

# **Mainframe Internet Integration**

**Prof. Dr. Martin Bogdan  
Prof. Dr.-Ing. Wilhelm G. Spruth**

**SS2013**

**Virtualisierung Teil 3**

**Logische Partitionen**

# Firmware

Wenn Sie einen Access Point für Ihr privates WLAN kaufen, erhalten Sie eine Box, die u.a. einen Microprozessor mit umfangreichen Code (häufig in EPROMs gespeichert) enthält. Das Betriebssystem ist in vielen Fällen eine Linux Variante.

Als Benutzer können sie ein derartiges Gerät konfigurieren (z.B. eine IP Adresse eingeben). Sie können es jedoch nicht programmieren oder eigenen Code installieren.

Ein derartiges Gerät könnte ganz in Hardware implementiert sein, ohne dass sie den Unterschied wahrnehmen würden. Microprozessor Code mit derartigen Eigenschaften ( vom Benutzer nicht programmierbar, änderbar oder einsehbar) wird als **Firmware** bezeichnet. Auch ein Administrator kann in der Regel Firmware nicht einsehen oder modifizieren, ggf. aber Konfigurations-Maßnahmen durchführen. Beispielsweise kann ein Internet Router einen in Firmware implementierten Browser enthalten, den ein Administrator benutzen kann, um Einstellungen vorzunehmen.

# HSA und Firmware

System z benutzt neben den CPUs weitere , unsichtbare Prozessoren, welche den Funktionsablauf steuern. Diese Prozessoren werden – im Gegensatz zu den CPUs – als „System Assist Prozessoren“ (**SAPs**) bezeichnet. Beispielsweise kann ein zEC12 Rechner 101 CPUs und 16 SAPs enthalten.

Auf SAPs können keine Anwendungen laufen, und ein Benutzer kann keine Programme für SAPs schreiben. SAPs werden z.B. für die Ein/Ausgabe Steuerung, für Fehler-Behandlungsroutinen – und für Virtualisierungszwecke – benutzt.

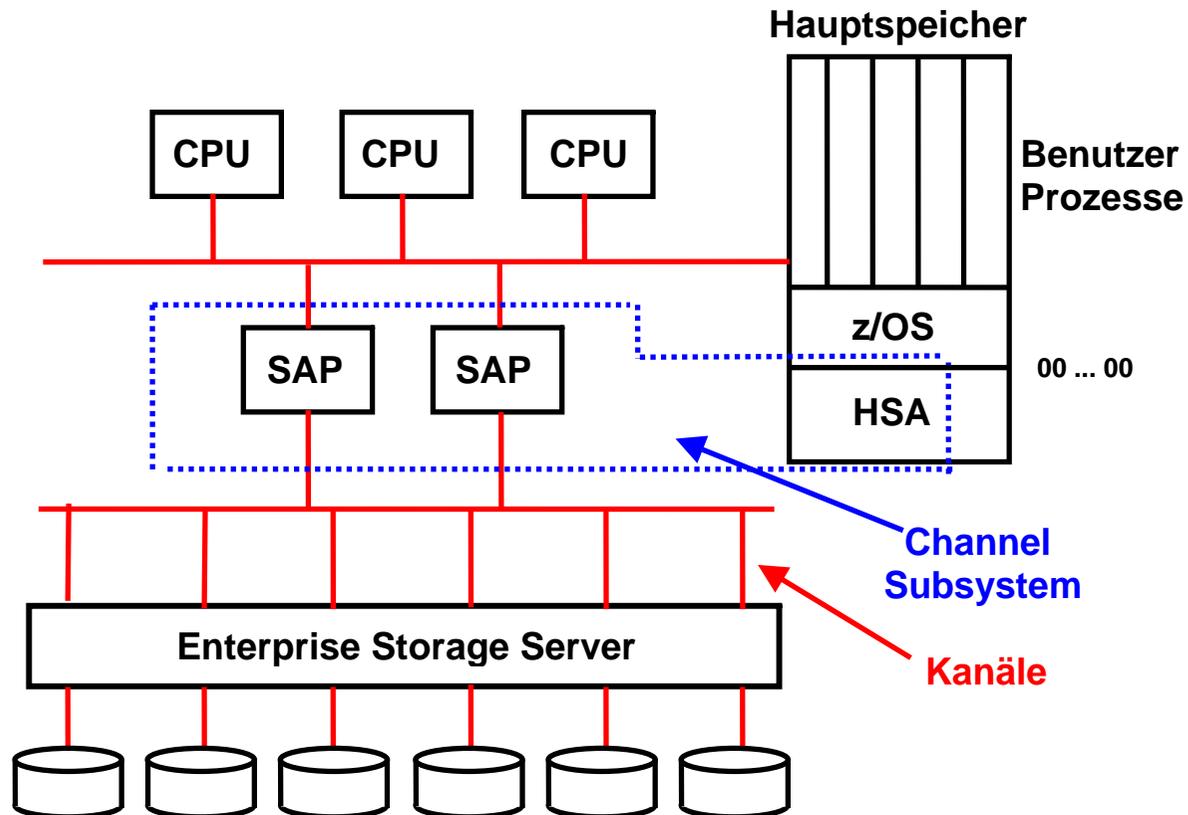
Der von den SAPs ausgeführte Code wird als **Firmware** bezeichnet. (Der Coupling Facility Control Code (CFCC) ist ebenfalls Firmware). Firmware wird ähnlich wie Microcode als Bestandteil der Hardware betrachtet. Der Unterschied zwischen Firmware und Microcode besteht darin, dass Firmware reguläre System z Maschinenbefehle unterstützt und ausführt. Der Firmware Code befindet sich in einem getrennten Speicher, der als Hardware System Area (**HSA**) bezeichnet wird. Firmware wird von IBM auch als „Licensed Internal Code“ (LIC) bezeichnet. Itanium (und DEC Alpha) verwendet statt LIC die Bezeichnung PAL (Privileged Architecture Library). Itanium enthielt ursprünglich einen in PAL implementierten x86 Emulator.

