

# **Einführung in das System z Mainframe**

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

**2. Juli 2007**

**Humboldt Universität Berlin**

**Teil 2**

**System z Architektur**

**G. Amdahl  
G.A. Blaauw  
F.P. Brooks**

**B.O. Evans**

# **/360 Architektur**

**7. April 1964**

- **8 Bit Byte**
- **Mehrzweckregister**
- **Byte Adressierung des Hauptspeichers**
- **Überwacher**
- **Kanalprogramm**
- **Lebensdauer**

## **Was ist seit 1964 passiert ?**

- **VAX**
- **HP Precision**
- **MIPS**
- **Itanium**
- **viele andere**

**Mit dem heutigen Wissen, was hätten Amdahl, Blaauw und Brooks 1964 besser machen sollen ?**

# **Architektur - Eigentümlichkeiten**

**EBCDIC - ASCII**

**JES2/3, TSO, WebSphere, Unix System Services**

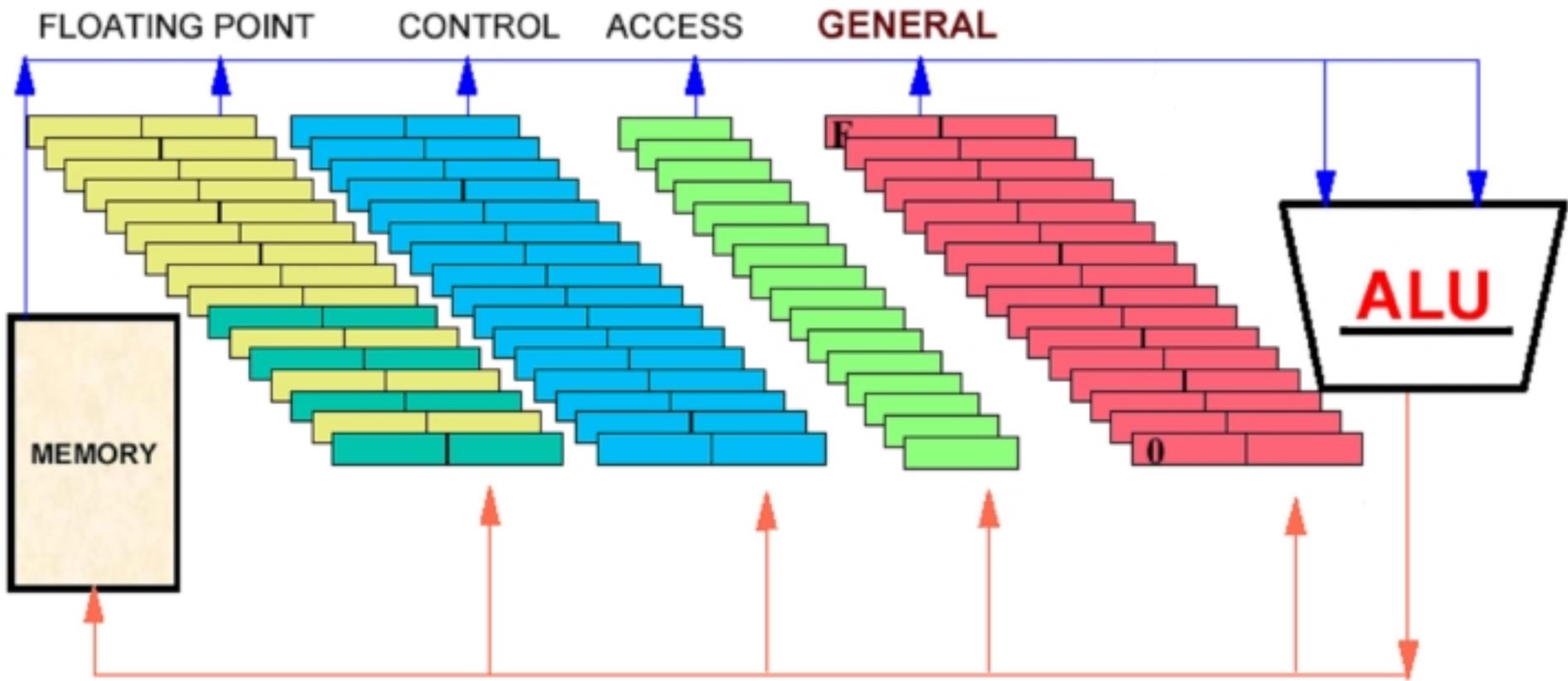
**Anwendungen in COBOL, PL/1, Assembler, REXX**

**Partitionierung, z/Linux**

**VSAM, IMS, DB/2 Datenbanken**

**3270 Bildschirmprotokoll**

**SNA / LU 6.2 und TCP/IP**



## zSeries Programmiermodell

Je 16 Gleitkomma, Steuer- und Mehrzweckregister, je 64 Bit, plus 16 Access Register

ASCII-Tabelle

	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
0	NUL	SOH	STX	ETX	EOT	ENQ	ACK	BEL	BS	HT	LF	VT	FF	CR	SO	SI
1	DLE	DC1	DC2	DC3	DC4	NAK	SYN	ETB	CAN	EM	SUB	ESC	FS	GS	RS	US
2	SP	!	"	#	\$	%	&	'	(	)	*	+	,	-	.	/
3	0	1	2	3	4	5	6	7	8	9	:	;	<	=	>	?
4	@	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O
5	P	Q	R	S	T	U	V	W	X	Y	Z	Ä	Ö	Ü	^	_
6	`	a	b	c	d	e	f	g	h	i	j	k	l	m	n	o
7	p	q	r	s	t	u	v	w	x	y	z	ä				



