Introduction to Virtualization

SHARE Anaheim, Session 14561

Romney White, IBM
romneyw@us.ibm.com
System z Architecture and Technology
Introduction to Virtualization
- Concept
- Server Virtualization Approaches
- Hypervisor Implementation Methods
- Why Virtualization Matters

Virtualization on System z
- Logical Partitions
- Virtual Machines
Virtualization Concept

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
- System z LPAR and z/VM®
- POWER™ Hypervisor
- VMware ESX Server
- Xen Hypervisor

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

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
Hypervisor Implementation Methods

**Trap and Emulate**
- **Virt Mach**
  - L
  - A
  - ST
  - PrivOp
  - L
  - ...
- **Trap**
  - Hypervisor PrivOp emulation code
- **Examples**
  - CP-67, VM/370
- **Benefits**
  - Runs unmodified OS
- **Issues**
  - Substantial overhead

**Translate, Trap, and Emulate**
- **Virt Mach**
  - L
  - A
  - ST
  - TrapOp
  - L
  - ...
- **Trap**
  - Hypervisor PrivOp emulation code
- **Examples**
  - VMware, Microsoft VS
- **Benefits**
  - Runs unmodified, translated OS
- **Issues**
  - Substantial overhead

**Hypervisor Calls (“Paravirtualization”)**
- **Virt Mach**
  - L
  - A
  - ST
  - Hcall
  - L
  - ...
- **Call**
  - Hypervisor service
- **Examples**
  - POWER Hypervisor, Xen
- **Benefits**
  - High efficiency
- **Issues**
  - OS must be modified to issue Hcalls

**Direct Hardware Virtualization**
- **Virt Mach**
  - L
  - A
  - ST
  - PrivOp
  - L
  - ...
- **Exit**
  - Hypervisor service
- **Examples**
  - System z LPAR, z/VM, Xen
- **Benefits**
  - High efficiency, runs unmodified OS
- **Issues**
  - Requires underlying hardware support
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

**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
Multi-dimensional virtualization technology

- System z 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**

**Instruction Execution Controls**

**Hardware or Firmware**

**Instruction Interpretation Handling**

**Virtualization Assists**

**Physical CPU Instruction Execution Unit**

**Instructions**
- Load
- Store
- Add
- SSCH
- TSCH

---

System z Virtualization Technology © 2013 IBM Corporation
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
LPAR Memory Partitioning

= 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

Logical Channel Subsystem Image 1

Logical Channel Subsystem Image 2

Logical Channel Subsystem Image 3

Logical Channel Subsystem Image 4

System z Physical Channel Subsystem

FICON Channel Path Transparently Shared by All Logical Partitions

OSA Ethernet Adapter Transparently Shared By Logical Partitions Configured to Channel Subsystem Image N

FICON Switch, CU - Devices, etc

Switches, Network Links, etc.
The I/O operations for each logical partition are multiplexed within the adapter/channel path and on the associated I/O interface.

- 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*

Logical CPU => Physical CPU Instruction Execution Unit

__Hardware or Firmware__

Instruction Interpretation Handling

Virtualization Assists

z/VM CPU STATE Descriptor

Instructions
Load
Store
Add
SSKE
SIGP
z/VM Virtual Processor Dispatching

**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
= 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
z/VM Disk Virtualization

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

ESS 750, ESS 800, DS8000, DS6000

Minidisk: real disk partitioning technology

TDISK: dynamic temporary disk allocation pool

Shared Application Code
z/VM LAN and Switch Virtualization

Load Balancer Aggregator / Multiplexer

LACP (Link Aggregation Control Protocol)

OSA

Real Switch

z/VM Virtual Switch

VM Switch Controller

System z LPAR
1. Send all Linux console output to a single CMS virtual machine

2. Use PROP and REXX to interrogate console messages

3. Issue hypervisor commands on behalf of Linux servers

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 continuous innovation in virtualization

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

Business Value: Scalability, Reliability, Robustness, Flexibility, ...

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