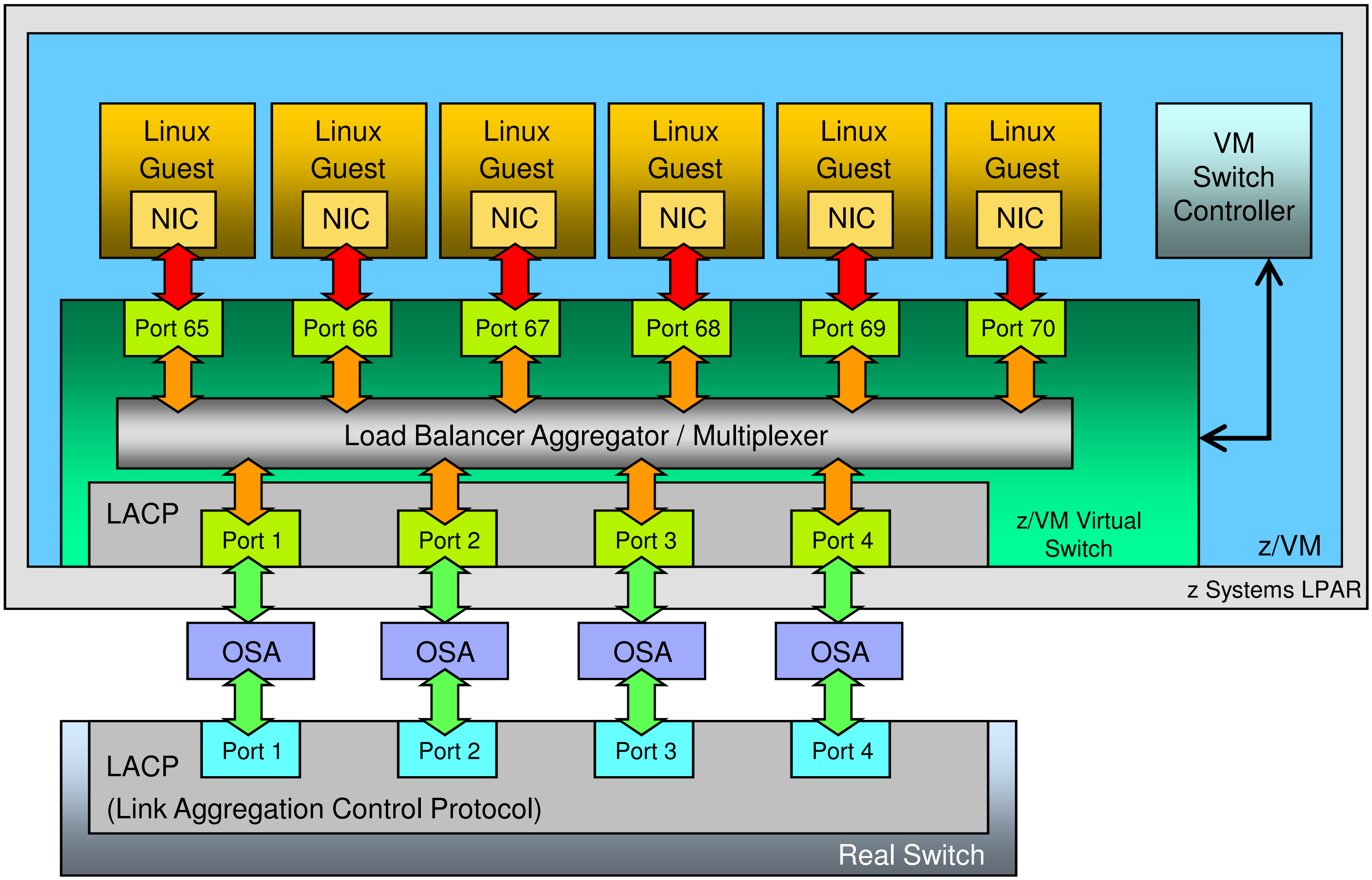

= the guest real memory pages associated with a virtual machine; that is, the sets of dynamically-allocated host virtual memory pages necessary to run a guest operating system in a virtual machine