EBCDIC-Tabelle

	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
0	NUL	SOH	STX	ETX	PF	HT	LC	DEL			SMM	VT	FF	CR	SO	SI
1	DLE	DC1	DC2	TM	RES	NL	BS	IL	CAN	EM	CC	CU1	IFS	IGS	IRS	IUS
2	DS	SOS	FS		BYP	LF	ETB	ESC			SM	CU2		ENQ	ACK	BEL
3			SYN		PN	RS	UC	EOT				CU3	DC4	NAK		SUB
4	SP										ø	.	<	(	+	
5	&										!	\$	*	)	:	~
6	-	/										.	%	_	>	?
7											:	#	@	'	=	*
8		a	b	c	d	e	f	g	h	i						
9		j	k	l	m	n	o	p	q	r						
A		-	s	t	u	v	w	x	y	z						
B																
C		A	B	C	D	E	F	G	H	I						
D		J	K	L	M	N	O	P	Q	R						
E	\		S	T	U	V	W	X	Y	Z						
F	0	1	2	3	4	5	6	7	8	9						



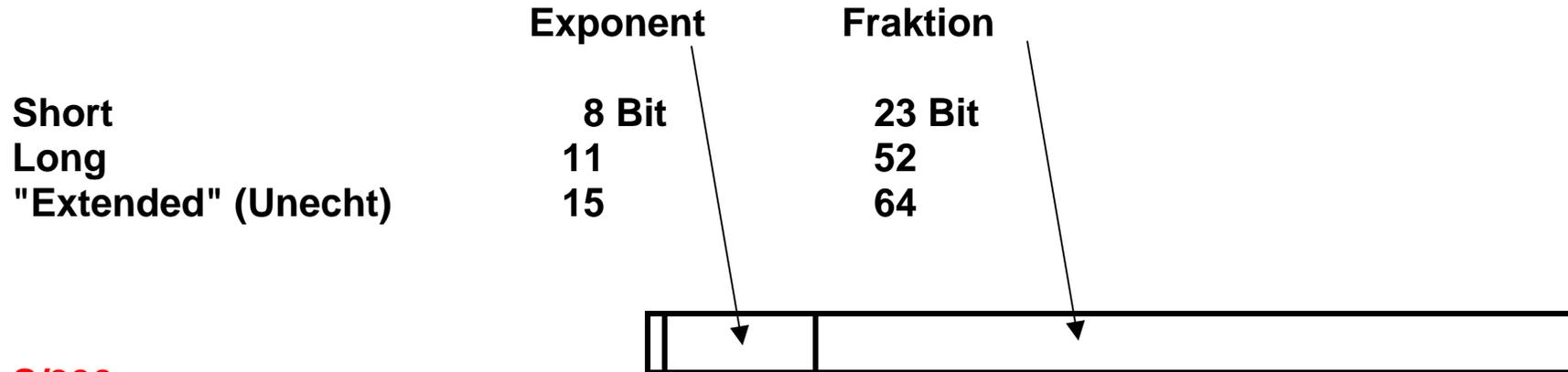
## ASCII- und EBCDIC Zeichentabellen

Beispiele:    ASCII    **R = Hex 52** ;  
                   EBCDIC **R = Hex D9** ;

Weltweit sind etwa 60% aller wirtschaftlich relevanten Daten im 8 Bit EBCDIC Standard (Extended Binary Coded Decimal Interchange Code) abgespeichert. Etwa 40% aller wirtschaftlich relevanten Daten sind im 7 Bit ASCII Standard (American Standard Code for Information Interchange) bzw. seiner 8 bit Erweiterung abgespeichert.

# GLEITKOMMA - STANDARDS

## IEEE 754



## S/390

	Exponent	Fraktion
Short	7 Bit	24 Bit
Long	7	56
Extended (Echt)	7	112

Zahlenbereich von etwa  $10^{-79}$  .....  $10^{+75}$

Die meisten Mikroprozessor - Architekturen verwenden den IEEE 754 Standard.

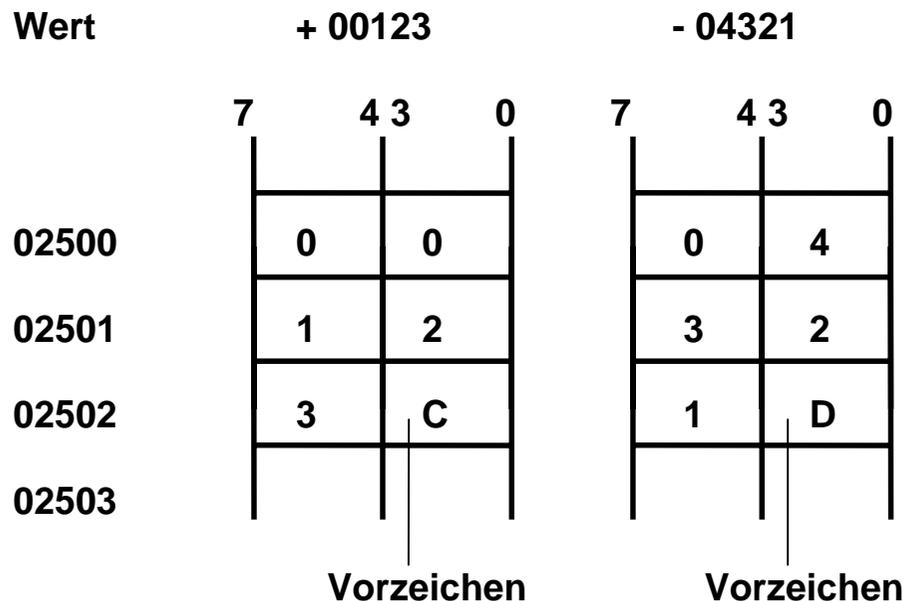
zSeries verwendet IEEE 754 zusätzlich zum S/390 Standard.