Heutige Implementierungen verwenden einen einzigen physischen Speicher, der in zwei getrennte logische Speicher aufgeteilt wird:

- Einen realen Speicher, der das Betriebssystem aufnimmt und in dem die virtuellen Speicher der Benutzerprozesse abgebildet werden.
- Eine Hardware System Area (HSA), die Firmware aufnimmt.

Die von den SAPs und ihrem HSA Code ausgeführte Ein/Ausgabe Steuerung wird als Channel Subsystem bezeichnet. Es bildet das virtuelle I/O Subsystem, mit dem der Betriebssystem Kernel glaubt zu arbeiten, auf die reale I/O Struktur ab. Unabhängig von System- und Benutzercode sind damit umfangreiche Optimierungen der Plattenspeicher Zugriffe möglich.

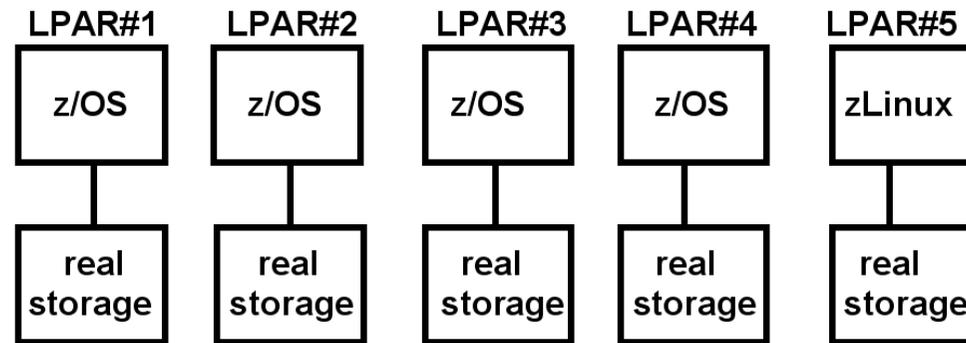
# zSeries Ein/Ausgabe Anschluss



Die HSA (Hardware System Area) ist ein Teil des System z physischen Hauptspeichers. Sie liegt außerhalb des Adressenraums, auf den die CPUs zugreifen können. Der zEC12 Rechner hat eine 32 GByte große HSA.

Der in der HSA laufende Code wird ähnlich wie der in der Coupling Facility laufende Code als Firmware bezeichnet. Er wird von SAPs ausgeführt

SAPs führen z.B. Channel Subsystem Code aus oder implementieren Diagnostische oder Fehlerbehandlungs-Aufgaben. Auch der PR/SM Hypervisor Code wird von SAPs ausgeführt.

