

Enterprise Computing

Einführung in das Betriebssystem z/OS

Prof. Dr. Martin Bogdan
Dr. rer. nat. Paul Herrmann
Prof. Dr.-Ing. Wilhelm G. Spruth

WS 2009/2010

Teil 10

Partitionierung und Virtualisierung

HP Superdome Cell Board



4 Itanium 2 CPU Chips
mit Kühlern

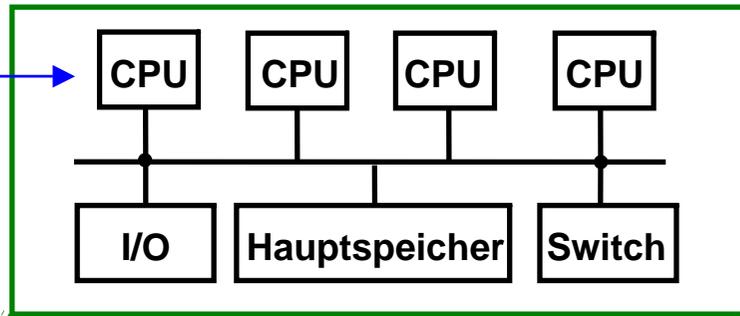
32 – 128 Gbyte
Hauptspeicher

I/O Bus
Anschlüsse
4 Steckkarten

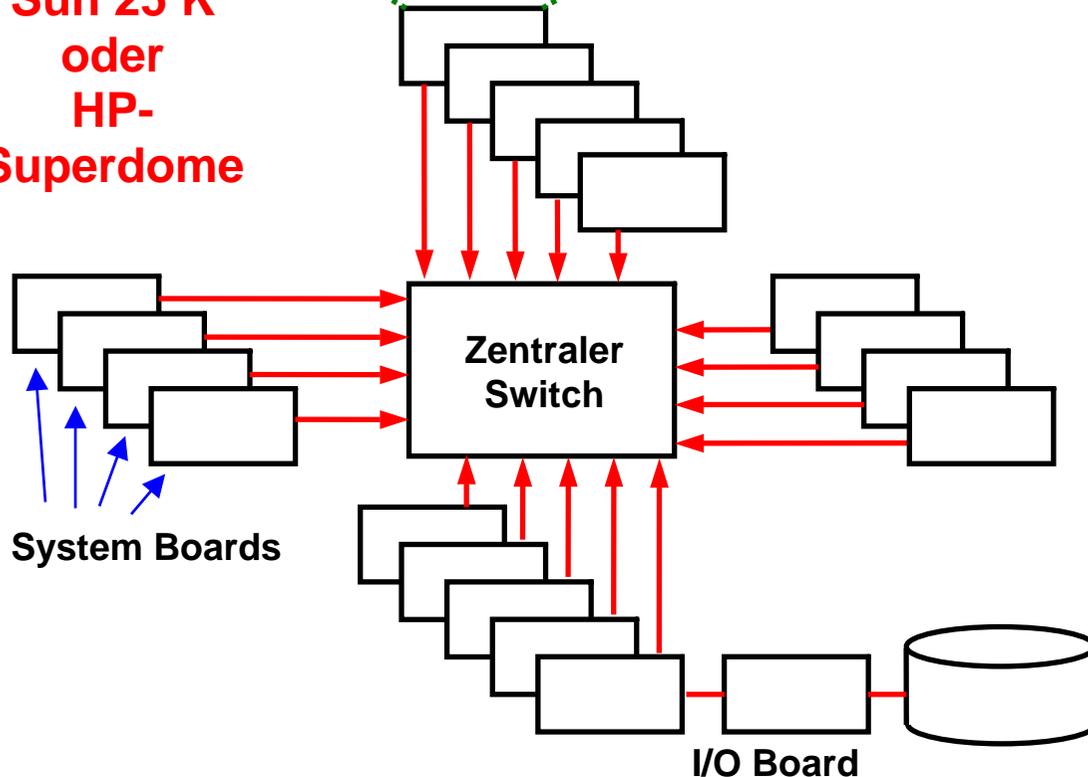
Die Großrechner der Firmen Sun und Hewlett Packard sind sehr ähnlich aufgebaut. Dargestellt ist ein System Board (auch als Cell Board bezeichnet). Es besteht aus einem Printed Circuit Board mit 4 CPU Chips (Dual Core), Hauptspeicher Dimms, sowie 4 Steckkarten, von denen PCI Bus Kabel zu I/O Cages führen.

Ein Sun oder HP Großrechner enthält bis zu 16 oder 18 System Boards.

4 CPU
chips
2/4 Core



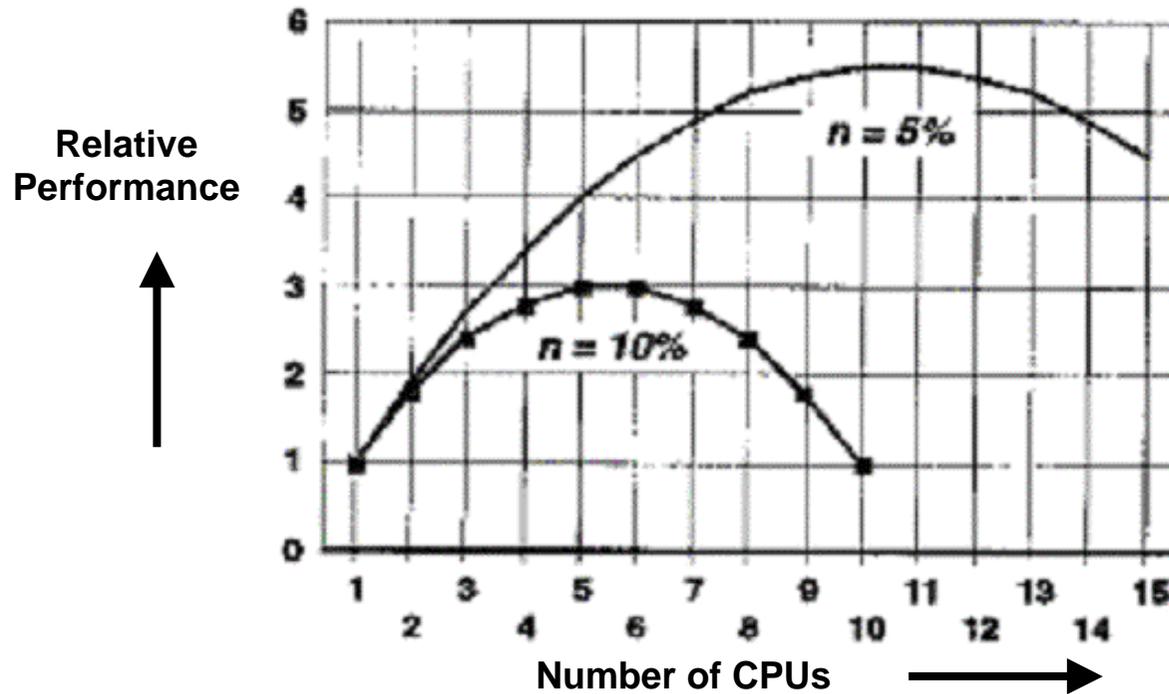
Sun 25 K
oder
HP-
Superdome



Ein Sun 25K oder HP Superdome Großrechner enthält 16 oder 18 System Boards, jedes mit 4 CPU Chips und bis zu 128 GByte Hauptspeicher, einem Anschluss an einen zentralen Switch sowie I/O Controller auf jedem System Board.

Die I/O Controller sind mit PCI Bus Kabeln mit einer Reihe von I/O Cages verbunden, in denen sich Steckkarten für den Anschluss von I/O Geräten, besonders Plattenspeichern, befinden.

Die CPUs eines jeden System Boards können nicht nur auf den eigenen Hauptspeicher, sondern auch auf den Hauptspeicher eines jeden anderen system Boards zugreifen. Hiermit wird eine „Non-Uniform Memory Architecture“ (NUMA) verwirklicht.



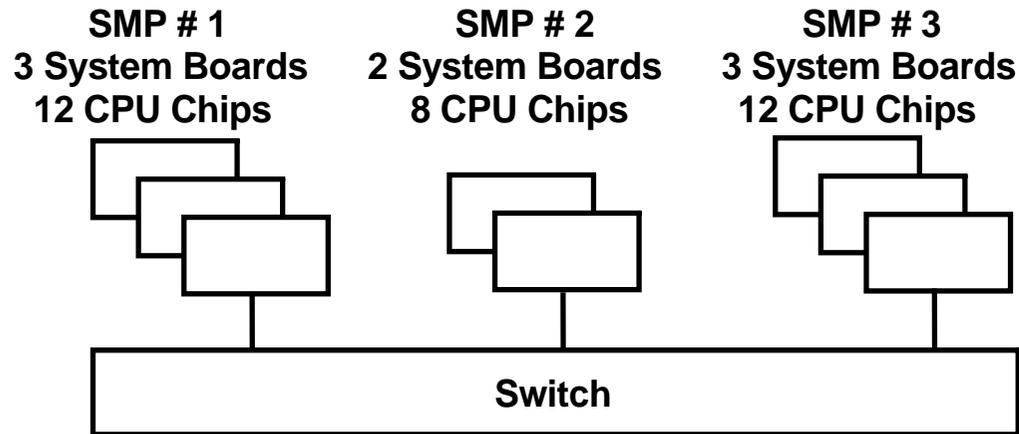
Partitioning a large System

Large Systems from Companies like Sun and Hewlett Packard consist of 64 or more CPUs, using dual or quad core chips.. Symmetric Multiprocessors (SMPs) do not scale to such large numbers. Therefore a system with 32 or 64 CPUs is usually partitioned into several SMPs which cooperate as a cluster.

Harte Partitionierung

Virtuelle Partitionierung

Logische Partitionen (LPAR)



Aufteilung eines Sun 25K oder HP Superdome Servers mit 8 System Boards in mehrere parallel laufende SMPs

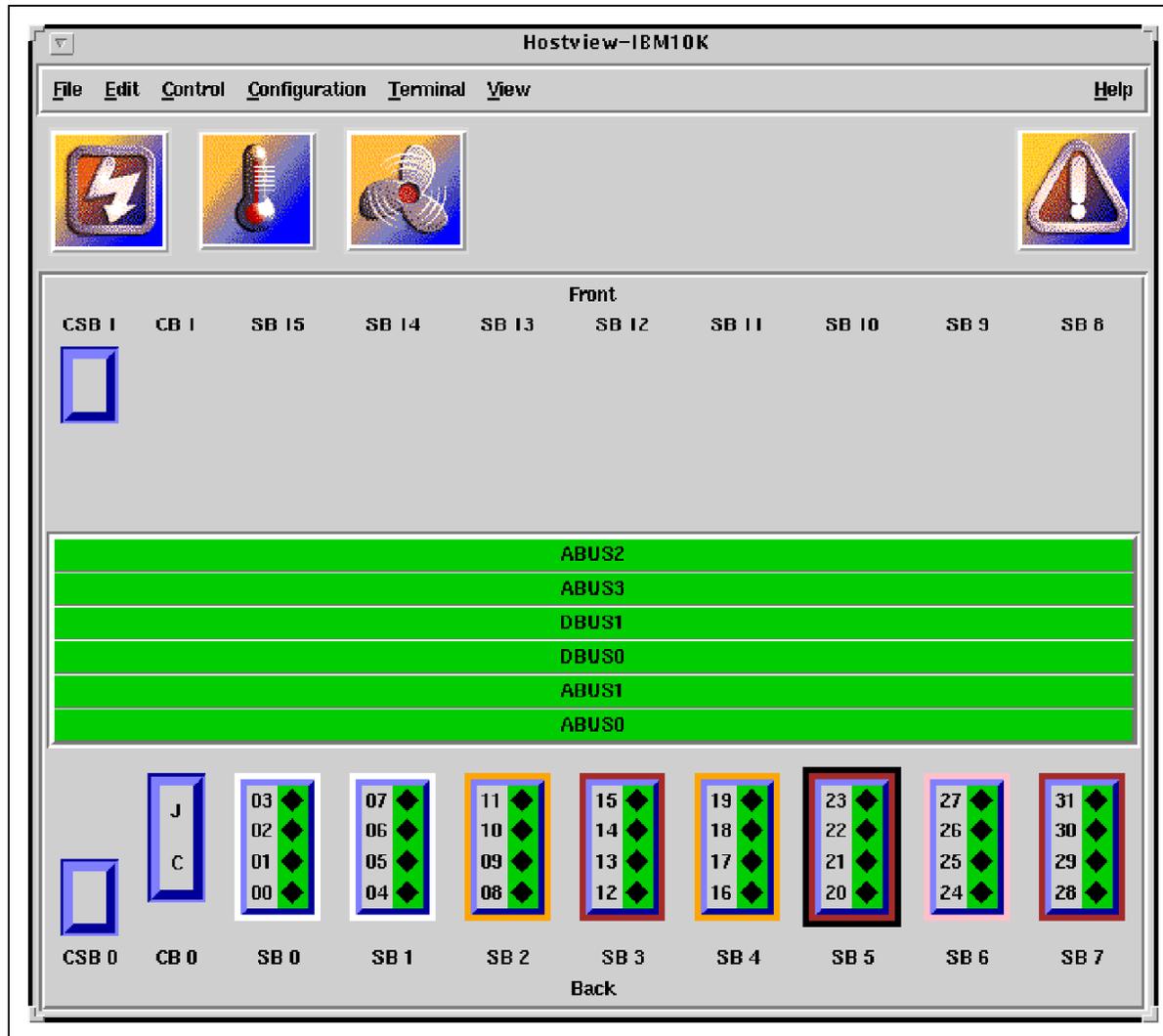
Eine Harte Partitionierung sei an Hand des hier dargestellten Beispiels erläutert. Vorhanden sind 8 System Boards mit je 4 CPU Chips pro System Board.

Die 8 System Boards werden in 3 Partitionen und 3 SMPS mit 3, 2 und 3 System Boards aufgeteilt. Jeder SMP hat ein eigenes Betriebssystem. Die Aufteilung erfolgt statisch mit Hilfe elektronischer Schalter.

Der System Administrator kann während des laufenden Betriebes die Zuordnung der System Boards zu den einzelnen SMPs ändern

Für Transaktionsanwendungen unter Unix sind kaum mehr als 12 – 16 CPUs pro SMP realistisch.

CPUs



Sun Fire Administrator Konsole

Dargestellt sind 8 System Boards SB0 .. SB7. Sie werden in 3 SMPs partitioniert:

SMP1 besteht aus: SB0, SB1, SB6

SMP2 besteht aus: SB2, SB4

SMP3 besteht aus: SB3, SB5, SB7

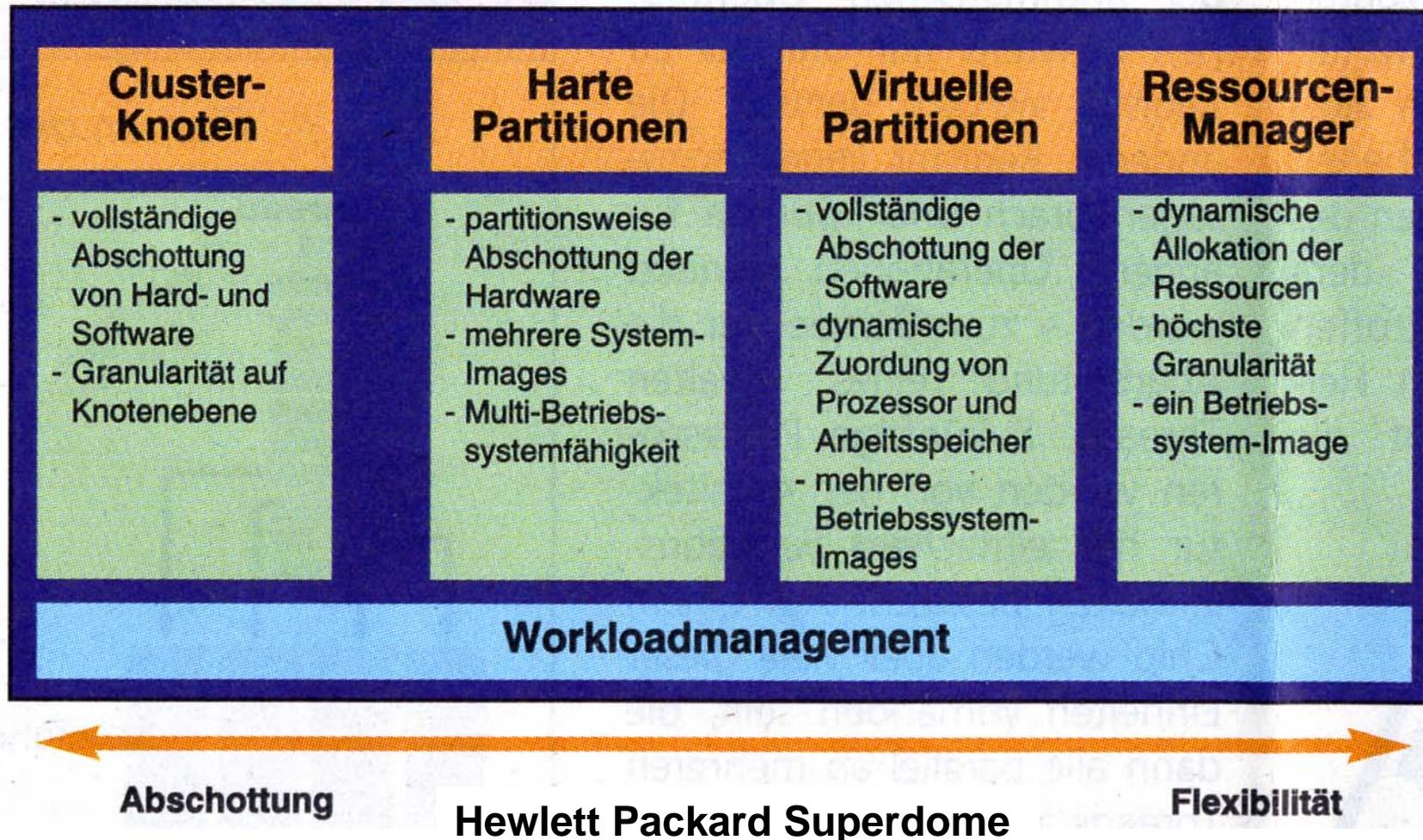
Angenommen, SMP2 ist überlastet, während SMP3 überschüssige Kapazität hat.