# Dezimalarithmetik

Darstellung als „gepackte dezimale Zeichenfolge“, 0 .. 31 Ziffern. 2 Ziffern pro Byte (eine Ziffer pro Nibble)

Hauptspeicheradresse der Zeichenfolge zeigt auf das werthöchste Byte. Bytes mit ansteigender Adresse beinhalten Ziffern mit abnehmenden Stellenwert. Das letzte Nibble enthält das Vorzeichen

Beispiel: Gepackte dezimale Zeichenfolge auf Hauptspeicheradresse 02500

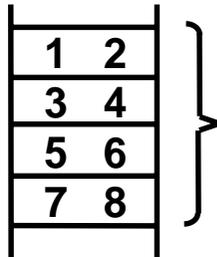


Identische Darstellung bei zSeries, BS/2000 und OS/400

Adresse

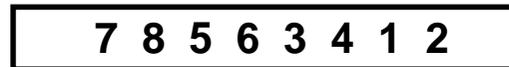
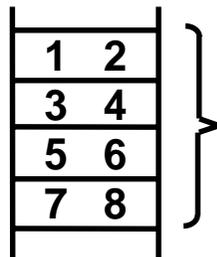
Haupt-  
speicher-  
inhalt

004710  
11  
12  
13



**Big Endian**

004710  
11  
12  
13



**Little Endian**

## Byte Ordering

Wenn Halbworte oder Worte im Hauptspeicher gespeichert sind, dann befindet sich an der adressierten Hauptspeicherstelle:

- Das wertniedrigste Byte bei Little Endian Rechnern
- Das werthöchste Byte bei Big Endian Rechnern

Die Bytes eines Halbwortes oder Wortes werden bei Little Endian Rechnern in umgekehrter Reihenfolge abgespeichert wie bei Big Endian Rechnern.

# Little Endian

Das wertniedrigste Byte wird adressiert. Beispiele:

DEC Alpha

DEC VAX

Intel Pentium, Pentium Pro

Intel 80860

# Big Endian

Das werthöchste Byte wird adressiert. Beispiele:

HP Precision

IBM ESA/390

IBM/Motorola PowerPC (mit little Endian Option)

MIPS 10000 (mit little Endian Option)

Motorola 68040

Motorola 88110 (mit little Endian Option)

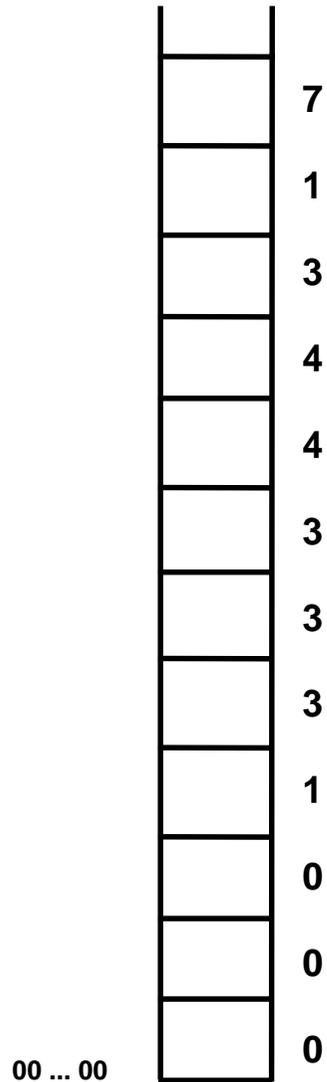
Sun Sparc

Beim Internet (TCP/IP) wird der Big Endian Standard eingesetzt (Network Byte Order).

# Hardware Protection Key Programm Status Word

**PSW Key:** Bits 8-11 form the access key for storage references by the CPU. If the reference is subject to key-controlled protection, the PSW key is matched with a storage key when information is stored or when information is fetched from a location that is protected against fetching.