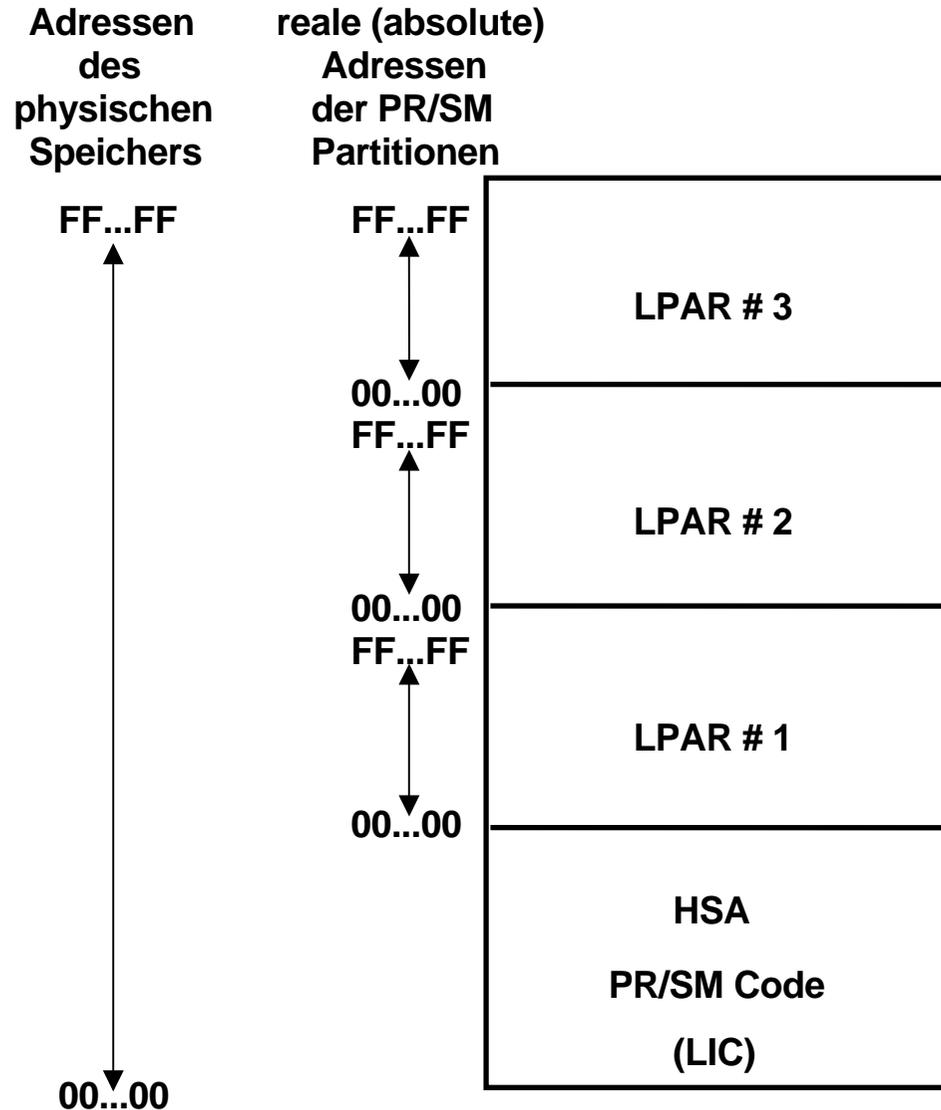


## Logical Partition (LPAR) und PR/SM

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 System z „PR/SM“ (Processor Resource/System Manager) Einrichtung verwendet ein anderes, als **LPAR** (Logical Partition) bezeichnetes Verfahren. PR/SM (ausgesprochen *pri`sm*) hat die Funktion eines Hypervisors (Host Kernel). PR/SM ist als Firmware implementiert, die in der HSA untergebracht ist und von SAPs ausgeführt wird. Für jedes Gastbetriebssystem wird eine eigene LPAR eingerichtet. Jede LPAR verfügt über einen eigenen getrennten realen Speicher. Die PR/SM Hardware stellt mehrfache (bis zu 60) LPARs mit getrennten realen Speicher zur Verfügung.

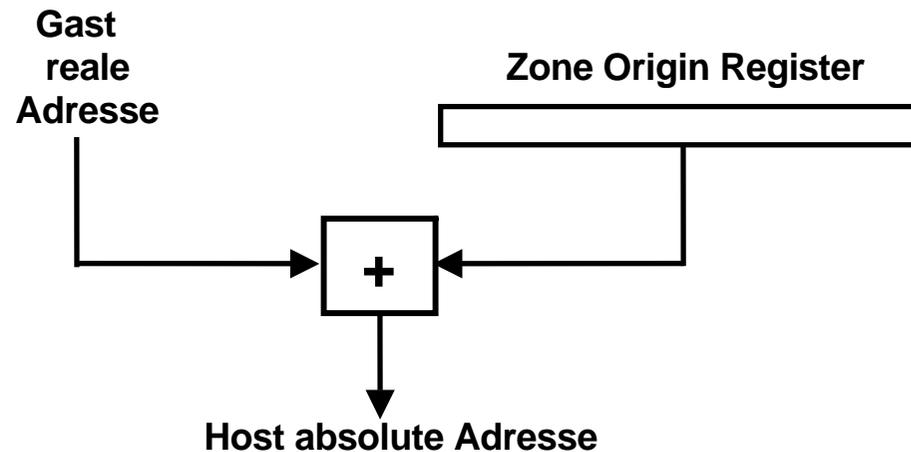
Da jede LPAR über einen eigenen realen Speicher verfügt, sind Shadow Tables oder eine doppelte Adressumsetzung nicht erforderlich.