Der Administrator beschließt, SMP3 das System Board SB5 wegzunehmen und SMP2 zuzuordnen.

Auf der dargestellten Konsole klickt der Administrator auf das SB5 Icon und zieht es rüber auf den SB2 Icon.

Die Granularität der Zuordnung auf die einzelnen SMPs (Partitionen) war ursprünglich in Einheiten von ganzen System Boards, ist aber bei neueren Systemen verfeinert worden.

Partitionskonzepte im Vergleich



Die Harte Partitionierung (Hardware partitioning) ist nicht sehr flexibel. HP Superdome und Sun 25K sowie das Nachfolgersystem Sun M9000 bieten deshalb zusätzlich eine „Virtuelle Partitionierung“ an, auch als „Software Partitioning“ bezeichnet. Bei den Sun und HP Servern hatte die virtuelle Partitionierung ursprünglich erhebliche Performance Probleme. In den letzten Jahren wurden deutliche Verbesserungen erreicht.

Eine Harte Partitionierung ist bei Mainframes nicht erforderlich und nicht verfügbar.



Hard partitioned system benefits include:

Flexibility:

- Modular growth
- Mixing of CPU types (Intel Itanium "Montvale" and Itanium "Montecito" processors)
- Mixing of operating systems (HP-UX, Windows, Linux* and OpenVMS*)

Availability:

- Electrical (HW & SW) fault isolation
- Online maintenance
- Online upgrades
- Significantly improved server uptime

HP Superdome

Aug. 2009

HP-UX Virtual Partitions (vPars) Software

Partition Manager - Complex zoo - Microsoft Internet Explorer provided by Hewlett-Packard

File Edit View Favorites Tools Help

Address <https://zoo7.fc.hp.com:50000/parmgr/jsp/complex.jsp>

Links My Projects, Etc Other MSL sites HP-UX websites HP Misc PM Stuff CHART @hp Employee Portal Customize Links

Complex: >> zoo Last Complex Scan: Tuesday, February 11, 2003 9:30:41 AM MST Refresh

Hardware Partitions Power and Cooling Cells I/O General

Partitions:

<input type="checkbox"/> 0 >> z000	<input type="checkbox"/> 1 >> z001	<input type="checkbox"/> 2 >> z002
<input type="checkbox"/> 3 >> z003	<input type="checkbox"/> 4 >> z004	<input type="checkbox"/> 5 >> z005
<input type="checkbox"/> 6 >> z006	<input type="checkbox"/> 7 >> z007	<input type="checkbox"/> 8 >> z008
<input type="checkbox"/> 9 >> z009	<input type="checkbox"/> 10 >> z0010	<input type="checkbox"/> 11 >> z0011
<input type="checkbox"/> 12 >> z0012	<input type="checkbox"/> 13 >> z0013	<input type="checkbox"/> 14 >> Available

The diagram shows three server cabinets: cab0, cab1, and cab8. Each cabinet has a 'CELLS' section at the top with 8 slots (0-7) and an 'I/O' section below with 3 chassis (0/1, 1/1, 2/1). Cab0 has partitions 0-7, cab1 has partitions 8-11, and cab8 has partitions 12-13. Connections are shown between the cells and the I/O chassis across the cabinets.

- + Complex
- + nPartition
- + Cell
- + I/O
- + Tools
- + Help

Selected Items:

Partitions	0
Cells	0
I/O Chassis	0
I/O Slots	0

Deselect All

Status Icon Legend

Active/OK	↑
Inactive	↓
Deconfigured	⚠
Failed	✖
Powering On	⚡
Powered-Off	⏻
Unknown	?

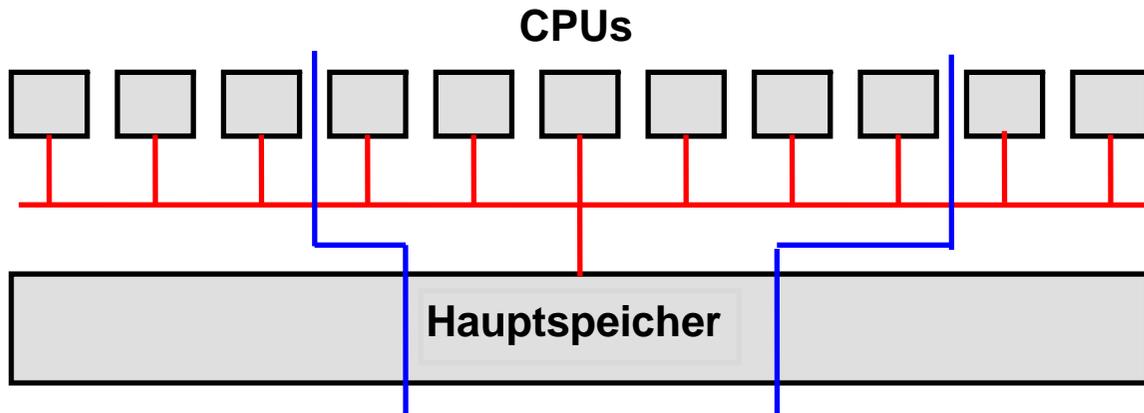
Done Internet

Hardware-centric management of hard-partitioned servers

Harte Partitionierung

Virtuelle Partitionierung

Logische Partitionen (LPAR)



Aufteilung eines Großrechners in mehrere SMPs

z/OS unterstützt symmetrische Multiprozessoren (SMP) mit bis zu 24 – 32 CPUs. Bei Unix, Linux und Windows Betriebs-systemen liegt die Grenze für Transaktions- und Datenbankanwendungen eher bei 12 – 16 CPUs.

Moderne Großrechner (Systeme) verfügen über wesentlich mehr CPUs (bis zu 256). Sie werden deshalb in mehrere SMPs aufgeteilt, die über einen zentralen Switch miteinander kommunizieren.

Der Systemadministrator kann den gesamten Hauptspeicher in unterschiedlichen Größen auf die einzelnen Hauptspeicher aufteilen.

Bei den Sunfire und HP Superdome Rechnern ist die Granularität der SMPs jeweils 4, 8, 12 oder 16 CPUs. System z und z/OS erlauben eine beliebig kleine Granularität

Virtuelle Partitionierung

Die Virtuelle Partitionierung mit Software ist in der Regel flexibler als die Hardware-Partitionierung. Allerdings ist durch den Einsatz von Software der Overhead größer, der für die Steuerung der Umgebung benötigt wird.

Die Virtuelle Partitionierung mit Hilfe von Software wird häufig auch als Logische Partitionierung oder als Virtualisierung bezeichnet.

Da die Hardware-Umgebung virtuell abgebildet wird, kann auch mit Hardware gearbeitet werden, die physisch gar nicht vorhanden ist. Weiterhin werden die Ressourcen einer virtuellen Maschine nur dann benötigt, wenn in der virtuellen Maschine Anwendungen laufen.

Emulator

Ein Emulator ist ein Software Paket, welches auf einem Rechner mit der Hardware-Architektur x (Host) einen Rechner mit der Hardware-Architektur y (Gast) emuliert. Jeder einzelne Maschinenbefehl eines Programms in der Architektur y wird durch den Emulator interpretiert.

Beispiele:

Hercules und ***FLEX-ES*** emulieren einen zSeries Rechner mit dem z/OS Betriebssystem auf einem Intel/AMD Windows oder Linux Rechner.

Microsoft VirtualPC Typ 1 (Firma Connectix) emuliert einen Intel/AMD mit dem Windows Betriebssystem auf einem Apple MAC PowerPC Rechner mit einem (früher verfügbaren) PowerPC Mikroprozessor.

Bochs ist ein in C++ geschriebener Open Source Emulator, der einen Rechner mit der die Intel/AMD Architektur auf vielen anderen Plattformen emuliert, z.B. PowerPC.

Die Emulation ist sehr Performance-aufwändig. Ein Performance Verlust um einen Faktor 10 oder noch schlechter ist häufig anzutreffen. Mehrere emulierte Gast Rechner auf einem Host Rechner sind zwar möglich, aber wegen des Performance Verlustes nicht üblich. Moderne Mikroprozessoren sind aber schnell genug, so dass der Betrieb von z/OS unter Hercules auf einem PC für Experimentierzwecke oder Software Entwicklung durch einen einzelnen Programmierer mit einer durchaus brauchbaren Performance erfolgt.

Virtuelle Maschine

Eine virtuelle Maschine ist etwas Ähnliches wie eine emulierte Maschine. Der Unterschied ist: Auf einem Host Rechner mit der Hardware-Architektur x wird ein (oder mehrere) Gast Rechner der **gleichen** Architektur abgebildet. Die meisten Maschinenbefehle der virtuellen Maschine brauchen nicht interpretiert zu werden. Der Leistungsverlust einer virtuellen Maschine hält sich deshalb in erträglichen Grenzen (bis zu 50 % unter VMWare, wenig mehr als 1 % unter z/OS PR/SM).

Beispiele für Hardware oder Betriebssysteme, die Virtuelle Maschinen unterstützen, sind:

VM/370, *z/VM* und der *PR/SM* Hypervisor für die System z und S/390 Hardware-Architektur.

Für den PowerPC Microprozessor existiert der *LPAR Hypervisor*.

VMWare, *XEN* und *Microsoft VirtualPC* Typ 2 bilden mehrere Windows oder Linux Gast Maschinen auf einem Windows oder Linux Host ab.

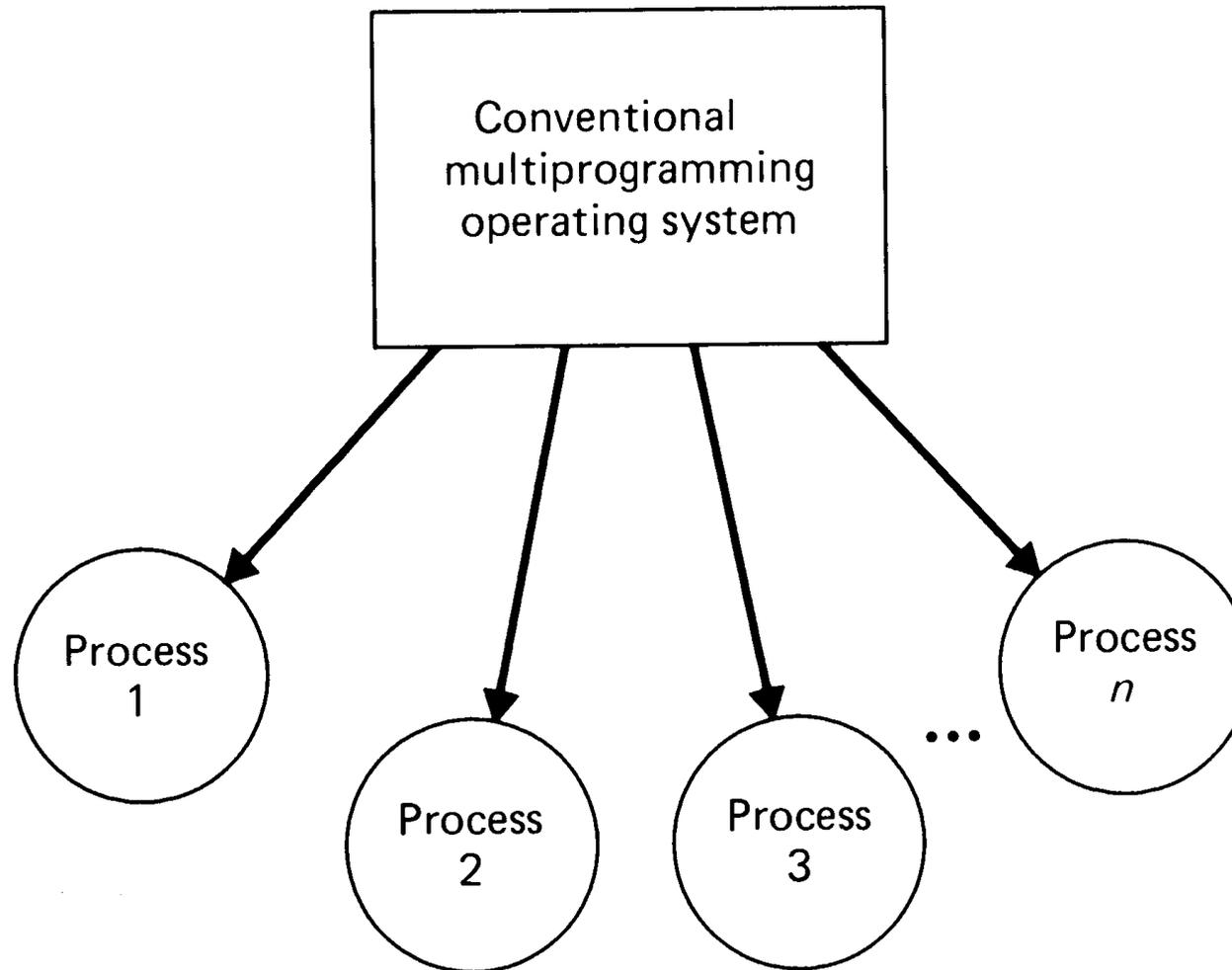
Intel *Virtualization Technology (VT)* für die Itanium Architecture (VT-i) sowie die X86 (Pentium) Architecture (VT-x).

AMD *Pacifica*, ebenfalls für die X86 (Pentium) Architecture.

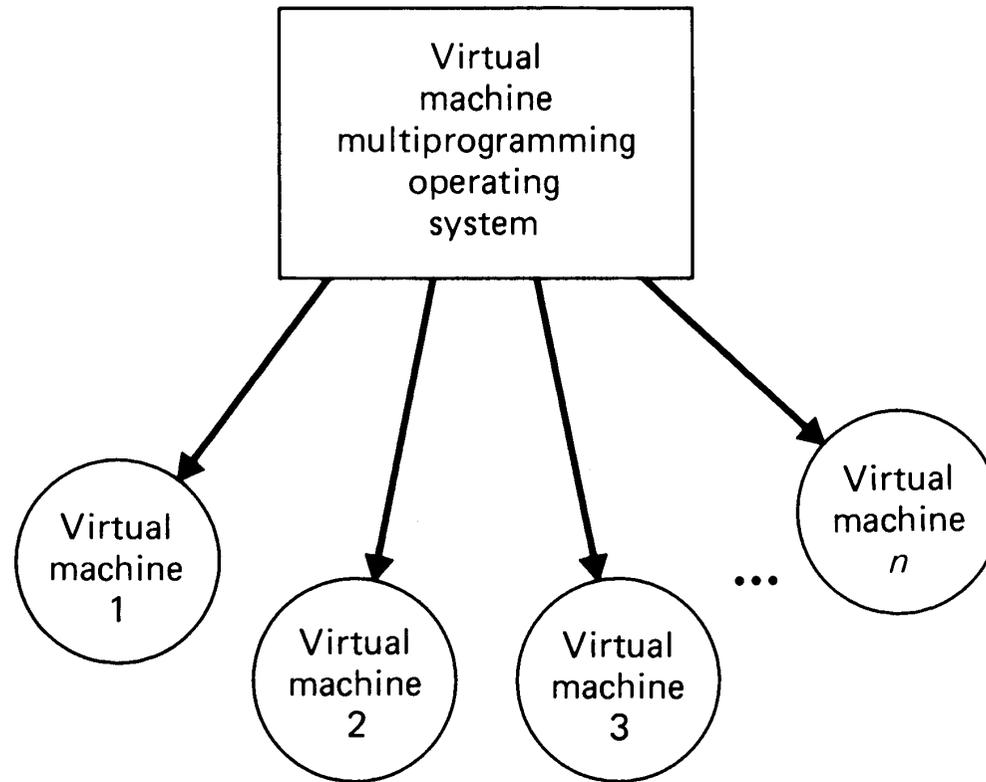
VirtualBox von Sun Microsystems ist eine Virtualisierungssoftware für Linux, Windows, Mac OS X, FreeBSD oder Solaris.

Paravirtualization wird von *Xen* und *Denali* implementiert.

Weiterführende Literatur: Joachim von Buttlar, Wilhelm G. Spruth: Virtuelle Maschinen. zSeries und S/390 Partitionierung. IFE - Informatik Forschung und Entwicklung, Heft 1/2004, Juli 2004. <http://www-ti.informatik.uni-tuebingen.de/~spruth/publish.html>



Unter einem normale multiprogrammierten Betriebssystem laufen zahlreiche Prozesse gleichzeitig ab. Jeder Prozess läuft in einem eigenen virtuellen Adressenraum.



Unter einem Virtual Machine Operating System laufen spezielle Prozesse, von denen jeder eine virtuelle Maschine implementiert. Jede virtuelle Maschine hat ihr eigenes Betriebssystem, unter dem Anwendungen der virtuellen Maschine laufen. Auf zwei virtuellen Maschinen können unterschiedliche Betriebssysteme laufen.

Host System und Gast System

Das Virtual Machine Operating Systems wird häufig als **Hypervisor** oder als **Host Betriebssystem** bezeichnet. Analog bezeichnet man das emulierte Betriebssystem als das **Gast-Betriebssystem**, welches auf einer **Gast-Maschine** (virtuelle Maschine) läuft. Das Host-Betriebssystem verfügt über einen **Host-Kernel (Überwacher)** und das Gast-Betriebssystem verfügt über einen **Gast-Kernel**.

Wir bezeichnen als Kernel denjenigen Teil des Betriebssystems, der im Kernel-Modus ausgeführt wird. Vielfach besteht der Hypervisor nur aus einem Host-Kernel. Es gibt aber Beispiele, wie VMware, wo dies nicht der Fall ist, und zusätzliche Betriebssystem-Funktionen in Anspruch genommen werden.

Die Begriffe virtuelle Maschine und Gast Maschine sowie die Begriffe Hypervisor, Host Kernel und Virtual Machine Monitor und Virtual Machine Operating System bedeuten jeweils das Gleiche und werden austauschbar verwendet.