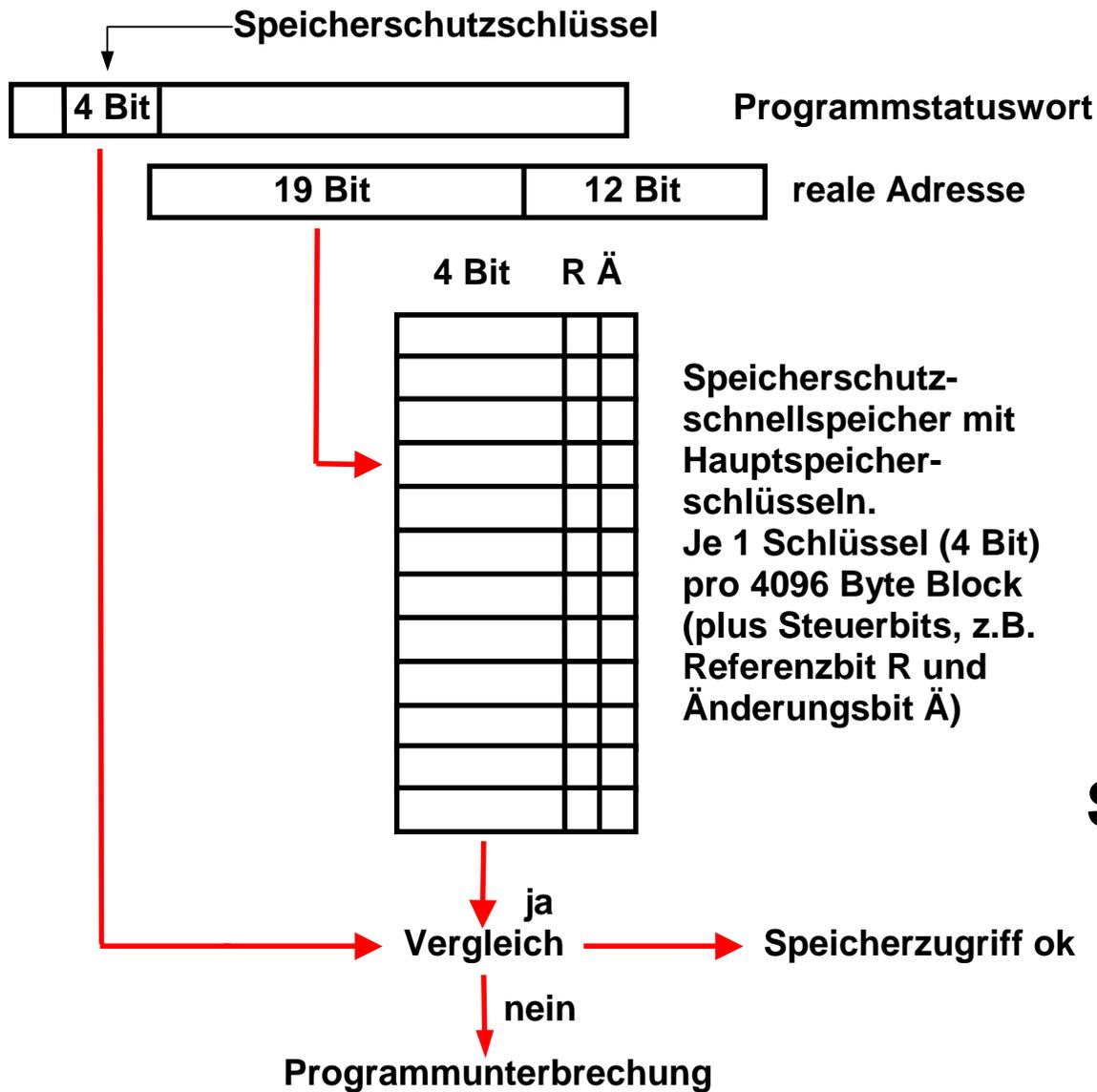
**System z Principles of Operation**



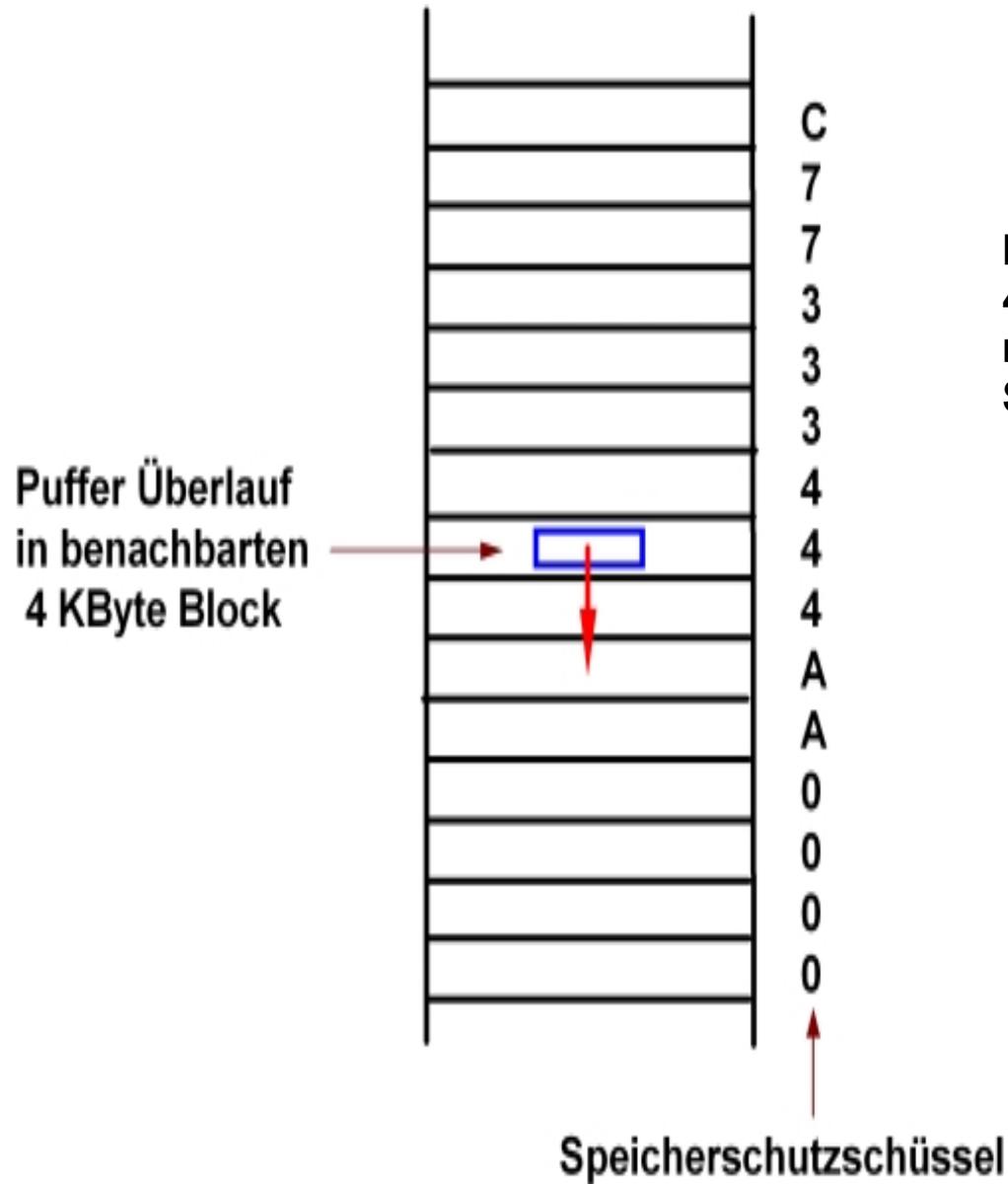
**Hauptspeicher aufgeteilt in  
Blöcke (Rahmen)  
zu je 4096 Bytes**

