Einführung in z/OS

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

WS 2008/2009

Teil 7

Virtuelle Maschinen, Partitionierung

Partitionierung

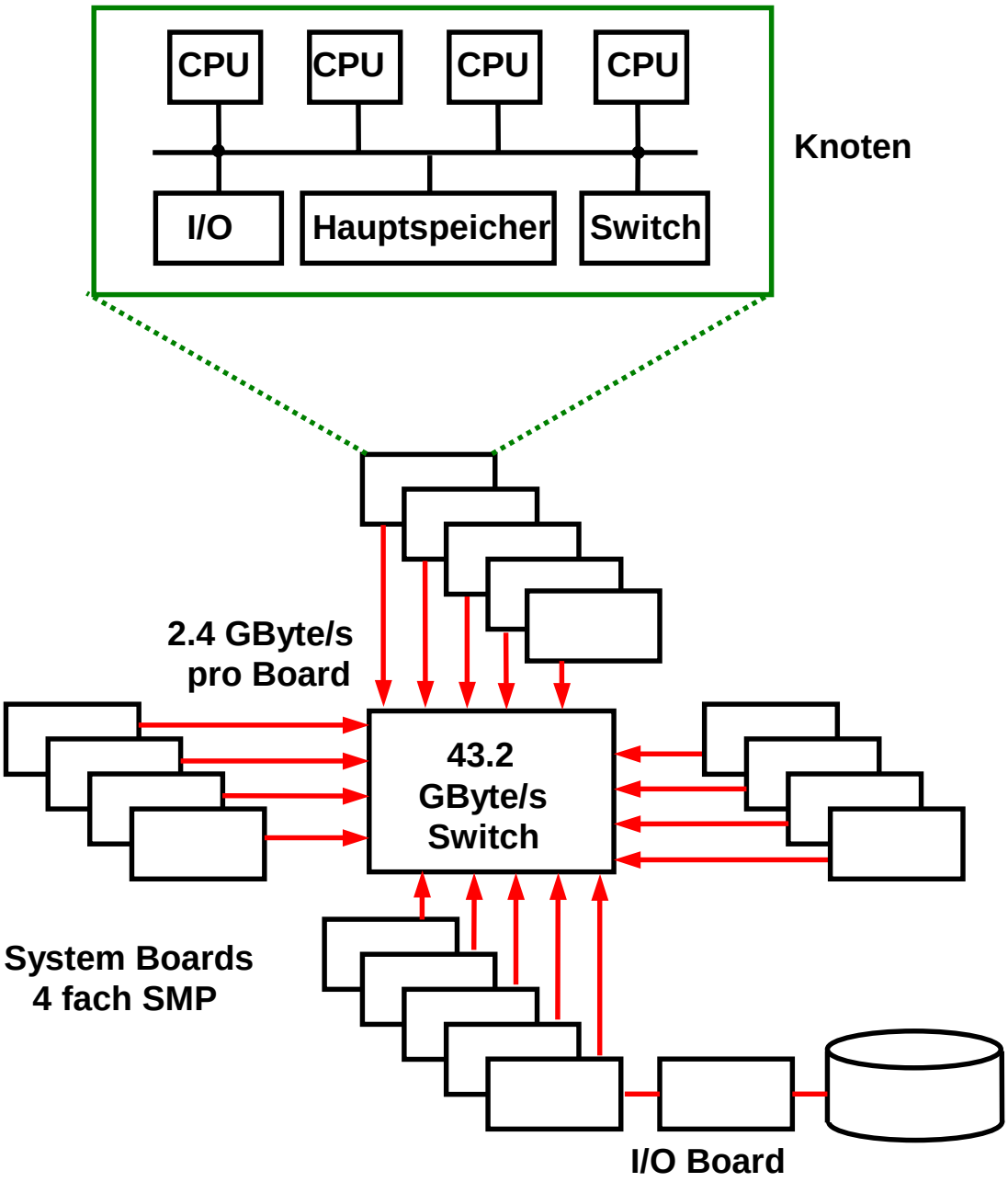
Virtuelle Maschinen

Literatur:

Joachim von Buttlar, Wilhelm G. Spruth:
Virtuelle Maschinen. zSeries und S/390 Partitionierung.
IFE - Informatik Forschung und Entwicklung, Heft 1/2004, Juli 2004

download unter

http://www-ti.informatik.uni-tuebingen.de/~spruth/publish.html

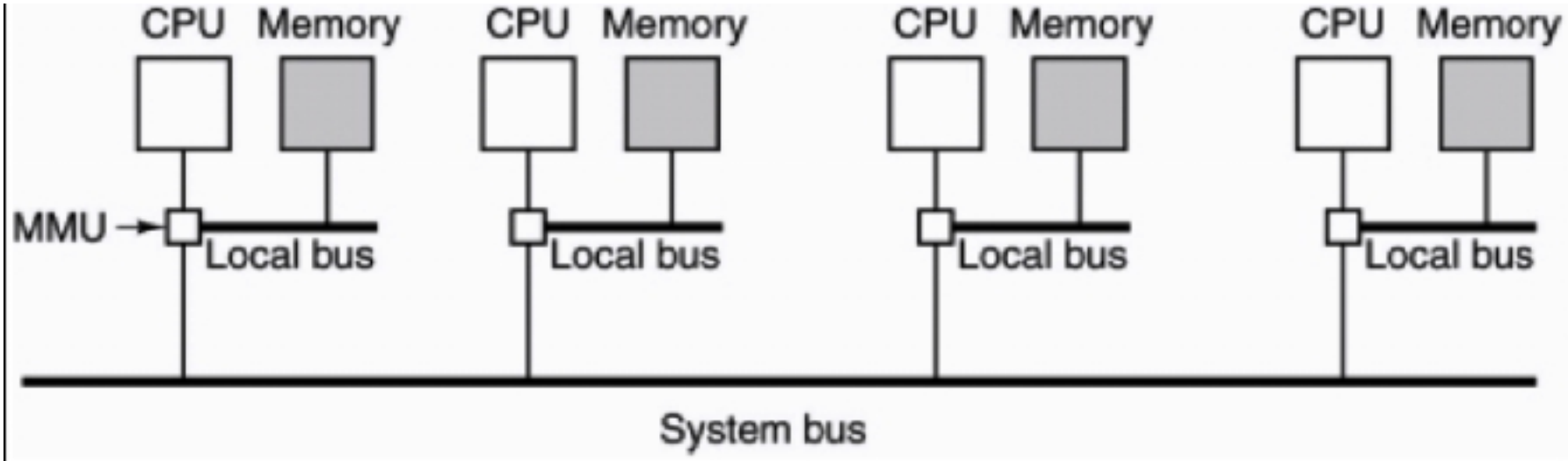


Sun E15K

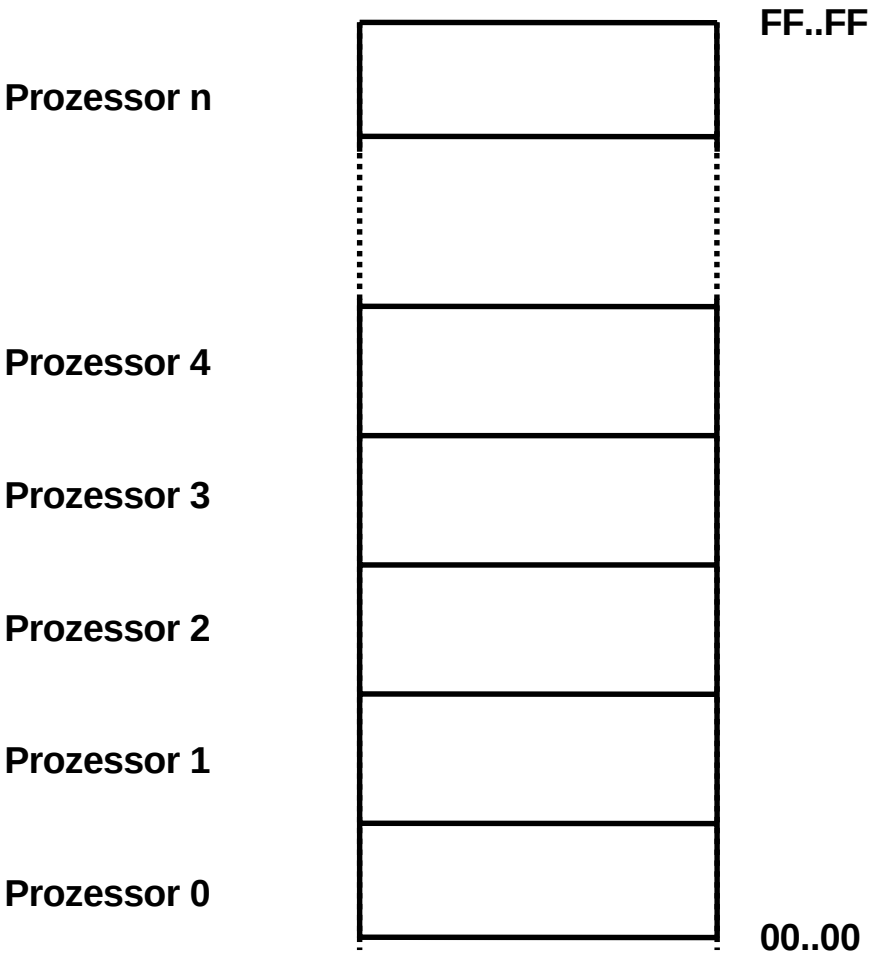
72 CPU´s

18 System Boards, je 4 CPU/System Board

I/O Controller auf jedem System Board



NUMA Rechner mit zwei Ebenen von Bussen

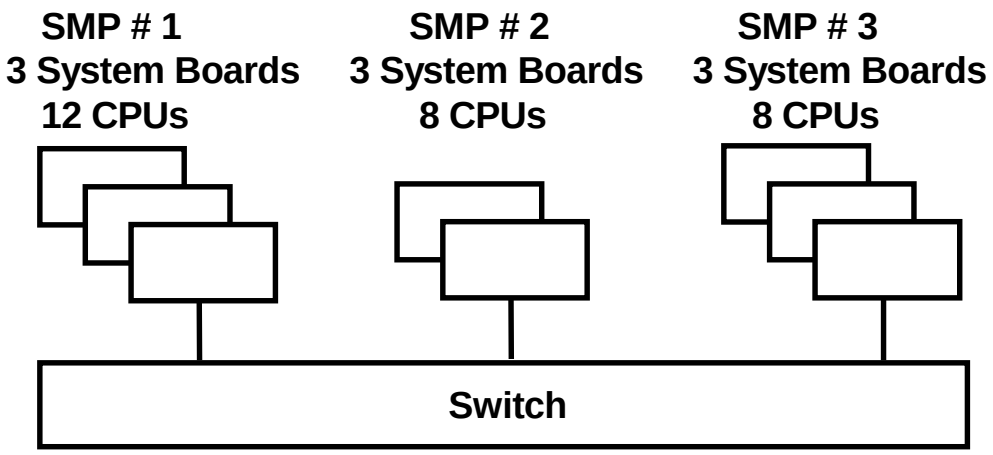


Non-uniform Memory Architecture
NUMA

Die Knoten eines Clusters haben jeweils einen eigenen lokalen Hauptspeicher.

Alle Hauptspeicher der Knoten bilden einen gemeinsamen realen Adressenraum. Jeder Knoten bildet automatisch einen Ausschnitt dieses Adressenraums auf die absoluten Adressen seines lokalen Hauptspeichers ab.

Harte Partitionierung



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

Harte Partition. 4 CPUs pro System Board

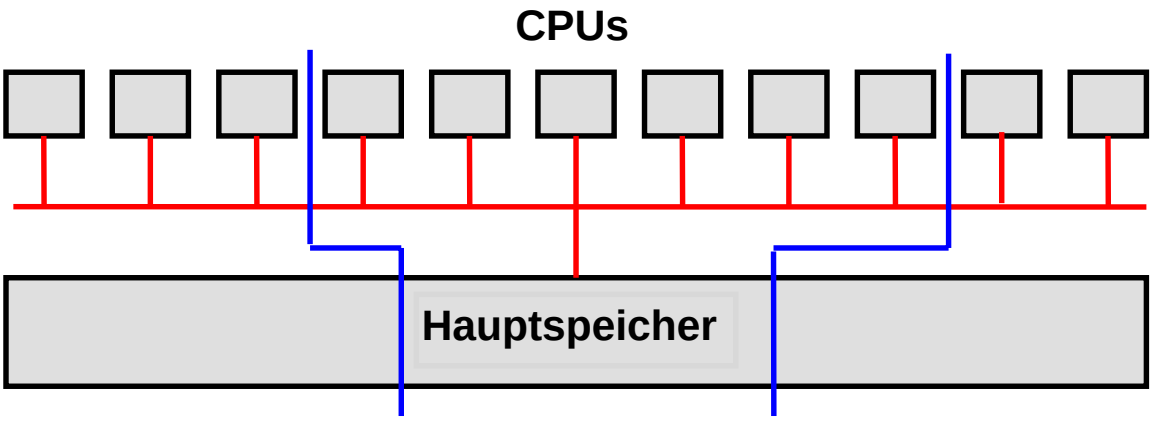
Jeder SMP hat ein eigenes Betriebssystem-

Aufteilung erfolgt statisch

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

Für Transaktionsanwendungen realistisch kaum mehr als 3 System Boards (12 CPUs) pro SMP

(Ausnahme: z/OS kann 24 – 32 CPUs in einem SMP betreiben)



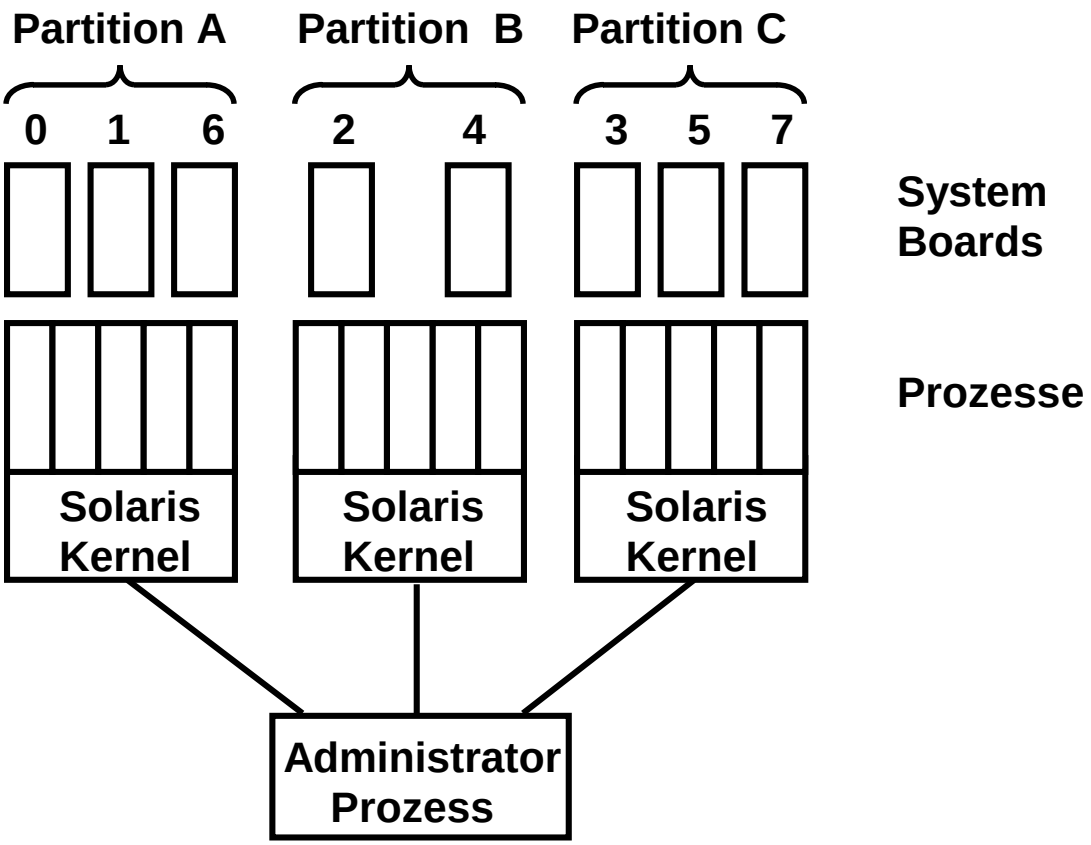
Aufteilung eines Großrechners in mehrere SMPs

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 CPUs.

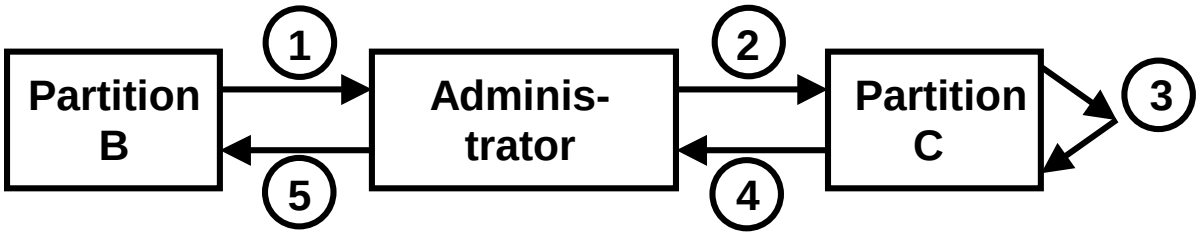
Moderne Großrechner (Systeme) verfügen über wesentlich mehr CPUs. Sie werden deshalb in mehrere SMPs aufgeteilt, die über einen zentralen Switch miteinander kommunizieren.

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

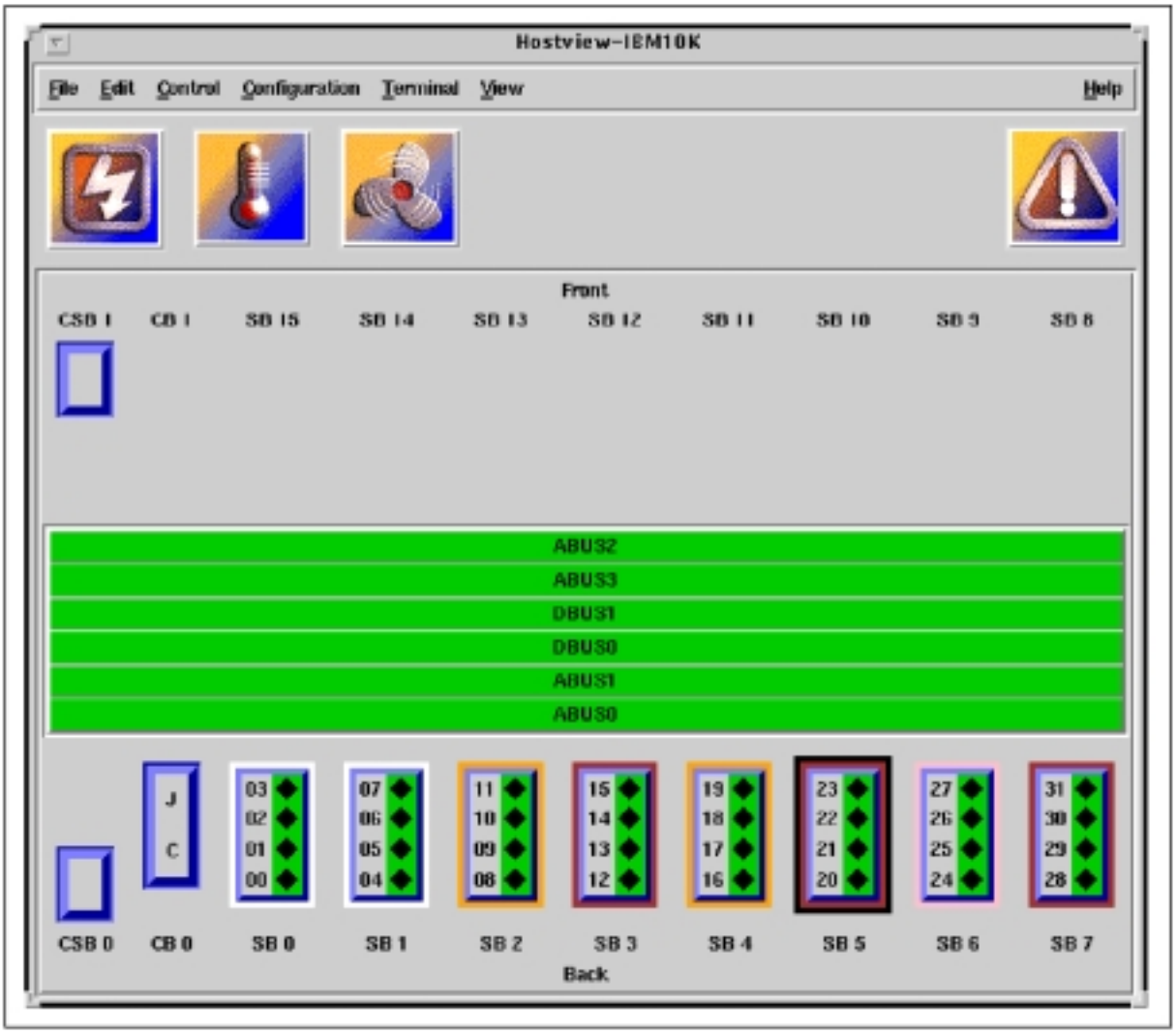
Bei den Sunfire und HP Superdome Rechnern ist die Granularität der SMPs jeweils 4, 8 oder 12 CPUs. zSeries und z/OS erlauben eine beliebig kleine Granularität



Solaris Cluster mit 8 System Boards, 32 CPUs



1. Partition B braucht mehr CPU Leistung
2. Administrator entscheidet, dass Partition C ein System Board freigeben muss
3. Partition C beendet alle Prozesse, die dieses System Board benutzen
4. Das ausgewählte System Board steht jetzt einer anderen Partition zur Verfügung
5. Administrator ordnet freies System Board der Partition B zu



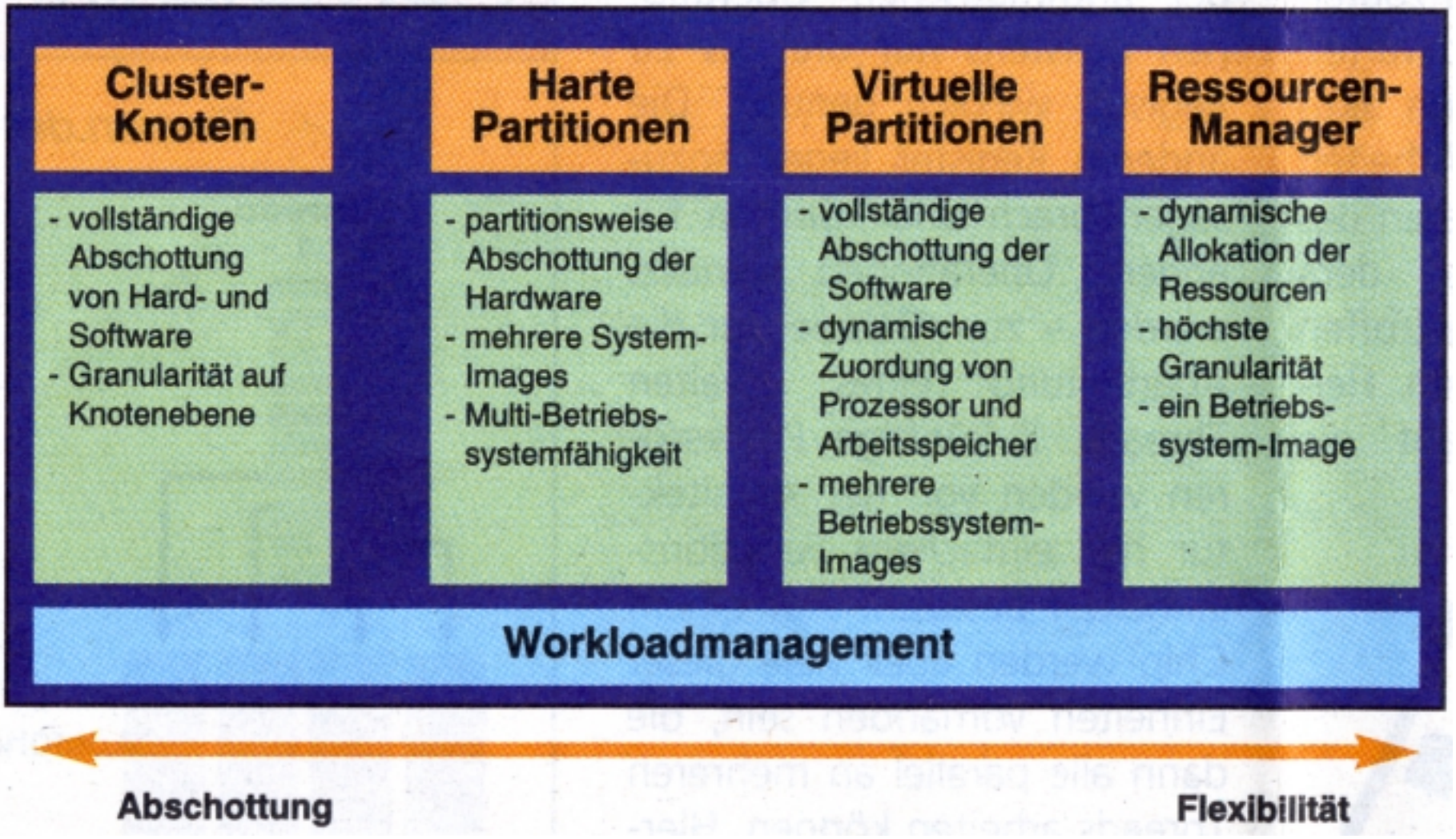
Sun E 10 000 Administrator Konsole

Dargestellt sind 8 Prozessor Boards SB0 .. SB7.