Überwacherstatus Problemstatus

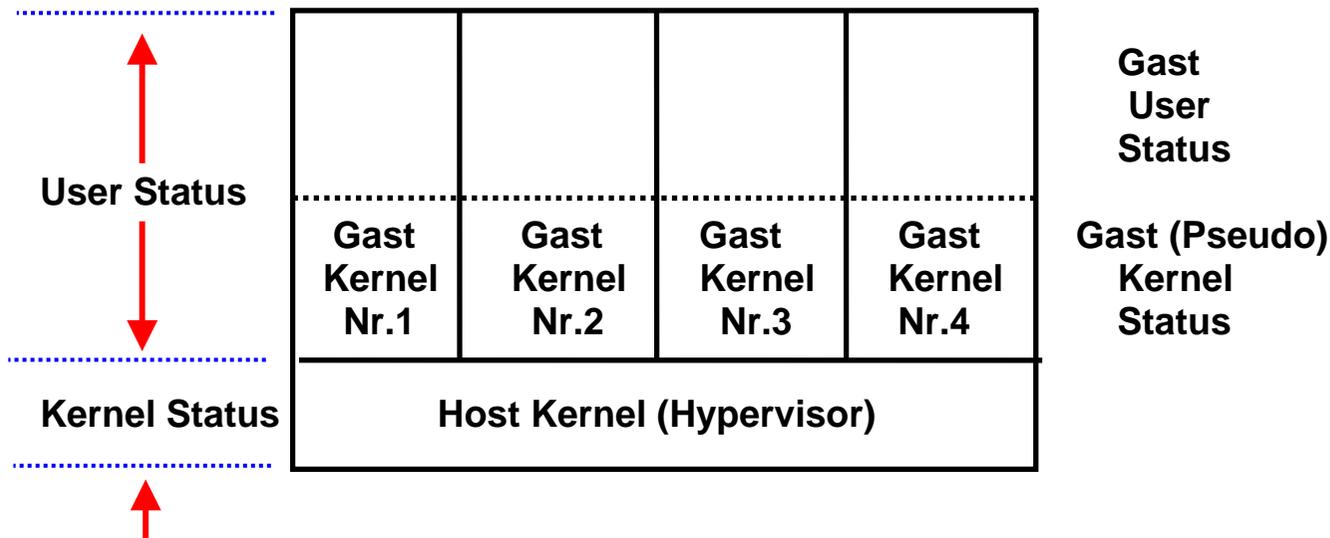
Definiert durch 1 Bit in einem Steuerregister in der Zentraleinheit
(Status Register, Programm Status Wort)

Der Überwacher (Kernel) läuft im Überwacherstatus
(Supervisor State, Kernel State).

Anwendungsprogramme laufen im Problemstatus
(Problem State, User State).

Auswirkungen:

- Bestimmte „**Privilegierte**“ Maschinenbefehle können nur im Überwacherstatus ausgeführt werden
- **Speicherschutz**. Im Problemstatus kann nur auf einen Teil des Hauptspeichers zugegriffen werden



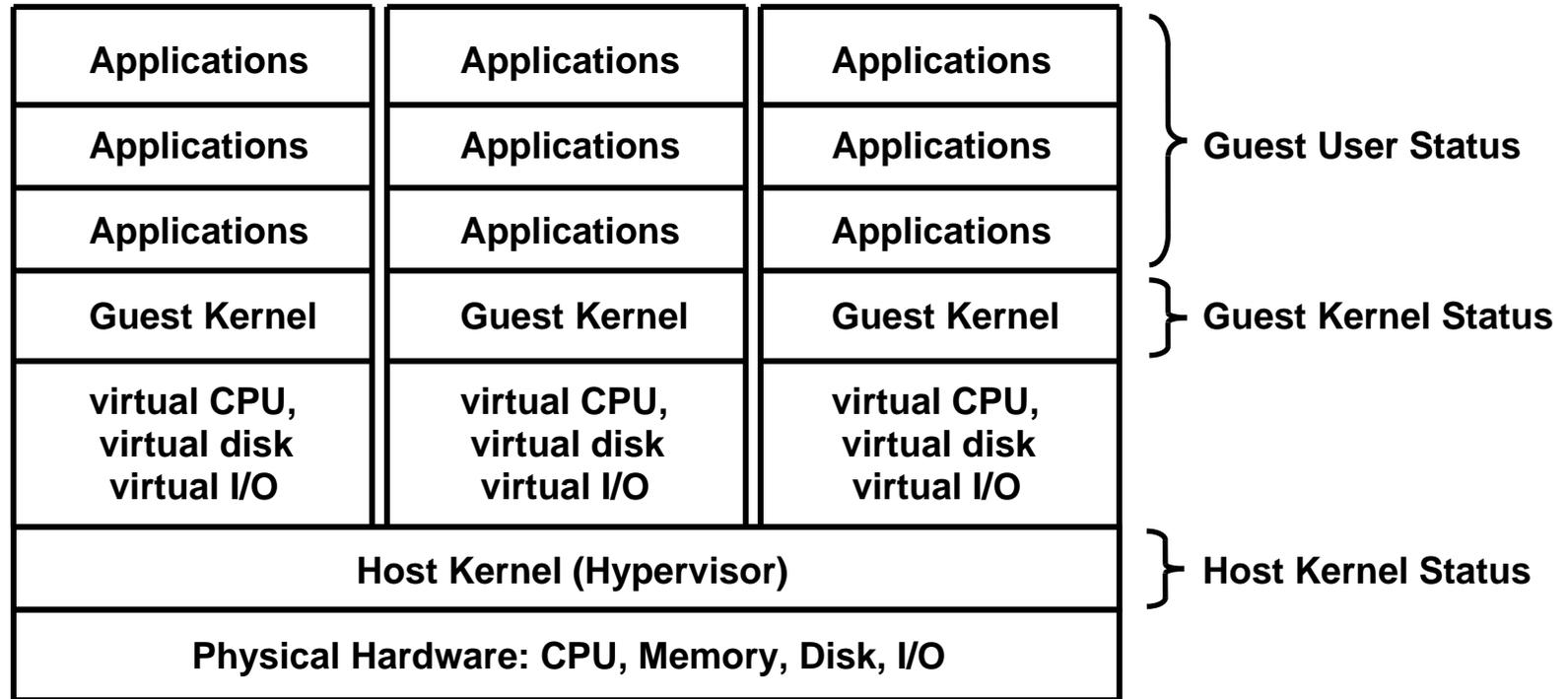
Gleichzeitiger Betrieb mehrerer Betriebssysteme auf einem Rechner

Der Ansatz, mehrere *Gast*-Betriebssysteme unter einem *Host*-Betriebssystem zu betreiben, wird als logische oder virtuelle Partitionierung bezeichnet

Der Host-Kernel übernimmt die Kontrolle über den Rechner immer dann, wenn eine Gast-Maschine einen Maschinenbefehl auszuführen versucht, der das korrekte Verhalten des Hosts oder einer anderen Gast-Maschine beeinflussen würde. Derartige Maschinenbefehle werden als **sensitive** Befehle bezeichnet. Der Host-Kernel interpretiert diese Befehle für den Gast und stellt sicher, dass die Semantik erhalten bleibt. Wenn Gast und Host die gleiche Hardware Architektur verwenden, können im Gegensatz zur Emulation fremder Architekturen alle nicht-sensitiven Maschinenbefehle eines Prozesses direkt von der CPU ausgeführt werden.

Eine Gast-Maschine Prozess führt hauptsächlich nicht-sensitive Befehle aus. Nur sensitive Maschinenbefehle werden emuliert. Da dies nur wenige sind, ist der Leistungsabfall durch Emulation gering. Im Gegensatz dazu ist eine vollständige Emulation sehr aufwendig.

Im einfachsten Fall sind alle sensitiven Maschinenbefehle gleichzeitig privilegierte Maschinenbefehle.



A Hypervisor (also called Host Kernel) runs on top of the hardware. The Hypervisor has the characteristics of a regular operating system kernel. It includes interrupt handlers, a timeslicing scheduler/dispatcher, I/O management, and virtual storage management.

It hosts several virtual Machines (Guest machines). In the simplest case, a timesliced part of the CPU, some memory and I/O, including disk, is permanently dedicated to a particular virtual machine. The virtual machine runs its own operating system including its operating system kernel (guest kernel). The virtual machine operating system runs multiple applications as user-processes in parallel.

Typ A

Gast 1 Betriebs- system	Gast 2 Betriebs- system	Gast 3 Betriebs- system	Gast 4 Betriebs- system
Hypervisor - Host Kernel, z.B. z/VM, XEN, ESX Server			
Hardware			
CPU	Hauptspeicher	Plattenspeicher	Netzwerk

Typ B

normal laufende Anwen- dungen	Gast 1 Betriebs- system	Gast 2 Betriebs- system	Gast 3 Betriebs- system
	VMware GSX Server, MS Virtual Server		
Host Betriebssystem, z.B. Windows			
Hardware			
CPU	Hauptspeicher	Plattenspeicher	Netzwerk

Alternativen für die Virtualisierung

Es existieren zwei Alternativen für die Implementierung.

Typ A ist die bisher beschriebene Konfiguration. Sie wird auf allen größeren Servern eingesetzt, besonders auch auf den Mainframes..

Für den PC existiert als zusätzlich Alternative **Typ B**. Der Host Kernel läuft als Anwendung unter einem regulären PC Betriebssystem, z.B. Windows. Er hat nicht alle Eigenschaften eines selbständigen Betriebssystem Kernels, sondern nutzt statt dessen teilweise Komponenten des darunterliegenden Windows Betriebssystems.

Der Vorteil von Typ B ist, dass Anwendungen auch nicht virtualisiert mit besserem Leistungsverhalten laufen können. Als Nachteil laufen virtuelle Maschinen bei Typ B langsamer als bei Typ A.

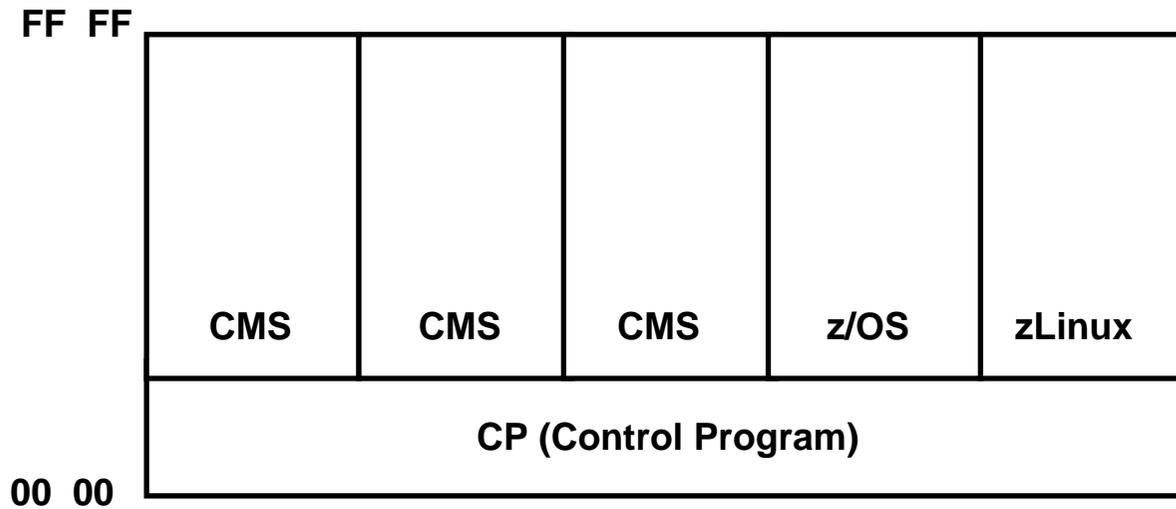
Gleichzeitiger Betrieb mehrerer Betriebssysteme auf einem Rechner

Der Ansatz, mehrere *Gast*-Betriebssysteme unter einem *Host*-Betriebssystem zu betreiben, wird als logische oder virtuelle Partitionierung bezeichnet

Der Host-Kernel übernimmt die Kontrolle über den Rechner immer dann, wenn eine Gast-Maschine einen Maschinenbefehl auszuführen versucht, der das korrekte Verhalten des Hosts oder einer anderen Gast-Maschine beeinflussen würde. Derartige Maschinenbefehle werden als **sensitive** Befehle bezeichnet. Der Host-Kernel interpretiert diese Befehle für den Gast und stellt sicher, dass die Semantik erhalten bleibt. Wenn Gast und Host die gleiche Hardware Architektur verwenden, können im Gegensatz zur Emulation fremder Architekturen alle nicht-sensitiven Maschinenbefehle eines Prozesses direkt von der CPU ausgeführt werden.

Eine Gast-Maschine Prozess führt hauptsächlich nicht-sensitive Befehle aus. Nur sensitive Maschinenbefehle werden emuliert. Da dies nur wenige sind, ist der Leistungsabfall durch Emulation gering. Im Gegensatz dazu ist eine vollständige Emulation sehr aufwendig.

Im einfachsten Fall sind alle sensitiven Maschinenbefehle gleichzeitig privilegierte Maschinenbefehle.



z/VM (VM/370) Betriebssystem

Unter z/VM (ehemals VM/370 und VM/390) laufen alle Anwendungen grundsätzlich auf virtuellen Maschinen. Der Host Kernel (Hypervisor) wird als CP (Control Programm) bezeichnet und läuft im Kernelstatus.

Für VM/370 wurde ein eigenes Gast Betriebssystem CMS (Conversational Monitor System) entwickelt. Dieses und alle anderen Gast Betriebssysteme (einschließlich ihrer Kernel Funktionen) laufen im Problemstatus. Privilegierte Maschinenbefehle (z.B. E/A) werden von CP abgefangen und interpretativ abgearbeitet.

CMS ist ein besonders für die Software Entwicklung ausgelegtes Einzelplatz Betriebssystem. Für 1000 gleichzeitige CMS Benutzer werden 1000 CMS Instanzen angelegt. Ähnliches ist mit zLinux möglich.

z/VM erreicht eine uneingeschränkte S/390 und System z Kompatibilität für alle Gast Betriebssysteme. Der Performance Verlust ist relativ gering (< 5 %).

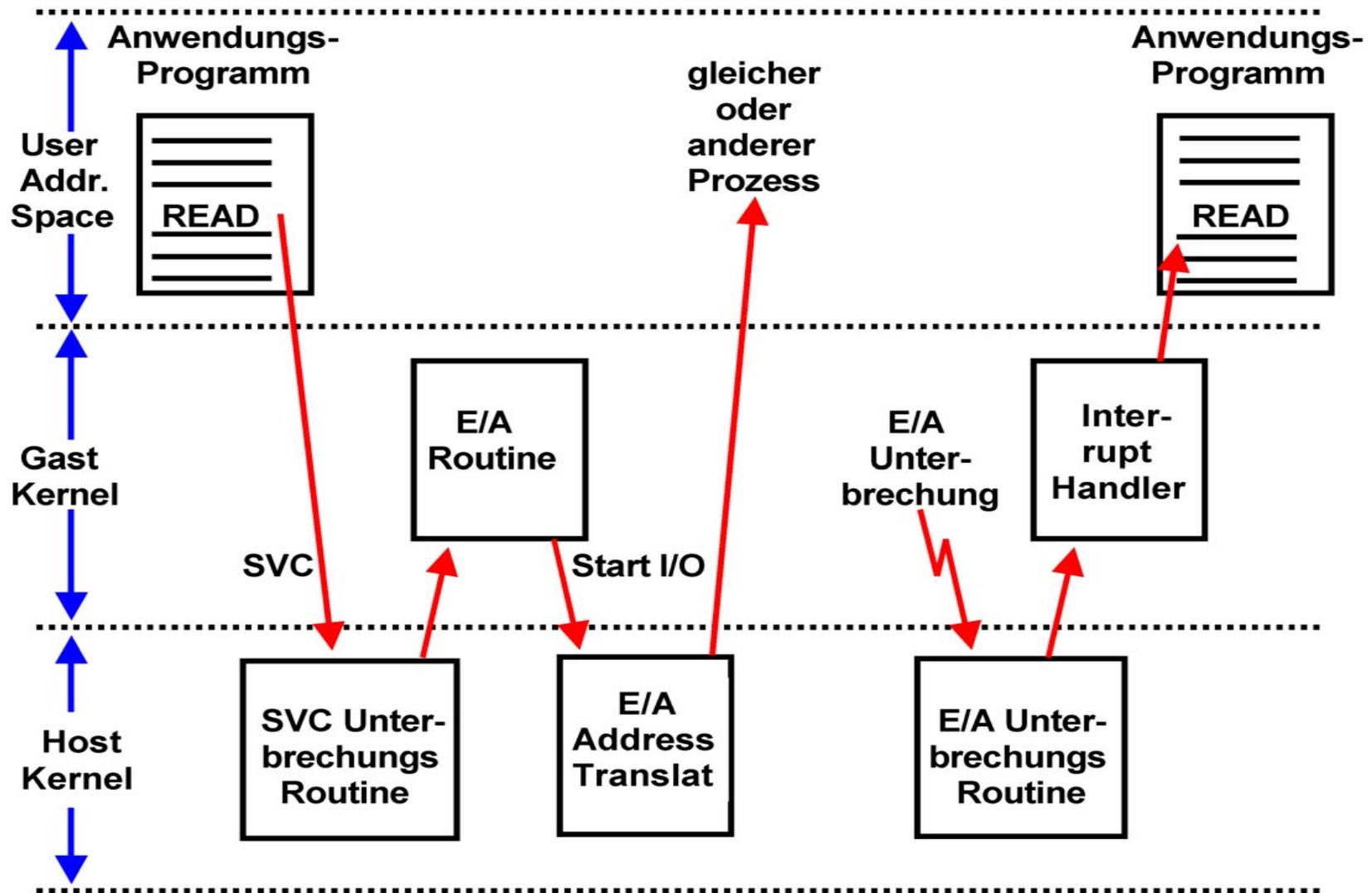
Steuerung der virtuellen Maschine

Alle Gast Maschinen laufen in einem eigenen virtuellen Adressenraum

Der Host Kernel Zeitscheiben-Scheduler übergibt die Kontrolle über die CPU(s) einer Gast Maschine.

Der Kernel der Gast Maschine läuft im User Mode (Problem Mode). Wenn das Programm des Gast Betriebssystems versucht, einen privilegierten Maschinenbefehl auszuführen, führt dies zu einer Programmunterbrechung. Privilegierte Maschinenbefehle sind häufig sensitive Maschinenbefehle.

Die Programmunterbrechungsroutine des Host Kernels interpretiert den privilegierten Maschinenbefehl der Gast Maschine soweit als erforderlich und übergibt die Kontrolle zurück an den Kernel des Gastbetriebssystems



Als Beispiel ist eine I/O READ Operation dargestellt, die von einem Anwendungsprogramm einer Gastmaschine ausgeführt wird.