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 Firmware 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

## Aufteilung des physischen Speichers in mehrere reale Speicher

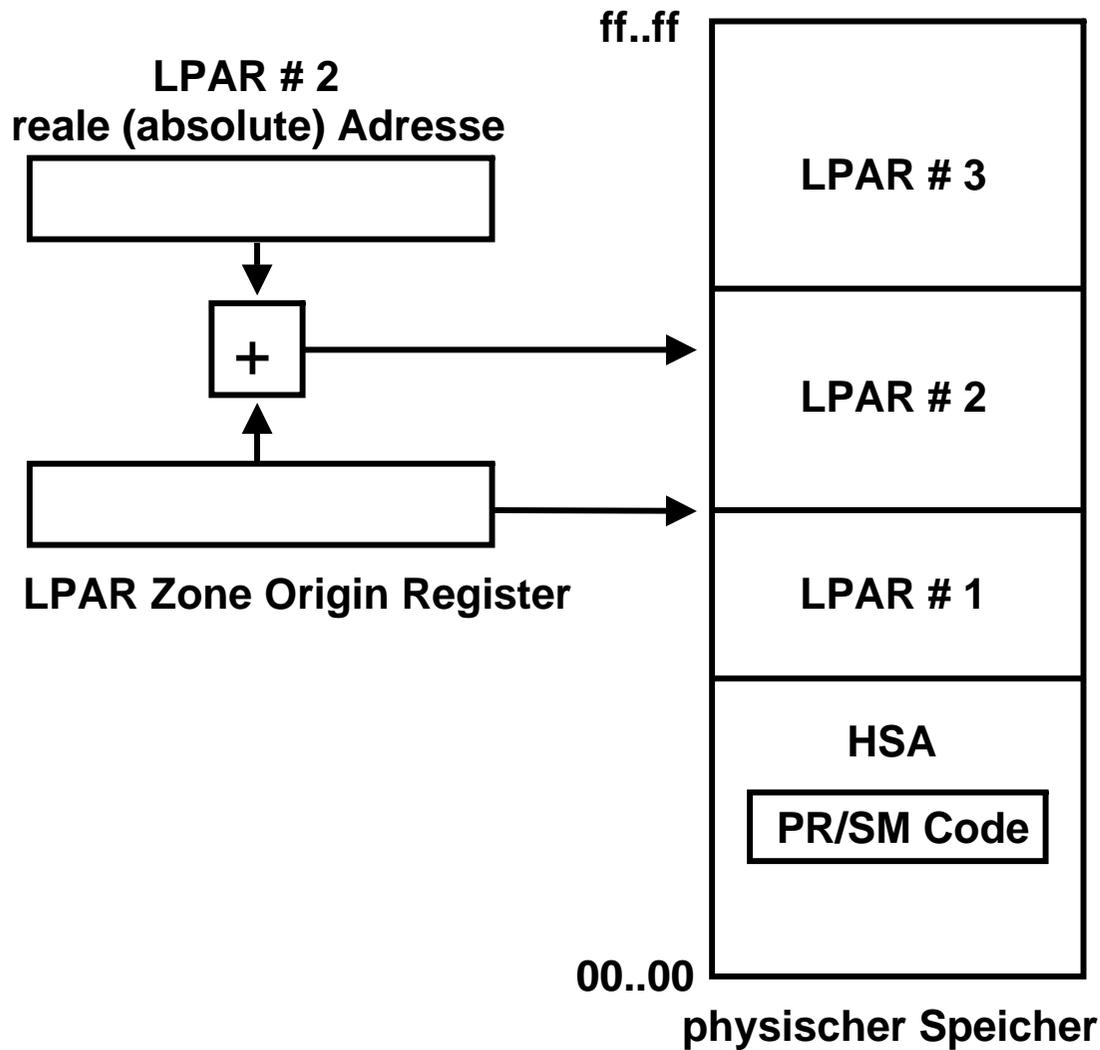


## Umsetzung der Gast Adressen in physische Adressen

In einem System z Server 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.