Cluster: SB0, SB1, SB6
SP2, SB4
SP3, SB5, SB7

Virtuelle Partitionierung

Partitionskonzepte im Vergleich



Virtuelle Partitionierung

Andere Bezeichnung: Logische 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.

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 benötig, wenn in der virtuellen Maschine Anwendungen laufen.

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

Emulator

Auf einem Rechner mit der Hardware-Architektur x (Host) wird ein Rechner mit der Hardware-Architektur y (Gast) emuliert.

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 emuliert einen Intel/AMD Windows Rechner auf einem Apple MAC PowerPC Rechner.**

Bochs **ist ein in C++ geschriebener Open Source Emulator, der die Intel/AMD Architektur auf vielen anderen Plattformen emuliert.**

Mehrere Gast Rechner auf einem Host Rechner sind möglich, aber nicht üblich.

Virtuelle Maschine

Auf einem Host Rechner mit der Hardware-Architektur x wird ein (oder mehrere) Gast Rechner der gleichen Architektur abgebildet.

Beispiele:

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

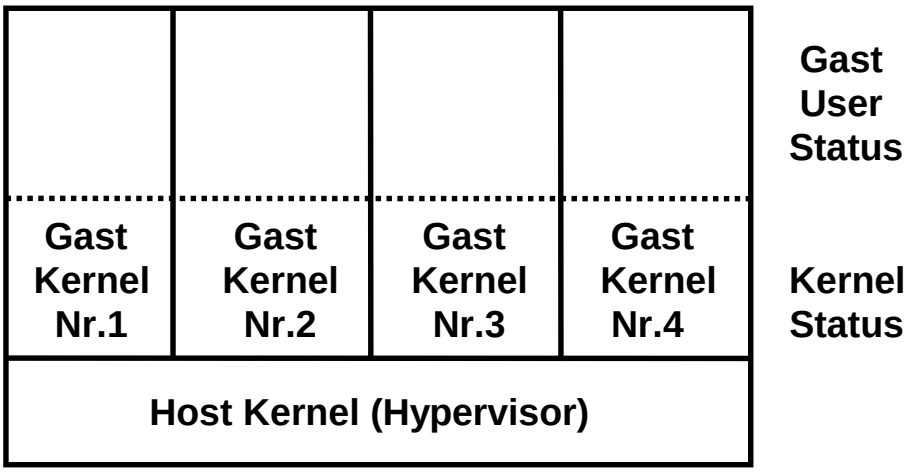
PowerPC LPAR Hypervisor

VMWare **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 IA-32 (Pentium) Architecture (VT-x)

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

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



Gleichzeitiger Betrieb mehrerer Betriebssysteme auf einem Rechner

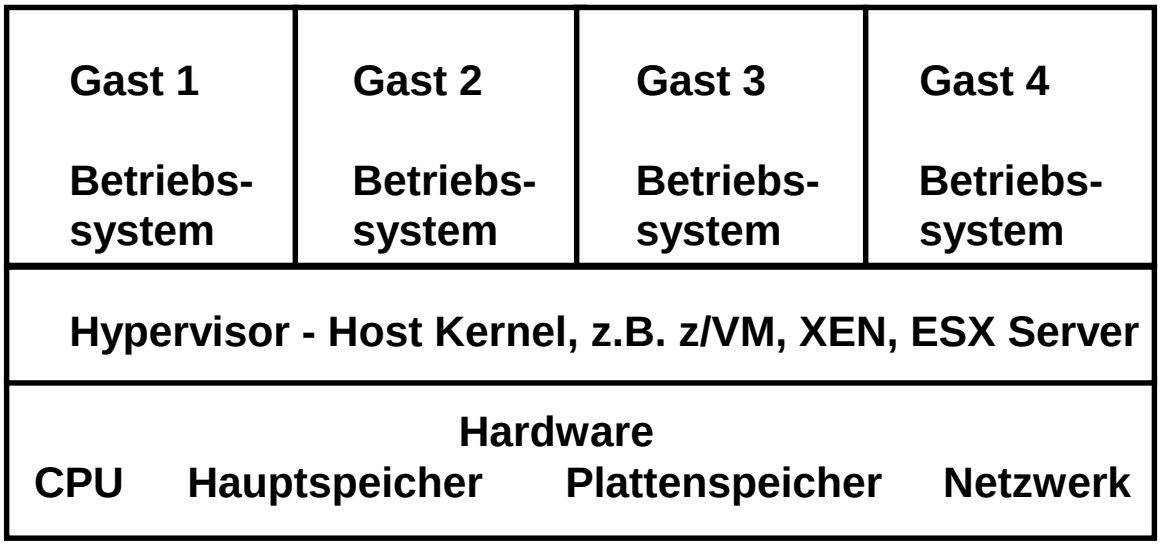
Ansatz: mehrere *Gast*-Betriebssysteme unter einem *Host*-Betriebssystem betreiben.

Dieser Vorgang wird als logische oder virtuelle Partitionierung bezeichnet

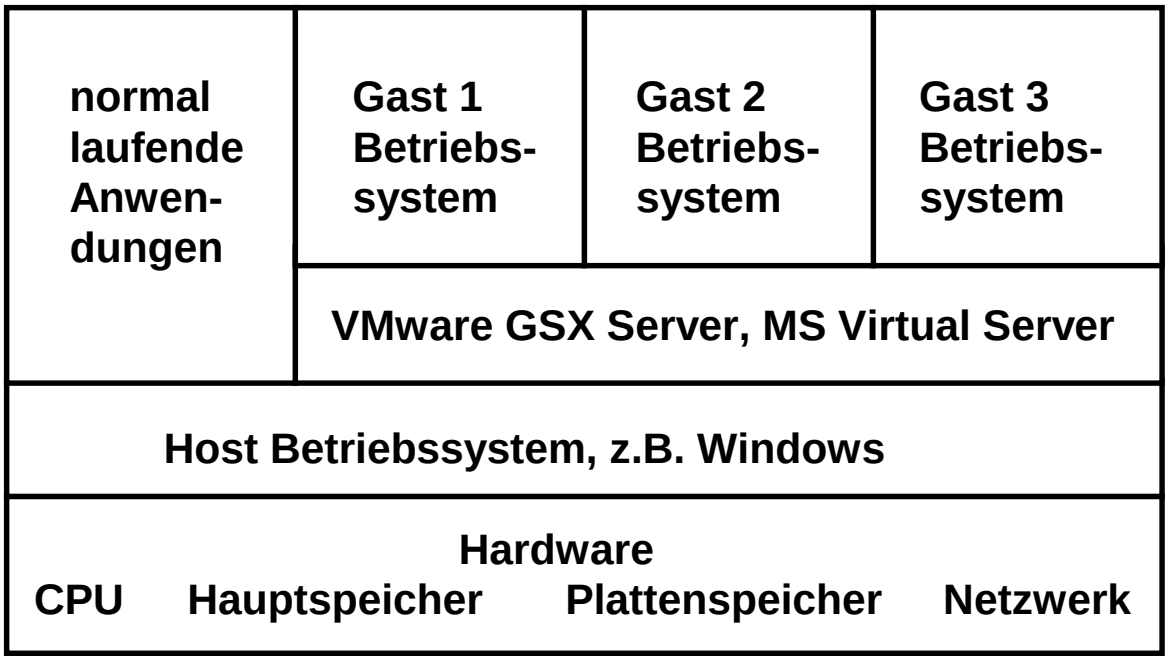
Das Host Betriebssystem verfügt über einen *Host Kernel* , auch als *Hypervisor* oder *Virtual Machine Monitor* bezeichnet.

Das Gast Betriebssystem verfügt über einen *Gast Kernel*

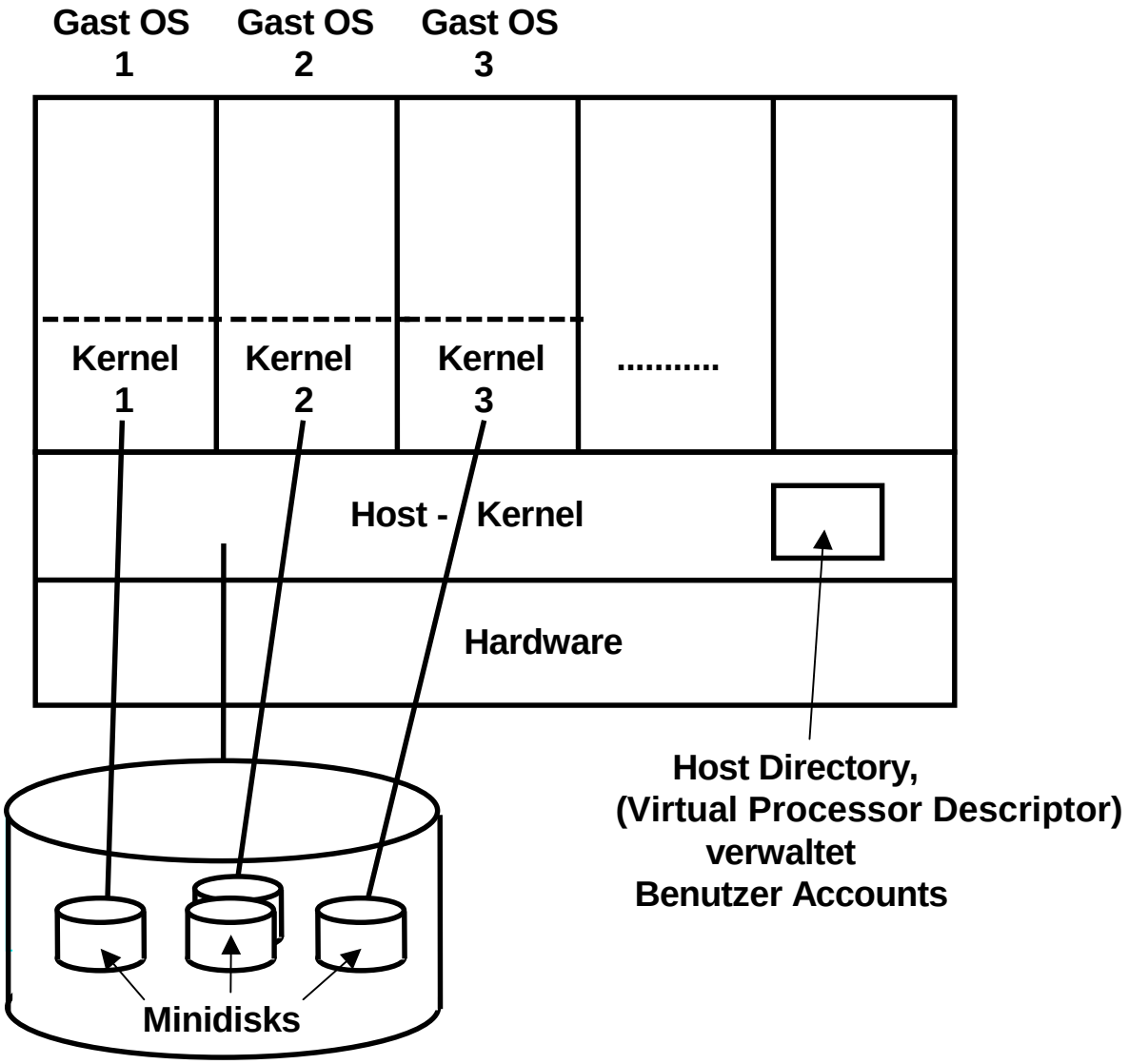
Typ A



Typ B



Alternativen für die Virtualisierung



Den Gast Betriebssystemen werden Ressourcen wie

- **CPU-Zeit,**
- **Aufteilung auf mehrere CPUs in einem Mehrfachrechner,**
- **Hauptspeicher,**
- **Ein-/Ausgabe-Geräte und -Anschlüsse**
  in der Regel fest zugeordnet.

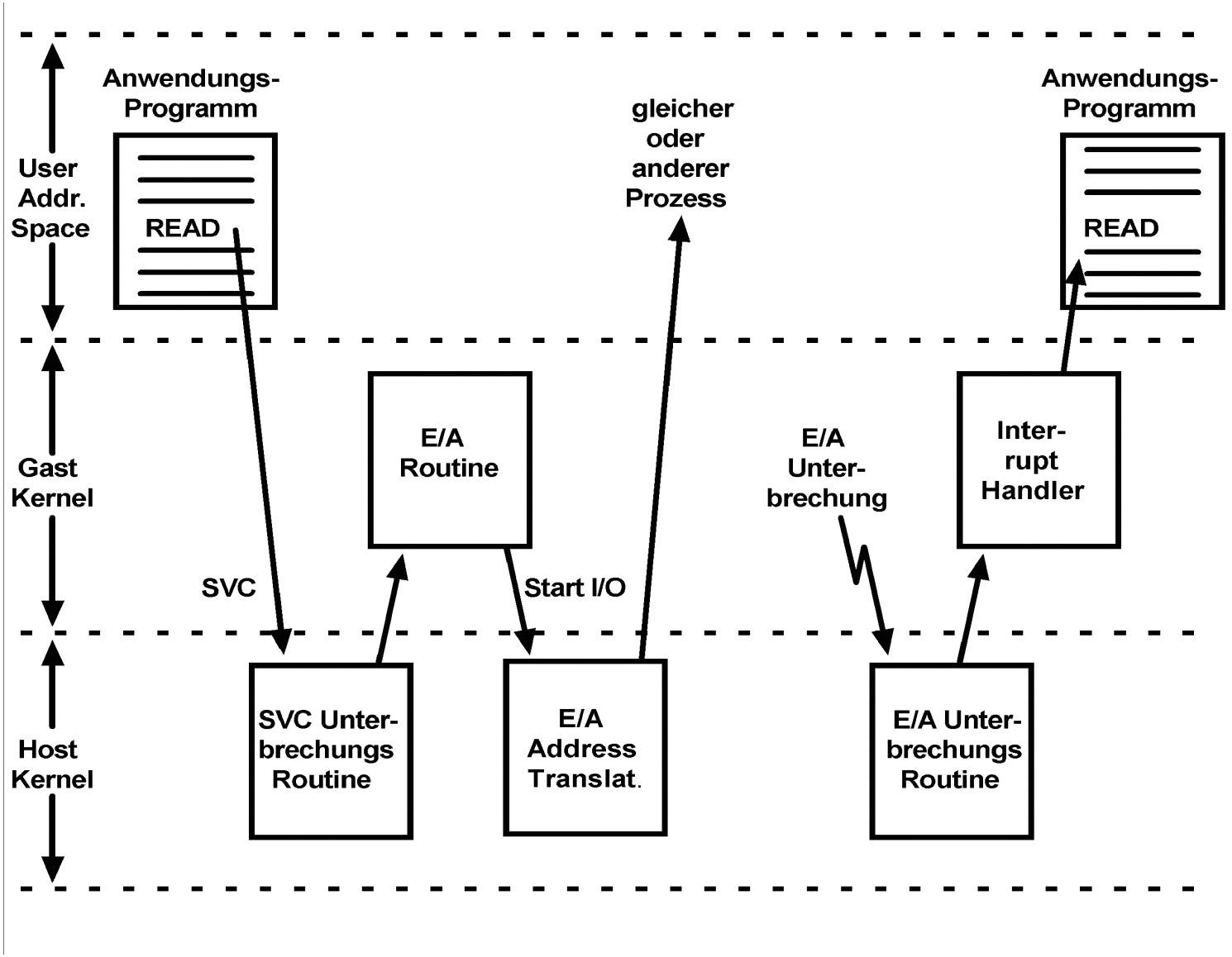
Steuerung der virtuellen Maschine

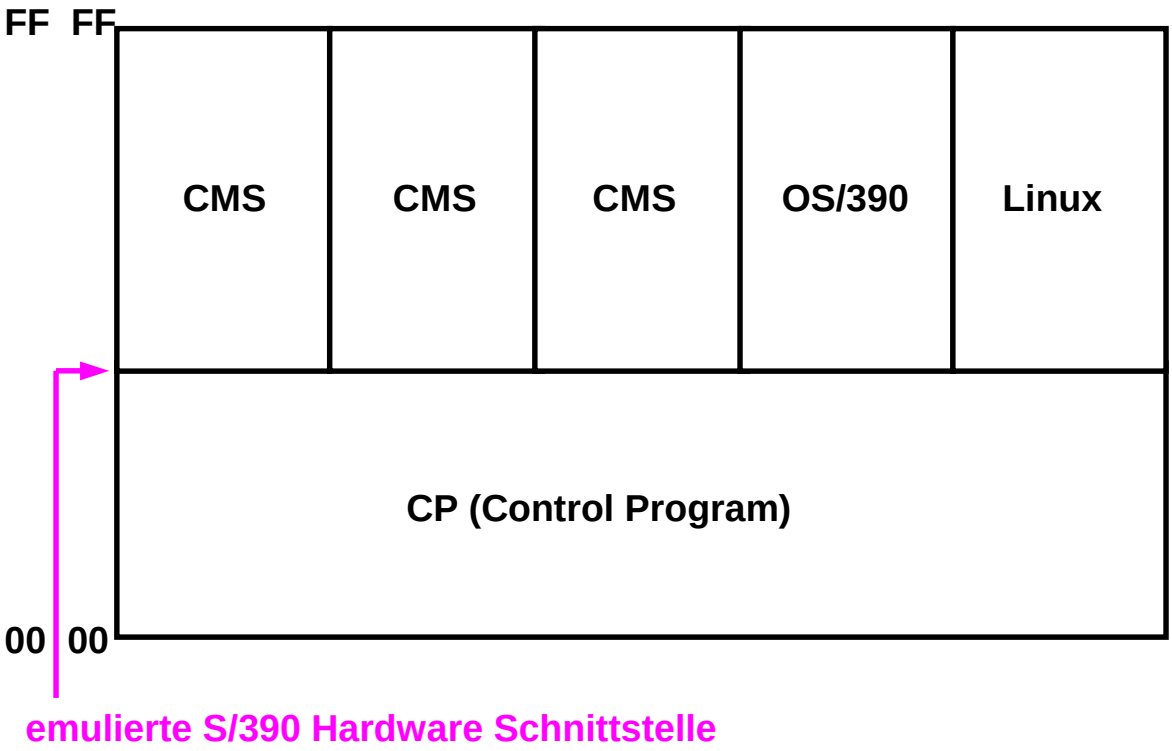
Alle Gast Maschinen laufen in einem eigenen virtuellen Adressenraum

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.

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


Anwendungs-
Programm
User
Addr.
Space
READ
gleicher
oder
anderer
Prozess
Anwendungs-
Programm
READ
Gast
Kernel
E/A
Routine
E/A
Unter-
brechung
Inter-
rupt
Handler
SVC
Start I/O
Host
Kernel
SVC Unter-
brechungs
Routine
E/A
Address
Translat.
E/A Unter-
brechungs
Routine



z/VM Betriebssystem

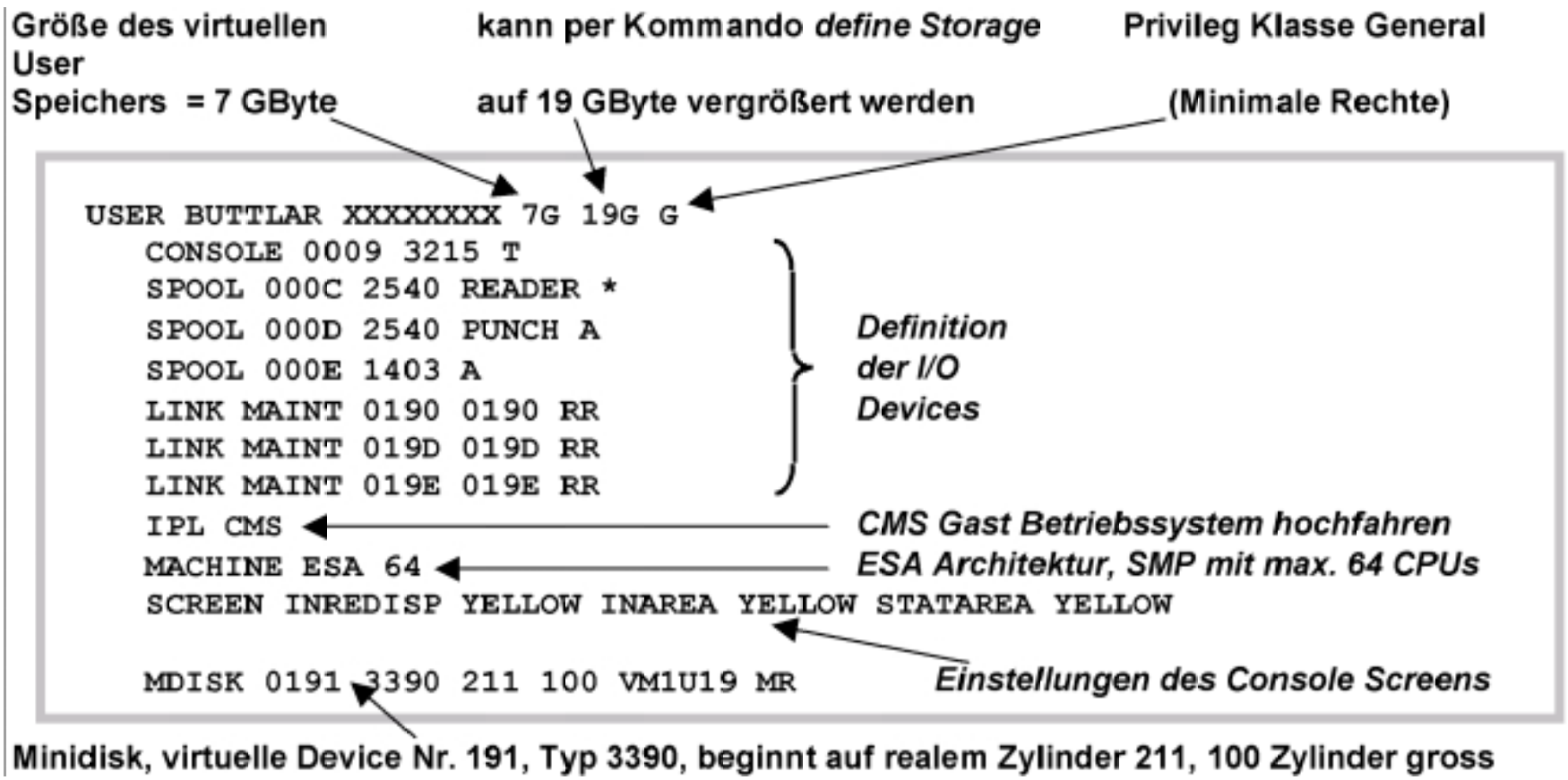
CP (Host Kernel – Hypervisor) läuft im Überwacherstatus.