Ausführung einer I/O Operation

Beispiel READ

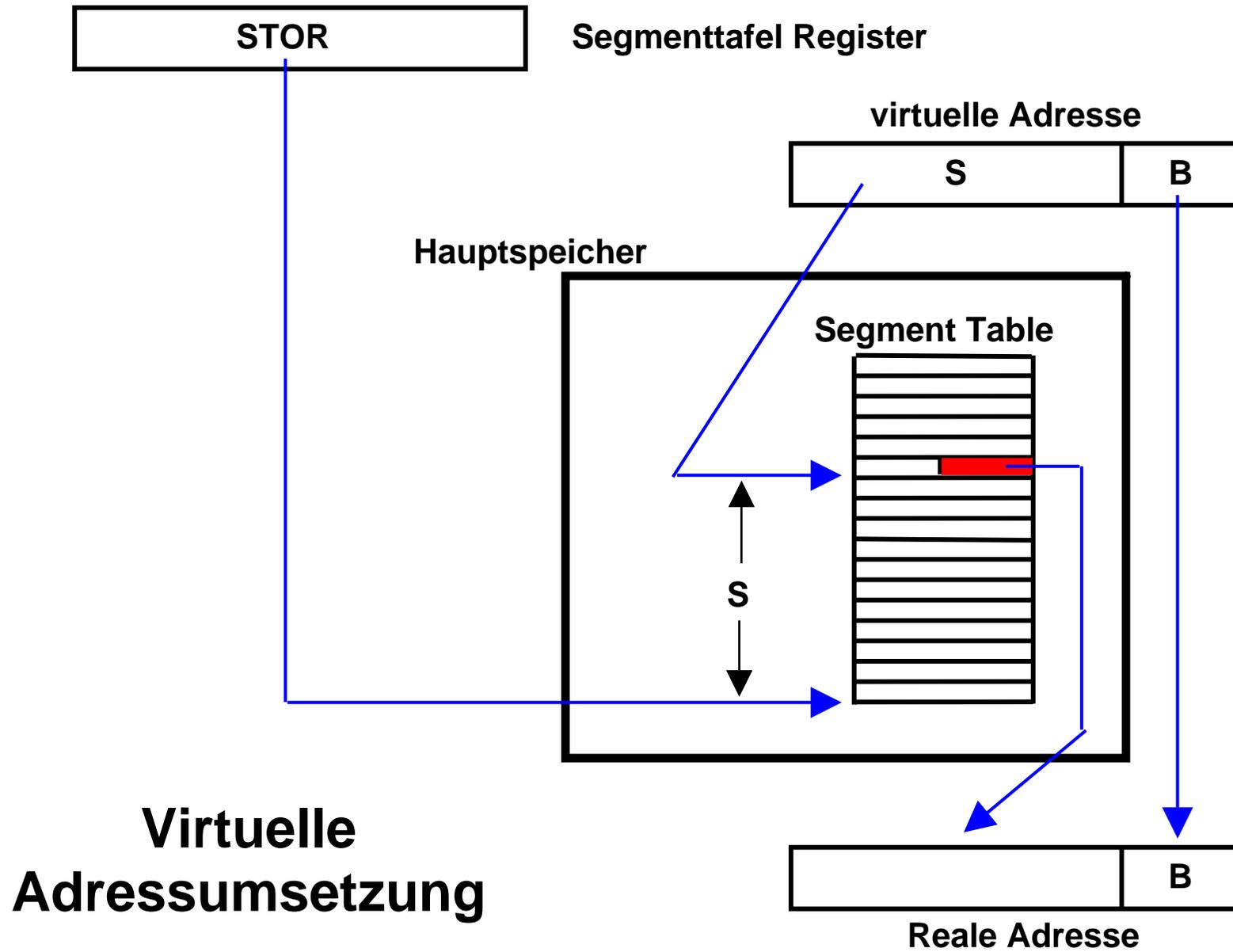
Die Ausführung einer (Plattenspeicher) READ Operation in einem Cobol oder C++ Anwendungsprogramm führt zum Aufruf einer Library Routine, welche mittels eines SVC (Supervisor Call) Maschinenbefehl in eine Routine des Kernels verzweigt. Diese führt die I/O Operation stellvertretend für den Anwendungsprozess aus.

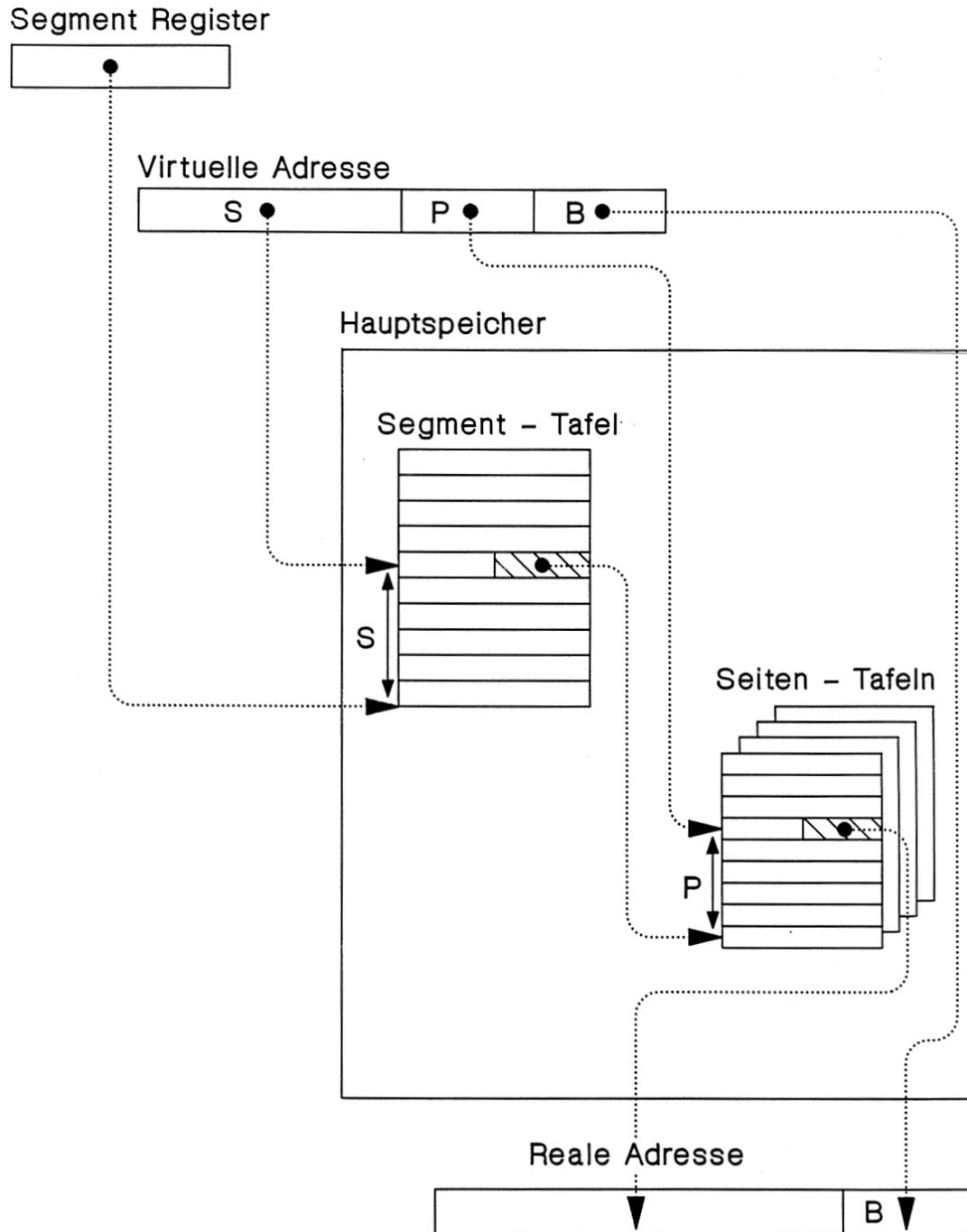
Eine virtuelle Maschine läuft als Prozess unter dem Host Kernel im User Status, einschließlich des Kernels des Gast Betriebssystems. Erfolgt die READ Operation in einer virtuellen Gast-Maschine, so bewirkt der SVC Befehl einen Aufruf nicht des Gast Kernels sondern des Host Kernels, da nur dieser im Kernel Status läuft. Nur dieser ist in der Lage, den eigentlichen I/O Driver auszuführen.

Auf der anderen Seite verwaltet der Gast Kernel die I/O Buffer oder die I/O Steuerblöcke (z.B. DCB, ECB, UCB unter z/OS). Der Host Kernel delegiert deshalb die Erstellung der I/O Anforderung an den Gast Kernel.

Die so erstellte I/O Anforderung arbeitet aber mit den virtuellen Adressen des Gast Kernels. Der Gast Kernel übergibt deshalb die entgeltige Fertigstellung an den Host Kernel (über einen SVC Aufruf), der die I/O Anforderung vervollständigt und an die Plattenspeicher-Steuereinheit weitergibt.

Der Abschluss der I/O Operation wird mit einer I/O Unterbrechung dem Host Kernel übergeben. Dieser übergibt die Unterbrechung an den entsprechenden Unterbrechungshandler des Gast Kernels. Letzterer benachrichtigt die Anwendung, welche die READ Operation auslöste, von deren erfolgreichen Abschluss.





Zweistufige Adressumsetzung , S/390, Pentium

In der S/390 Architektur
übernimmt Control
Register 1 (CR1) die
Rolle des Segment
Registers (STOR,
Segment Table Origin
Register)

Virtuelle Adressumsetzung Pentium, S/390

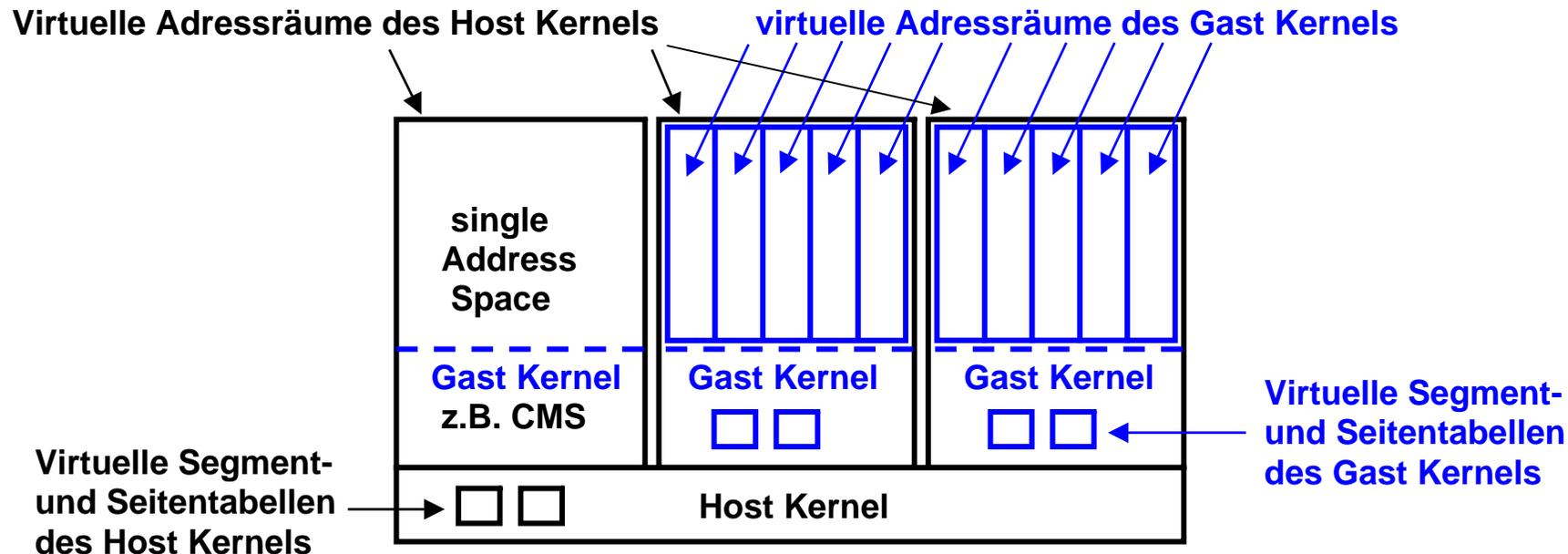
Jeder Prozess und jede virtuelle Maschine hat ihren eigenen virtuellen Adressenraum.

Die Adressumsetzung erfolgt durch eine Segment Tabelle (in der praktischen Implementierung zwei Tabellen, Segment and Page Table), die im Kernel Bereich des Hauptspeichers untergebracht sind. Die Anfangsadresse der Segmenttabelle steht in einem Control Register der Zentraleinheit, z.B. CR Nr, 1 bei der System z Architektur oder CR Nr. 3 bei der Pentium Architektur.

Zur Leistungsverbesserung werden die gängigen Adressen in einem Adressumsetzungspuffer gecached.

Die Adressumsetzung erfolgt bei jedem Hauptspeicherzugriff durch Hardware mit Unterstützung durch die Tabellen im Kernel Bereich . Sie kann durch den Programmierer nicht beeinflusst werden.

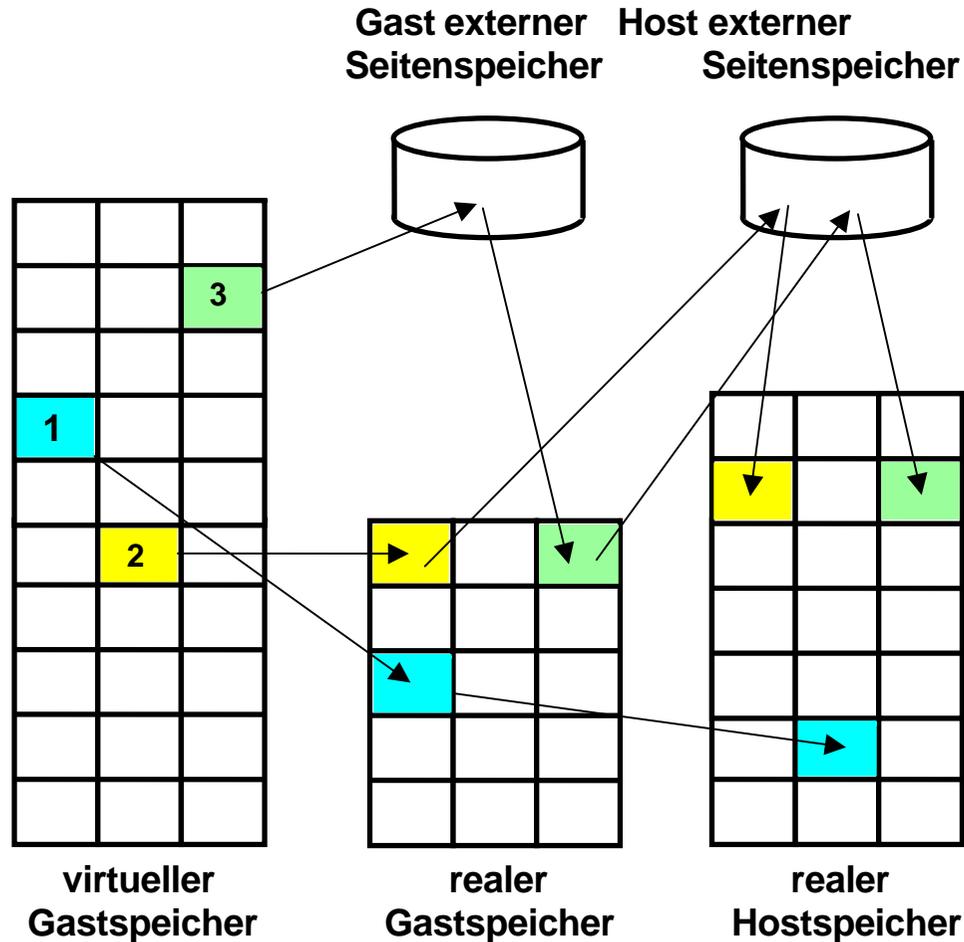
Problem: Eine Gastmaschine, die mit Virtueller Adressumsetzung konfiguriert ist, will ihre eigene Adressumsetzung machen.



Virtuelle Maschinen mit virtueller Adressumsetzung

Bei der Einführung von VM/370 war CMS das einzige verfügbare Gast-Betriebssystem. CMS ist (ähnlich wie DOS 6.22) ein single User Betriebssystem und verwendet keine virtuelle Adressumsetzung.

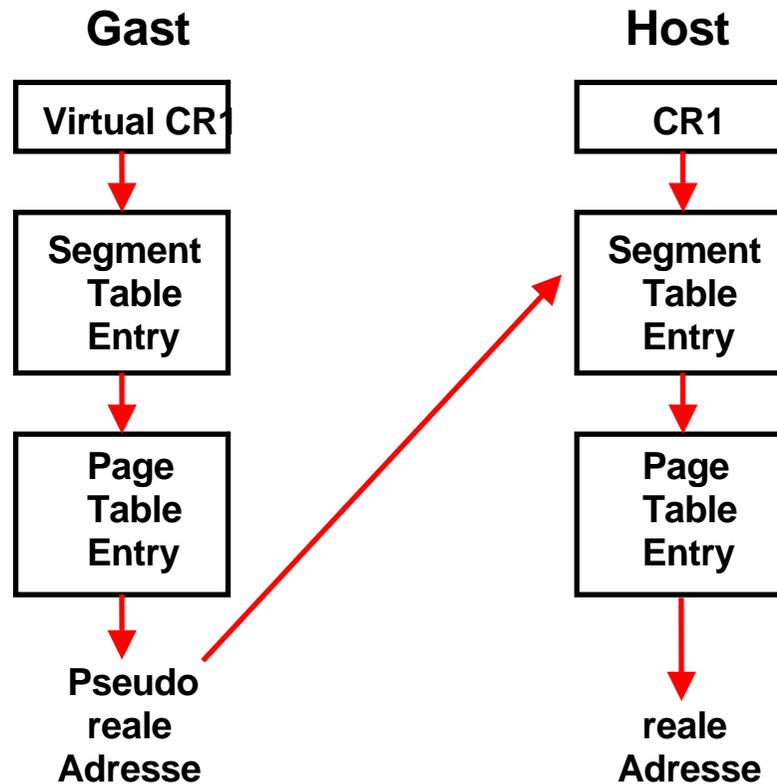
Das OS/370 Betriebssystem hat eine eigene virtuelle Adressumsetzung. Wenn es als Gast Betriebssystem eingesetzt werden soll, läuft es in einem einzigen virtuellen Adressenraum des Host Kernels. Es will aber seine eigenen Prozesse in eigenen virtualisierten Adressräumen laufen lassen. Wir brauchen eine Virtualisierung der virtuellen Adressenräume des Gastbetriebssystems.