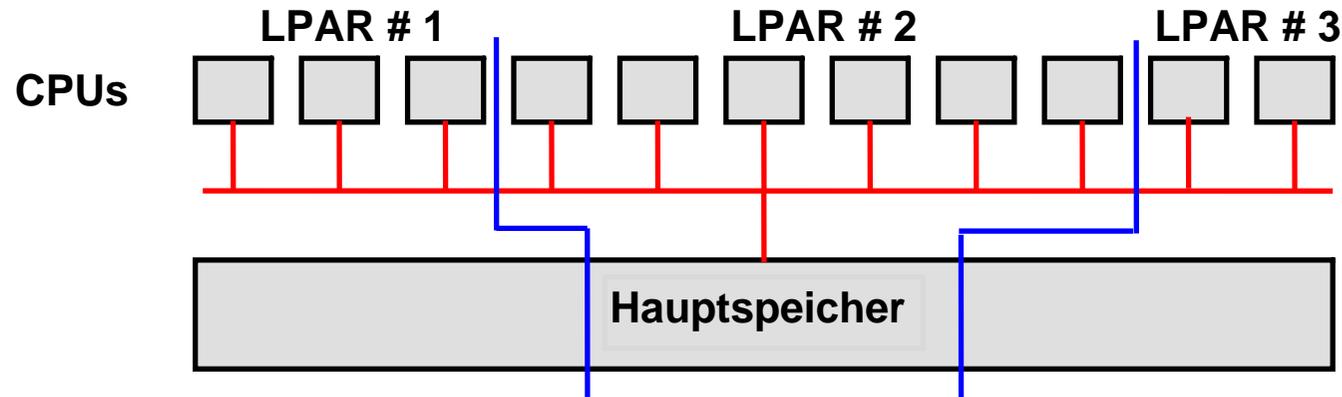
PR/SM verfügt ähnlich wie ein Betriebssystem Kernel über eine Zeitscheibensteuerung. Damit ist es möglich, auf einem Rechner mit nur einer einzigen CPU mehrere LPARs zu betreiben.



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 Hauptspeicher-Adresse definiert, die dieser LPAR zugeordnet ist.

## LPAR Speicherplatz-Verwaltung Basis Fall



## Aufteilung eines Großrechners in mehrere SMPs

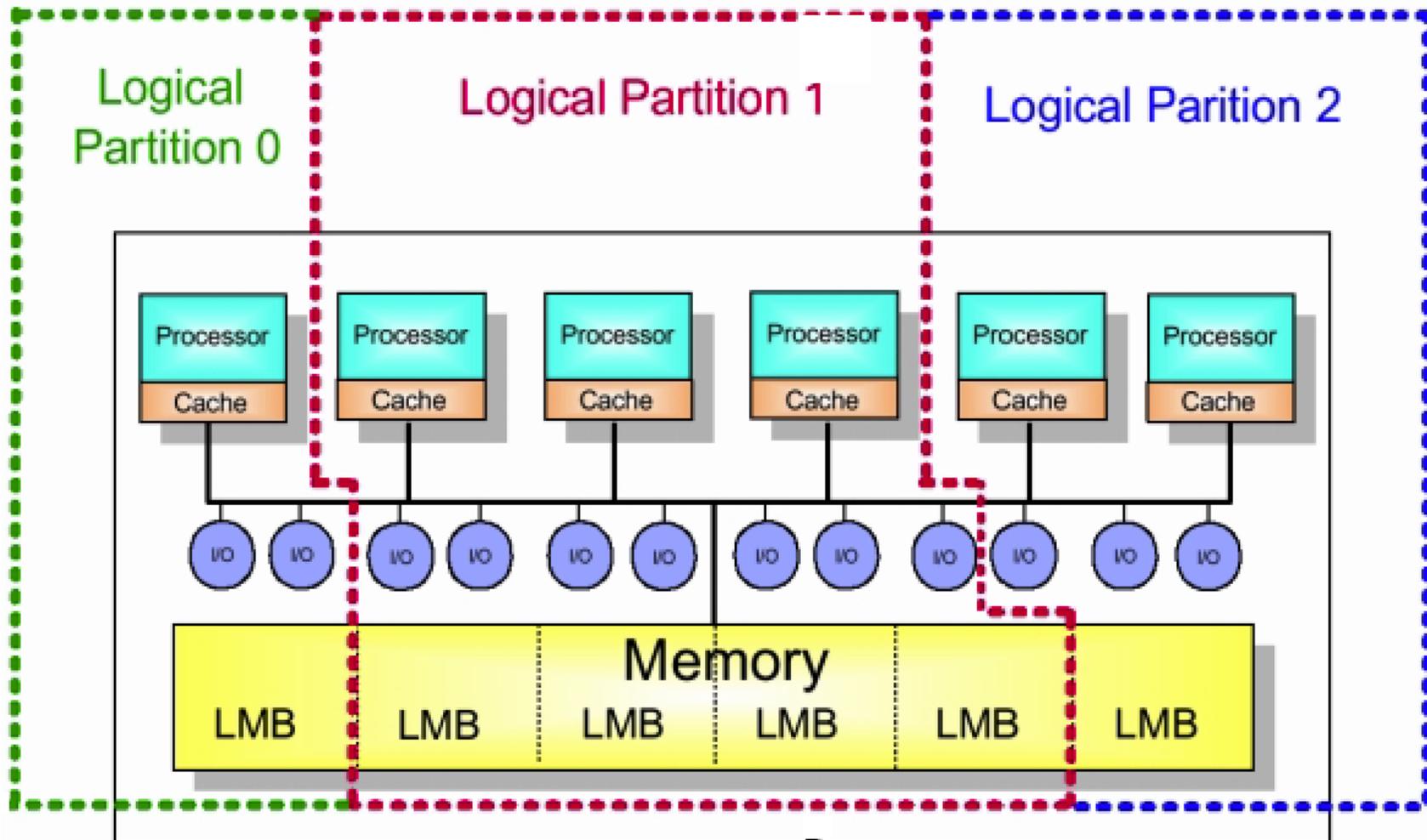
z/OS unterstützt symmetrische Multiprozessoren (SMP) mit bis zu 64 CPUs. Realistisch sind bis zu 24 – 32 CPUs. Bei Unix, Linux und Windows Betriebssystemen liegt die Grenze für Transaktions- und Datenbank Anwendungen eher bei 12 CPUs.