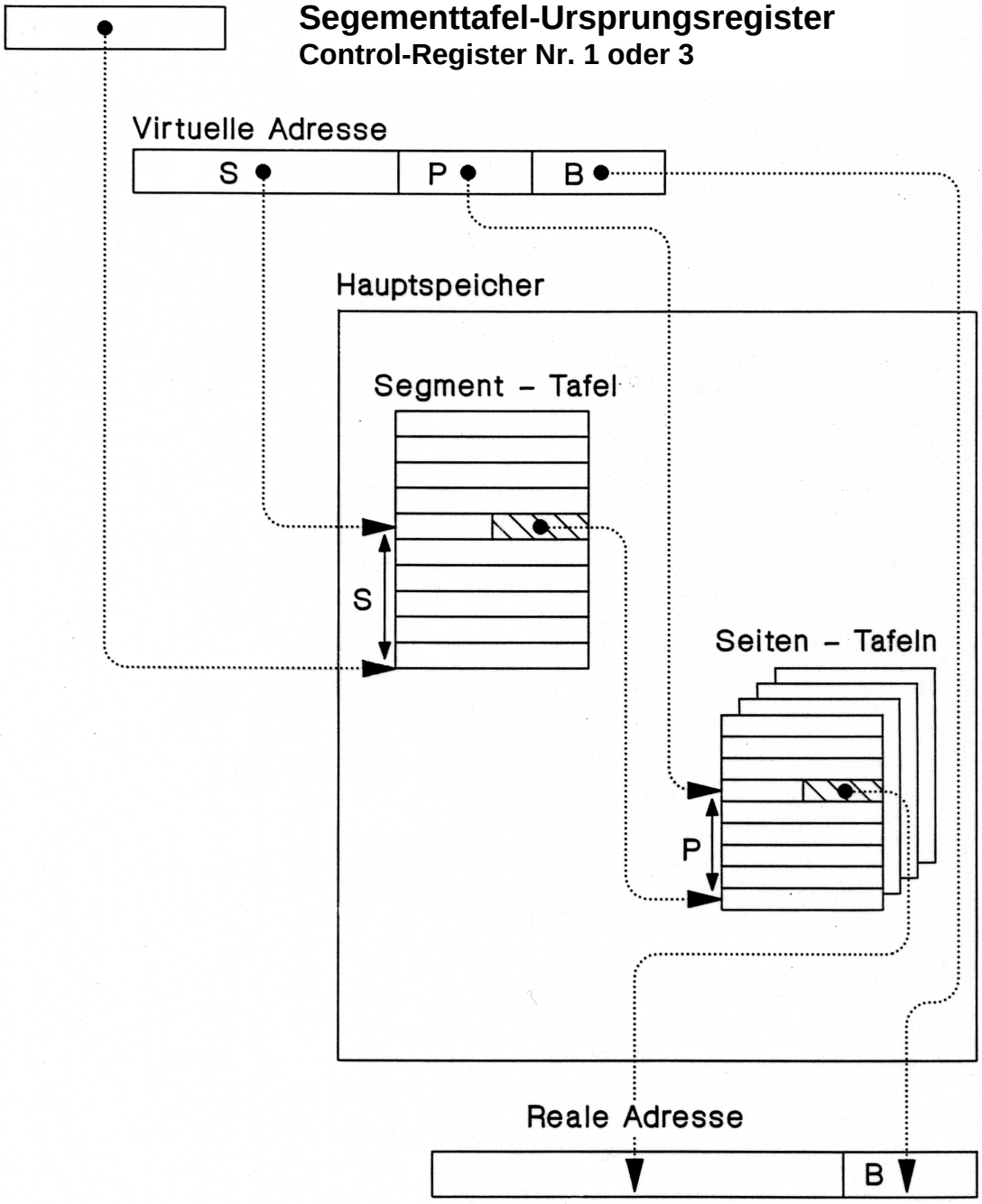
CMS und alle anderen Gast Betriebssysteme (einschließlich ihrer Kernel Funktionen) laufen im Problemstatus. Privilegierte Maschinenbefehle (z.B. E/A) werden von CP abgefangen und interpretativ abgearbeitet.

Volle S/390 Kompatibilität für alle Gast Betriebssysteme. Geringer Performance Verlust (< 5 %).

CMS (Conversational Monitor Program) ist ein besonders für die Software Entwicklung ausgelegtes Einzelplatz Betriebssystem. Für 1000 gleichzeitige CMS Benutzer werden 1000 CMS Instanzen angelegt. Ähnliches ist mit Linux/390 möglich.

Plattenspeicherplatz wird allen Gastbetriebssystemen in der Form virtueller „Minidisks“ statisch zugeordnet. Hauptspeicherplatz wird dynamisch verwaltet.

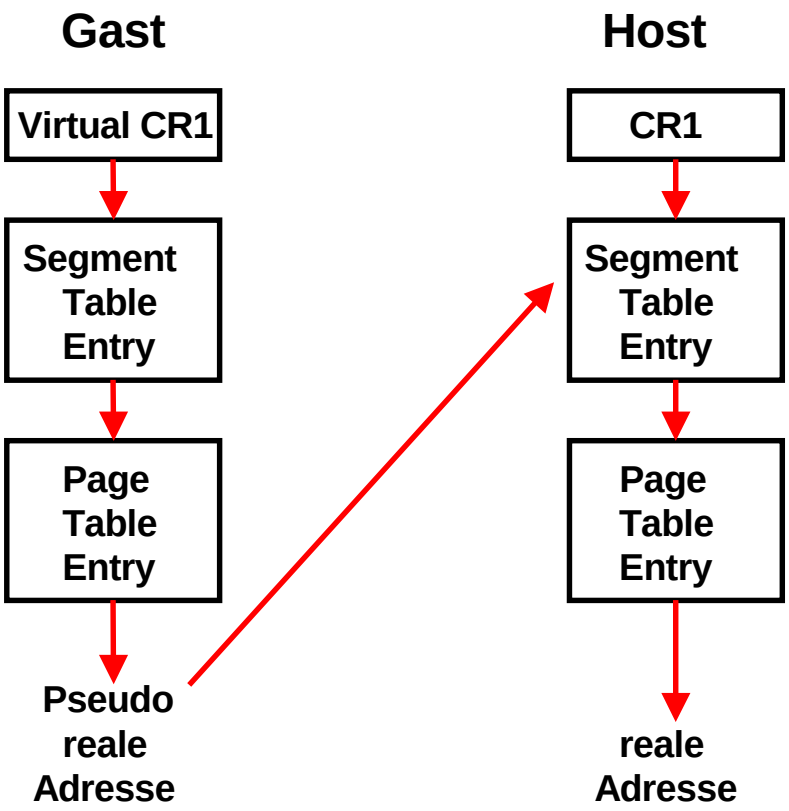

Größe des virtuellen User Speichers = 7 GByte
kann per Kommando define Storage auf 19 GByte vergrößert werden
Privileg Klasse General (Minimale Rechte)
USER BUTTLAR XXXXXXXX 7G 19G G
CONSOLE 0009 3215 T
SPOOL 000C 2540 READER *
SPOOL 000D 2540 PUNCH A
SPOOL 000E 1403 A
LINK MAINT 0190 0190 RR
LINK MAINT 019D 019D RR
LINK MAINT 019E 019E RR
Definition der I/O Devices
IPL CMS
CMS Gast Betriebssystem hochfahren
MACHINE ESA 64
ESA Architektur, SMP mit max. 64 CPUs
SCREEN INREDISP YELLOW INAREA YELLOW STATAREA YELLOW
Einstellungen des Console Screens
MDISK 0191 3390 211 100 VM1U19 MR
Minidisk, virtuelle Device Nr. 191, Typ 3390, beginnt auf realem Zylinder 211, 100 Zylinder gross



Virtuelle Adressumsetzung
Pentium, S/390

Adressumsetzung erfolgt bei jedem Hauptspeicherzugriff durch Hardware

Zur Leistungsverbesserung werden die gängigen Adressen in einem Adressumsetzpuffer gecached



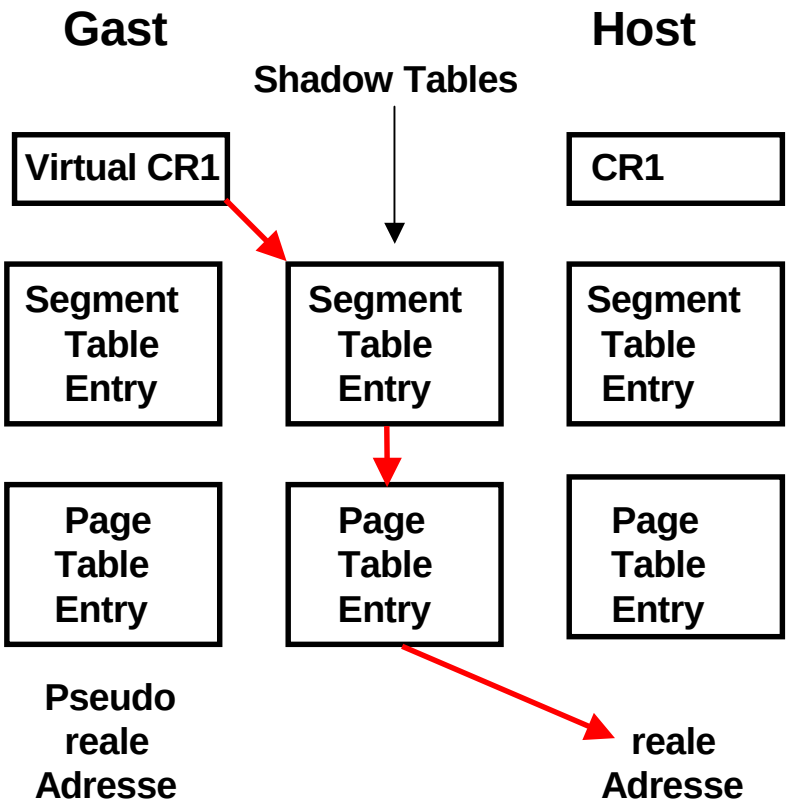
Problem derAdressübersetzung für die Virtuelle Maschine

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 Adressübersetzung 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.

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

Sensitive Maschinenbefehle (1)

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 IA32 Architektur

Im Vergleich zu VM/370 sind die VMware ESX und GSX Server benachteiligt, weil einige kritische Eigenschaften in der IA32 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 Kontrollregister 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 IA32 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.

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.

G.J. Popek, R.P. Goldberg: Formal Requirements for Virtualizable Third Generation Architectures. Comm. ACM, Vol. 17, Nr. 7, Juli 1974, S. 412-421.

http://www.cap-lore.com/CP.html

Paravirtualization.

Hierbei ist die Architektur der virtuellen Maschine nicht vollständig mit der Host Architektur identisch. Die Benutzerschnittstelle (Application Binary Interface, ABI) ist die gleiche. Der Gast-Kernel unterstellt jedoch Abweichungen zu der IA32 Architektur. Dies verbessert das Leistungsverhalten, erfordert aber Änderungen des Gast-Kernels. Hiervon ist nur ein sehr kleiner Teil des Kernel-Codes betroffen. Derzeitig existiert ein funktionsfähiger Linux-Port (XenoLinux), dessen ABI mit dem eines nicht-virtualisierten Linux 2.4 identisch ist. An Portierungen für Windows XP und BSD wird gearbeitet.

Ein ähnlicher Ansatz wird von *Denali verfolgt*. Als Gast-Betriebssystem dient *Ilwaco*, eine speziell an den Denali Hypervisor angepasste BSD Version. Denali unterstützt nicht das vollständige IA32 ABI. Es wurde für Netzwerk-Anwendungen entwickelt und unterstellt Einzelbenutzer-Anwendungen. Mehrfache virtuelle Adressräume sind unter Ilwaco nicht möglich.

VMware

VMware ist ein Produkt mit Virtual Machine Hypervisor Funktionalität für die IA32 Architektur und ist in zwei Versionen verfügbar: *VMware Workstation* und *ESX Server*.

VMware Workstation ist die ursprüngliche Version. Es handelt sich um ein als Host-OS bezeichnetes Anwendungsprogramm, das entweder unter Windows oder unter Linux auf einer IA32 Plattform im Benutzer-Modus läuft, und als Kombination die Funktionalität eines Host-Kernels bereitstellt. Eine zusätzliche, als *VMDriver* bezeichnete Komponente wird in Windows oder Linux integriert. Es bestehen Ähnlichkeiten mit der Art, mit der eine virtuelle Maschine in einem DOS-Fenster unter Windows abläuft. VMware Workstation nutzt die existierenden Ein-/Ausgabe Einrichtungen des Host-Betriebssystems. Wenn ein Gast-Betriebssystem eine Ein-/Ausgabe Operation ausführt, wird diese von VMDriver abgefangen und unter Benutzung geeigneter Systemaufrufe interpretiert. Ebenso werden Ein-/Ausgabe Unterbrechungen unter Beteiligung des VMDrivers bearbeitet.

Das Host-OS kann Seiten auslagern, die einer spezifischen virtuellen Maschine zugeordnet sind. Lediglich ein kleiner Satz an Seiten wird ständig im Hauptspeicher gehalten. Dies bedeutet, dass VMware Workstation vom Host-Betriebssystem wie ein normaler Benutzer-Prozess behandelt wird. Falls der Algorithmus des Hosts zur Seitenersetzung nicht optimal arbeitet, kann dies zu einer VMware Leistungsverschlechterung führen.

Die Nutzung von Host-Betriebssystem Komponenten verursacht einen Leistungsverlust, insbesondere bei Ein-/Ausgabe-Operationen. VMware stellt deshalb mit seinem ESX Server einen eigenen Host-Kernel zur Verfügung. Dieser benötigt keine Unterstützung durch das Host-Betriebssystem, läuft auf der realen Hardware und hat vergleichbare Funktionen wie der VM/370 Host-Kernel. Es werden bis zu 64 virtuelle Maschinen unterstützt. Gast-Betriebssysteme, die unter dem ESX Host–Kernel laufen, verfügen über einen mit der Adresse 0 beginnenden linearen virtuellen Adressenraum. *Die* Adressumsetzung erfolgt ähnlich wie bei VM/370 mit Hilfe von *shadow page table*s und einer als *pmap* bezeichneten Datenstruktur. Gast-Maschinenbefehle, die Gast-Seitentabellen oder den TLB abändern, werden vom ESX Host–Kernel abgefangen und interpretiert. Der TLB enthält hierzu immer die in den shadow page tables enthaltenen Adressumsetzungen. Die wichtigsten Ein-/Ausgabe-Treiber sind in bezug auf maximale Leistung im Gastbetrieb-Modus optimiert. Die derzeitige Version ist in der Lage, einen symmetrischen Multiprozessor (SMP) mit bis zu 2 CPUs zu unterstützen.

Im Vergleich zu VM/370 sind der ESX Server und VMware benachteiligt, weil einige kritische Eigenschaften in der IA32-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-Modus ausgeführt werden können.

Dies sei an einem Beispiel erläutert. Wenn ein Gast ein Kontroll-Register schreibt, muss der Host-Kernel diesen Maschinenbefehl abfangen, damit nicht das reale Kontroll-Register des Hosts verändert wird. Der Host-Kernel wird jetzt nur die Effekte der Instruktion für diesen Gast simulieren. Liest der Gast anschließend diese Kontroll-Register 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 Kontroll-Registers, der nur für den Host sichtbar ist.

Da die IA32 Architektur diese Bedingung nicht erfüllt, ist es nicht möglich, wie unter VM/370 alle Maschinenbefehle einfach im Benutzer-Modus auszuführen, und auf Programmunterbrechungen zu vertrauen, wenn auf privilegierten Maschinenstatus Information zugegriffen wird. Beispielsweise:

> *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*

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.

Weitere Implementierungen

Ein anderes Produkt für die IA32 Plattform ist VirtualPC von Microsoft. VirtualPC wurde ursprünglich von Connectix entwickelt.

Die Firma SW-Soft setzt ihre Virtuozzo Virtual Private Servers (VPS) Software vor allem für Hosting Services bei Hosting Providern ein. Als Konkurrent tritt die Firma Ensim mit ihrer konzeptuell ähnlichen Extend Software auf. SW-Soft bezeichnet seine Produkte als *virtuelle Server*. Der als Virtual Environment (VE) bezeichnete Gast ist vom gleichen Typ wie das Host-Betriebssystem, normalerweise Linux. Er erscheint nach außen hin wie ein kompletter Server, verfügt aber über keinen eigenen Kernel, sondern benutzt stattdessen die Kernel Funktionen des Hosts. Zwischen dem Host-Betriebssystem und den virtuellen Environments befindet sich eine Zwischenschicht, die die Steuerung und Verwaltung der VEs übernimmt. Sie erweckt den Anschein, als ob eine Gruppe von Anwendungsprozessen auf getrennten Instanzen des Kernels läuft. Die VEs sind gegeneinander abgeschottet, können unabhängig voneinander gestartet und beendet werden und besitzen je ein eigenes Dateisystem.

Mehrere Projekte adressieren das Problem der fehlenden Virtualisierungseinrichtungen der Host-Architektur, indem sie das Gast-Betriebssystem abändern.

Xen ist ein GNU Open Source Software Virtual Machine Monitor, der derzeitig von der Systems Research Group des University of Cambridge Computer Laboratory entwickelt wird und ebenfalls auf der IA32 Plattform läuft. Um die Probleme mit der IA32 Architektur zu umgehen, wird, im Gegensatz zu VMware, ein als *Paravirtualization* bezeichneter Ansatz verwendet. Hierbei ist die Architektur der virtuellen Maschine nicht vollständig mit der Host Architektur identisch. Die Benutzerschnittstelle (Application Binary Interface, ABI) ist die gleiche. Der Gast-Kernel unterstellt jedoch Abweichungen zu der IA32 Architektur. Dies verbessert das Leistungsverhalten, erfordert aber Änderungen des Gast-Kernels. Hiervon ist nur ein sehr kleiner Teil des Kernel-Codes betroffen. Derzeitig existiert ein funktionsfähiger Linux-Port (XenoLinux), dessen ABI mit dem eines nicht-virtualisierten Linux 2.4 identisch ist. An Portierungen für Windows XP und BSD wird gearbeitet.

Ein ähnlicher Ansatz wird von *Denali verfolgt*. Als Gast-Betriebssystem dient *Ilwaco*, eine speziell an den Denali Hypervisor angepasste BSD Version. Denali unterstützt nicht das vollständige IA32 ABI. Es wurde für Netzwerk-Anwendungen entwickelt und unterstellt Einzelbenutzer-Anwendungen. Mehrfache virtuelle Adressräume sind unter Ilwaco nicht möglich.

Disco ist ein Hypervisor für einen ccNUMA Cluster mit MIPS R10000 Prozessoren und IRIX 5.3 als Gast-Betriebssystem. Auch die MIPS Architektur gestattet keine vollständige Virtualisierung des virtuellen Adressenraums des Kernels. Ähnlich Xen wurde der IRIX 5.3 Gast-Kernel für einen Betrieb unter Disco leicht abgeändert.

Plex86 ist ebenfalls ein Open Source Projekt. Als Gast Betriebssystem wird ein abgeändertes Linux verwendet, welches z.B. keine Ein-/Ausgabe Unterstützung enthält. Ein-/Ausgabe Operationen werden von einem Hardware Abstraction Layer direkt an den Plex86 Hypervisor weitergereicht.

Einrichtungen der Hardware Architekturzur Verbesserung des Virtualisierungs-Leistungsverhaltens

VM/370 Interpretive Execution Facility

1981

zSeries PR/SM und LPAR

1995

PowerPC LPAR

2002

Pentium Virtualisation Technology (VT-x)
Itanium Virtualisation Technology (VT-i)
Vanderpool

2006

AMD Pacifica

2006

Interpretive Execution Facility (VM/370)
Intel Virtualisation Technology (Vanderpool)
AMD Pacifica

Einführung eines Host/Gast Modus
zusätzlich zum Kernel/User Modus.

	Kernel Modus	User Modus
Host Modus	+	—
Gast Modus	+	+

Mögliche Zustände beim Einsatz des Host/Gast Modus

Intel Virtualisation Technology (VT) Bezeichnung für die IA32 (Pentium) Architektur:

- **Host Modus** VMX root operation
- **Gast Modus** VMX non-root operation

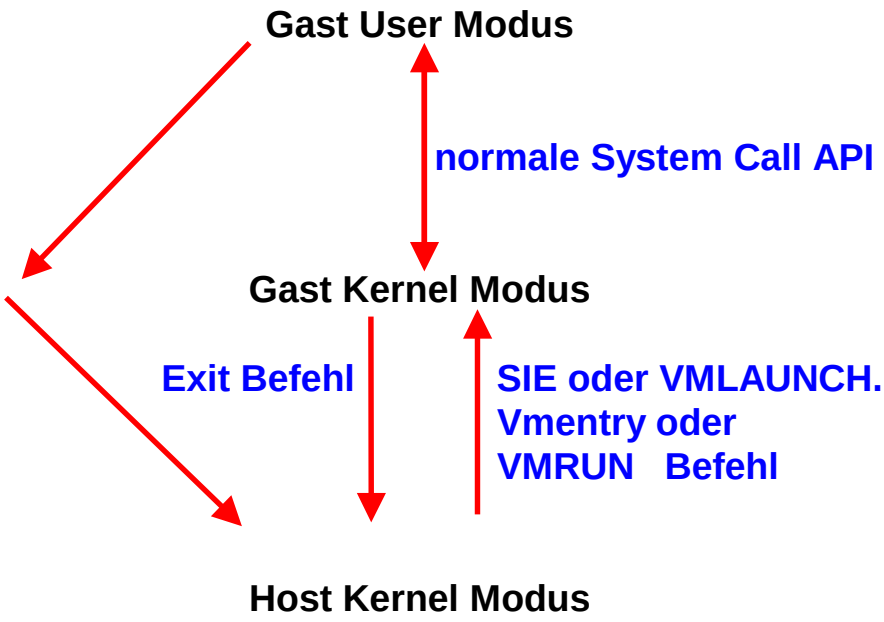
Der Host-User Modus hat keine Bedeutung.