Adressumsetzung zwischen Gast-Kernel und Host-Kernel

Drei alternative Möglichkeiten

1. Der gewünschte Rahmen befindet sich im (pseudo-)realen Speicher des Gastes und auch im realen Speicher des Rechners selbst (realer Hostspeicher).
2. Der gewünschte Rahmen befindet sich im realen Speicher des Gastes aber nicht im realen Hostspeicher. Der Host-Kernel löst eine Fehlseitenunterbrechung aus und lädt den Rahmen in den realen Hauptspeicher.
3. Der gewünschte Rahmen befindet sich nicht im realen Speicher des Gastes. Der Gast-Kernel löst eine Fehlseitenunterbrechung aus, die vom Host-Kernel abgefangen wird. Dieser lädt den Rahmen in den realen Hauptspeicher und gleichzeitig auch in den realen Gastpeicher.

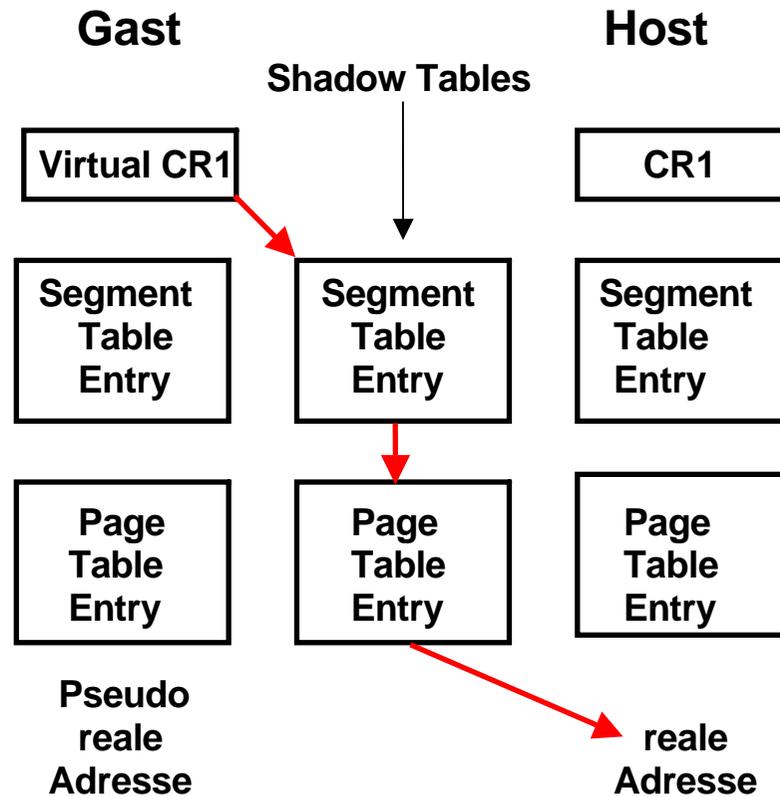


Problem der Adressübersetzung für die Virtuelle Maschine

Die virtuelle Adressumsetzung mit Hilfe von Segment- und Seitentafeln erfolgt durch Hardware und kann durch Software nicht beeinflusst werden.

Ein Gast Betriebssystem verfügt in seinem Kernel Bereich über eigene Segment- und Seitentabellen, mit denen es seine Adressumsetzung beschreibt.

Der Adressraum, den der Gast als real ansieht, ist jedoch in Wirklichkeit virtuell aus Sicht des Host Kernels und wird von diesem ebenfalls über Segment- und Seitentabellen beschrieben. Somit stehen in den Seitentabellen des Gastes stehen die falschen Werte.



Shadow Page Tables unter VM/370 und VMware

VM/370 und VMware erstellen anhand der Gast- und ihrer eigenen Segment- und Seitentabellen sogenannte *Shadow Tables*, die direkt die virtuellen Adressen des Gastes auf reale Adressen des Hosts abbilden. Diese Tabellen liegen im Speicherbereich des Host-Kernels. Wenn immer eine Gast Maschine (in Wirklichkeit ein Prozess des Host Kernels) die Verfügungsgewalt über die CPU bekommt, muss der Host Kernel die Shadow Tables mit neu berechneten korrekten Werten füllen. Bei jeder Änderung in den Segment- und Seitentabellen einträgen des Gastes werden diese Änderungen in den Shadow Tables nachvollzogen.

Sensitive Maschinenbefehle

Es muss verhindert werden, dass bei der Ausführung eines Maschinenbefehls innerhalb einer virtuellen Maschine das Verhalten einer anderen virtuellen Maschine beeinflusst wird.

Die Ausführung von nicht-sensitiven Maschinenbefehlen beeinflusst nicht das Verhalten einer anderen virtuellen Maschine.

Die Ausführung von sensitiven Maschinenbefehlen kann das Verhalten einer anderen virtuellen Maschine beeinflussen.

**Einfachste Implementierung: Alle sensitiven Maschinenbefehle sind gleichzeitig auch privilegierte Maschinenbefehle und können nur vom Host-Kernel ausgeführt werden.
Beispiel VM/370.**

Probleme der x86 Architektur

Im Vergleich zu VM/370 sind die VMware ESX und GSX Server benachteiligt, weil einige kritische Eigenschaften in der x86 Architektur fehlen. Für den Betrieb von Gast-Maschinen ist es erforderlich, dass alle Maschinenbefehle, welche den privilegierten Maschinenstatus abändern oder auch nur lesen, nur im Kernel Status ausgeführt werden können.

Wenn ein Gast ein Controlregister schreibt, muss der Host Kernel diese Instruktion abfangen, damit nicht das reale Kontrollregister des Hosts verändert wird. Der Host Kernel wird jetzt nur die Effekte der Instruktion für diesen Gast simulieren. Liest der Gast anschließend diese Kontrollregister wieder aus, so muss diese Instruktion ebenfalls abgefangen werden, damit der Gast wieder den Wert sieht, den er vorher in das Register geschrieben hat (und nicht etwa den realen Wert des Kontrollregisters, der nur für den Host sichtbar ist).

Da die x86 Architektur diese Bedingung nicht erfüllt, ist es nicht möglich, wie unter VM/370, alle Maschinenbefehle einfach im User Mode auszuführen, und auf Programmunterbrechungen zu vertrauen wenn auf privilegierten Maschinenstatus Information zugegriffen wird.

VMware's ESX Server überschreibt hierzu dynamisch Teile des Gast-Kernels und schiebt Unterbrechungsbedingungen dort ein, wo eine Intervention des Host-Kernels erforderlich ist. Als Folge hiervon tritt ein deutlicher Leistungsverlust auf, besonders bei Ein-/Ausgabe-Operationen. Manche Funktionen sind nicht vorhanden oder können nicht genutzt werden.

Kompatibilitätsprobleme treten auf; es kann sein, dass bestimmte Anwendungen nicht lauffähig sind.

Sensitive Maschinenbefehle (2)

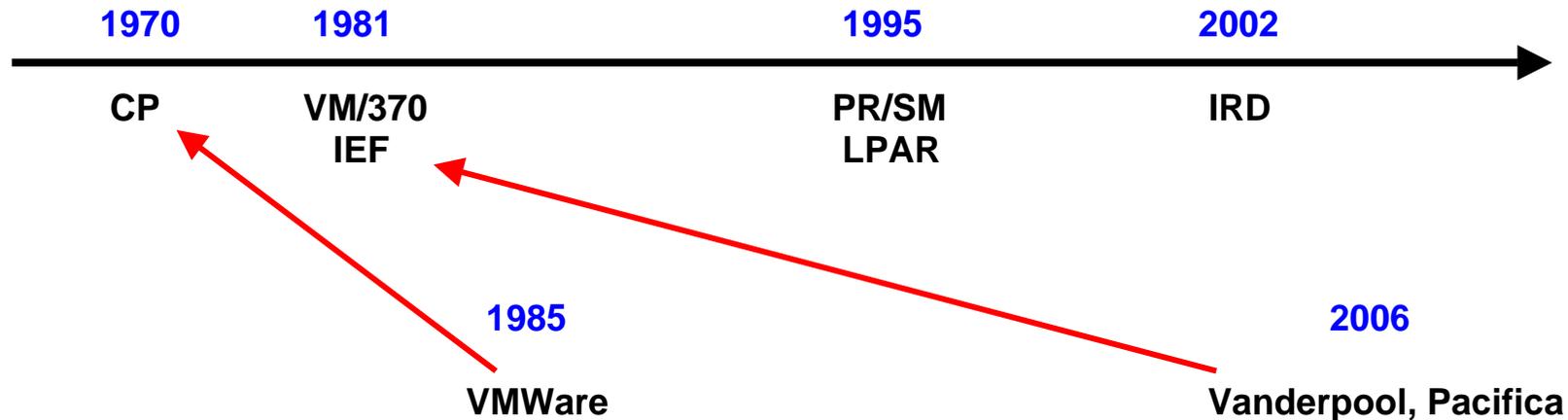
Many models of Intel's machines allow user code to read registers and get the value that the privileged code put there instead of the value that the privileged code wishes the user code to see.

Sensitive Maschinenbefehle können eine andere virtuelle Maschine beeinflussen.

VMware's ESX Server überschreibt hierzu dynamisch Teile des Gast-Kernels und schiebt Unterbrechungsbedingungen dort ein, wo eine Intervention des Host-Kernels erforderlich ist. Als Folge hiervon tritt ein deutlicher Leistungsverlust auf, besonders bei Ein-/Ausgabe-Operationen. Manche Funktionen sind nicht vorhanden oder können nicht genutzt werden.

Kompatibilitätsprobleme treten auf; es kann sein, dass bestimmte Anwendungen nicht lauffähig sind.

Die Entwicklung der Virtualisierung



Das VM/370 Betriebssystem mit dem CP Hypervisor und dem CMS Gast-Betriebssystem entstand 1970. 1985 Gründeten einige VM/370 Entwickler die Firma VMware.

Bereits 1981 hatte IBM für alle S/370 Rechner eine Erweiterung der Hardware Architektur eingeführt, die als „Interpretive Execution Facility“ (IEF) bezeichnet wurde, und zu einer erheblichen Leistungsverbesserung der virtuellen Maschinen führte. Eine vergleichbare Architekturerweiterung wurde von Intel und AMD im Jahre 2006 für die x86 (Pentium) Architektur unter dem Namen Vanderpool bzw. Pacifica eingeführt.

VT-x (Virtualisation Technology for x86) ist eine alternative Bezeichnung für Vanderpool. Eine ähnliche Einrichtung für Itanium hat den Namen VT-i . AMD Virtualization, abgekürzt AMD-V ist eine alternative Bezeichnung für Pacifica.

Bereits vorher im Jahre 1995 hatte die S/390 Architektur eine weitere grundlegende Verbesserung unter dem Namen „Logische Partition“ (LPAR) eingeführt. Dies wurde 2002 durch eine nochmalige Erweiterung, den „Intelligent Resource Director“ (IRD) ergänzt. Der PowerPC führte LPARs 2002 ein.

Experten gehen davon aus, dass x86 weitere 10 Jahre benötigen wird, um eine mit LPAR und IRD vergleichbare Funktionalität zu erreichen.

	Kernel Modus	User Modus
Host Modus	+	-
Gast Modus	+	+

Mögliche Zustände beim Einsatz des Host/Gast Modus

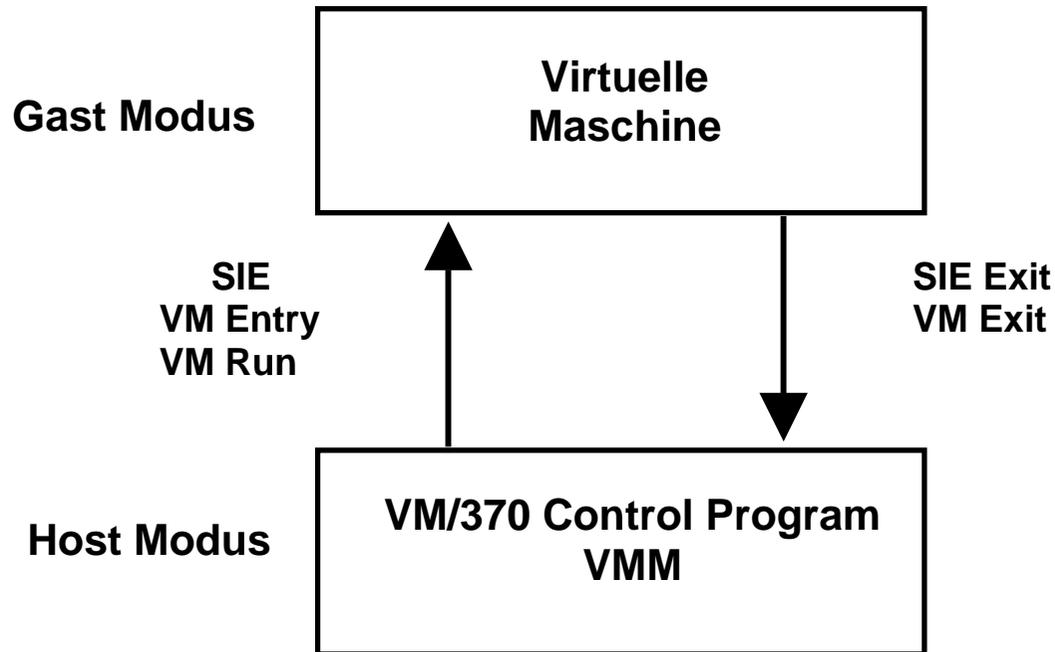
Jeder Rechner unterscheidet zwischen Kernel Modus (supervisor state) und User Modus (problem state). Das unterschiedliche Verhalten wird durch ein Bit im Status Register der CPU (Program Status Word, PSW) definiert.

Interpretive Execution Facility, Vanderpool und Pacifica führten zusätzlich noch eine Unterscheidung Host Modus/Gast/Modus mit Hilfe eines zweiten Bits im Statusregister ein. Damit sind die Kombinationen Host Modus/Kernel Modus, Gast Modus /Kernel Modus und Gast Modus / User Modus möglich. Der Hypervisor läuft grundsätzlich im Host Modus; die Kombination Host Modus / User Modus hat in Bezug auf virtuelle Maschinen keine Bedeutung.

Das Problem der sensitiven Maschinenbefehle kann nun dadurch gelöst werden, dass die Hardware sich im Gast Modus anders verhält als im Host Modus. Auch kann die Gast Maschine in ihrem Kernel Modus solche privilegierten Maschinenbefehle direkt selbst ausführen, die nicht sensitiv sind – sie braucht hierfür nicht mehr die Unterstützung des Host Kernels.

Vanderpool bezeichnet den Host Kernel Modus, in dem der Virtual Machine Monitor (VMM) ausgeführt wird, als *VMX Root Operation*. Jeder Gast läuft in *VMX non-Root Operation*.

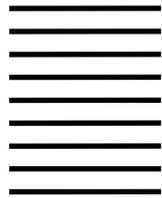
Der Aufruf eines Gastes erfolgt durch einen VM Entry Befehl (VM Run bei AMD VT). Die Rückkehr zum VMM erfolgt durch einen VM Exit Befehl.



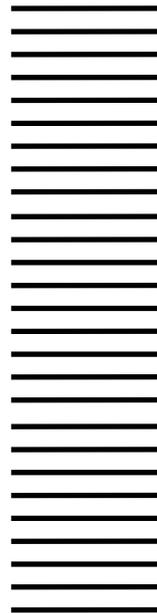
Im Gastmodus können privilegierte, aber nicht sensitive Maschinenbefehle vom Kernel des Gast Betriebssystems abgearbeitet werden. Das Problem der sensitiven nicht-privilegierten x86 Architektur wird durch zusätzliche Vanderpool bzw. Pacifica Hardware gelöst, die diese Befehle bei ihrer Dekodierung identifiziert und zwecks Ausführung an den VMM übergibt.

Host Programm

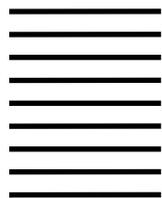
Gast Programm



SIE, VMENTRY,
VMRUN
Befehle



Exit Befehle

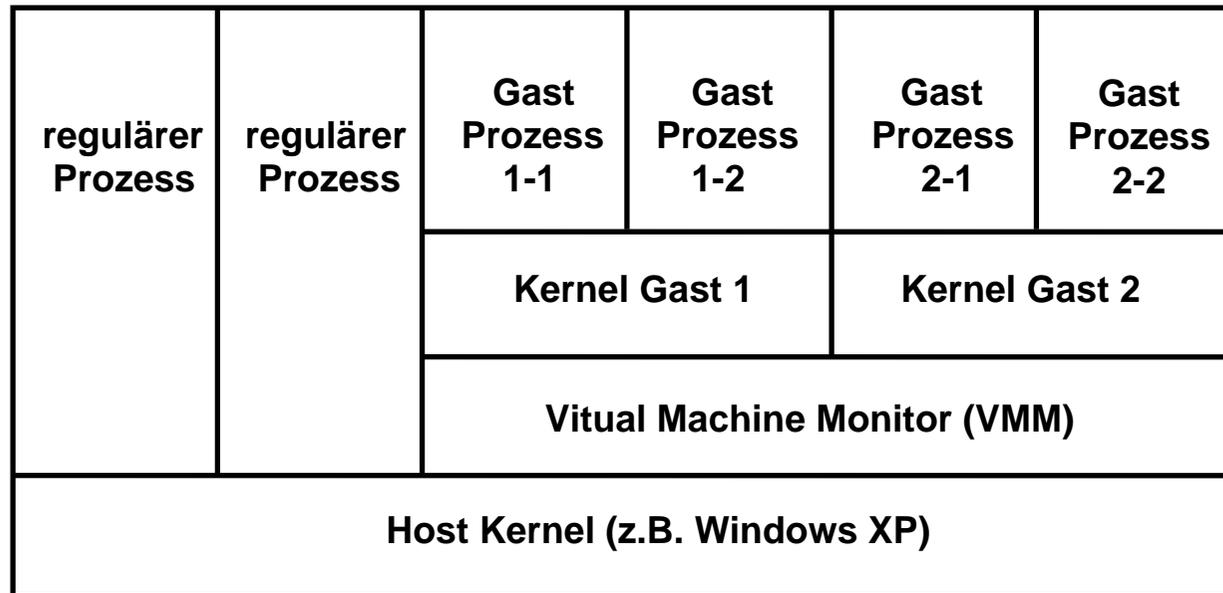


SIE, VMENTRY und VMRUN sind Maschinenbefehle, die nur im Host Modus ausführbar sind.