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.

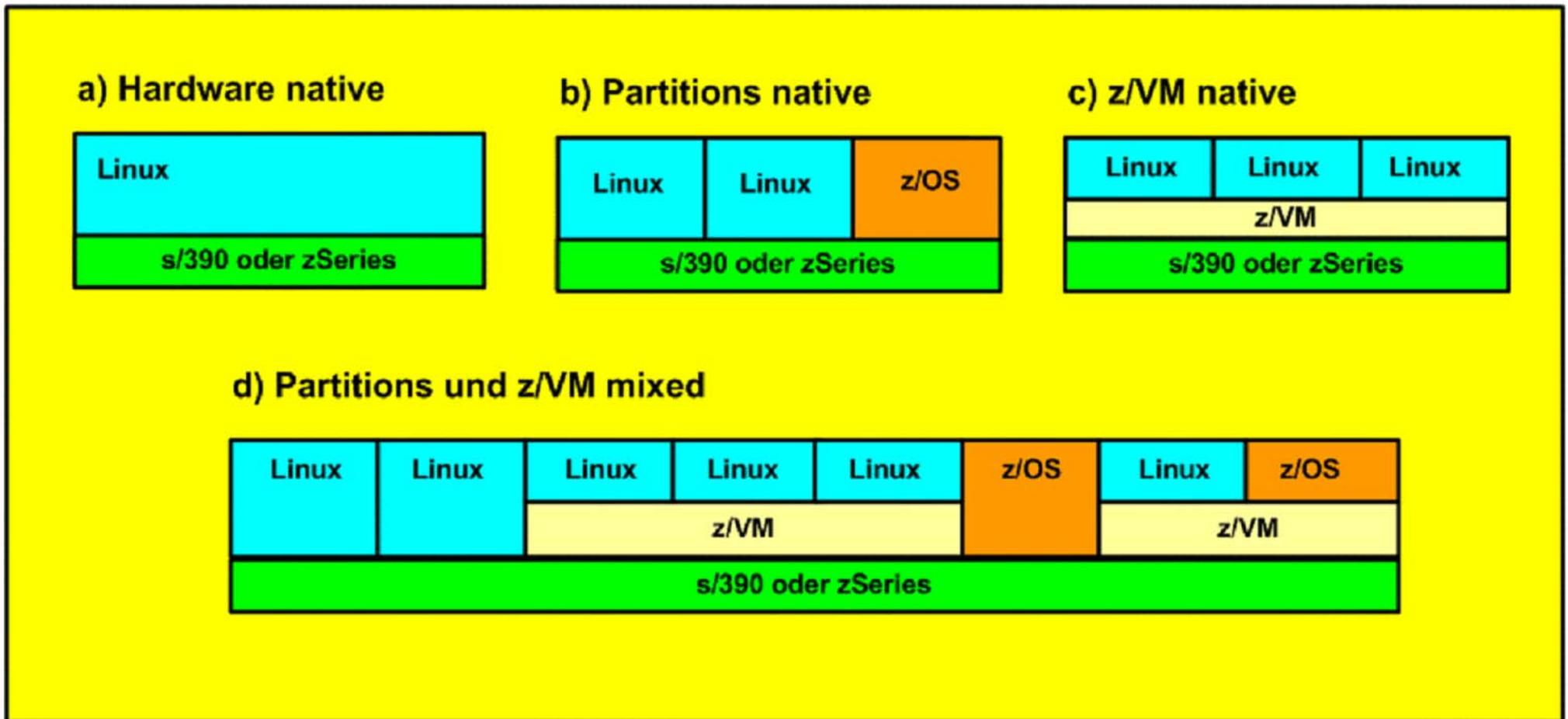
Der Systemadministrator kann den gesamten Hauptspeicher in unterschiedlichen Größen auf die einzelnen Hauptspeicher aufteilen.