Der Übergang vom Host Modus in den Gast Modus bewirkt den Start einer Virtuellen Maschine. Dies erfolgt durch spezielle Maschinenbefehle:

S/370	**Start Interpretive Execution (SIE)**
Intel Pentium VT-x	**VMLAUNCH**

Für die Ablaufsteuerung werden spezielle Maschinenbefehle benutzt, die im Gast Modus nicht sichbar sind:

S/370	**Millicode**
Intel Pentium VT-x	**VMX Befehle**



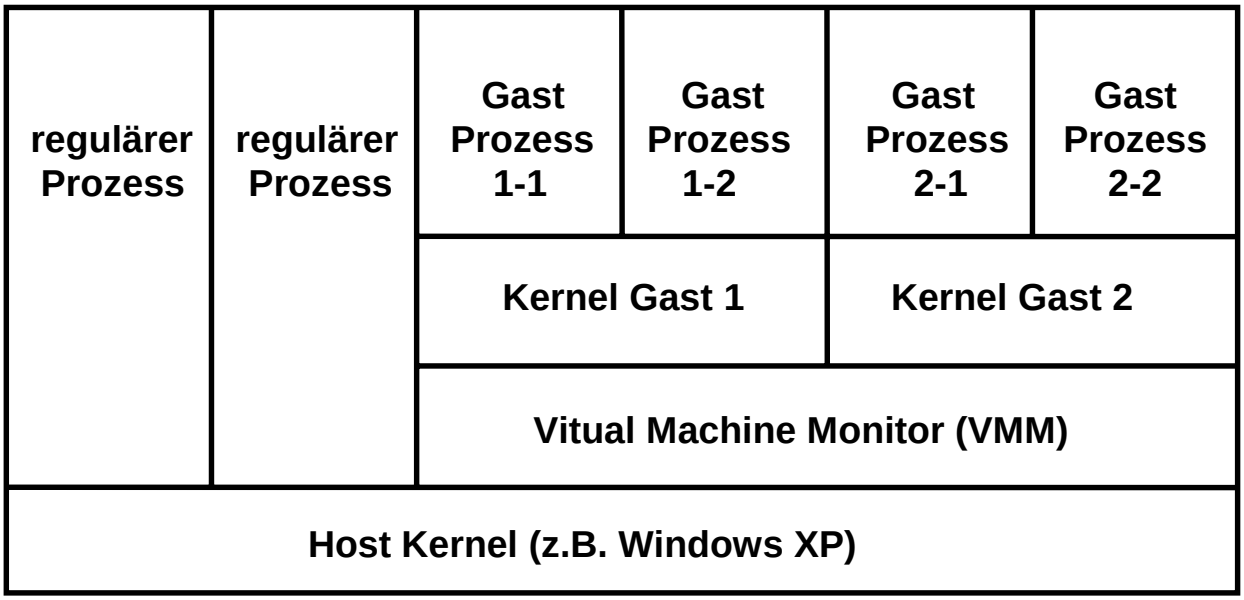
Die nicht-privilegierten Maschinenbefehle des Gastes werden vom Gast im User Modus direkt ausgeführt.

Die privilegierten Maschinenbefehle des Gastes werden in zwei Gruppen geteilt. Ein State Descriptor (Virtual-Machine Control Structure, VMCS) spezifiziert die Aufteilung.

Die meisten privilegierten Maschinenbefehle des Gast-Kernels werden innerhalb des Gast-Modus ausgeführt. Maschinenbefehle oder Zustände, die weiterhin die Unterstützung durch den Host-Kernel erfordern, führen zu einer *Interception*. Diese bewirkt einen Transfer in den Host-Kernel 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 virtualsiert unmittelbar unter dem Host Betriebssystem laufen



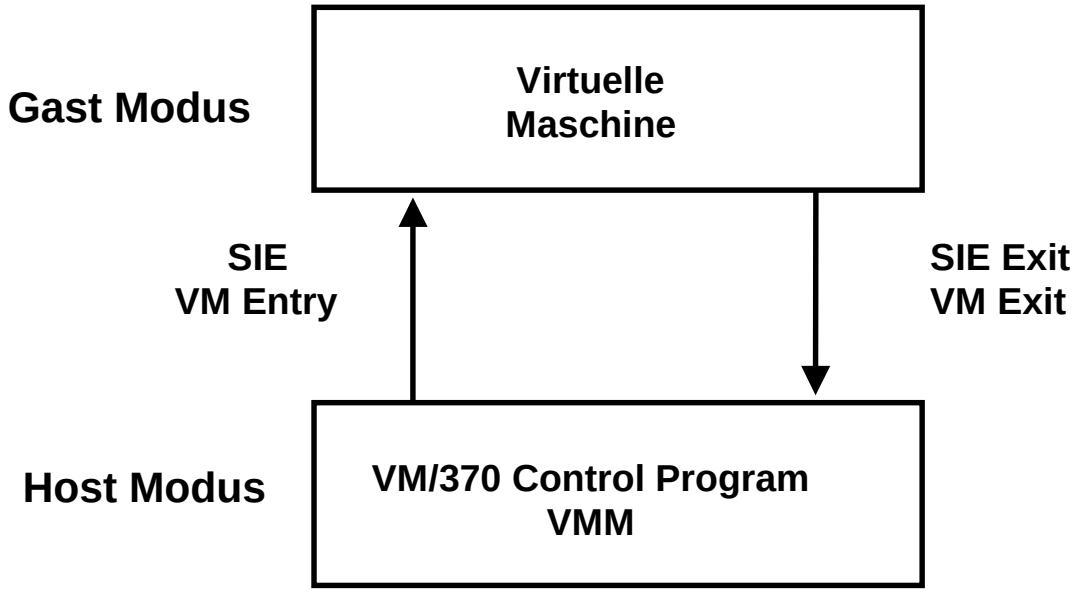
Mögliche Vanderpool Konfiguration

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.

Es wird erwartet, dass existierende Virtual Machine Monitore wie VMware, XEN, Virtual PC usw. in Zukunft die Vanderpool Technologie nutzen werden. Damit entfallen viele bisherige IA32 Probleme.

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. Die Rückkehr zum VMM erfolgt durch einen VM Exit Befehl.

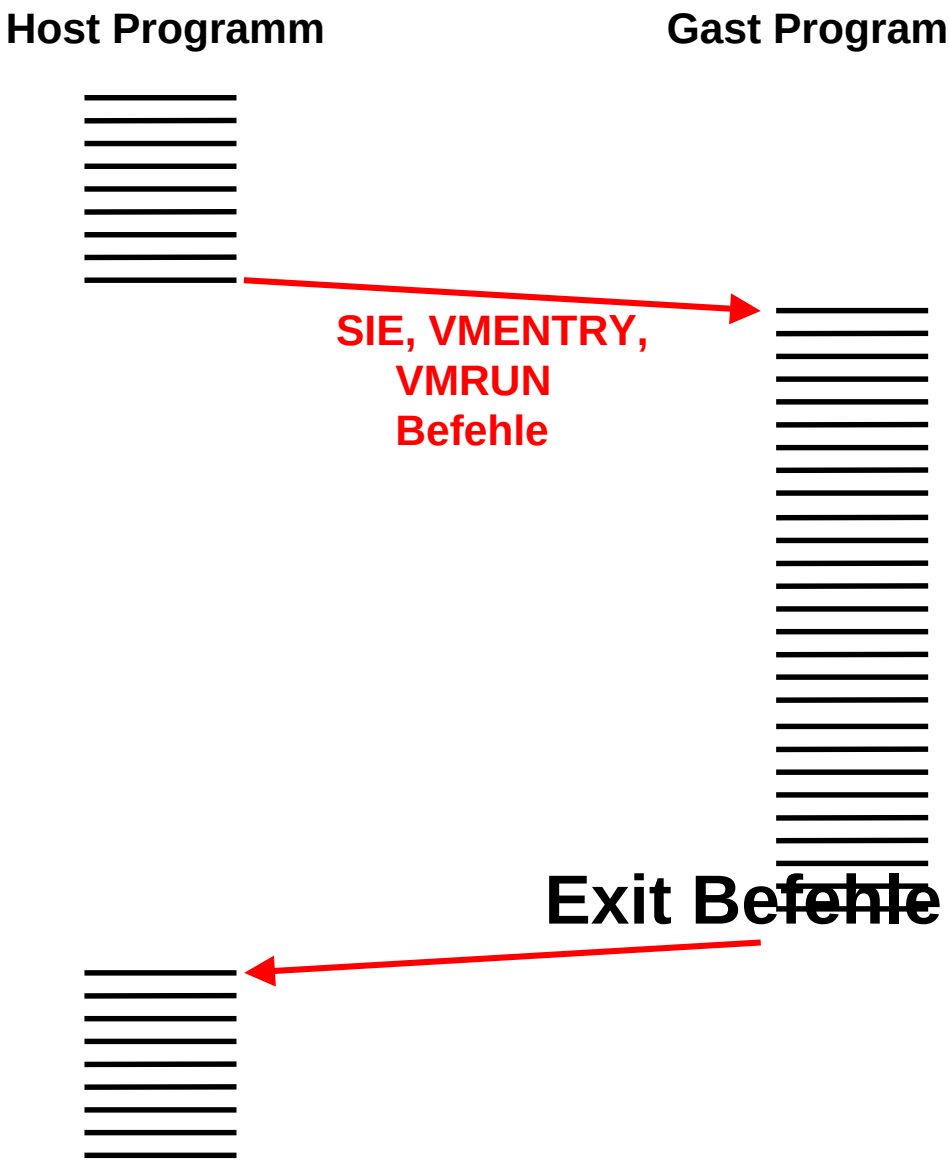


Im Gastmodus können privilegierte, aber nicht sensitive Maschininenbefehle vom Kernel des Gast Betriebssystems abgearbeitet werden. Das Problem der sensitiven nicht-privilegierten IA32 Architektur der IA32 Architektur wird durch zusätzliche Vanderpool Hardware gelöst, die diese Befehle bei ihrer Dekodierung identifiziert und zwecks Ausführung an den VMM übergibt.

Beispiel Intel-VT

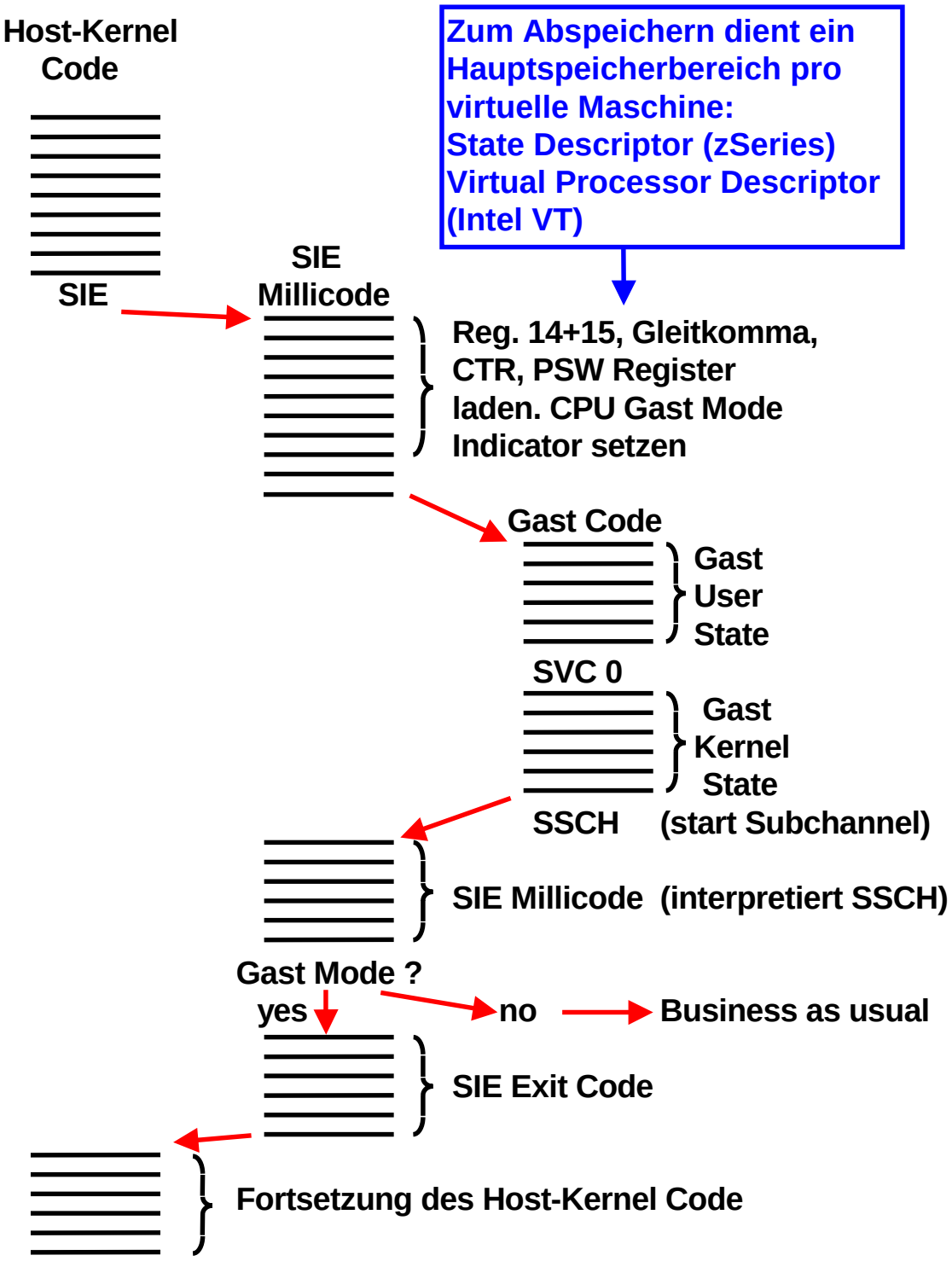
Bei der Initiierung eines neuen Gast-Betriebssystems wird mit einem VMPTLRD Kommando ein 4 KByte Speicherblock reserviert (Virtual Processor Descriptor bei Intel-VT, State Descriptor bei z/VM). In diesem Bereich wird der State einer virtuellen Maschine gespeichert, wenn sie nicht aktiv ist. Dieser Bereich bleibt während der Lebensdauer der virtuellen Maschine erhalten und wird bei Beendigung mit einem VMCLEAR Befehl gelöscht.

Der Übergang vom Host-Modus in den Gast-Modus erfolgt mit einem VMENTRY Befehl. Hiervon existieren zwei Ausprägungen: VMLAUNCH und VMRESUME. VMRESUME lädt den Prozessor State von dem Virtual Processor Descriptor. VMLAUNCH macht das Gleiche und erstellt eine “Virtual Machine Control Structure (VMCS)”.

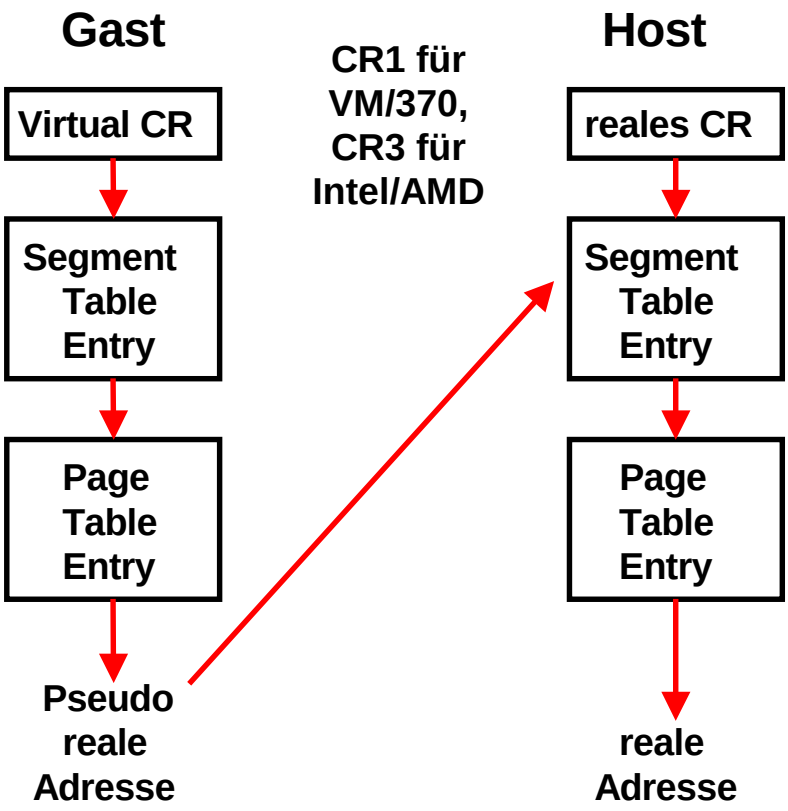


Übergang vom Host- in den Gast-Modus

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



Ausführung der SIE Instruktion

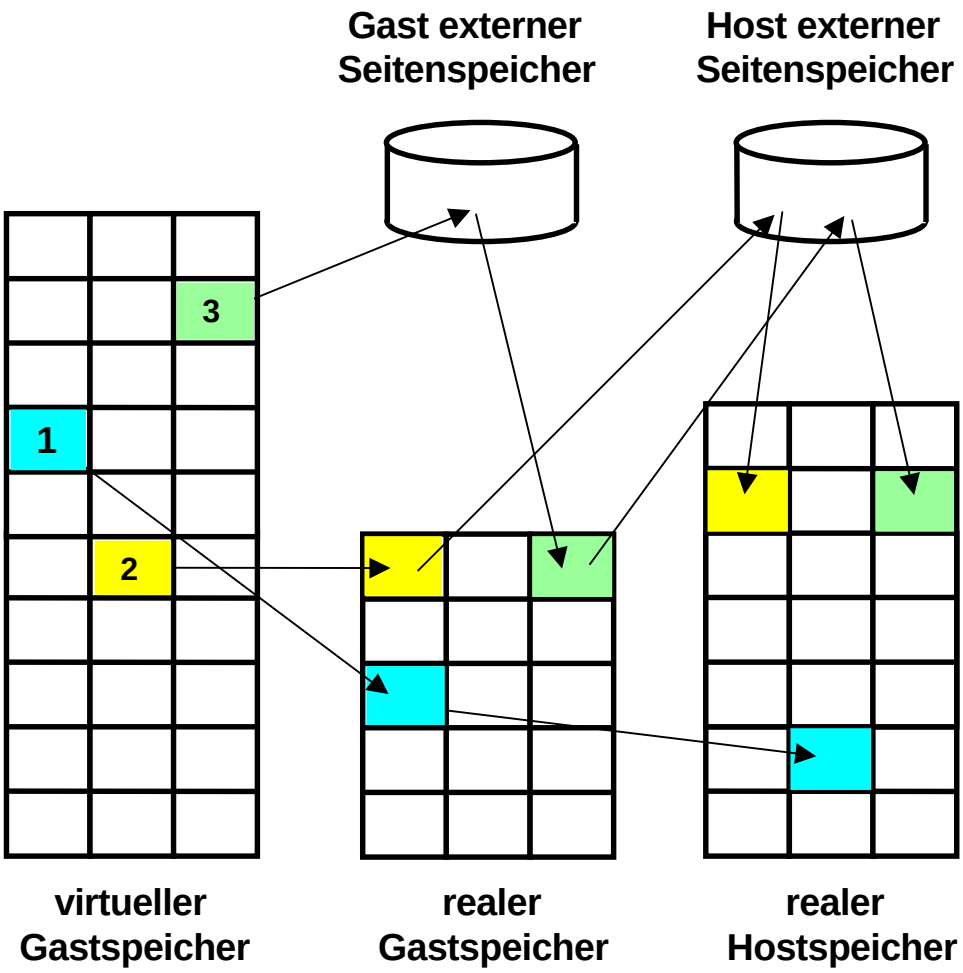


AMD Pacifica Virtuelle Adressübersetzung

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 Segement- und Seitentabellen des Gast-Kernels in eine pseudo-reale Adresse umgesetzt. Anschließend übersetzt die gleiche Hardware die pseudo-reale Adresse mit Hilfe von Segement- 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.



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 Gastspeicher.