Übergang vom Host- in den Gast-Modus

Der Kernel des VM/370 Betriebssystems ist gleichzeitig der Hypervisor. Der Rechner läuft immer im Virtualisierungsmodus.

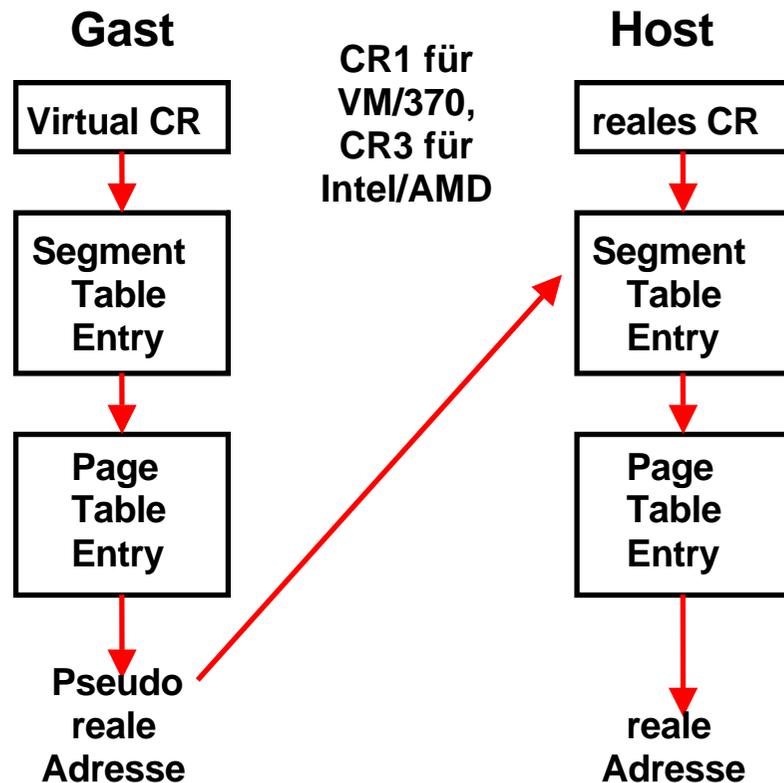
Der Vanderpool Virtual Maschine Monitor (VMM) ist eine Erweiterung zu einem vorhandenen Betriebssystem Kernel, z.B. dem Windows XP Kernel. Beide zusammen bilden den Hypervisor. Anwendungen können deshalb wahlweise im Virtualisierungs-modus unter einem Gastbetriebssystem oder nicht virtualisiert unmittelbar unter dem Host Betriebssystem laufen



Mögliche Vanderpool Konfiguration

Der Rechner befindet sich entweder im Virtualisierungsmodus oder auch nicht. Im Virtualisierungsmodus ist er in der Lage „Virtual Machine Execution“ (VMX) Operationen durchzuführen und mit Gast Maschinen zu arbeiten. Ein VMXON Befehl versetzt den Rechner in den Virtualisierungsmodus und aktiviert den Virtual Machine Monitor; mit VMXOFF wird der Virtualisierungsmodus wieder verlassen.

existierende Virtual Machine Monitore wie VMware, XEN, Virtual PC usw. nutzen die Vanderpool Technologie. Damit entfallen die meisten bisherigen x86 Probleme.



AMD Pacifica Virtuelle Adressübersetzung

Die Übersetzung der virtuellen in die reale Adresse erfolgt typischerweise in Hardware durch einen Mealy Automaten mit horizontalen Microcode. Beim Einsatz der VM/370 Interpretive Execution Facility oder der AMD Pacifica Technologie erfolgt die Übersetzung im Gastmodus in zwei Schritten. AMD verwendet hierfür die Bezeichnung „Nested Page Tables (NPT)“.

Nicht virtualisiert arbeitet der Rechner wie bisher. Im Gastmodus wird durch die Pacifica Hardware zunächst die virtuelle Adresse mit Hilfe der Segment- und Seitentabellen des Gast-Kernels in eine pseudo-reale Adresse umgesetzt. Anschließend übersetzt die gleiche Hardware die pseudo-reale Adresse mit Hilfe von Segment- und Seitentabellen des Host-Kernels in echte reale Adressen.

Dies ist weniger aufwendig als das Arbeiten mit Shadow Tables. Der erhöhte Übersetzungsaufwand wird durch größere Adress-umsetzpuffer (DLAT) kompensiert.

Hypervisor für x86

Hypervisor (Virtual Machine Monitor) Software in Unterstützung von Intel VT-x oder AMD V ist von einer großen Anzahl von Herstellern verfügbar. Die wichtigsten sind:

- **Hyper-V - Windows Server Virtualization**
- **Microsoft Virtual Server — supports hardware assisted virtualization.**
- **Microsoft Windows Server 2008 .**
- **Oracle VM - uses the Xen hypervisor**
- **Sun xVM - xVM is based on Xen on x64**
- **VirtualBox supports both AMD-V and VT-x.[13]**
- **VMware Workstation — Workstation 5.5 supports Intel VT-x.[14]**
- **VMware Fusion - Virtualization product for Mac OS X for use on Macintosh systems equipped with Intel Core, Core 2 Duo and Xeon processors.**
- **VMware ESX Server - Enterprise Virtualization Server, needs hardware support when running x64 and x32 virtual machine simultaneously.**
- **VMware Server - Supports Intel VT-x and AMD-V.**
- **Xen — Xen 3.0.2 supports Intel VT-x and AMD-V.**

Harte Partitionierung

Virtuelle Partitionierung

Logische Partitionen (LPAR)

Kommentar

Jurassic Park mit Zukunft

Wenn schon als Techno-Dinosaurier verspottet, dann kann man seine Mainframe-Produkte auch gleich nach den Weltherrschern von einst benennen – meint IBM und wählt mit Raptor und T-Rex besonders furchteinflößende Vertreter. Big Blue kann sich das kecke Spiel mit dem eigenen Image leisten: Immer deutlicher wird, dass die Architektur der Zeit nicht hinterher hinkt, sondern ihr sogar voraus eilt. Bei der logischen Partitionierung etwa geben Experten ihr einen zehnjährigen Entwicklungsvorsprung, und der Workload-Manager, der Ressourcen ziel- und situationsabhängig nutzt, fällt bei Standard-Servern viel primitiver aus. Eben solche Techniken sind aber die Voraussetzung für das On-Demand-Computing – die Nutzung von Rechenleistung ganz nach Bedarf. Neidische Blicke auf die Privilegierten, die sich einen T-Rex leisten können, sind aber unnötig: Stets ist der Open-Systems-Tross durch die technologischen Breschen, welche die Rechner-Dinos schlugen, nachgefolgt. Frank-Michael Kieß

Bei der logischen
Partitionierung geben
Experten ihr (zSeries)
einen zehnjährigen
Entwicklungs-Vorsprung

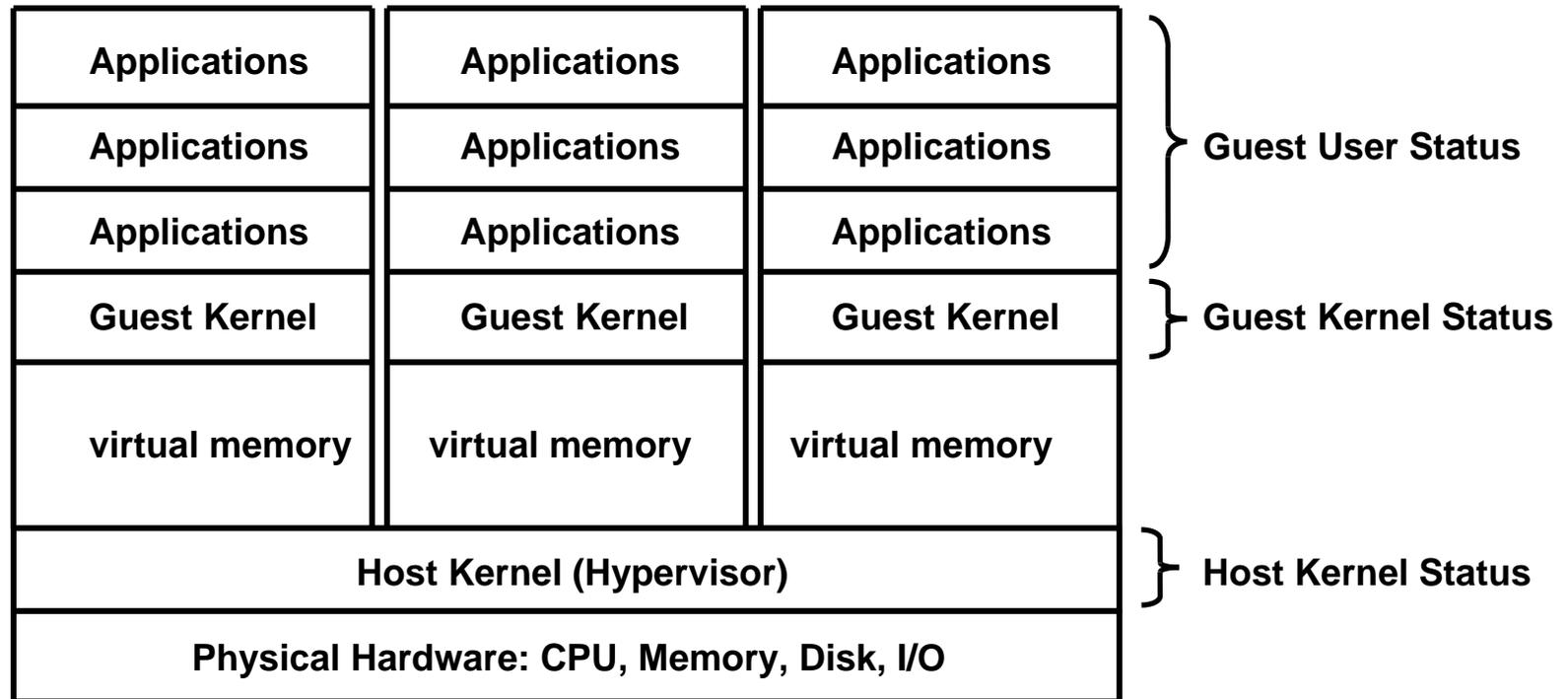
Computer Zeitung
19.5.2003, S. 1

Virtuelle Maschinen in realen Adressräumen

Logical Partition (LPAR) und PR/SM

Bei den virtuellen Maschinen wird das Gast-Betriebssystem mit seinen Anwendungen in einem virtuellen Adressraum des Host-Kernels abgebildet. Nach erfolgter virtueller Adressumsetzung teilen sich alle Gast-Systeme einen einzigen realen Adressraum. Die Adressumsetzung bedingt Shadow Tables oder eine doppelte Adressumsetzung wie im Fall von Pacifica und der z/VM Interpretive Execution Facility.

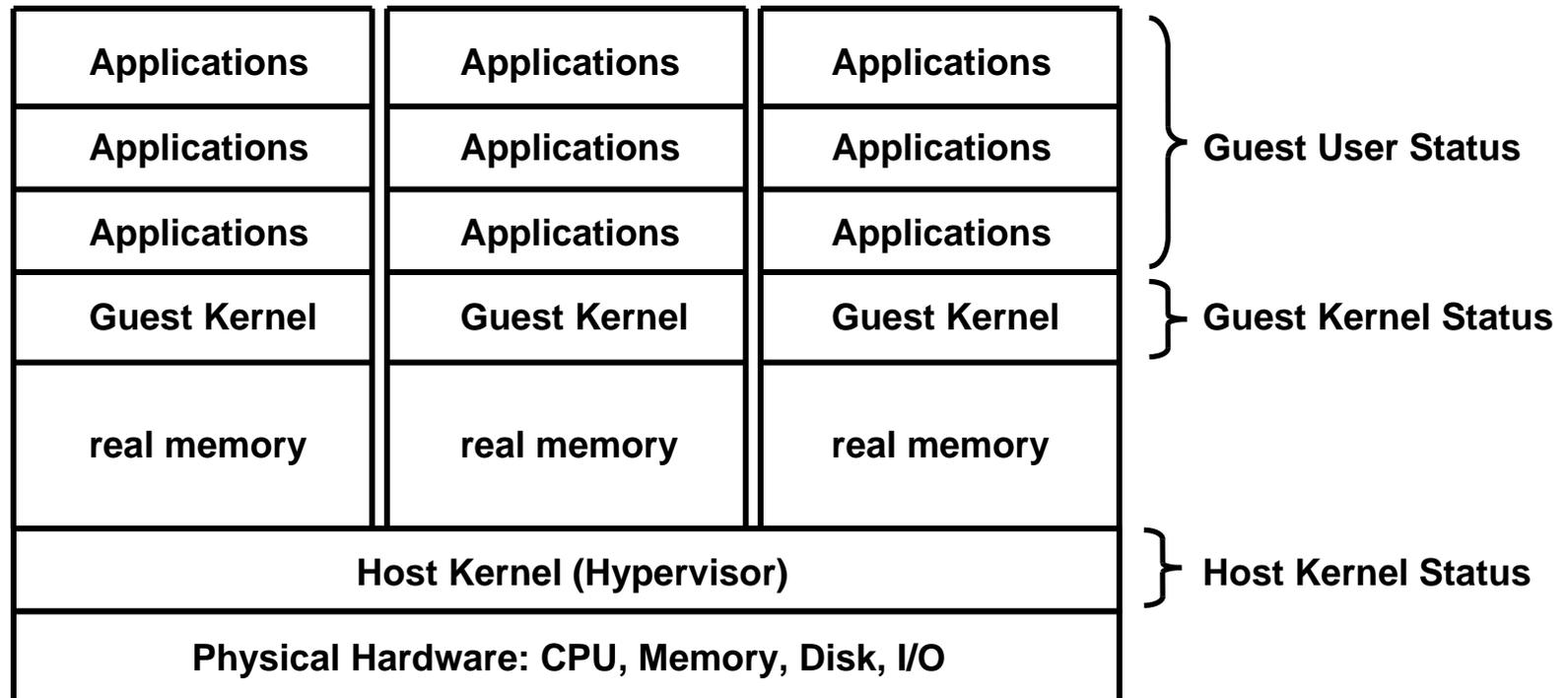
Die zSeries PR/SM (Processor Resource/System Manager) Einrichtung verwendet ein anderes, als LPAR (Logical Partition) bezeichnetes Verfahren. Für jedes Gastbetriebssystem wird eine eigene LPAR eingerichtet. Die PR/SM Hardware stellt mehrfache (bis zu 60) getrennte reale Speicher zur Verfügung. Die realen Speicher können in einem einzigen physischen Speicher abgebildet werden (Regelfall). Die Aufteilung des physischen Speichers auf die einzelnen LPARs erfolgt statisch, und wird bei der erstmaligen Inbetriebnahme des Systems festgelegt. Ein Host-Kernel (Hypervisor) läuft in einem weiteren getrennten Speicher. Er wird als Teil der Hardware betrachtet und führt Micro-Code (hier als Millicode bezeichnet) aus.



Bei der Virtualisierung unterhält der Host Kernel für jede Gast-Maschine einen virtuellen Adressenraum.

Innerhalb dieses Adressenraums führt die virtuelle Maschine ihrerseits eine Adressumsetzung virtueller in pseudo-reale Adressen durch.

Dies führt zu Komplikationen und Performance Overhead.

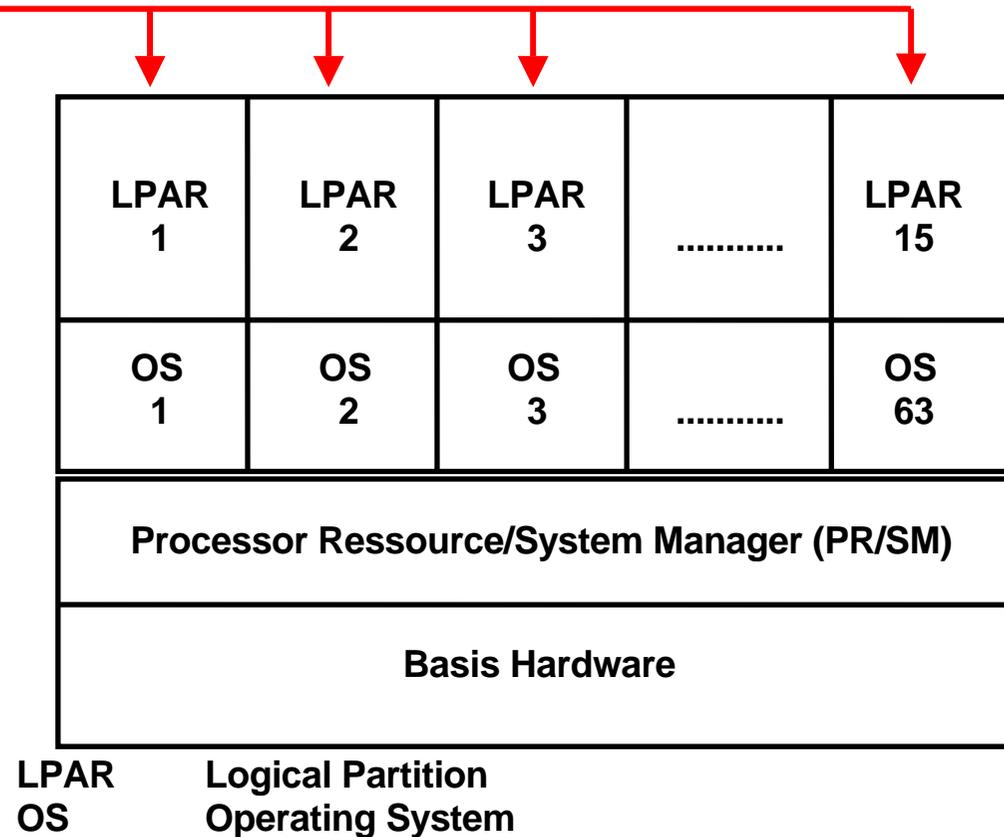


Bei den logischen Partitionen hat jede Gastmaschine einen eigenen realen Speicher. Auch der Host Kernel hat einen eigenen realen Speicher (Hardware System Area, HSA).