**Jedem Block wird vom Kernel  
eine Speicherschutznummer  
zwischen 0 ... 15 zugeordnet**

## **System z Speicherschutz**

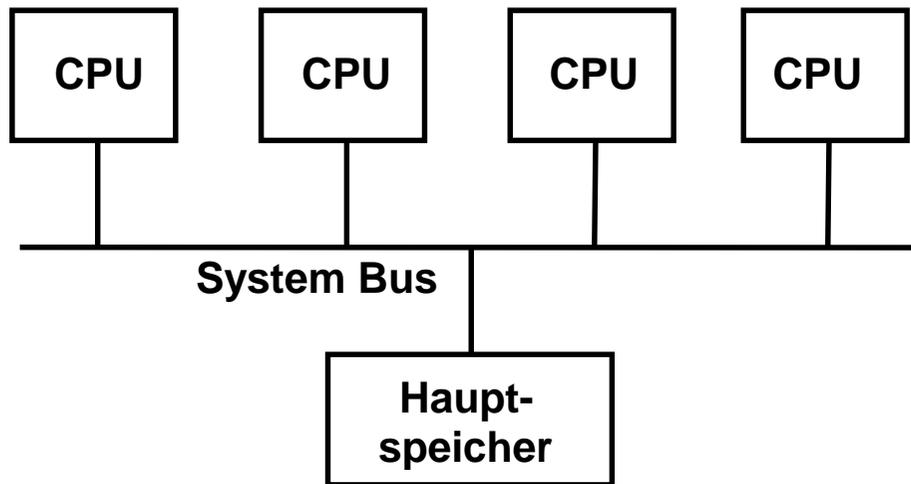


## zSeries, S/390 Speicherschutz



Hauptspeicher aufgeteilt in  
4 KByte Blöcke (Seitenrahmen)  
mit unterschiedlichen  
Speicherschutzschlüsseln

**Verhinderung des  
Puffer Überlaufs  
(Buffer Overflow)**



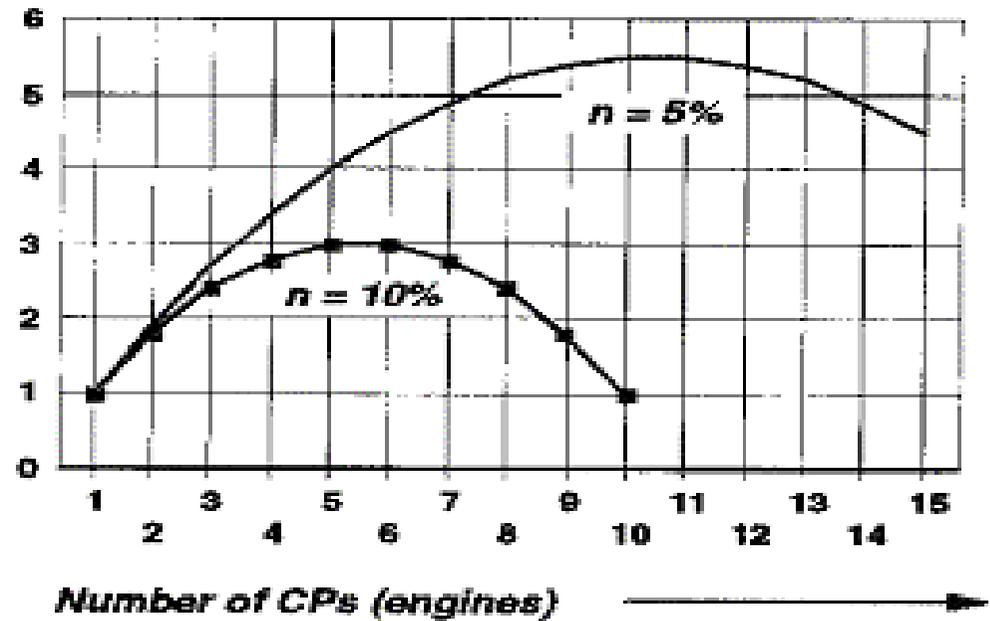
**Symmetrischer  
Multiprozessor,  
SMP**