Logical Partition
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

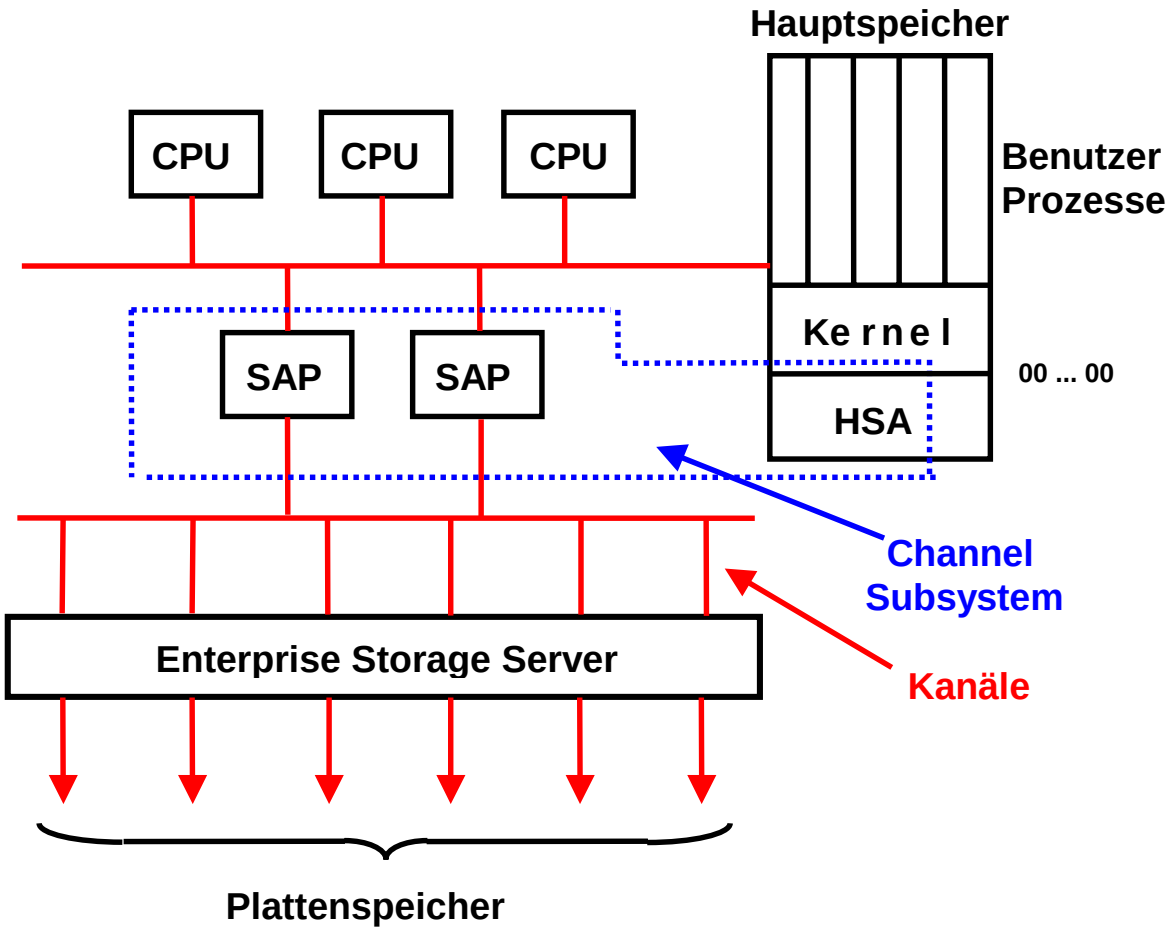
Virtuelle Maschinen in realen Adressräumen

Logical Partition (LPAR) und PR/SM

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

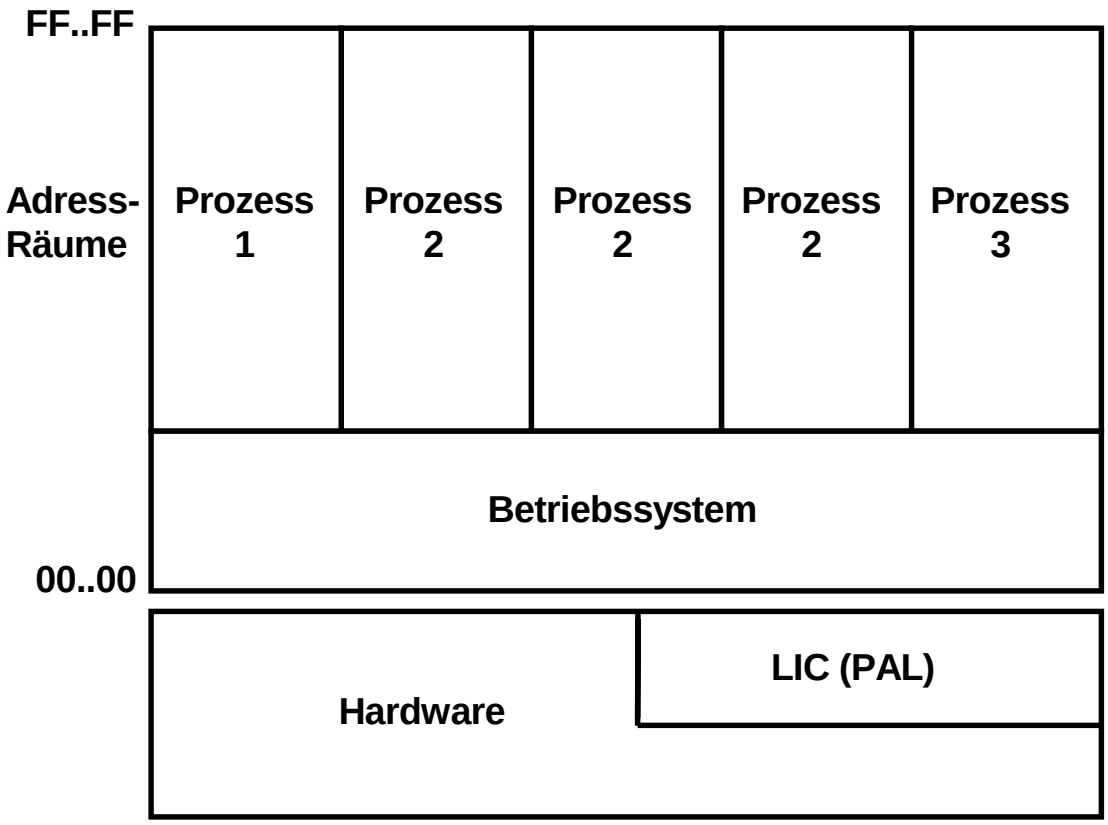
Die zSeries PR/SM (Processor Resource/System Manager) Einrichtung verwendet ein anderes, als LPAR (Logical Partition) bezeichnetes Verfahren. Für jedes Gastbetriebssystem wird eine eigene LPAR eingerichtet. Die PR/SM Hardware stellt mehrfache (bis zu 64) getrennte reale Speicher zur Verfügung. Die realen 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 µCode (hier als Millicode bezeichnet) aus.

zSeries Ein/Ausgabe Anschluss



Die *HSA* (Hardware System Area) ist ein Teil des System z Hauptspeichers. Sie liegt außerhalb des Adressenraums, auf den die CPUs zugreifen können. 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 umfang-reiche Optimierungen der Plattenspeicherzugriffe möglich.

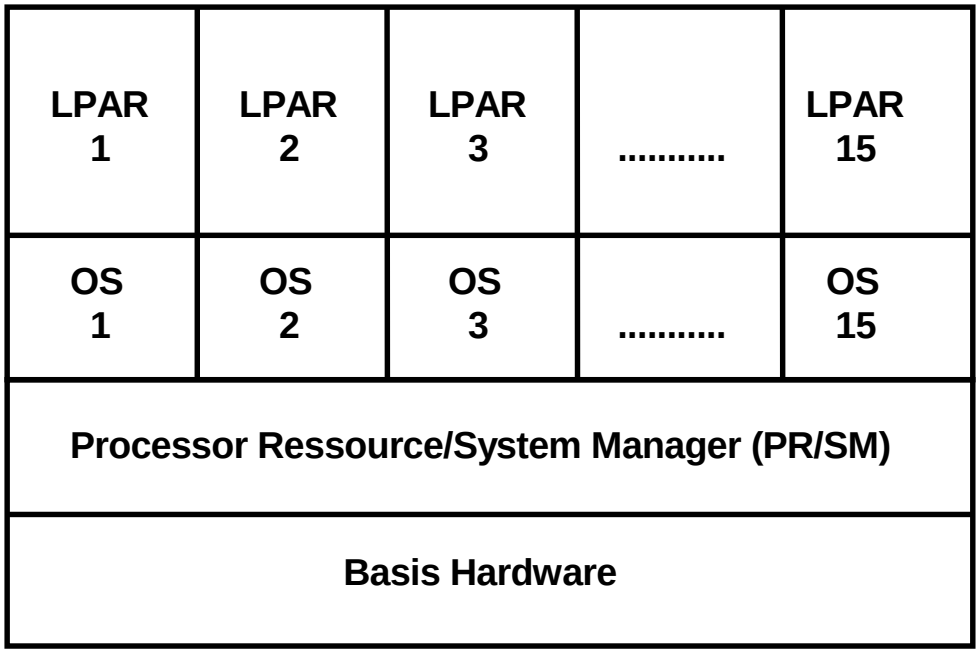


System z Struktur

LIC (Licensed Internal Code, auch als Firmware bezeichnet,) ist Maschinencode, der von der CPU ausgeführt wird, aber in einem getrennten Speicher untergebracht und ausgeführt wird, außerhalb des architekturmäßig zugänglichen realen Hauptspeichers. Häufig bestehen Teile aus regulären S/390 Maschinenbefehlen.

Itanium (und DEC Alpha) verwendet statt LIC die Bezeichnumg PAL (Privileged Architecture Library).

894 instructions (668 implemented entirely in hardware)



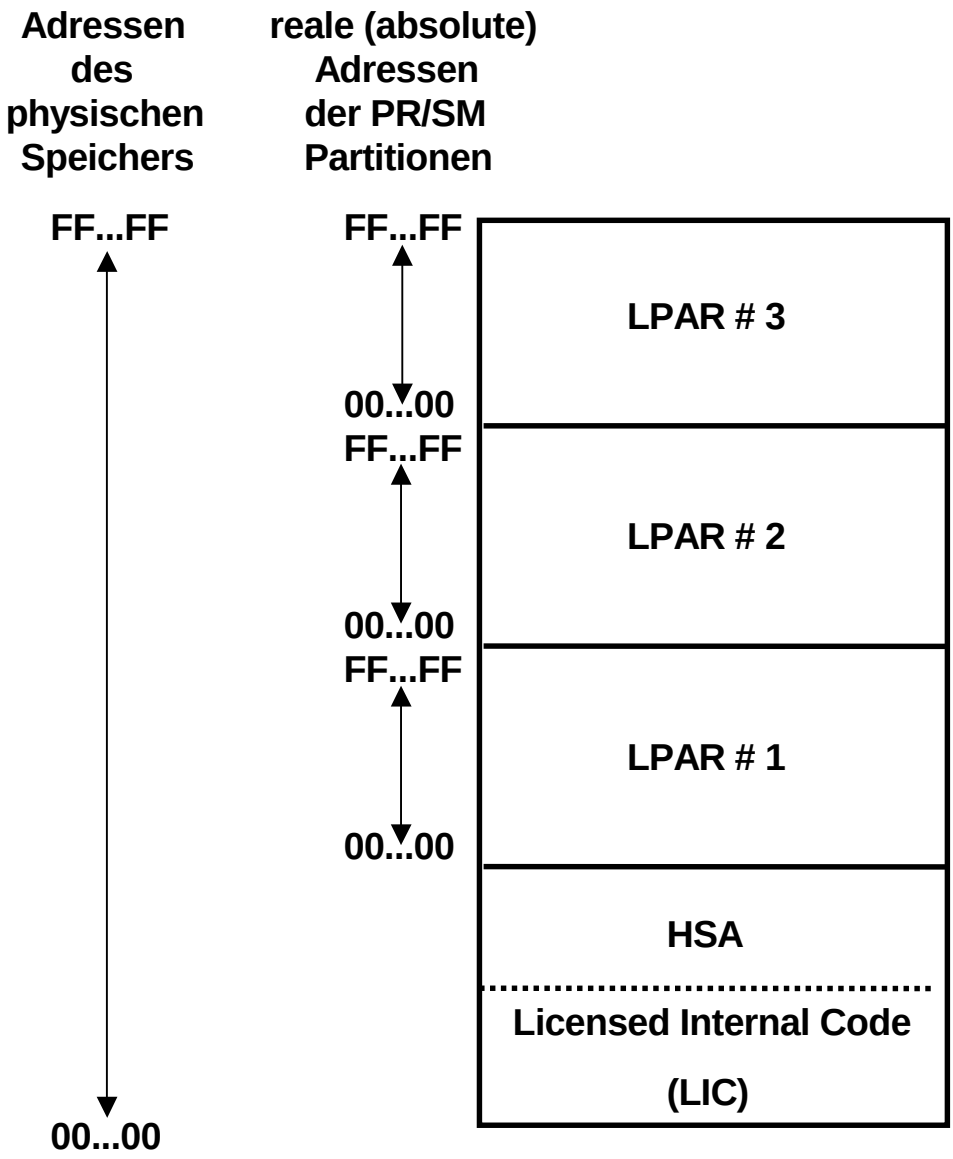
LPAR **Logical Partition**
OS **Operating System**

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).

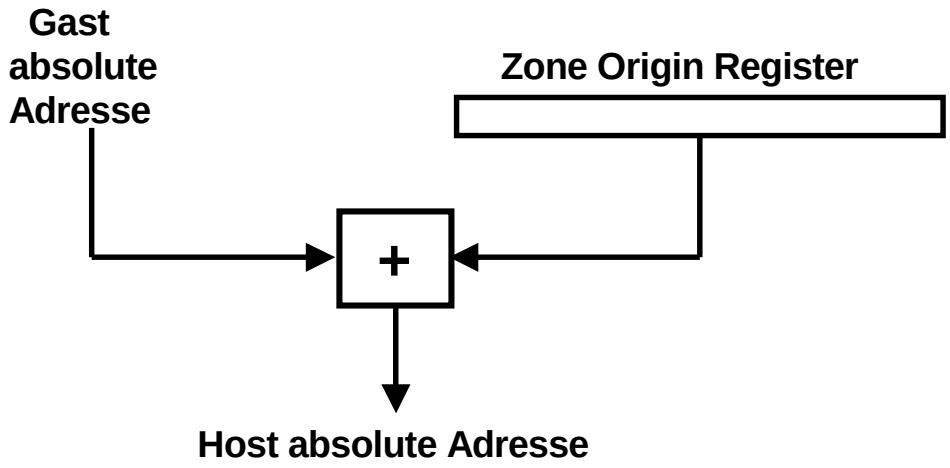
S/390 Rechner anderer Hersteller verfügen über vergleichbare Einrichtungen: Hitachi MLPF, Amdahl Multiple Domain Facility. Die Itanium Virtualization Technology implementiert einige Funktionen mit Hilfe von PAL Code.



Aufteilung des physischen Speichers in mehrere reale Speicher

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

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

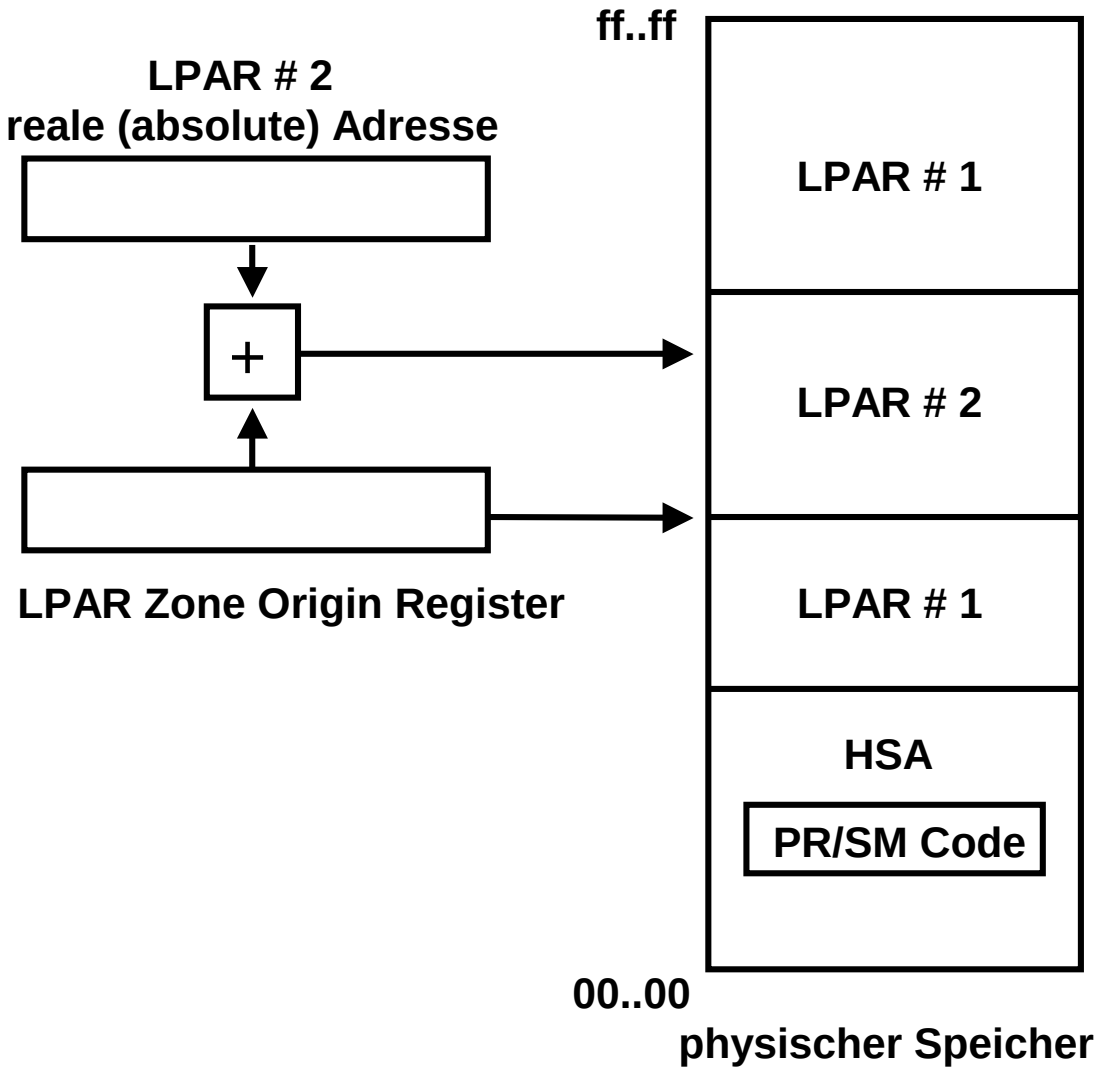


Umsetzung der Gast Adressen in physische Adressen

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.

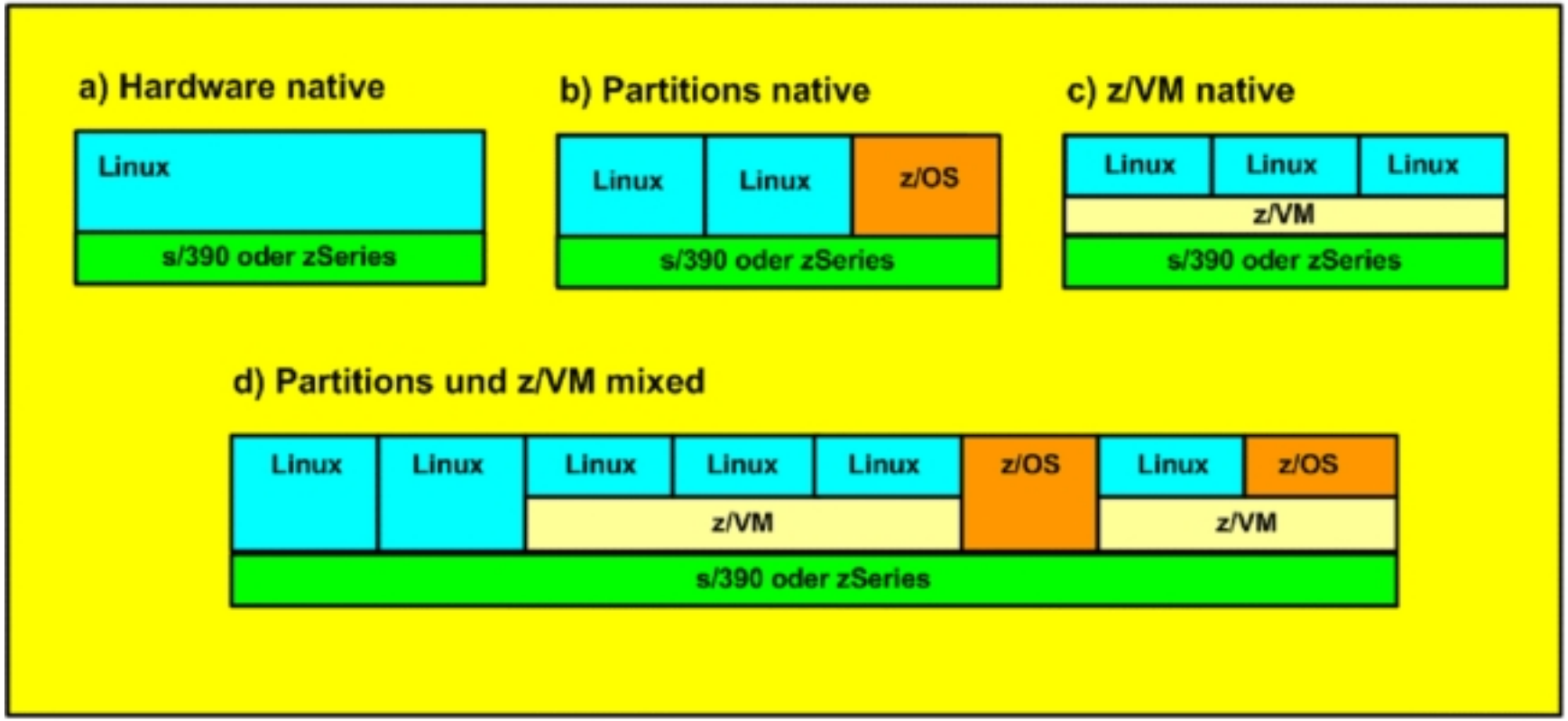
Es ist möglich, die CPUs den einzelnen LPARs dynamisch zuzuordnen. In diesem Fall ist es die Aufgabe des PR/SM Hypervisors sicherzustellen, dass die korrekten Adressenwerte in die Zone Origin und Zone Limit Register geladen werden. PR/SM verfügt ähnlich wie ein Bestriebssystem Kernel über eine Zeitscheibensteuerung.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



LPAR Speicherplatz-Verwaltung
Basis Fall

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

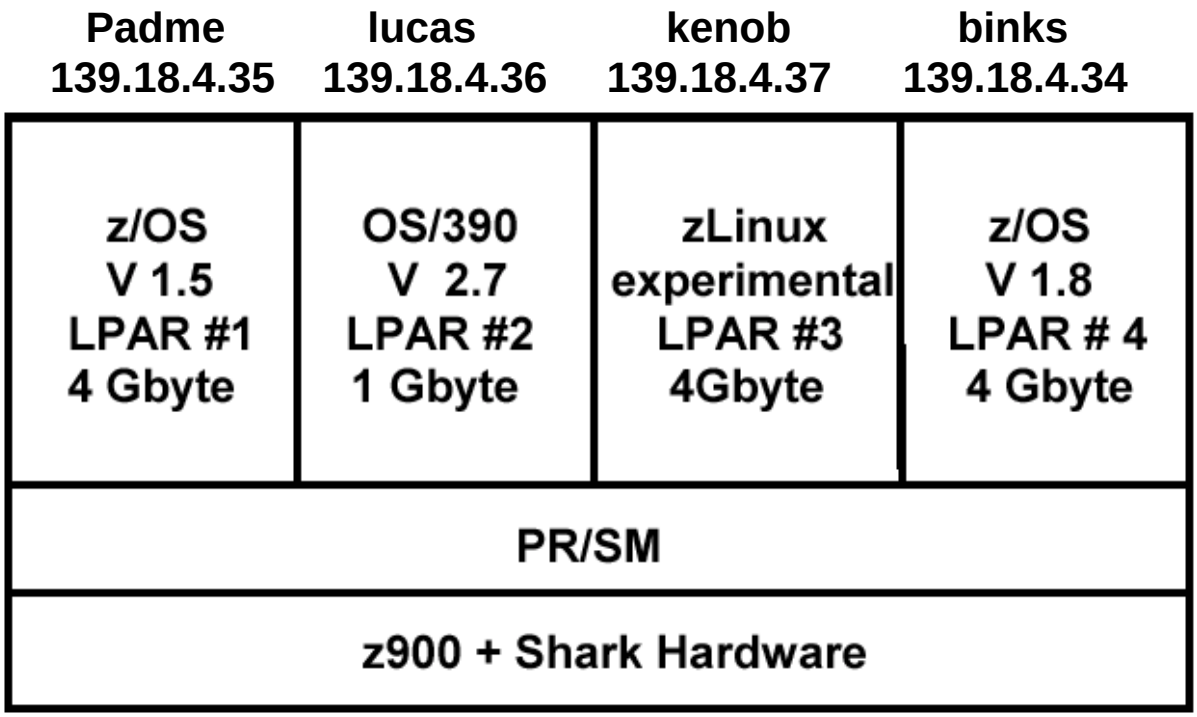
Wenn die CPU einer anderen LPAR zugeordnet wird, wird das LPAR Zone Origin Register neu geladen.