Dadurch entfällt die doppelte Adressumsetzung. Performance Unterschied Gast – Host nur noch $< 1,5\%$.

Benötigt werden zusätzliche Hardware-Einrichtungen für jede LPAR. Dies erklärt die Begrenzung auf 60 LPARs beim z10 System (42 000 virtuelle z/VM Maschinen).

Jede logische Partition hat ihren eigenen, getrennten, realen Speicher

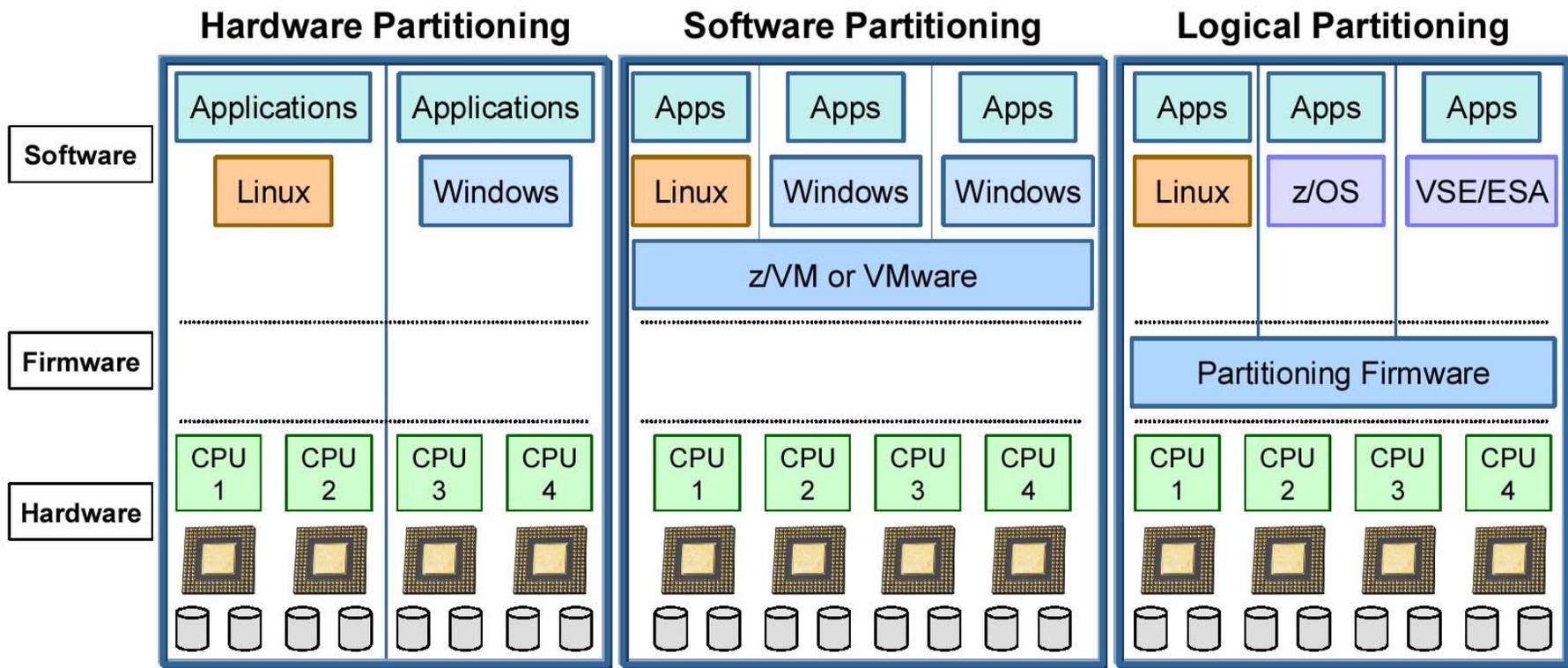


IBM PR/SM und LPAR

PR/SM ist eine Hardware Einrichtung (implementiert in LIC Code), welche die Aufgabe eines Hypervisors übernimmt, und die Partitionierung eines physischen Rechners in mehrere logische Rechner (LPAR. Logical Partition) ermöglicht. Jeder logische Rechner hat sein eigenes Betriebssystem, seinen eigenen unabhängigen realen Hauptspeicherbereich und seine eigenen Kanäle und Ein/Ausgabe-Geräte.

PR/SM schedules die einzelnen Betriebssysteme auf die einzelnen CPU's eines Symmetrischen Multiprozessors (SMP).

S/390 Rechner anderer Hersteller verfügen über vergleichbare Einrichtungen: Hitachi MLPF, Amdahl Multiple Domain Facility.



- BladeCenter
- xSeries

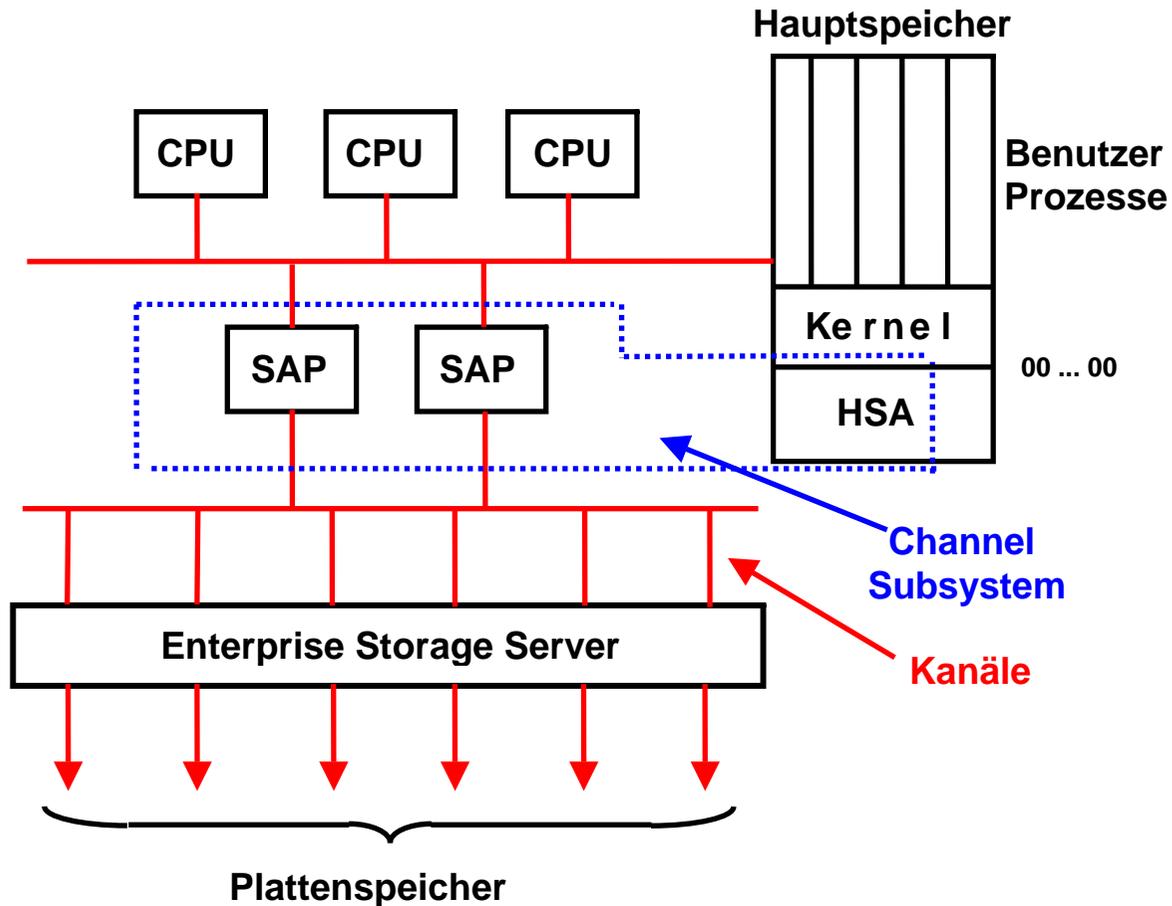
- z/VM on zSeries
- VMware and Microsoft Virtual Server on xSeries & BladeCenter

- LPAR on zSeries with Intelligent Resource Director
- LPAR on iSeries & pSeries

Hardware Partitioning ist unter anderem für die Großrechner der Firmen Sun und Hewlett Packard verfügbar. Software Partitioning (auch als Virtual Partitioning bezeichnet) arbeitet mit einem Hypervisor (Host Kernel) und nutzt im Prinzip keine zusätzliche Hardware Unterstützung (Ausnahme Interpretive Execution Facility, Vanderpool und Pacifica).

Logical Partitioning benutzt Firmware (Microcode) für die Implementierung der Hypervisor Funktionalität.

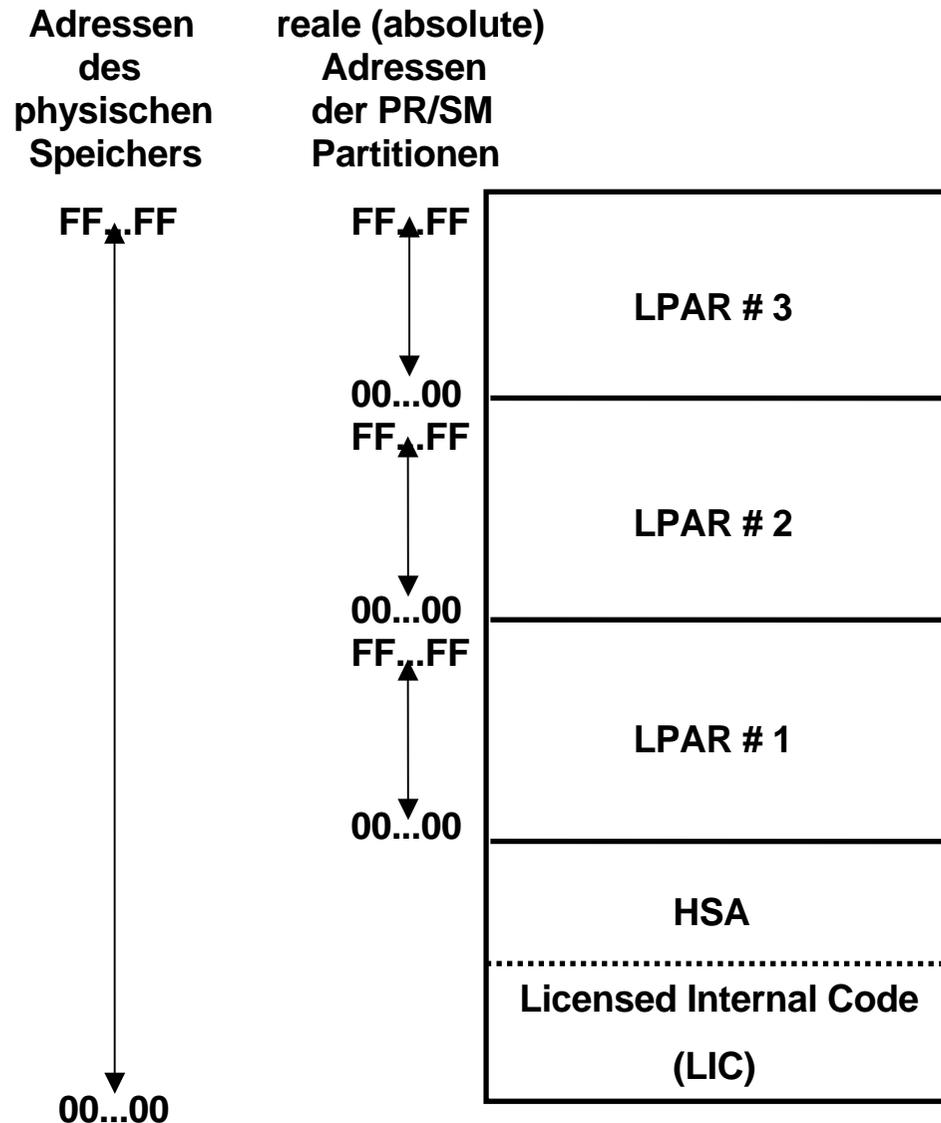
zSeries Ein/Ausgabe Anschluss



Die *HSA* (Hardware System Area) ist ein Teil des System z Hauptspeichers. Sie liegt außerhalb des Adressraums, auf den die CPUs zugreifen können. Der z10 EC Rechner hat eine bis zu 16 GByte große HSA (typisch 3 – 5 GByte).

Das *Channel Subsystem* besteht aus SAP Prozessoren und Code in der HSA. Es bildet das virtuelle E/A Subsystem, mit dem der Betriebssystem Kernel glaubt zu arbeiten, auf die reale E/A Struktur ab.

Unabhängig von System- und Benutzercode sind damit umfangreiche Optimierungen der Plattenspeicherzugriffe möglich.



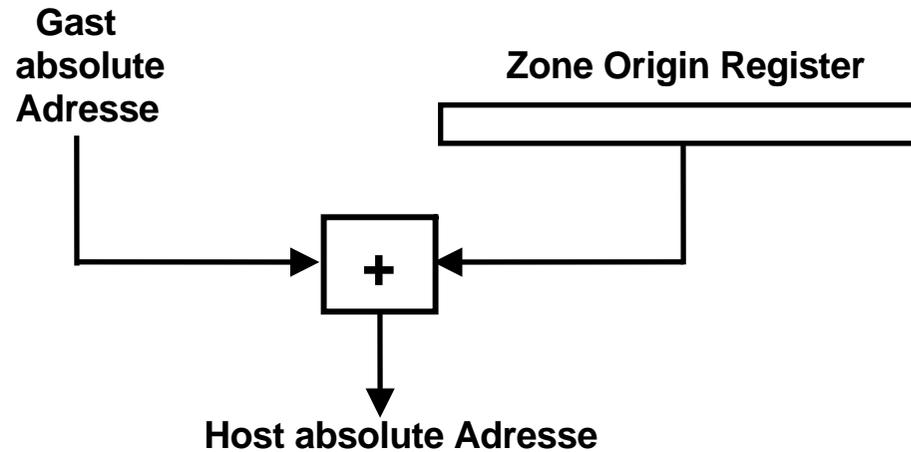
Es wäre denkbar, jeder LPAR einen getrennten physischen Hauptspeicher zuzuordnen. System z Rechner verwenden statt dessen einen einzigen physischen Hauptspeicher, der mittels spezieller Logik in einzelnen Partitionen aufgeteilt wird. Jede LPAR erhält eine eigene Hauptspeicher-Partition, deren kontinuierlicher Adressenbereich mit der Adresse 00 .. 00 anfängt.

In der HSA ist Licensed Internal Code (LIC) untergebracht, der unter anderem den PR/SM Hypervisor Code enthält. Die HSA enthält weiterhin Code für die I/O Ablaufsteuerung, dazugehörige Datenstrukturen sowie Code für umfangreiche Diagnostische und Fehlerbehandlungsfunktionen

Anmerkung: In der zSeries Architektur besteht ein (kleiner) Unterschied zwischen realen und absoluten Adressen.

zSeries Principles of Operations, IBM Form No. SA22-7832-00, Abb. 3-15)

Aufteilung des physischen Speichers in mehrere reale Speicher

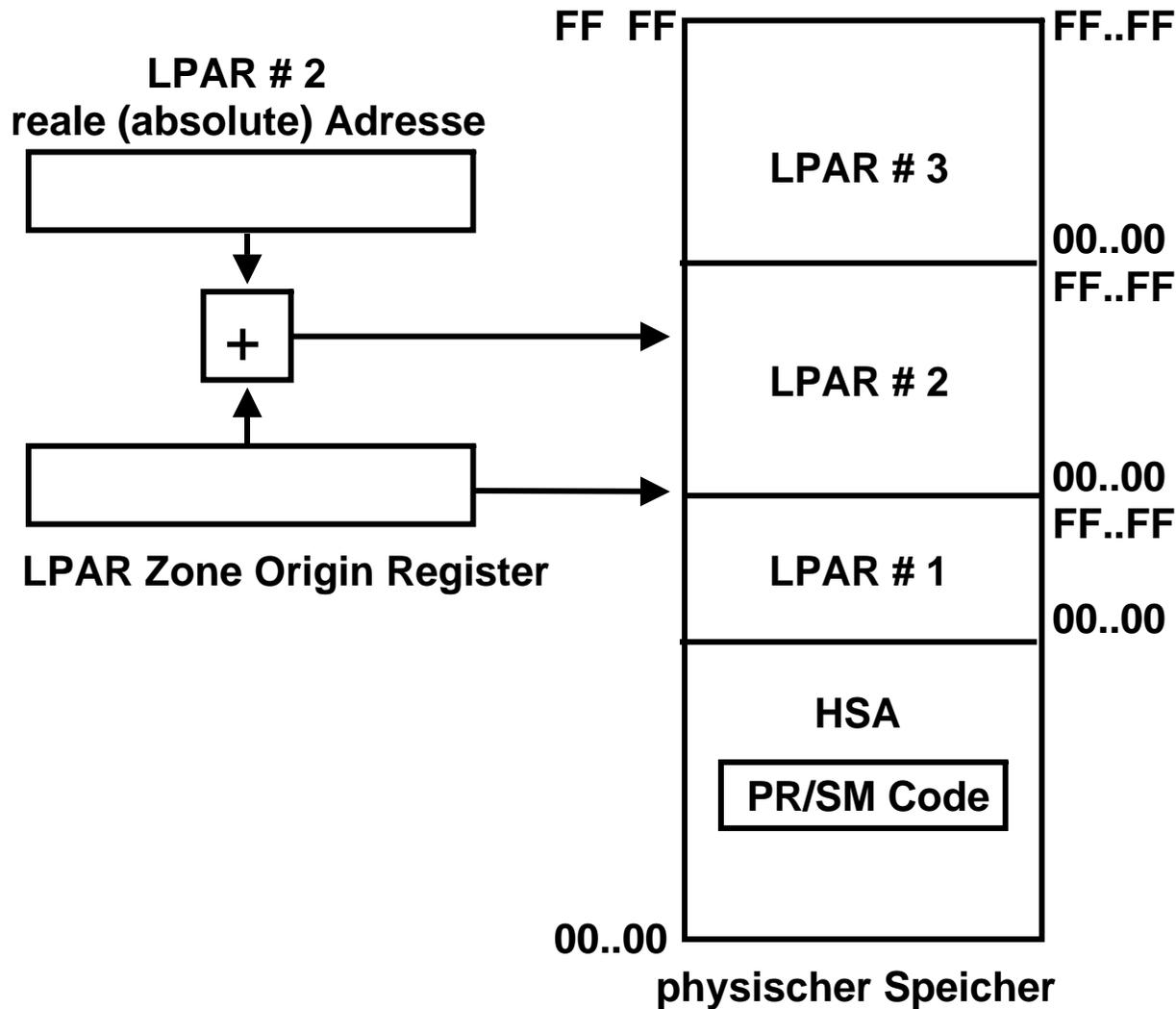


Umsetzung der Gast Adressen in physische Adressen

In einem Mehrrechnersystem sind im einfachsten Fall jeder LPAR eine oder mehrere CPUs fest zugeordnet. Jede CPU verfügt über ein „Zone Origin“ Register, in das bei der erstmaligen Konfiguration des Systems der korrekte Adressenwert geladen wird..