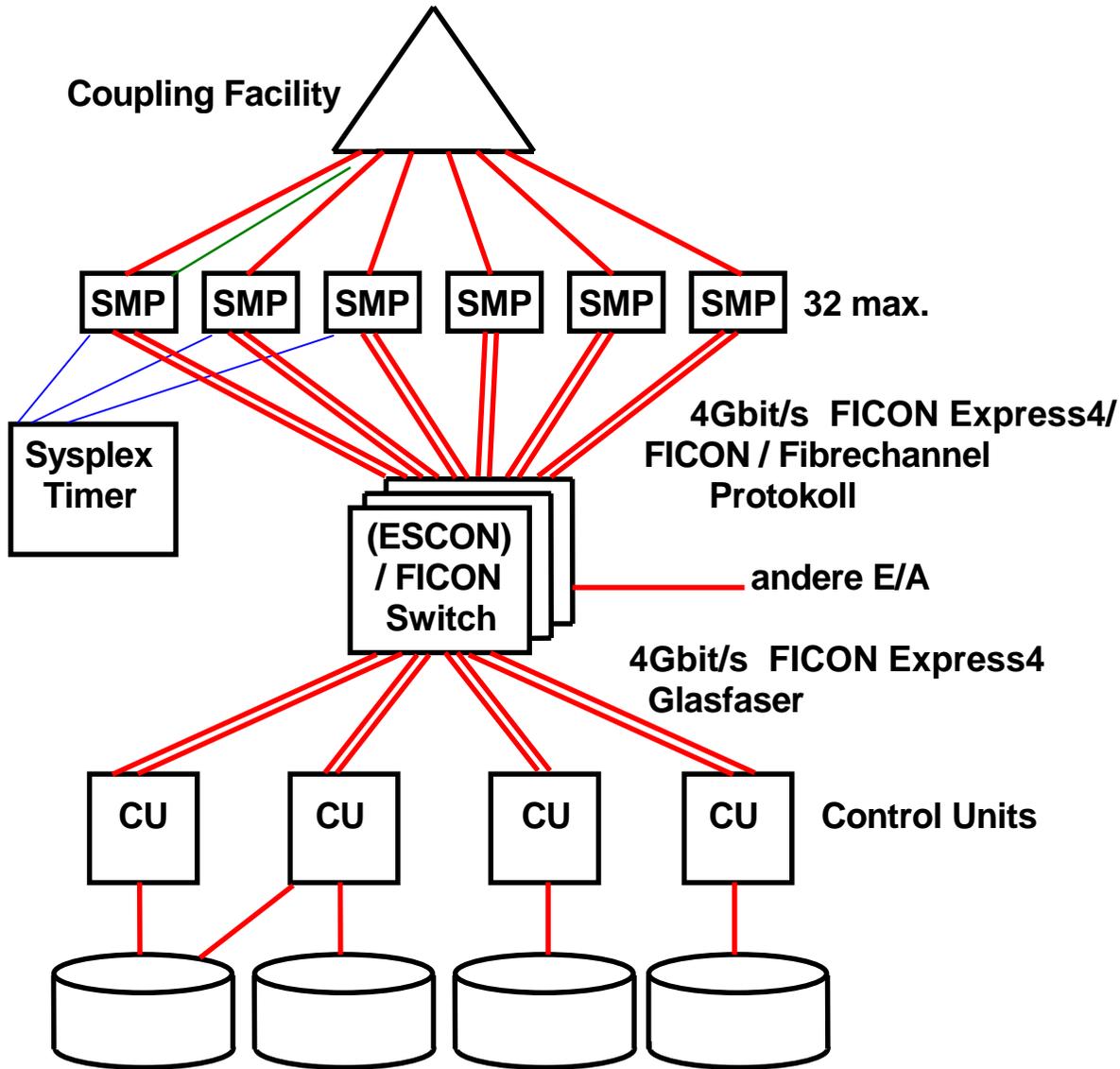
**Eine z/OS Instanz**

**z/OS bis zu 32 CPUs**

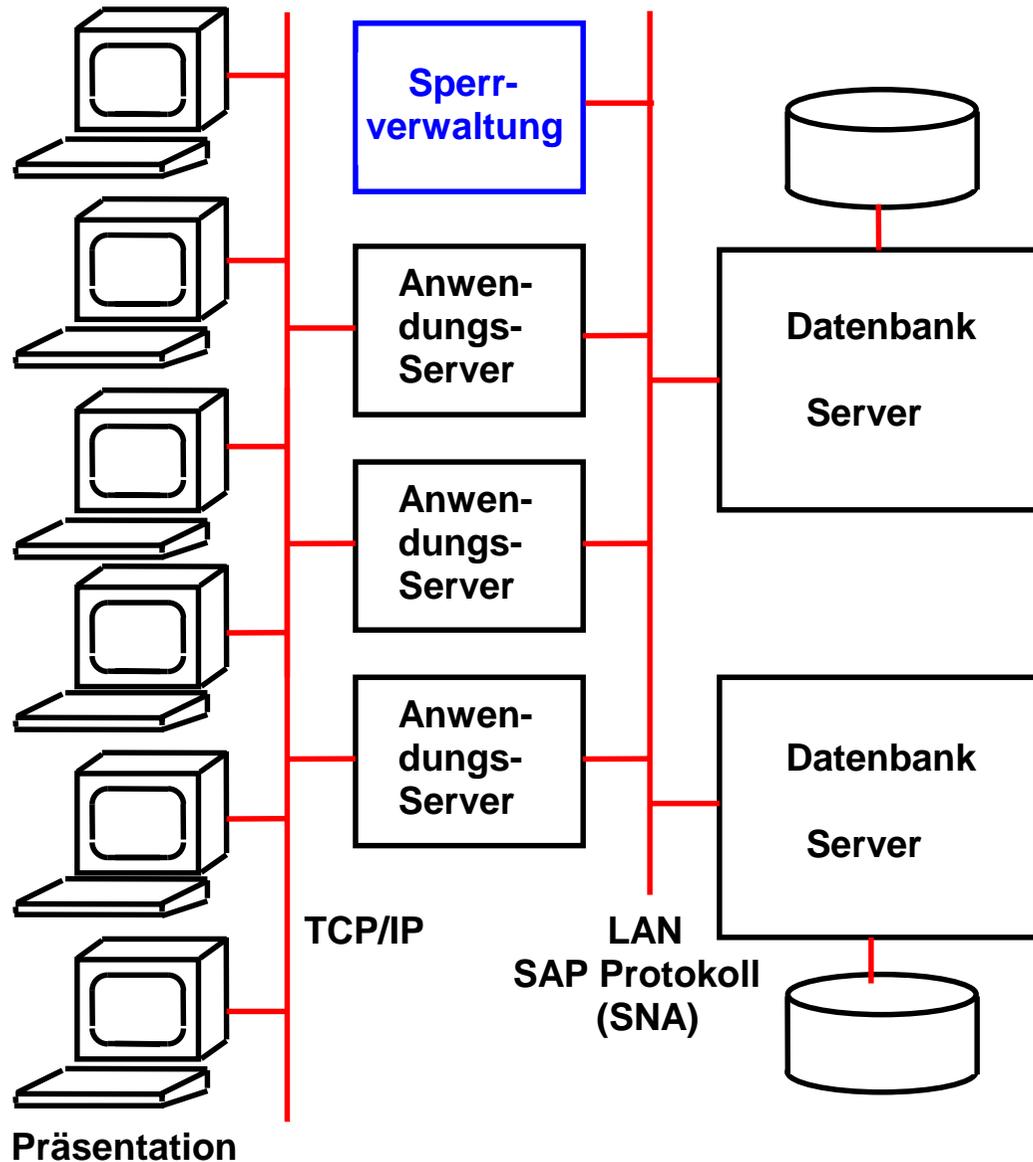
**andere Plattformen etwa  
die Hälfte für  
Transaktions- und  
Datenbankanwendungen**