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).

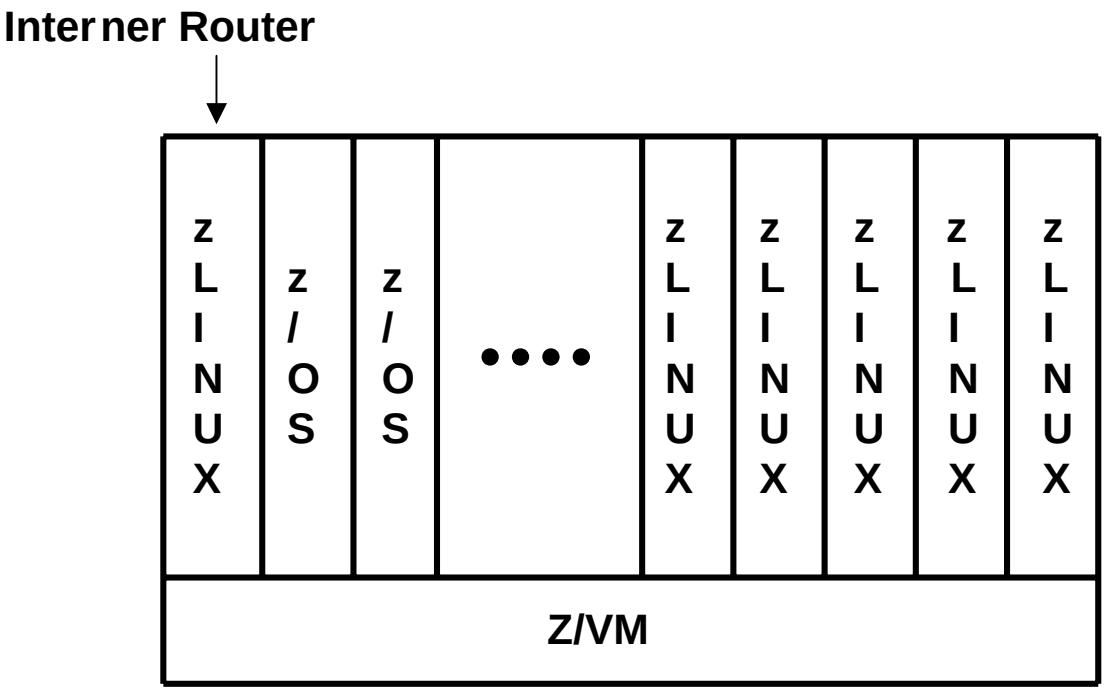
Rechner – Konfiguration
jedi.informatik.uni-leipzig.de



IBM z900 Enterprise Server

- 9 CPUs
- 32 GByte Hauptspeicher
- 1500 GByte Plattenspeicher
- Cryptographic coprocessor mit triple DES Support
- Hardware-assisted data compression
- ESCON 17MB/sec channel

padme.informatik.uni-leipzig.de
lucas.informatik.uni-leipzig.de
kenob.informatik.uni-leipzig.de
binks.informatik.uni-leipzig.de



kenob.informatik.uni- leipzig.de

Auf der z/VM virtuellen Maschine laufen theoretisch beliebig viele virtuelle Maschinen. Wir haben etwa 15 virtuelle Maschinen installiert

PR/SM and LPAR Sicherheit und Isolation

Zertifikat der USA Regierung:

„LPARs haben äquivalente Sicherheits-Eigenschaften wie physikalisch getrennte Rechner“.

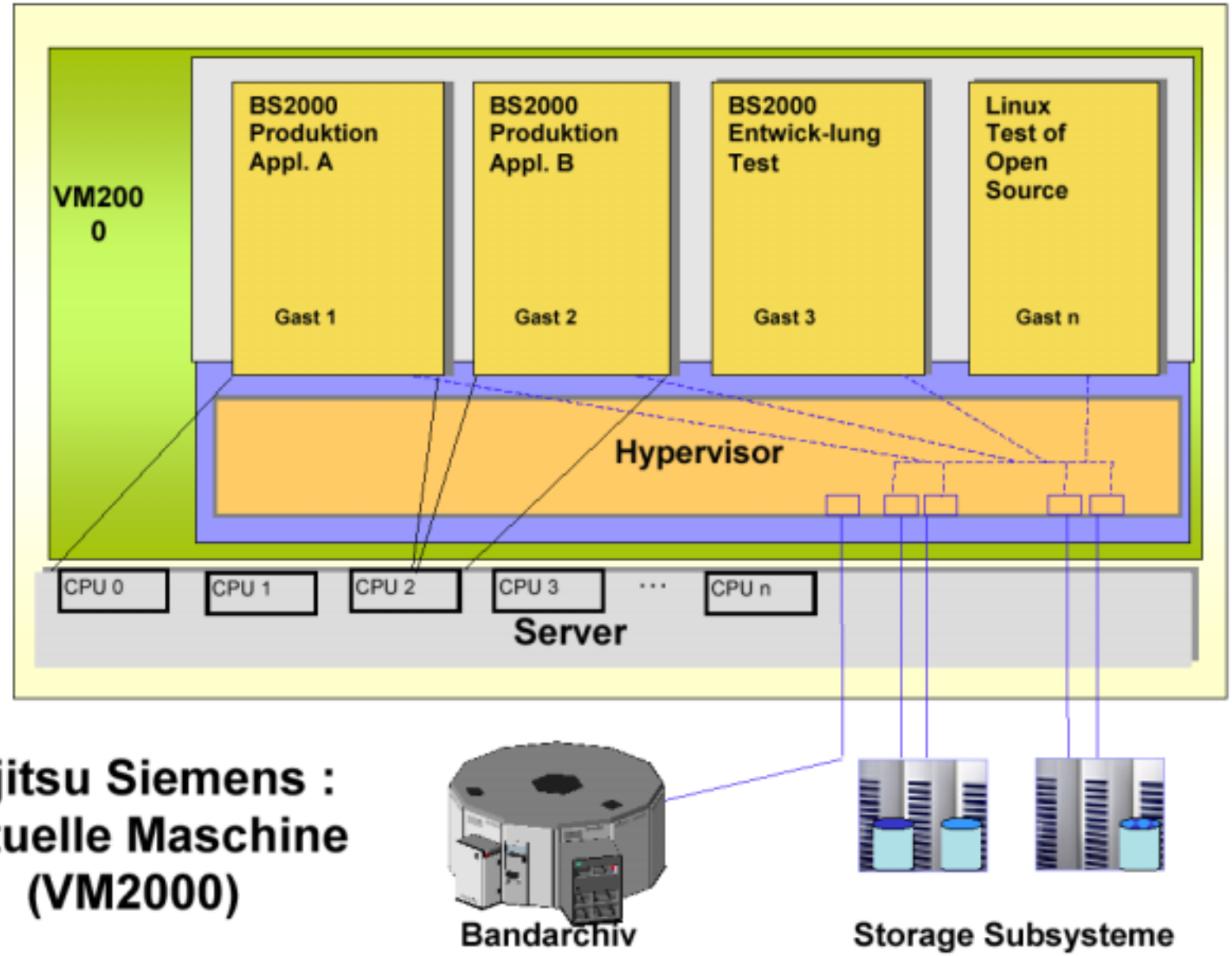
Das Bundesamt für Sicherheit in der Informationstechnik (BSI) stellt IBM für den Processor Resource/System Manager (PR/SM) des Mainframes z900 das weltweit höchste Sicherheitszertifikat für einen Server aus. Die Bescheinigung nach dem internationalen Standard Common Criteria (CC) für die Stufen EAL4 und EAL5 wurde auf der CeBIT 2003 an IBM verliehen. Der z900 ist der erste Server, der nach der Evaluierungsstufe EAL5 für seine Virtualisierungstechnologie zertifiziert wurde.

Die Zertifizierung des BSI bescheinigt, dass Programme, die auf einem IBM eServer zSeries z900 in verschiedenen logischen Partitionen (LPAR) laufen, ebenso gut voneinander isoliert sind, wie auf getrennten physikalischen Servern.

Beispielsweise können Web-Anwendungen und Produktions-Anwendungen, die in getrennten logischen Partitionen laufen, komplett voneinander isoliert betrieben werden. Dies ist obwohl sie die physikalischen Ressourcen des zSeries Servers gemeinsam nutzen.

IBM Presseinformation SG/94/2003, CeBIT 2003: 14. März 2003 -

es 0611 ww6 wgs 03-03



Fujitsu Siemens : Virtuelle Maschine (VM2000)

Fujitsu Siemens BS2000/OSD

Derzeitig BS2000/OSD-Version 7.0.

Die Hardware war ursprünglich fast S/370 kompatibel. Einige Inkompatibilitäten bei privilegierten Maschinen-befehlen und im I/O Bereich. Seit 1987 volle S/370 Kompatibilität.

1996 erfolgte die Portierung von BS2000/OSD auf die Sparc Architektur.

Obwohl das Betriebssystem jetzt auf unterschiedlichen Hardware-Architekturen (S-Server mit S/390-Architektur und SR2000-Server für die Sparc-Architektur) läuft, wird für BS2000-Anwendungen der objektkompatible Ablauf garantiert.

Für S/390 programmierte Anwendungen können ohne Neuübersetzung auf Rechnern mit RISC-Architektur eingesetzt werden.

Sesam ist die Alternative zu DB2, Universal Transaction Monitor (UTM) ist die Alternative zu CICS.

Modell S200 mit bis zu 15 CPUs wurde Nov. 2006 angekündigt. Keine 64 Bit Version.

OS/390 und z/OS bis zu Version 1.5 läuft auf Fujitsu Siemens Rechnern, aber BS2000/OSD läuft nicht auf IBM Rechnern.

Intelligent Resource Director
IRD

Dynamische Verwaltung der realen Speichergrößen

LPAR CPU Management

Dynamic Channel Path Management

Channel Subsystem Priority Queuing

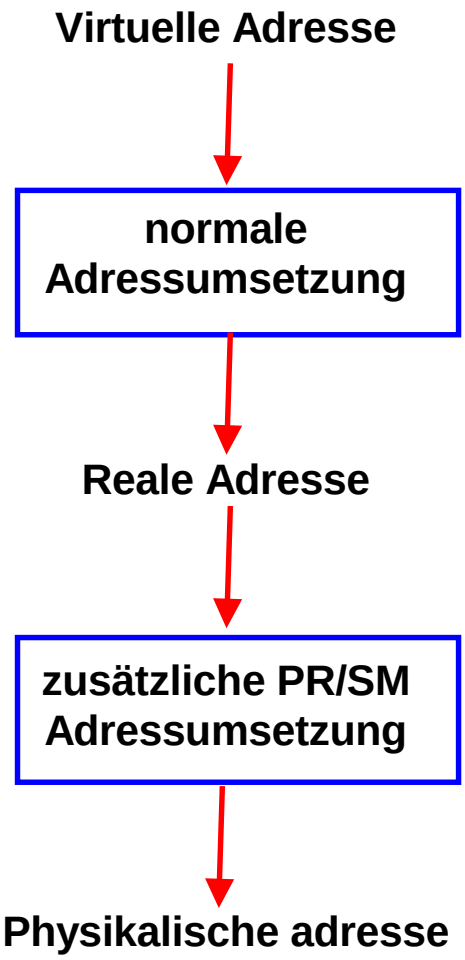
Gemeinsame Nutzung von Krypto Coprozessoren und I/O Geräten durch mehrere LPAR´s ist möglich (EMIF).

IRD Speicherplatzverwaltung

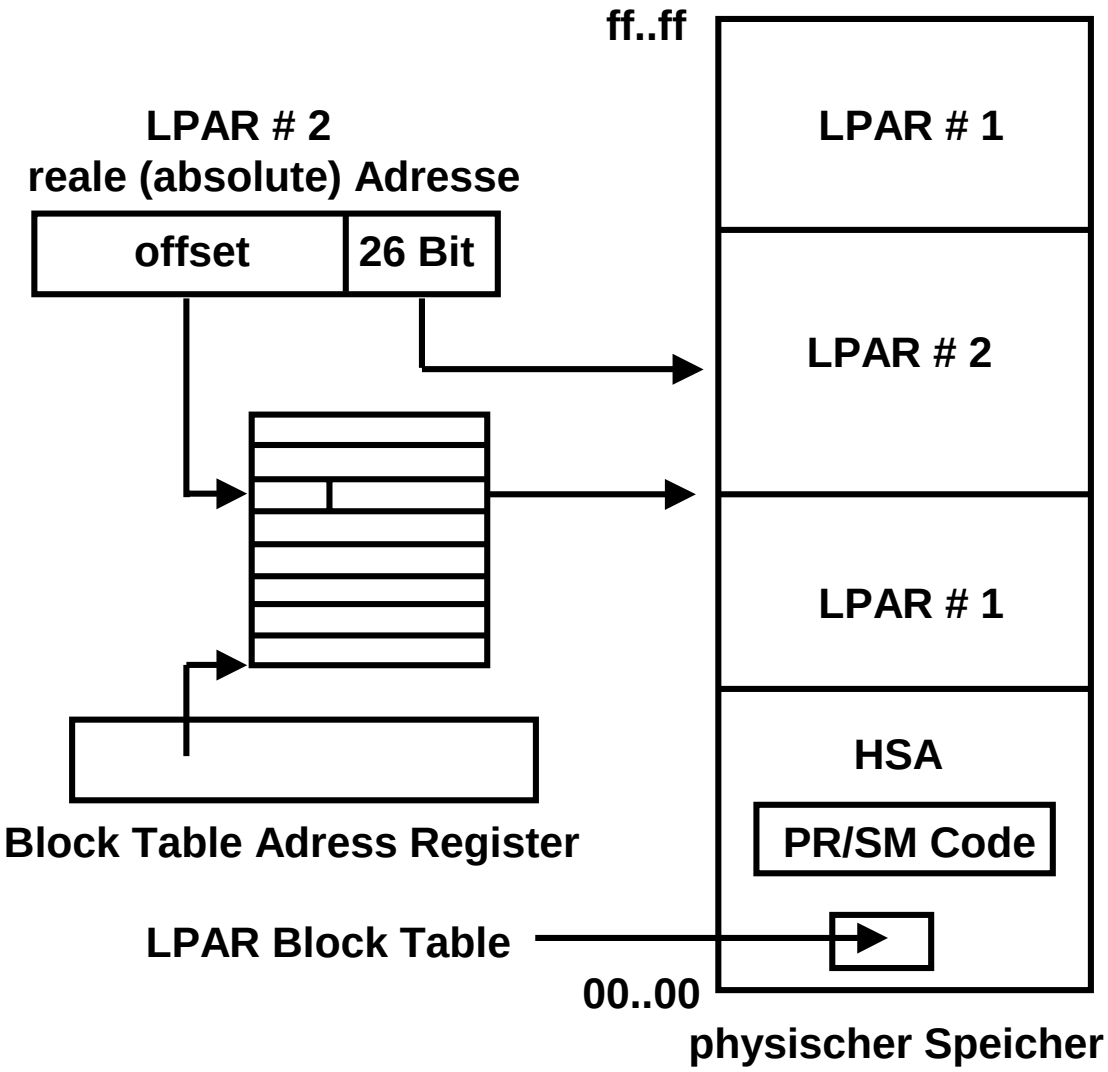
Bei der als IRD (Intelligent Resource Director) bezeichneten Erweiterung von PR/SM werden die mehrfachen realen Speicher der einzelnen LPARs in einem einzigen physischen Speicher abgebildet. Mit einem zusätzlichen Mechanismus, der ähnlich wie die virtuelle Speicher-platzverwaltung arbeitet, kann der den einzelnen LPARs zugeteilte Platz dynamisch in Blöcken von je 64 Mbyte vergrößert und verkleinert werden.

Hierbei kann nicht mehr sichergestellt werden, dass sich die Menge der einer LPAR zugeteilten 64 Mbyte Blöcke in einem zusammenhängenden Teil des physischen Speichers befindet. Damit für die LPARs die Illusion eines kontinuierlichen Adressenraums gewahrt bleibt, wird ein Ansatz ähnlich der virtuellen Speicherplatzverwaltung benutzt. Spezifisch wird ähnlich wie bei den Seitentabellen der kontinuierliche reale Adressenraum einer LPAR in eine diskontinuierliche Menge von 64 MByte Blöcken innerhalb des physischen Speichers mit Hilfe einer Block Tabelle umgesetzt.

Die Größe des den einzelnen LPARs zugeordneten physikalischen Speicherbereiches kann mit Hilfe einer weiteren Adressenumsetzung dynamisch variiert werden



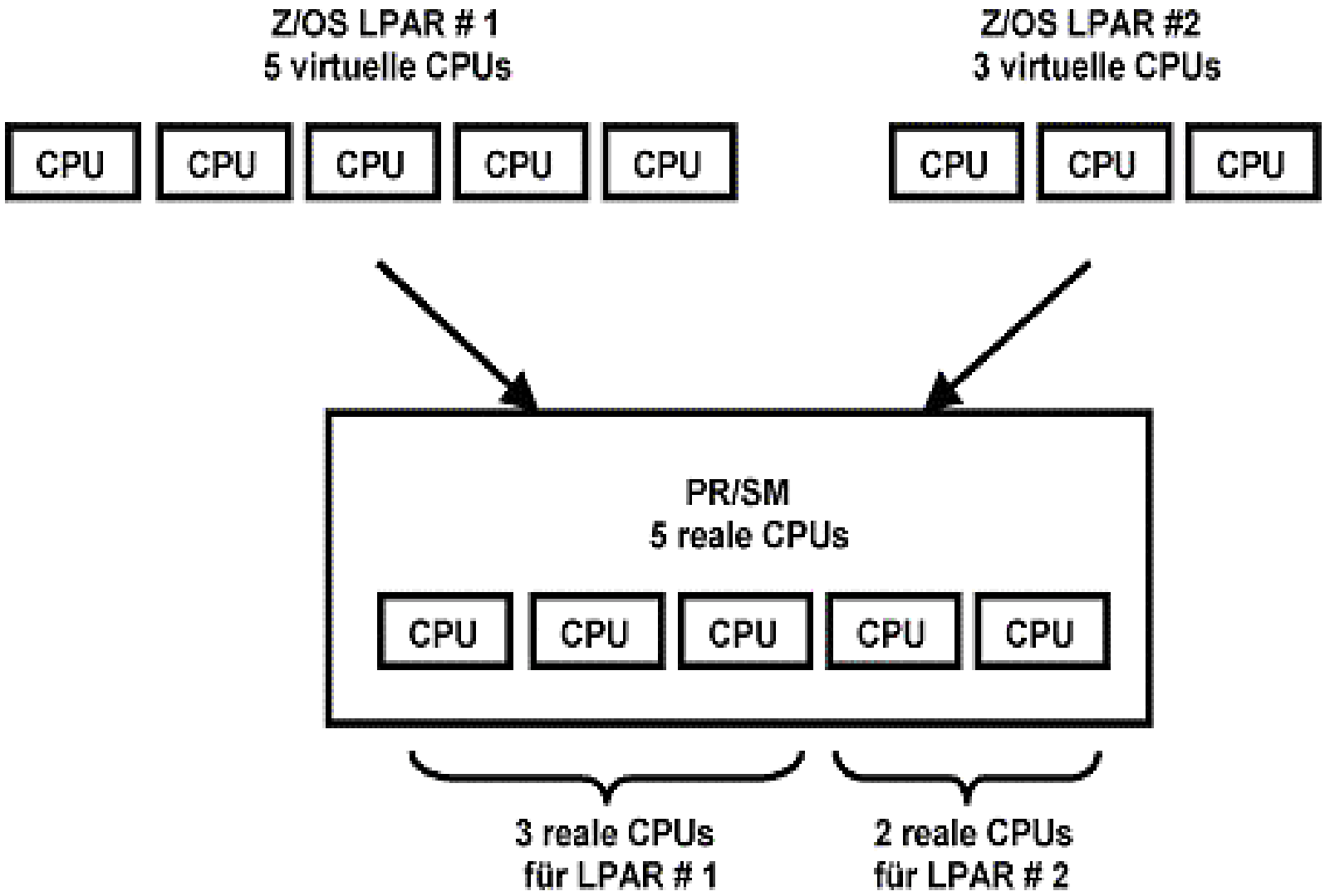
Die Seiten und Rahmen der zusätzlichen PR/SM Adressumsetzung haben eine Größe von 64 Mbyte, gegenüber 4 KByte bei der normalen virtuellen Adressumsetzung