Die Aufteilung des physischen Speichers auf die einzelnen LPARs erfolgt (im einfachsten Fall) statisch, und wird bei der erstmaligen Inbetriebnahme des Systems festgelegt.



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 im einfachsten Fall 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.



## zLinux auf dem Mainframe

zLinux ist ein populäres Mainframe Betriebssystem. Es wird aber fast immer zusätzlich zu z/OS eingesetzt, und zwar entweder in einer z/VM virtuellen Maschine, oder in einer eigenen LPAR, oder beides. LPARs können parallel zu virtuellen Maschinen unter z/VM eingesetzt werden. Dargestellt sind unterschiedliche Alternativen.

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

leia	binks	tipc049		padme	
z/OS V. 1.8	z/OS V. 1.8	zLinux Suse SLES 10	z/OS z/OS V. 1.8	z/OS V. 1.8 DB2 V. 2.9	WebSphere
<b>LPAR #1</b> 2 Gbyte	<b>LPAR #2</b> 2 Gbyte	<b>z/VM</b> <b>LPAR #3</b> 4GByte		<b>LPAR #4</b> 4GByte	<b>LPAR #5</b> 4GByte
PR/SM					

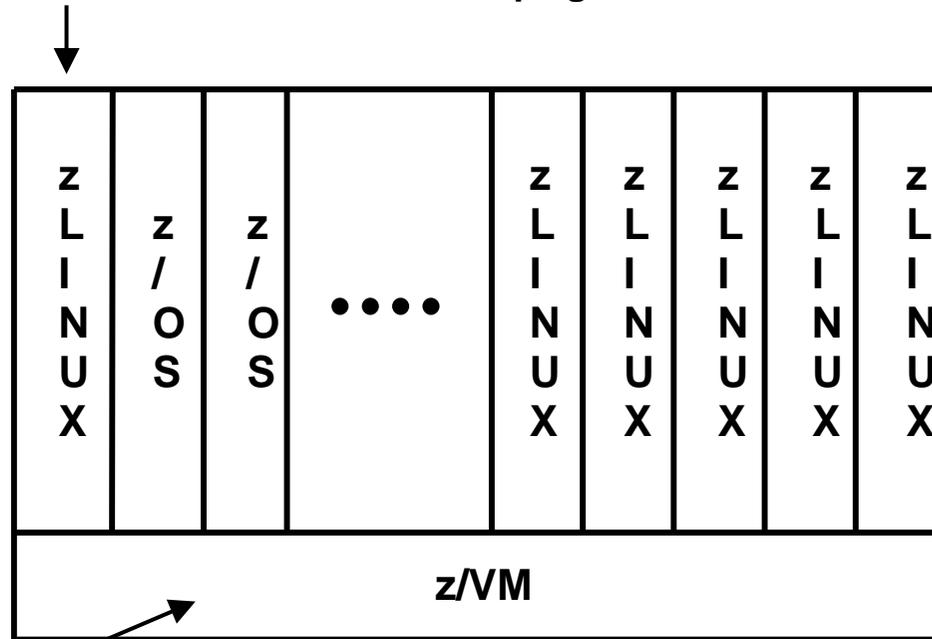
## Konfiguration des Rechners jedi.informatik.uni-leipzig.

Ein Beispiel ist die Konfiguration des Mainframe Rechners des Lehrstuhls Technische Informatik. Wir betreiben 5 LPARs. Jede LPAR hat einen eigenen Ethernet Adapter für Anschluss an das Internet :

LPAR # 1 - 139.18.4.30	leia.informatik.uni-leipzig.de	z/OS 1.8
LPAR # 2 - 139.18.4.34	binks.informatik.uni-leipzig.de	z/OS 1.5
LPAR # 3 - 139.18.4.49	tipc049.informatik.uni-leipzig.de	zVM mit zLinux und z/OS
LPAR # 4 - 139.18.4.35	padme.informatik.uni-leipzig.de	z/OS 1.8
LPAR # 5 - 139.18.4.33	Diplomarbeiten und Forschungsprojekte	z/OS 1.12

Wir haben bisher keine Notwendigkeit gesehen, eine eigene LPAR für zLinux einzurichten. Auf dem Mainframe Rechner an der Uni Tübingen existiert eine eigene zLinux LPAR.