*Relative  
Power  
Output*





# Sysplex mit Coupling Facility

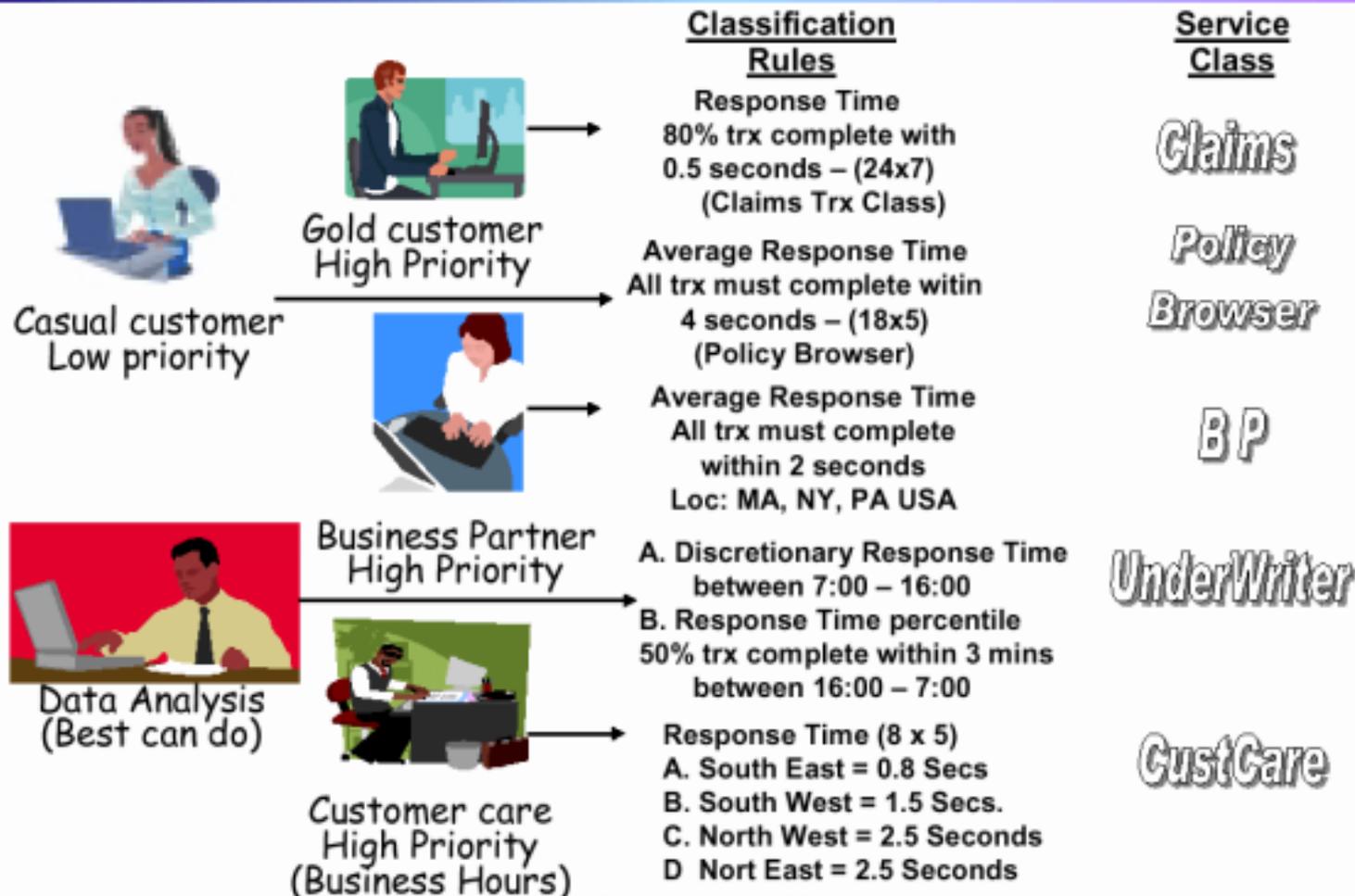


**Getrennter Rechner  
für die SAP  
Sperrverwaltung**

**You cannot scale a  
transaction  
processing system,  
if you do not solve the  
locking problem**

**Jim Gray,  
Andreas Reuter, 1993**

# Mainframe Policy Driven Workload Management



# Nutzung der Java Threads

The existing application isolation mechanisms, such as class loaders, do not guarantee that two arbitrary applications executing in the same instance of the JVM will not interfere with one another. Such interference can occur in many places. For instance, mutable parts of classes can leak object references and can allow one application to prevent the others from invoking certain methods. The internalized strings introduce shared, easy to capture monitors. Sharing event and finalization queues and their associated handling threads can block or hinder the execution of some application. Monopolizing of computational resources, such as heap memory, by one application can starve the others.

*Grzegorz Czajkowski, Laurent Daynès:  
Multitasking without Compromise: a Virtual Machine Evolution.  
<http://research.sun.com/projects/barcelona/papers/oopsla01.pdf>*

Java gives the virtuoso thread programmer considerable freedom, but it also presents many pitfalls for less experienced programmers, who can create complex programs that fail in baffling ways.

*Bo Sandén: Coping with Java Threads. IEEE Computer, Vol. 37, Nr. 4, April 2004, p. 20.*