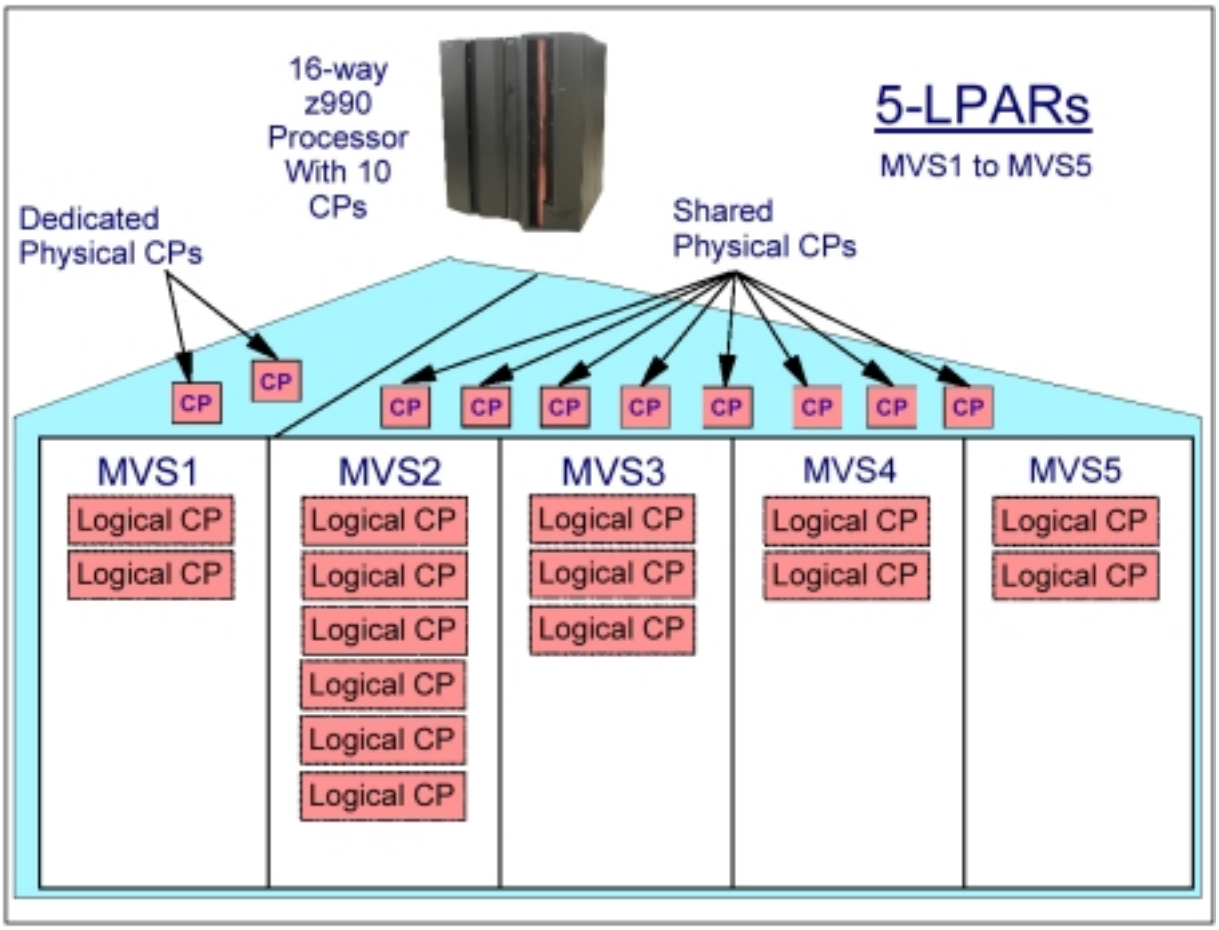


IRD dynamische Speicherplatz-Verwaltung

Aufteilung des physischen Speichers in 64 Mbyte große Blöcke

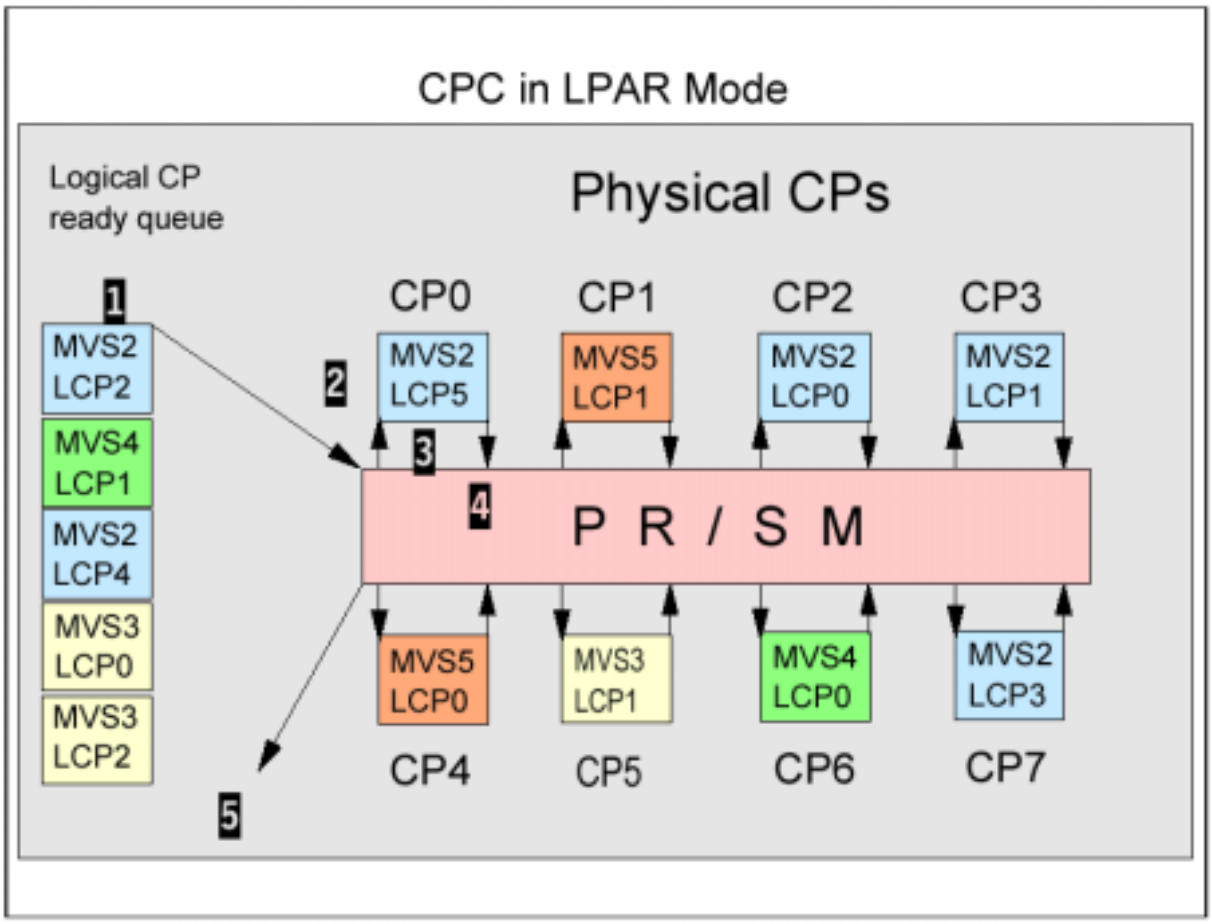
An Stelle des Zone Origin Registers besteht ein Block Table Adress Register. Ähnlich einer einstufigen virtuellen Address Translation. Es fehlt ein Fehlseiten-vergleichbarer Mechanismus. 1 Block Table für jede LPAR.


Z/OS LPAR # 1
5 virtuelle CPUs
CPU
CPU
CPU
CPU
CPU
Z/OS LPAR #2
3 virtuelle CPUs
CPU
CPU
CPU
PR/SM
5 reale CPUs
CPU
CPU
CPU
CPU
CPU
3 reale CPUs
für LPAR # 1
2 reale CPUs
für LPAR # 2



Physische CPUs (CPs) können dediziert oder gemeinsam genutzt (shared) sein. Bei einer LPAR mit dedizierten CPUs ist eine logische CPU permanent einer physischen CPU zugeordnet. Dies bedeutet weniger Overhead.

Gemeinsam genutzte (shared CPUs) erzeugen mehr Overhead. Dies wird überkompensiert, weil eine LPAR CPU Kapazität nutzen kann, die eine andere LPAR gerade nicht benötigt. Wenn ein Betriebssystem in den Warte- (wait) Zustand versetzt wird, gibt es die benutzten CPU(s) frei.

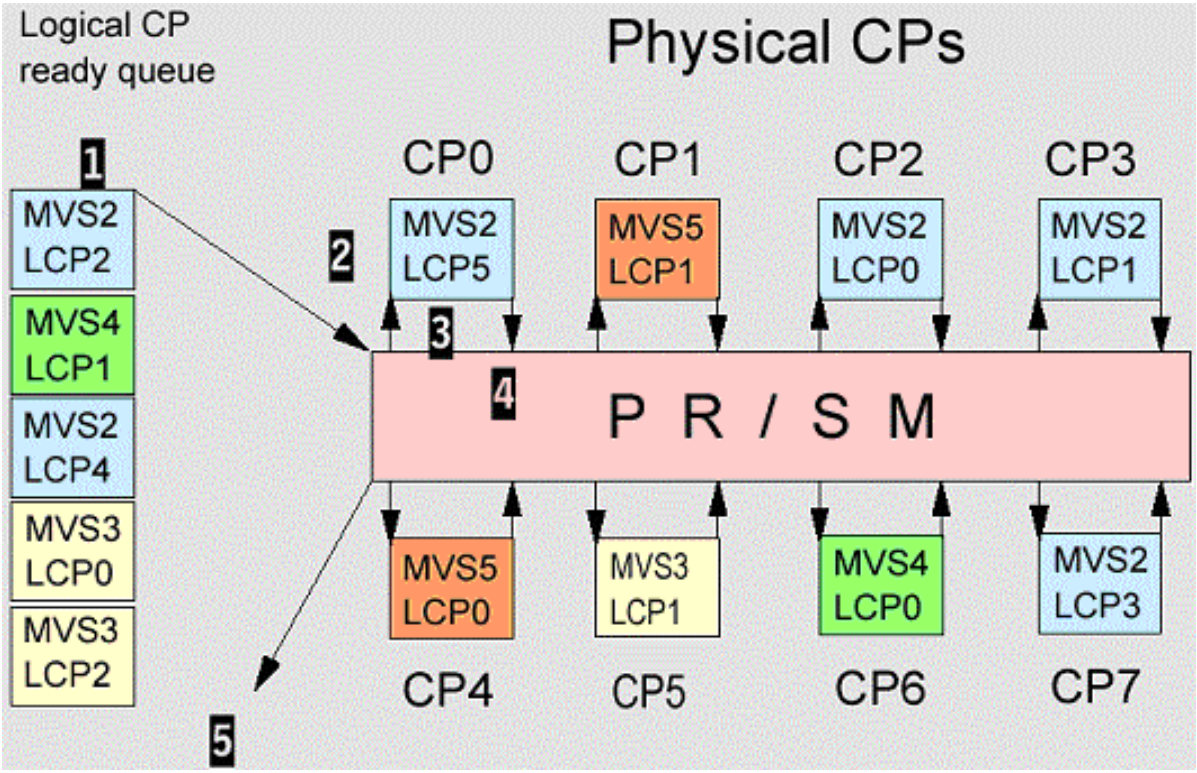


LPAR dispatching

Den vier LPARs mit den Betriebssystemen MVS2, MVS3, MVS4 und MVS5 stehen 8 physische CPUs zur Verfügung (CP0 .. CP7).

Jedes der Betriebssysteme ist als Multiprozessor konfiguriert und glaubt, über eine bestimmte Anzahl (logischer) CPUs zu verfügen. Die Anzahl der logischen CPUs übertrifft die Anzahl der physisch vorhandenen CPUs.

Der PR/SM Hypervisor ordnet die logischen CPUs den physisch vorhandenen CPUs zu.



1. **The next logical CP to be dispatched is chosen from the logical CP ready queue based on the logical CP weight.**
2. **LPAR LIC dispatches the selected logical CP (LCP5 of MVS2 LPAR) on a physical CP in the CPC (CP0 abovel).**
3. **The z/OS dispatchable unit running on that logical processor (MVS2 logical CP5) begins to execute on physical CP0. It executes until its time slice (generally between 12.5 and 25 milliseconds) expires, or it enters a wait, or it is intercepted for some reason.**
4. **the logical CP keeps running until it uses all its time slice. At this point the logical CP5 environment is saved and control is passed back to LPAR LIC, which starts executing on physical CP0 again.**
5. **LPAR LIC determines why the logical CP ended execution and requeues the logical CP accordingly. If it is ready with work, it is requeued on the logical CP ready queue and step 1 begins again.**

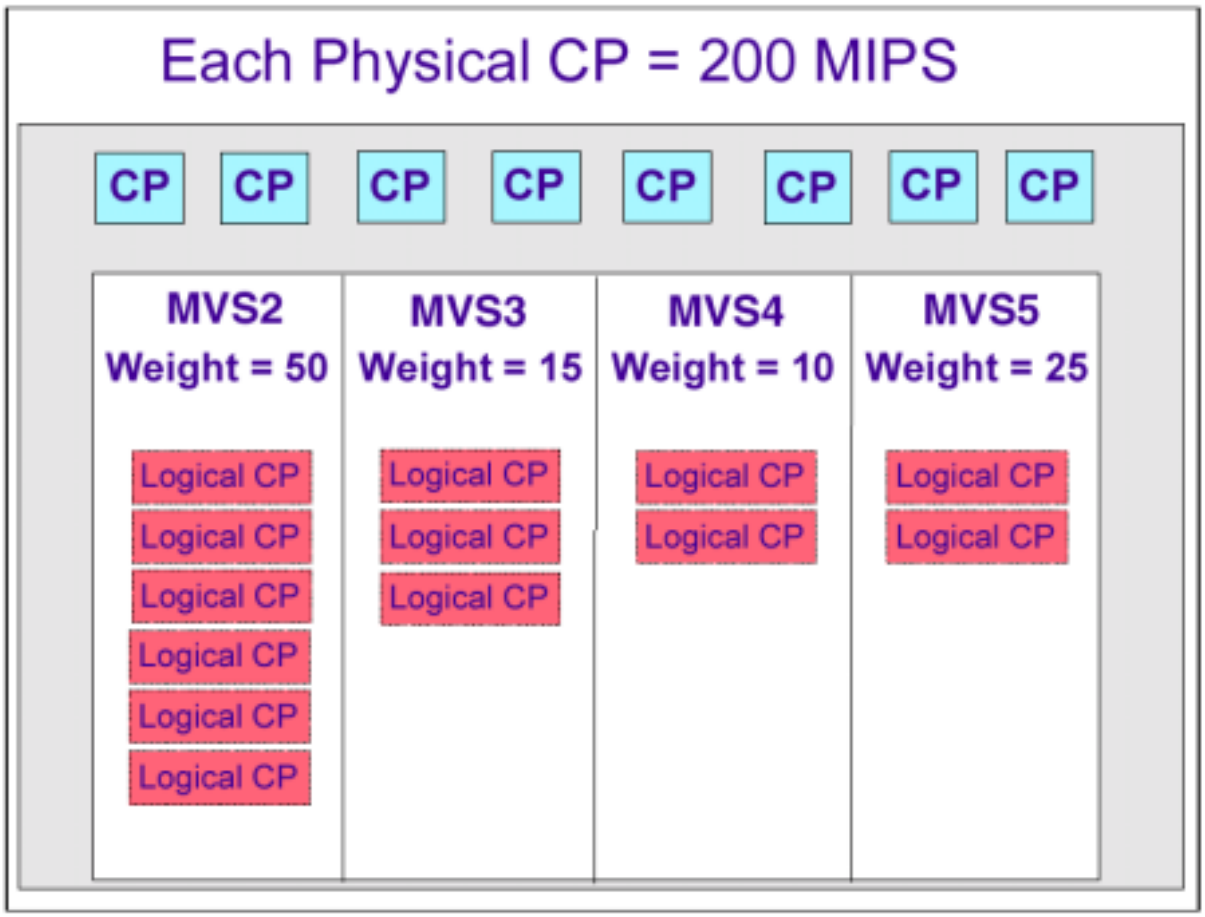
LPAR dispatching

The code that provides the LPAR dispatching function is a part of the PR/SM hypervisor.

The steps that occur in dispatching a logical CPare:

1. **The next logical CP to be dispatched is chosen from the logical CP ready queue based on the logical CP weight.**
2. **LPAR LIC dispatches the selected logical CP (LCP5 of MVS2 LPAR) on a physical CP in the CPC (CP0 abovel).**
3. **The z/OS dispatchable unit running on that logical processor (MVS2 logical CP5) begins to execute on physical CP0. It executes until its time slice (generally between 12.5 and 25 milliseconds) expires, or it enters a wait, or it is intercepted for some reason.**
4. **the logical CP keeps running until it uses all its time slice. At this point the logical CP5 environment is saved and control is passed back to LPAR LIC, which starts executing on physical CP0 again.**
5. **LPAR LIC determines why the logical CP ended execution and requeues the logical CP accordingly. If it is ready with work, it is requeued on the logical CP ready queue and step 1 begins again.**

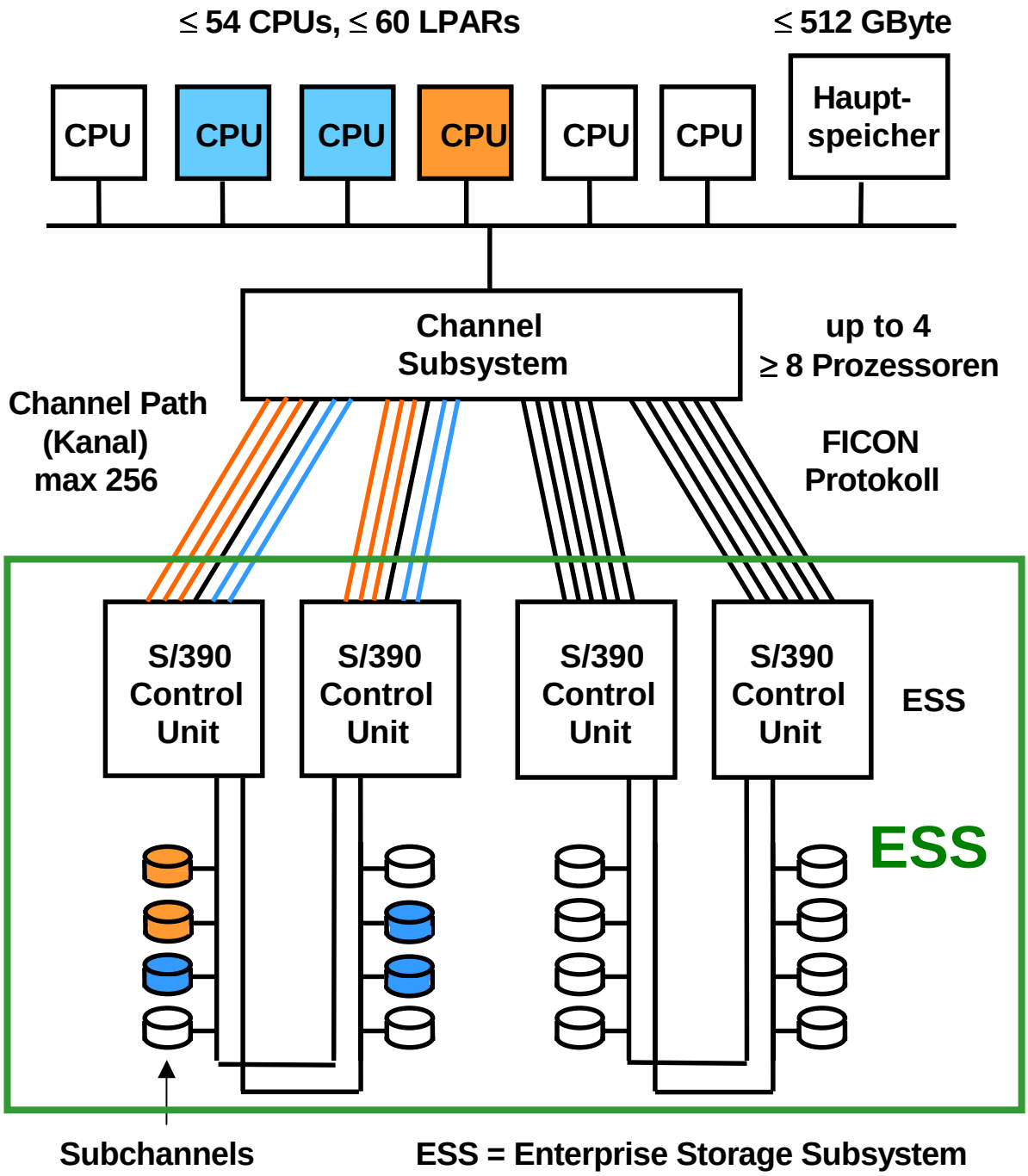
This process occurs on each physical CP.



LPAR weights are used to control the distribution of shared CPs between LPs.

LPAR weights determine the guaranteed (minimum) amount of physical CP resource an LP should receive (if needed). This guaranteed figure may also become a maximum when all the LPs are using all of their guaranteed amount (for example, if all LPs were completely CPU-bound).

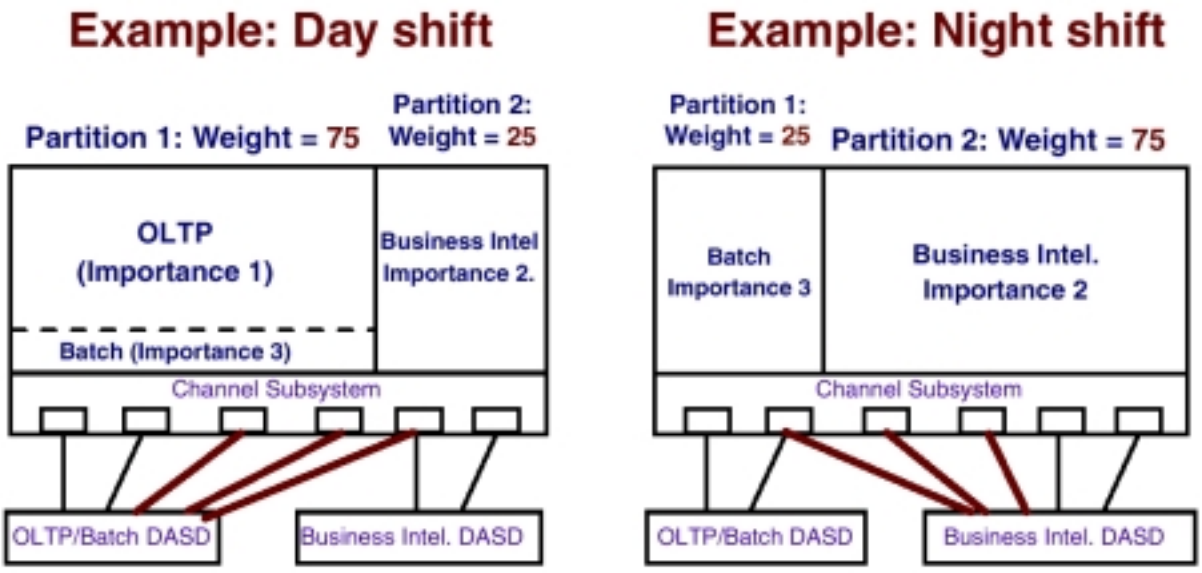
An LPAR may use less than the guarantee if it does not have much work to do. Similarly, it can use more than its weight if the other LPARs are not using their guaranteed amount.



System z Plattenspeicher Anschluss

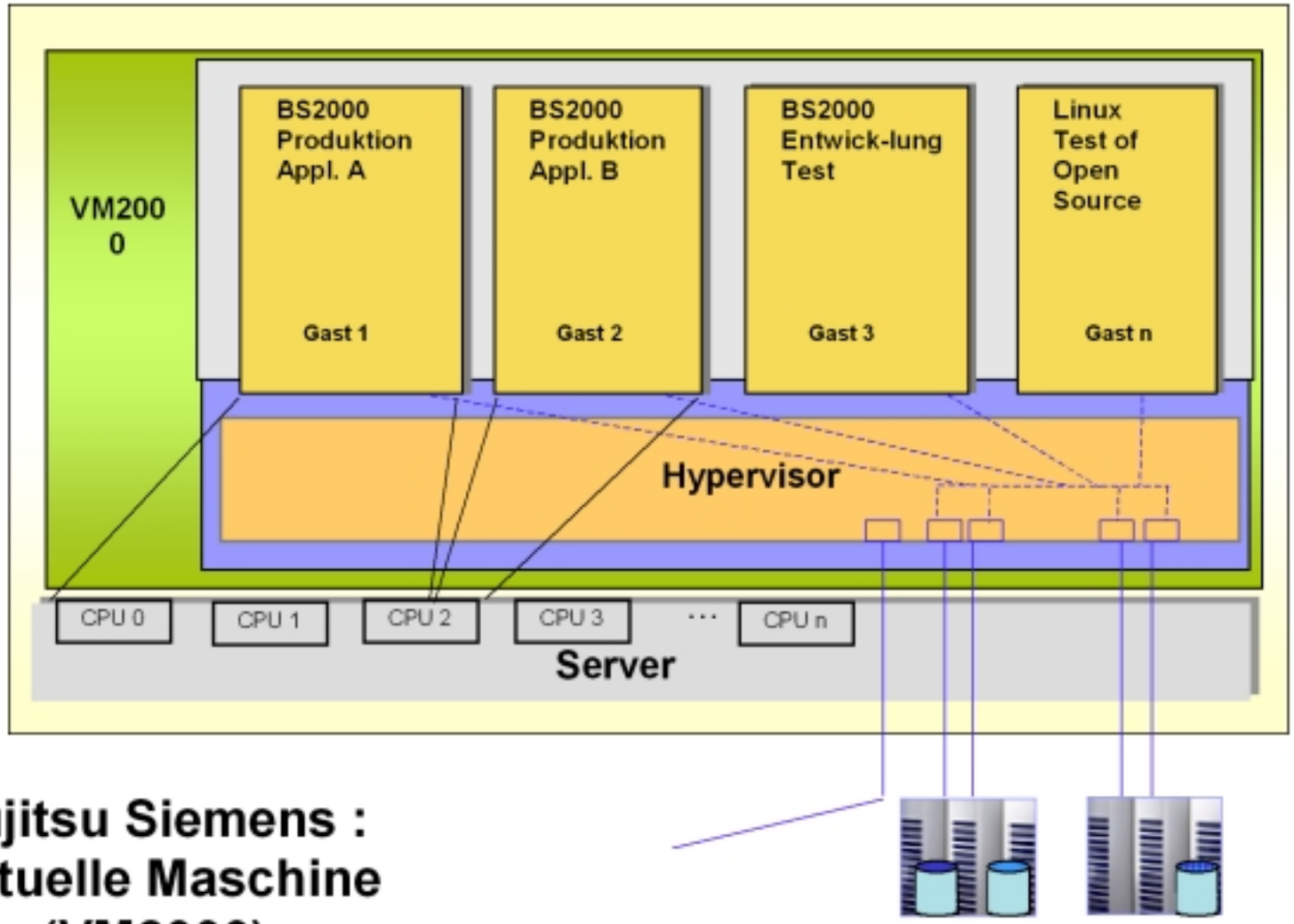
Plattenspeicher sind den LPARs fest zugeordnet. IRD kann Kanäle bei Bedarf dynamisch den einzelnen LPARs zuordnen.