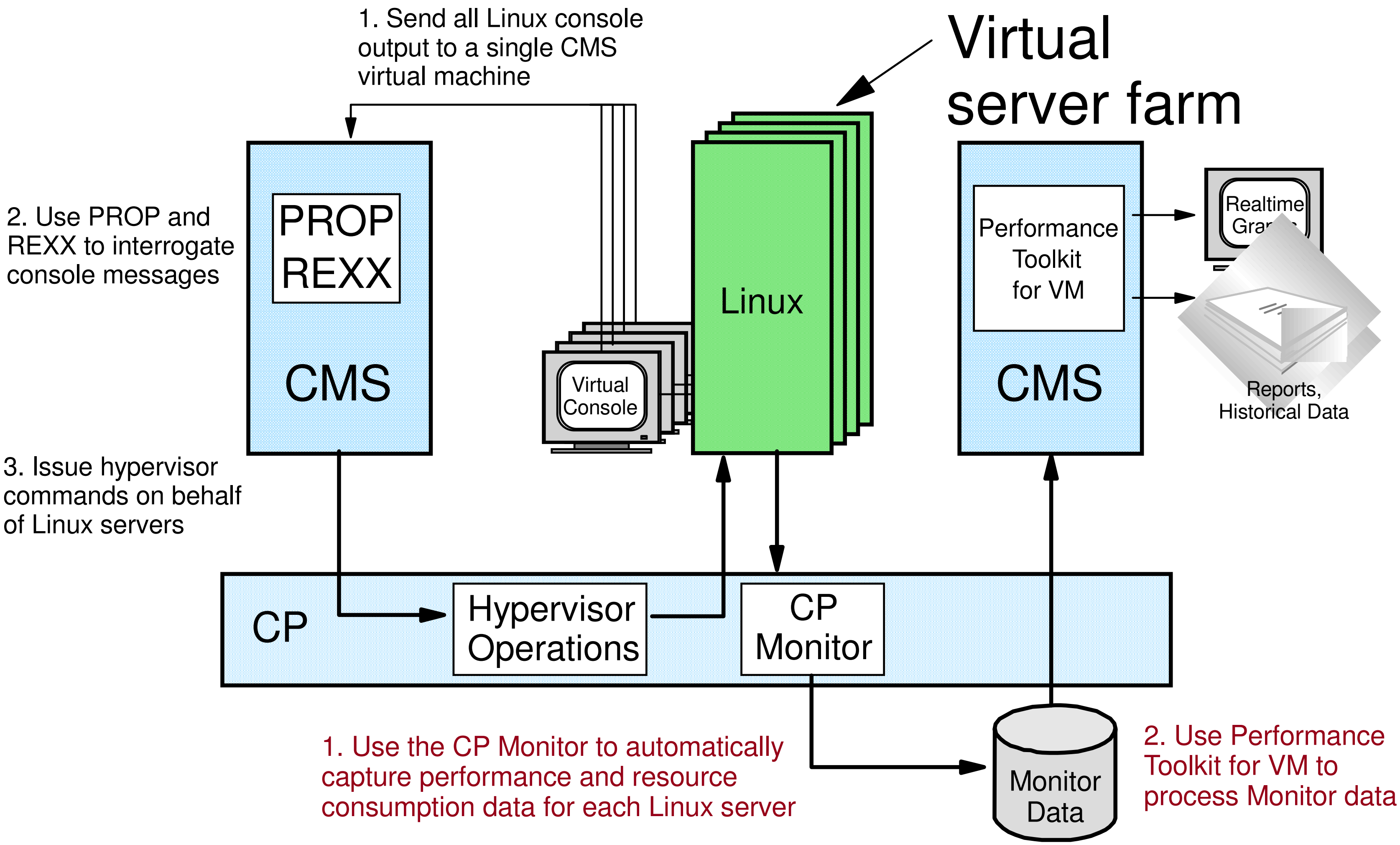


TDISK: dynamic temporary disk allocation pool

Minidisk: real disk partitioning technology

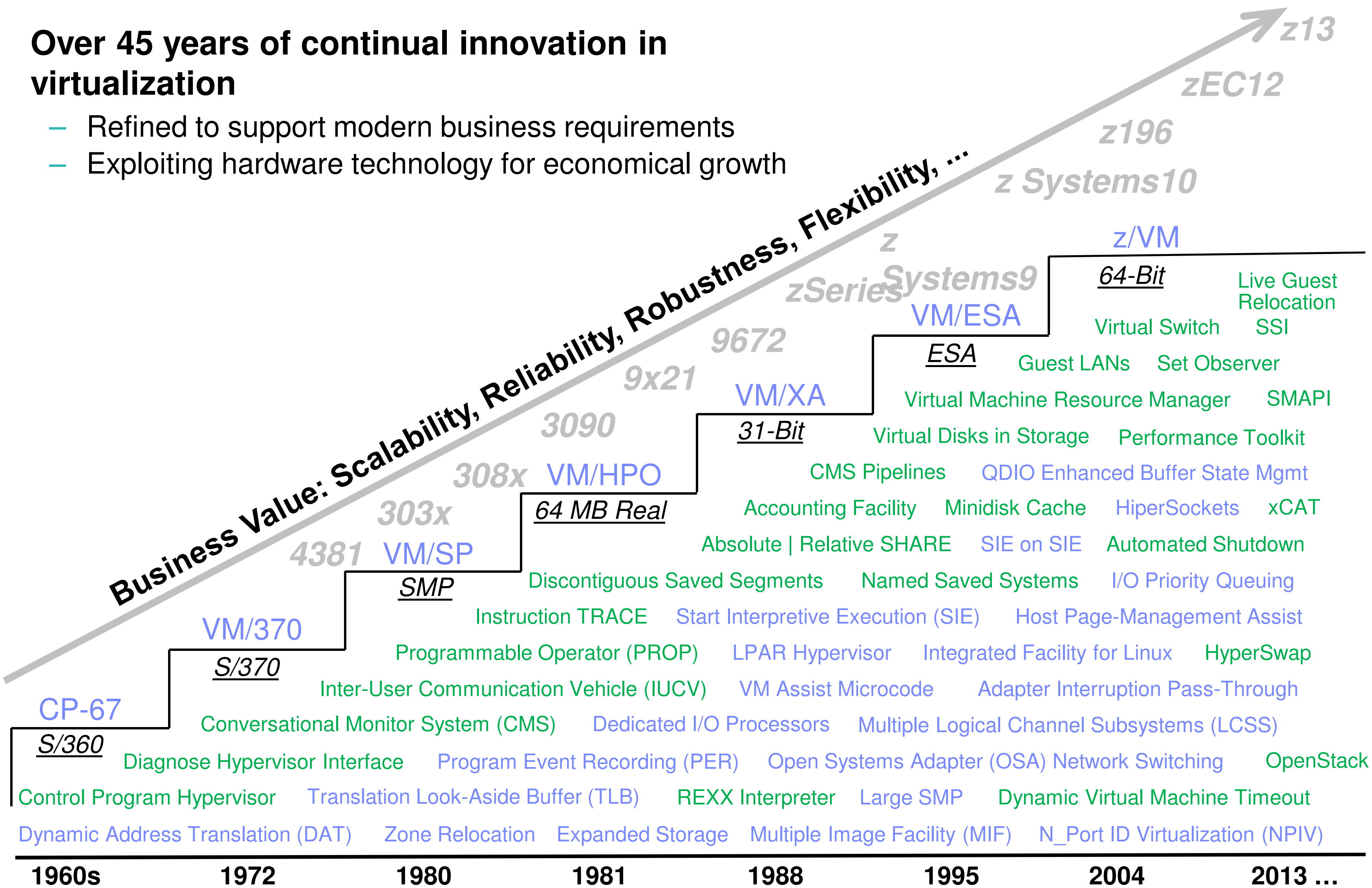
Legend:
R/W = Read/Write
R/O = Read-Only


Linux Guest
NIC
Linux Guest
NIC
Linux Guest
NIC
Linux Guest
NIC
Linux Guest
NIC
Linux Guest
NIC
VM Switch Controller
Port 65
Port 66
Port 67
Port 68
Port 69
Port 70
Load Balancer Aggregator / Multiplexer
LACP
Port 1
Port 2
Port 3
Port 4
z/VM Virtual Switch
z/VM
z Systems LPAR
OSA
OSA
OSA
OSA
Port 1
Port 2
Port 3
Port 4
LACP
(Link Aggregation Control Protocol)
Real Switch


1. Send all Linux console output to a single CMS virtual machine
Virtual server farm
2. Use PROP and REXX to interrogate console messages
PROP
REXX
CMS
Virtual Console
Linux
Performance Toolkit for VM
CMS
Realtime
Reports, Historical Data
3. Issue hypervisor commands on behalf of Linux servers
CP
Hypervisor Operations
CP Monitor
Monitor Data
1. Use the CP Monitor to automatically capture performance and resource consumption data for each Linux server
2. Use Performance Toolkit for VM to process Monitor data

Over 45 years of continual innovation in virtualization

- Refined to support modern business requirements
- Exploiting hardware technology for economical growth



IBM z Systems – a comprehensive and sophisticated suite of virtualization function