Ein weiteres „Zone Limit“ Register in jeder CPU stellt sicher, dass der Gast innerhalb des ihm zugewiesenen physischen Adressenbereiches bleibt.

Es ist möglich, die CPUs den einzelnen LPARs dynamisch zuzuordnen. In diesem Fall ist es die Aufgabe des PR/SM Hypervisors sicherzustellen, dass die korrekten Adressenwerte in die Zone Origin und Zone Limit Register geladen werden. PR/SM verfügt ähnlich wie ein Betriebssystem Kernel über eine Zeitscheibensteuerung. Damit ist es möglich, auf einem Rechner mit nur einer einzigen CPU mehrere LPARs zu betreiben.

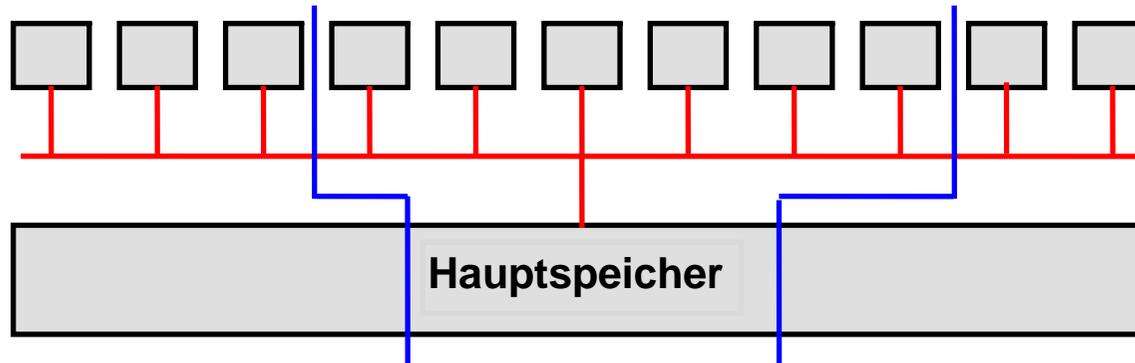


Jede CPU hat ein LPAR Relocate Register, als LPAR Zone Origin Register bezeichnet.. Hier ist die Anfangsadresse des Bereiches im physischen Speicher enthalten, der dieser LPAR zugeordnet ist.

Zusätzlich existiert ein Zone Limit Register, welches die maximale reale Hauptspeicheradresse definiert, welche dieser LPAR zugeordnet ist.

Wenn die CPU einer anderen LPAR zugeordnet wird, wird das LPAR Zone Origin Register neu geladen.

LPAR Speicherplatz-Verwaltung Basis Fall



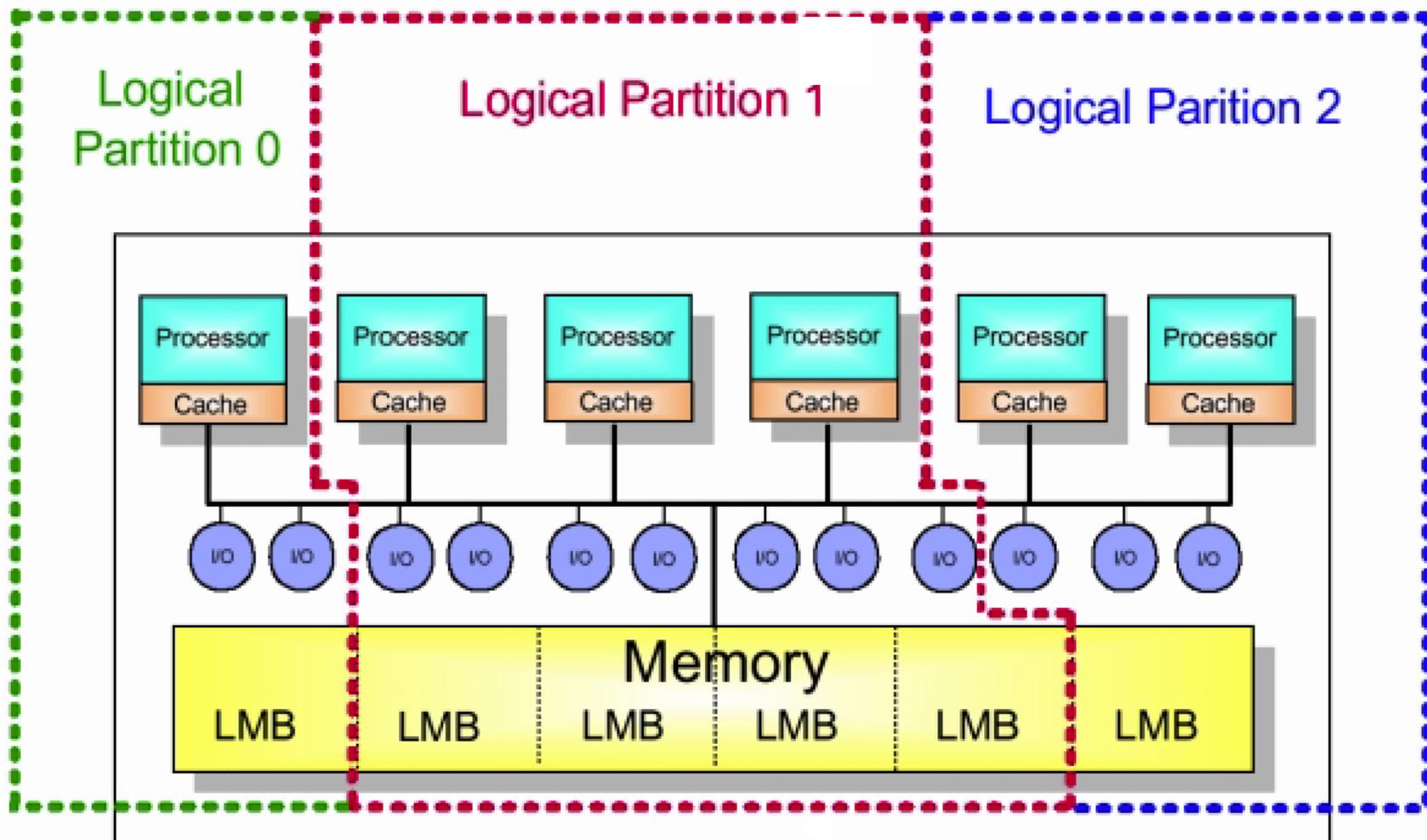
Aufteilung eines Großrechners in mehrere SMPs

z/OS unterstützt symmetrische Multiprozessoren (SMP) mit bis zu 64 CPUs. Realistisch sind bis zu 24 – 32 CPUs. Bei Unix, Linux und Windows Betriebssystemen liegt die Grenze für Transaktions- und Datenbank Anwendungen eher bei 12 CPUs.

Moderne Großrechner (Systeme) verfügen über wesentlich mehr CPUs. Sie werden deshalb in mehrere SMPs aufgeteilt, die über einen zentralen Switch miteinander kommunizieren.

Die CPUs werden im einfachsten Fall auf die LPARs aufgeteilt und diesen fest zugeordnet.

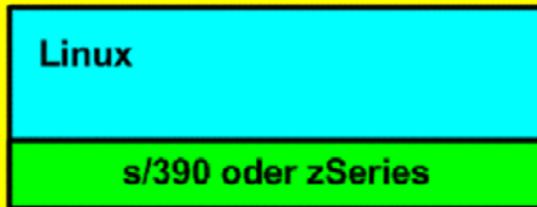
Der Systemadministrator kann den gesamten Hauptspeicher in unterschiedlichen Größen auf die einzelnen Hauptspeicher aufteilen.



Der vorhandene physische Hauptspeicher besteht aus 64 MByte großen „Logischen Memory Blöcken (LMB)“. Die feste Aufteilung des vorhandenen physischen Hauptspeichers erfolgt in LMB Einheiten.

Die vom Channel Subsystem verwalteten I/O Anschlüsse (Kanäle) und die damit verbundenen Platten- und Magnetbandspeicher werden den einzelnen LPARs fest zugeordnet. Das Gleiche gilt z.B. auch für Ethernet Anschlüsse. Hat ein Rechner z.B. 6 Ethernet Adapter, so kann man der ersten LPAR 4 Adapter, der zweiten LPAR 0 Adapter und der dritten LPAR 2 Adapter zuordnen.

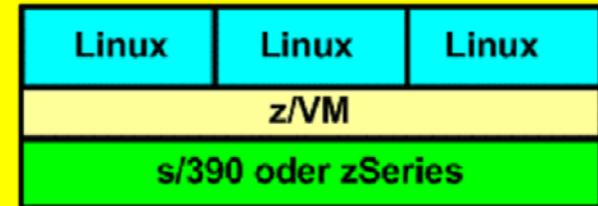
a) Hardware native



b) Partitions native



c) z/VM native



d) Partitions und z/VM mixed



Kombination LPAR und z/VM

Der Möglichkeiten a) und c) werden in der Praxis selten eingesetzt. Am häufigsten findet man die Varianten b) und d).

padme	lucas	tipc049		binks	leia
z/OS V. 1.5	OS/390 V. 2.7	zLinux Suse SLES 10	Expanded Storage 2 GByte	z/OS V. 1.8 DB2 V. 2.9	z/OS V. 1.8 WebSphere
LPAR #1 2 Gbyte	LPAR #2 2 Gbyte	z/VM LPAR #3 4GByte		LPAR #4 4GByte	LPAR #5 4GByte
PR/SM					

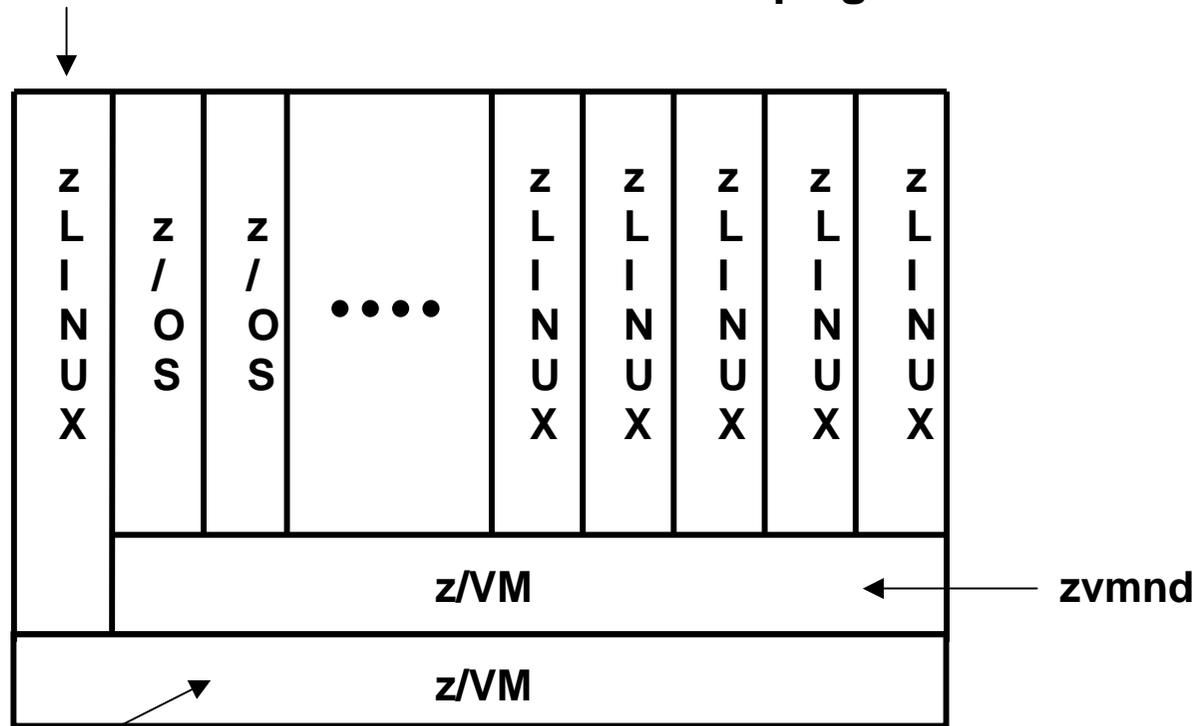
Konfiguration des Rechners jedi.informatik.uni-leipzig.

5 LPARs. je ein Ethernet Adapter für Anschluss an das Internet :

LPAR # 1 - 139.18.4.35 padme.informatik.uni-leipzig.de
 LPAR # 2 - 139.18.4.36 lucas.informatik.uni-leipzig.de
 LPAR # 3 - 139.18.4.49 tipc049.informatik.uni-leipzig.de
 LPAR # 4 - 139.18.4.49 binks.informatik.uni-leipzig.de
 LPAR # 5 – Datenbankpraktikum und Diplomarbeiten

z/OS 1.5
 OS/390 2.7
 zVM mit zLinux
 z/OS 1.8
 z/OS 1.8

Interner Router kenob.informatik.uni-leipzig.de = 139.18.4.37

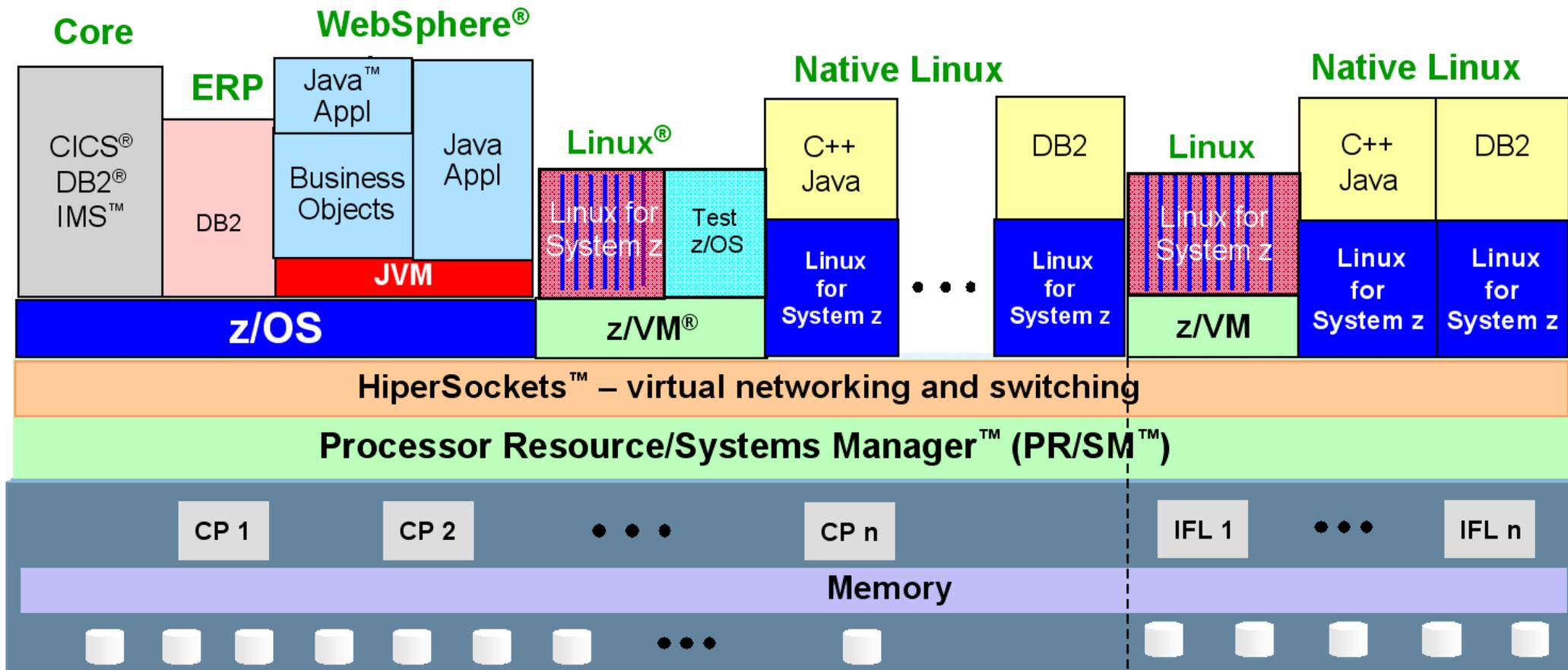


ticp049.informatik.uni-leipzig.de

kenob.informatik.uni-leipzig.de

Auf der LPAR # 3 - 139.18.4.49 laufen unter z/VM zwei Gast Maschinen. Auf der ersten Gastmaschine läuft zLinux. Dieses dient als Router für alle Gastmaschinen, die unter dem z/VM Host laufen. Auf der zweiten Gastmaschine läuft nochmals z/VM. Wir haben also einen z/VM Gast, der auf einem z/VM Host läuft.

Unter z/VM Gast sind etwa 20 virtuelle Maschinen installiert, teils z/OS 1.8, teils SLES v10 (Suse zLinux Enterprise Edition) . Diese sind an ein internes Netz 10.x.x.x angeschlossen, welches der zLinux Router unterhält.



- Up to 60 logical partitions on PR/SM; 100's to 1000's of virtual servers on z/VM
- Virtual networking for memory-speed communication, as well as virtual layer 2 and layer 3 networks supported by z/VM