S/390 Control Units werden vom ESS abgebildet (emuliert).



Dynamic Channel Path Management (DCM)

Dynamic Channel Path Management (DCM) is designed to dynamically adjust the channel configuration in response to shifting workload patterns.

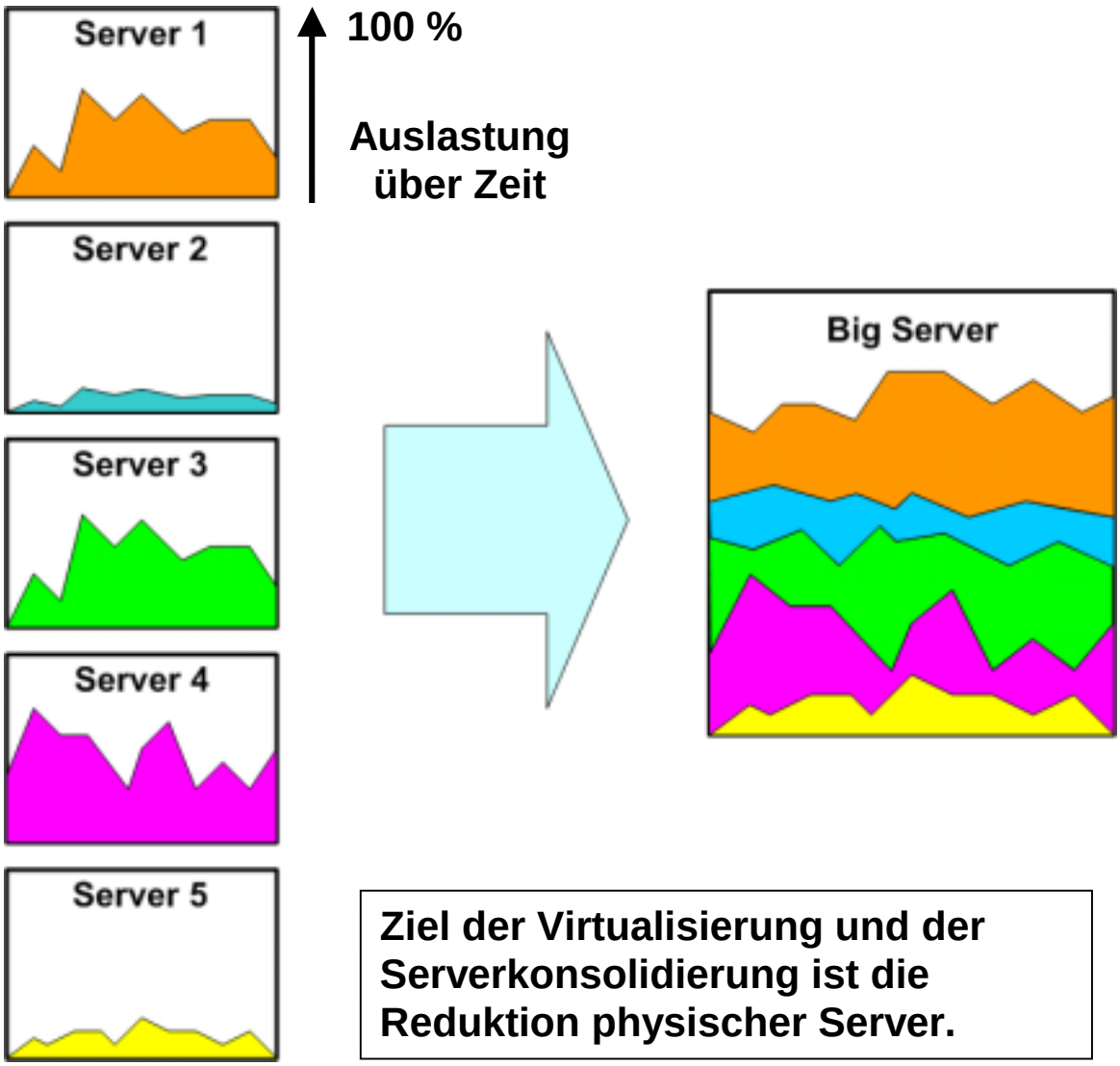


Fujitsu Siemens : Virtuelle Maschine (VM2000)

Bandarchiv

Storage Subsysteme

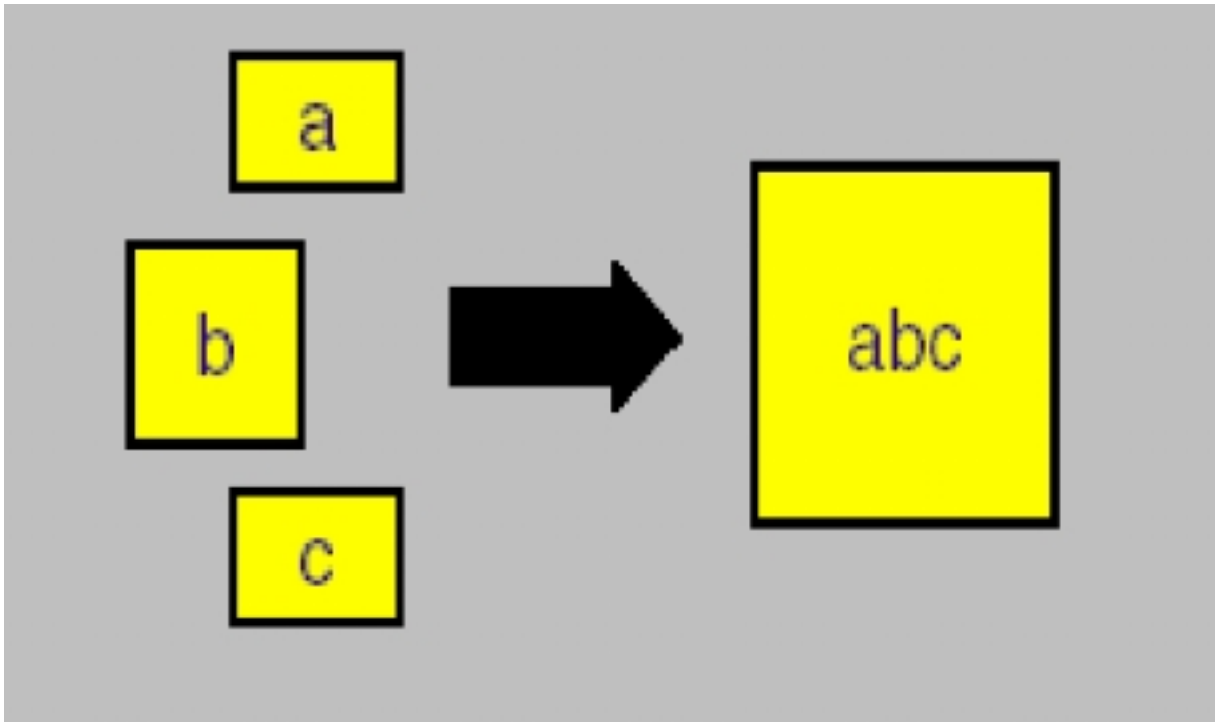
Rezentralisierung



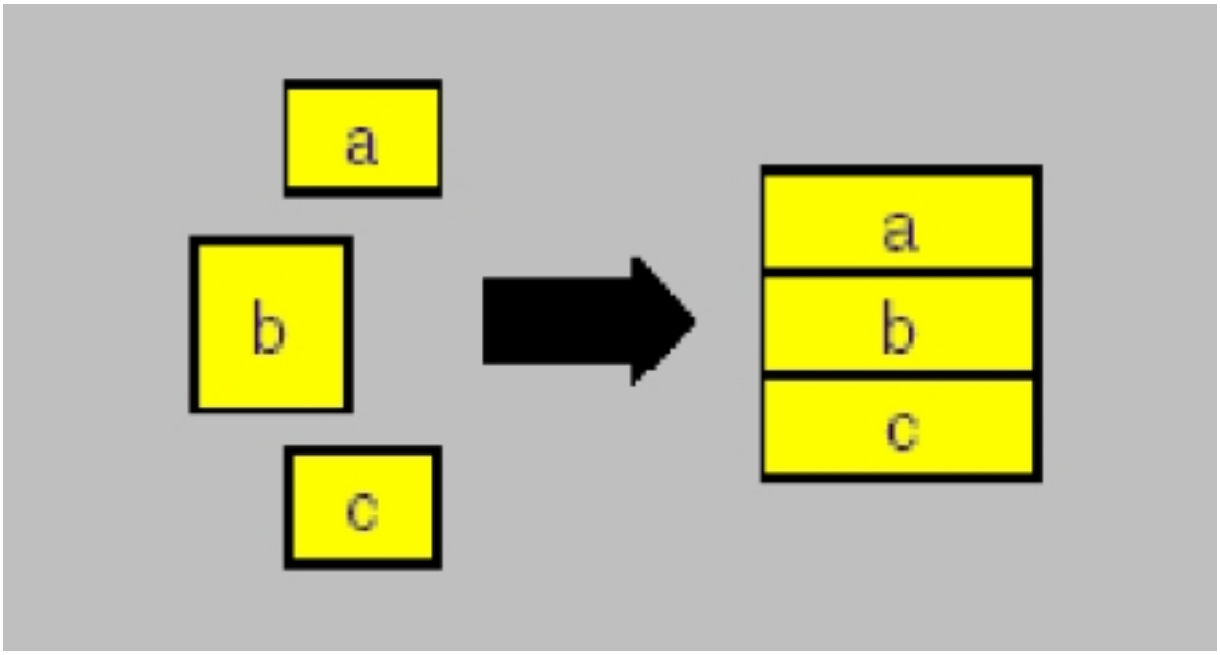
Ziel der Virtualisierung und der Serverkonsolidierung ist die Reduktion physischer Server.

Es soll durch den Pooling-Effekt und die gemeinsame Nutzung von Ressourcen eine bessere Auslastung erreicht werden. Der Platzbedarf und der Stromverbrauch werden deutlich reduziert. Die Verwaltung wird vereinheitlicht und effizienter.

Ein Schlüsselbegriff in diesem Zusammenhang ist die bessere Nutzung des so genannten "White Spaces", das sind die freien Kapazitäten einer bestimmten Umgebung, die nur genutzt werden können, wenn die entsprechenden Ressourcen von mehreren Anwendungen parallel gebraucht werden können.

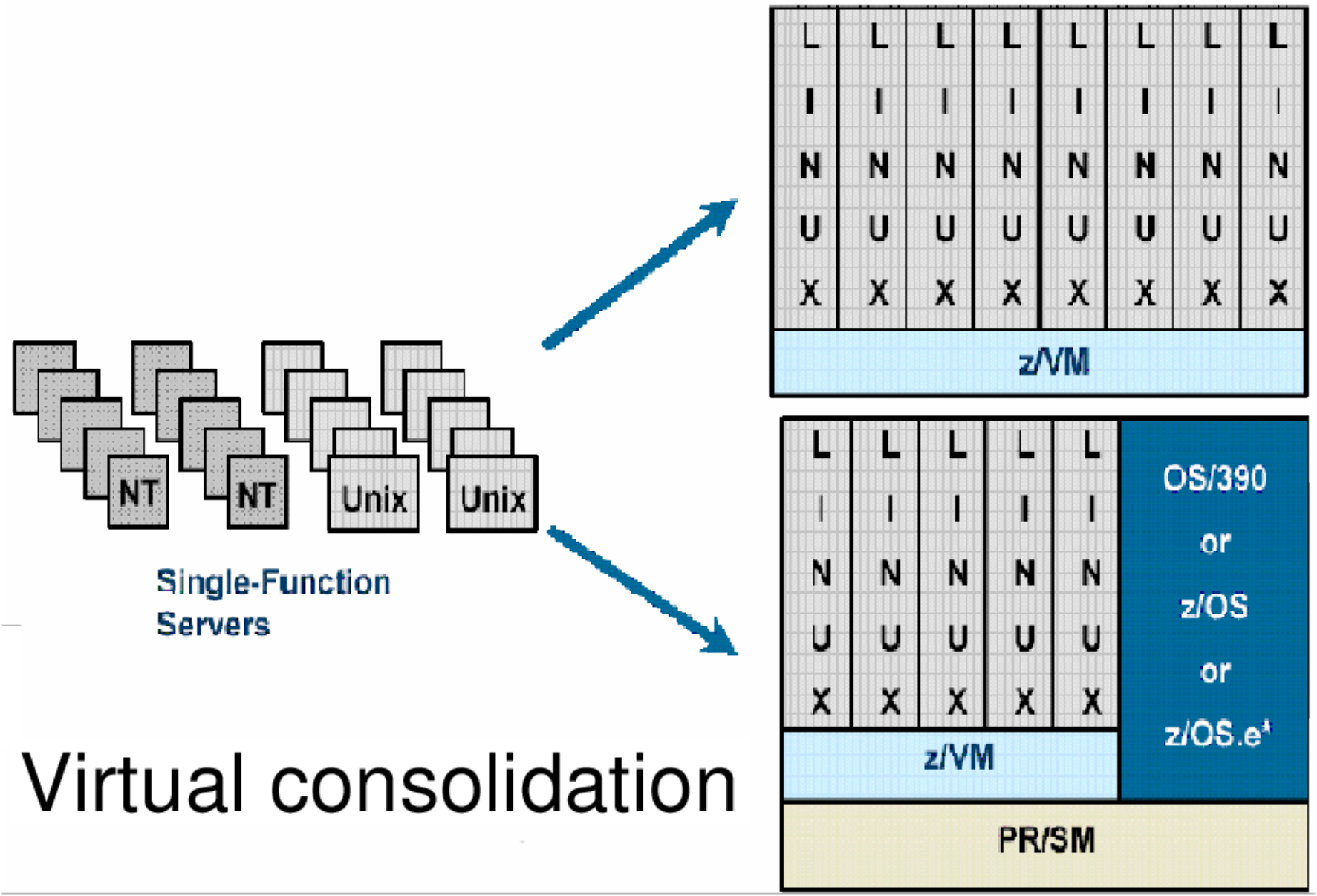


Volle Konsolidierung



Virtuelle Konsolidierung

Virtual consolidation



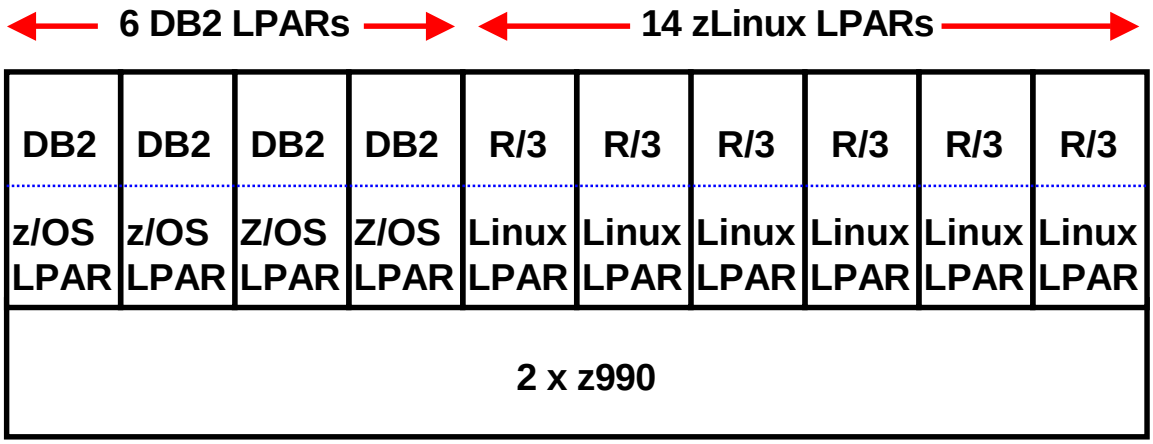
Beispiel: SAP R/3

August 2004

Fa. Endress+Hauser, Weil am Rhein.

Zwei zSeries 990 Mainframes, zusammen 328 GByte Hauptspeicher.

19 Produktions-Systeme SAP R/3, verteilt auf 14 logische Partitionen (LPARs), laufen unter zLinux . Weitere 20 Test-Systeme SAP R/3. Die angebundenen SAP-R/3 Datenbanken auf Basis von DB2 sind auf 6 logische Partitionen verteilt.



Fa. Endress+Hauser, Weil am Rhein.

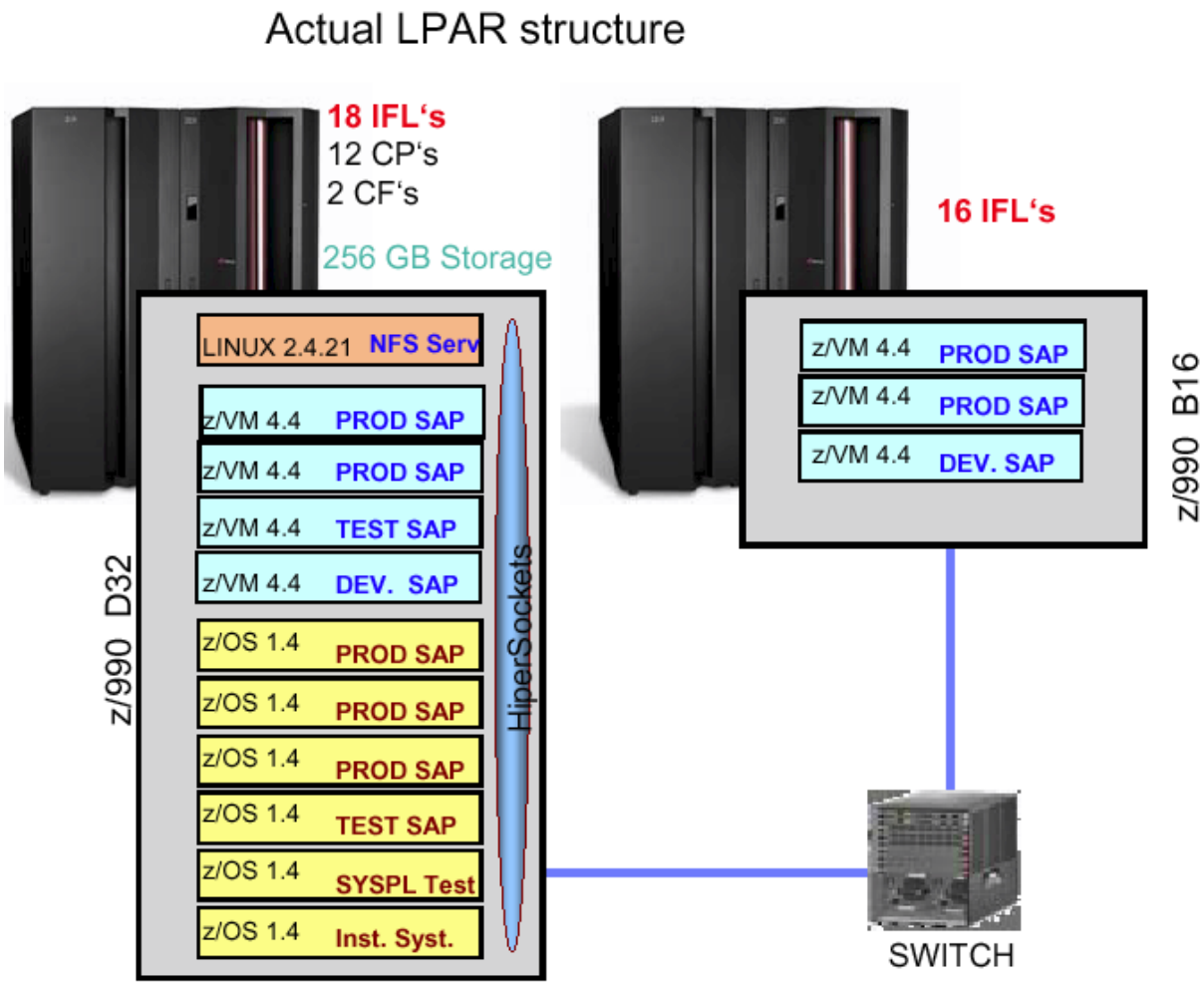
August 2004

Weltweit führender Anbieter von Messgeräten und Automationslösungen für industrielle Verfahrenstechnik.

Auf dem primären z990 Server sind 20 Prozessoren als Integrated Facilities for Linux (IFLs) konfi guriert, während mit den restlichen zwöf Prozessoren hauptsächlich SAP Datenbankserver unter z/OS ausgeführt werden. Der kleinere z990 verfügt über 16 IFLs für SAP Anwendungen und 128 GB Haupt-speicher, halb so viel wie auf dem primären Server.

Mit nur zwei Servern betreibt Endress+Hauser InfoServe 19 separate SAP Produktionssysteme und 20 SAP Test-umgebungen und unterstützt 3.700 nominierte Benutzer (drunter bis zu 2.500 gleichzeitige Benutzer). Diese Be-nutzer arbeiten in 35 Tochtergesellschaften an 71 Standorten weltweit und erhalten eine durchschnittliche Reaktionszeit von 0,5 Sekunden für SAP Anwendungen.

Die IBM HiperSockets-Technologie bietet extrem schnelle und sichere Konnektivität zwischen den SAP Anwendungs- und Datenbankservern.



SAP/R3 und zLinux
Fa. Endress+Hauser, Weil am Rhein.

August 2004

Zwei zSeries 990 Mainframes, zusammen 328 GByte Hauptspeicher.

19 Produktions-Systeme SAP R/3, verteilt auf 14 logische Partitionen (LPARs), laufen unter zLinux . Weitere 20 Test-Systeme SAP R/3. Die angebundenen SAP-R/3 Datenbanken auf Basis von DB2 sind auf 6 logische Partitionen verteilt.