Interner zLinux Router kenob.informatik.uni-leipzig.de = 139.18.4.49

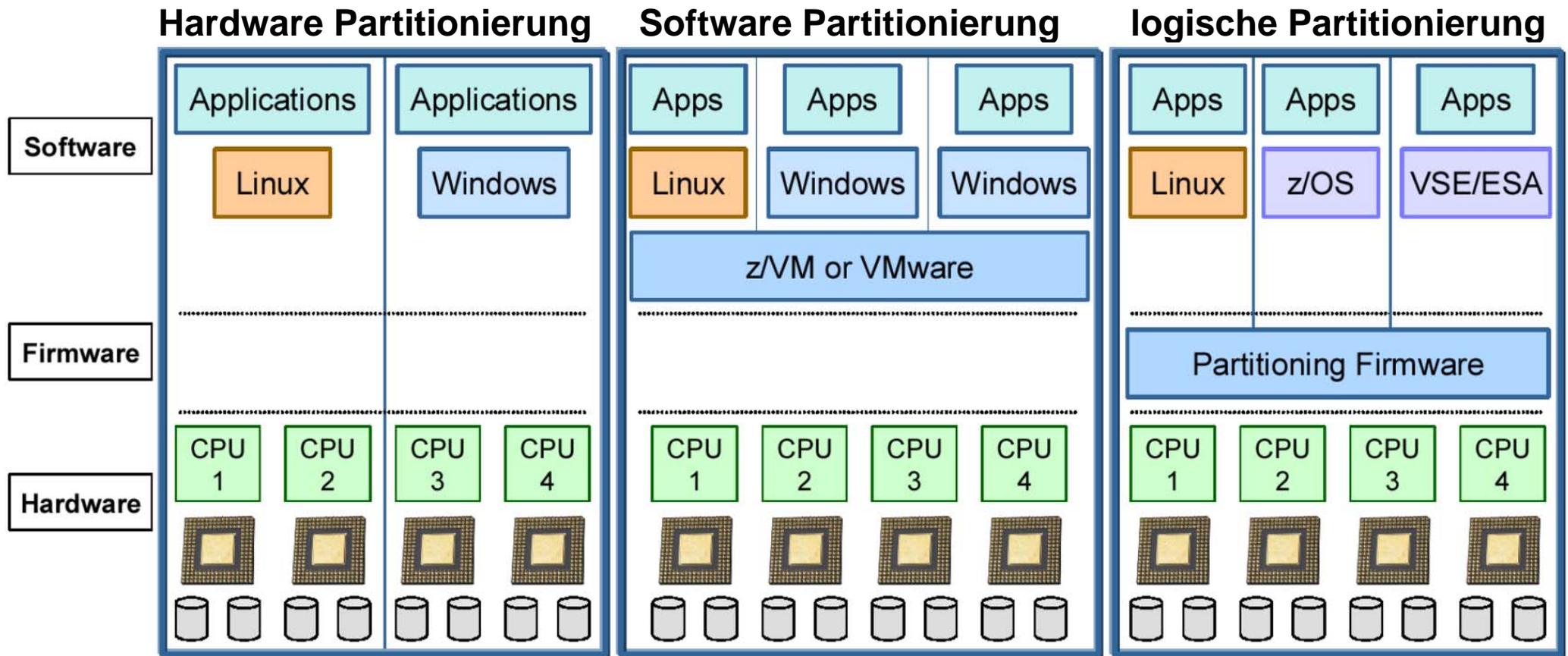


ticp049.informatik.uni-leipzig.de

**kenob.informatik.uni-leipzig.de**

Auf der LPAR # 3 - 139.18.4.49 laufen unter z/VM mehrere Gast Maschinen. Auf der ersten Gastmaschine läuft zLinux. Dieses dient als Router für alle weiteren Gastmaschinen.

Unter z/VM Gast sind etwa 20 virtuelle Maschinen installiert, teils z/OS, teils SLES v10 (Suse zLinux Enterprise Edition). Diese sind an ein internes Netz 10.x.x.x angeschlossen, welches der zLinux Router unterhält. Für einen Zugriff ruft man z/VM auf, und benutzt dort das „Dial VM“ Kommando, um sich mit einer der virtuellen Maschinen zu verbinden.



## Zusammenfassung: Unterschiedliche Formen der Partitionierung

**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 benötigt im Prinzip keine zusätzliche Hardware Unterstützung (Ausnahme Interpretive Execution Facility, Vanderpool und Pacifica).

**Logical Partitioning** benutzt Firmware für die Implementierung der Hypervisor Funktionalität. Weiterhin sind die virtuellen Maschinen nicht mehr in virtuellen Adressräumen des Host Kernels untergebracht.

## 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  
(System z) einen  
zehnjährigen  
Entwicklungs-  
vorsprung**

**Computer Zeitung  
19.5.2003, S. 1**