<http://www-ti.informatik.uni-tuebingen.de/~spruth/DiplArb/jmueller.pdf>

Adressen  
des  
physikalischen  
Speichers

FF...FF

00...00

reale (absolute)  
Adressen  
der PR/SM  
Partitionen

FF...FF

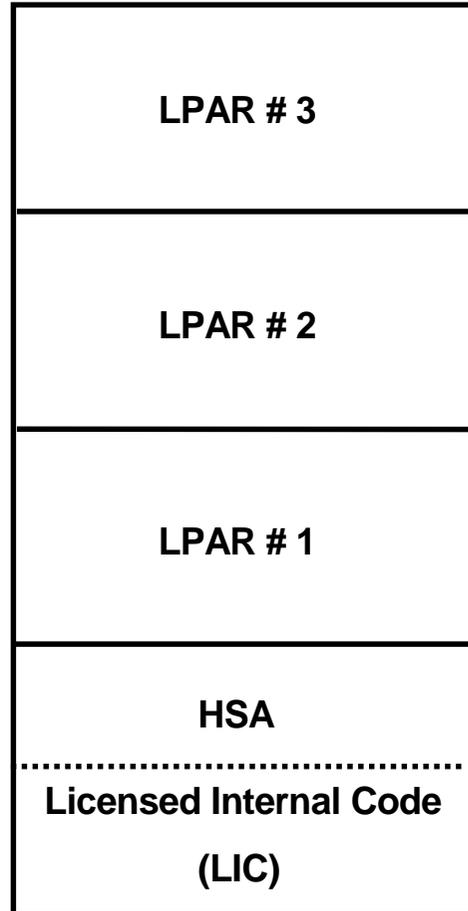
00...00

FF...FF

00...00

FF...FF

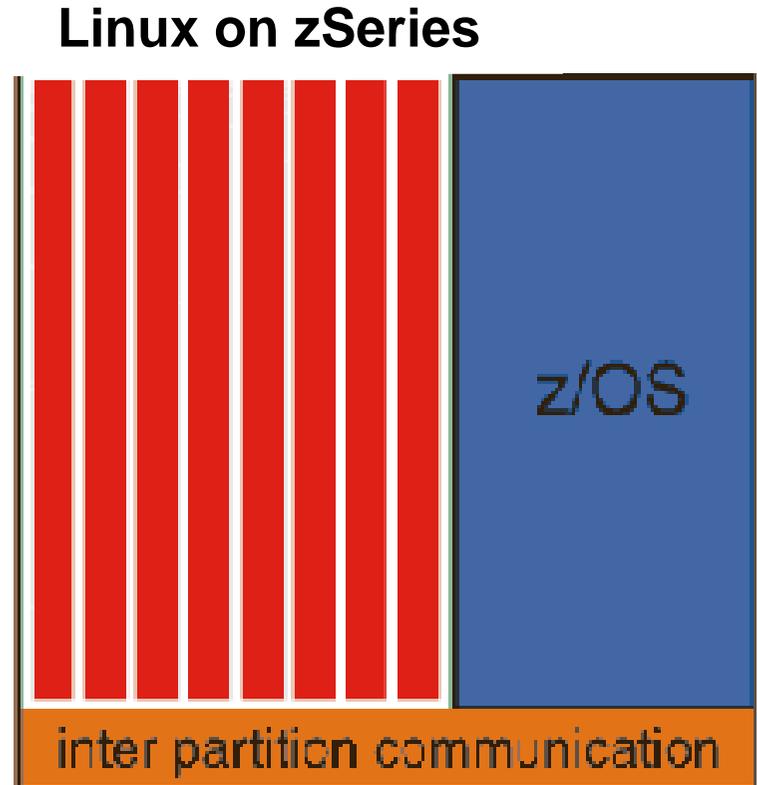
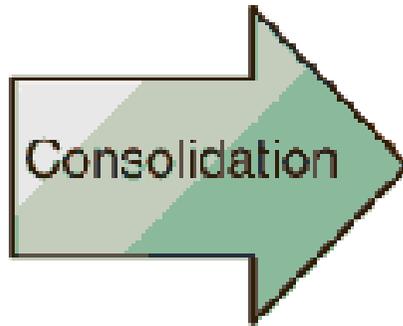
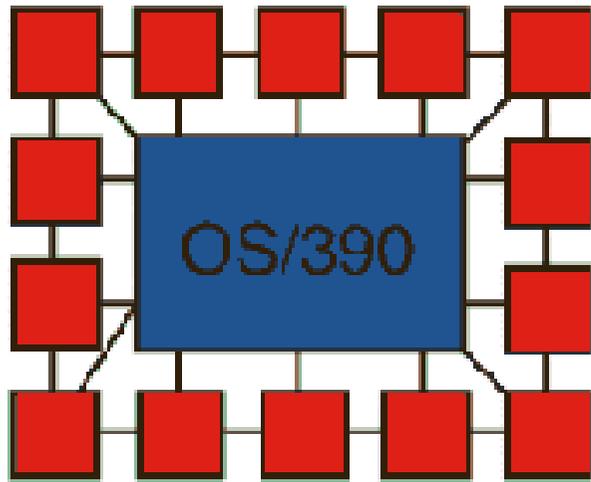
00...00



## zSeries LPAR PR/SM Virtualisierung.

Im Gegensatz zu allen  
anderen Plattformen  
arbeiten die LPARs mit  
einem realen Speicher.

Dynamische LPAR  
Speicherzuteilung

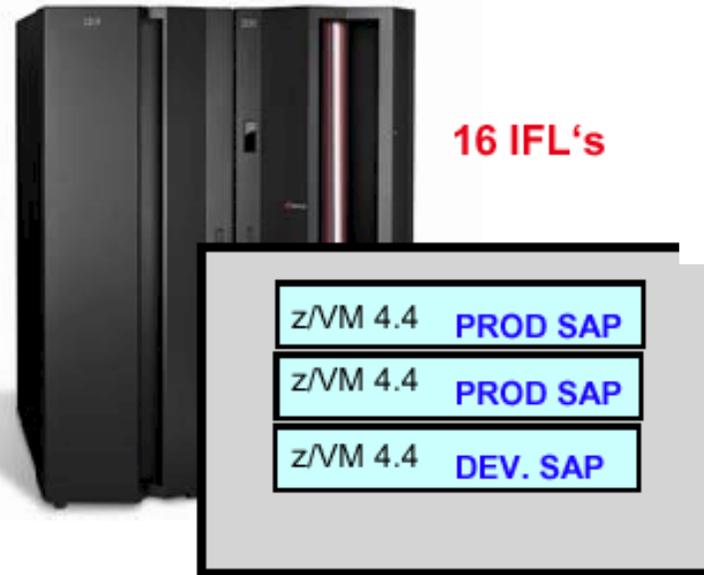
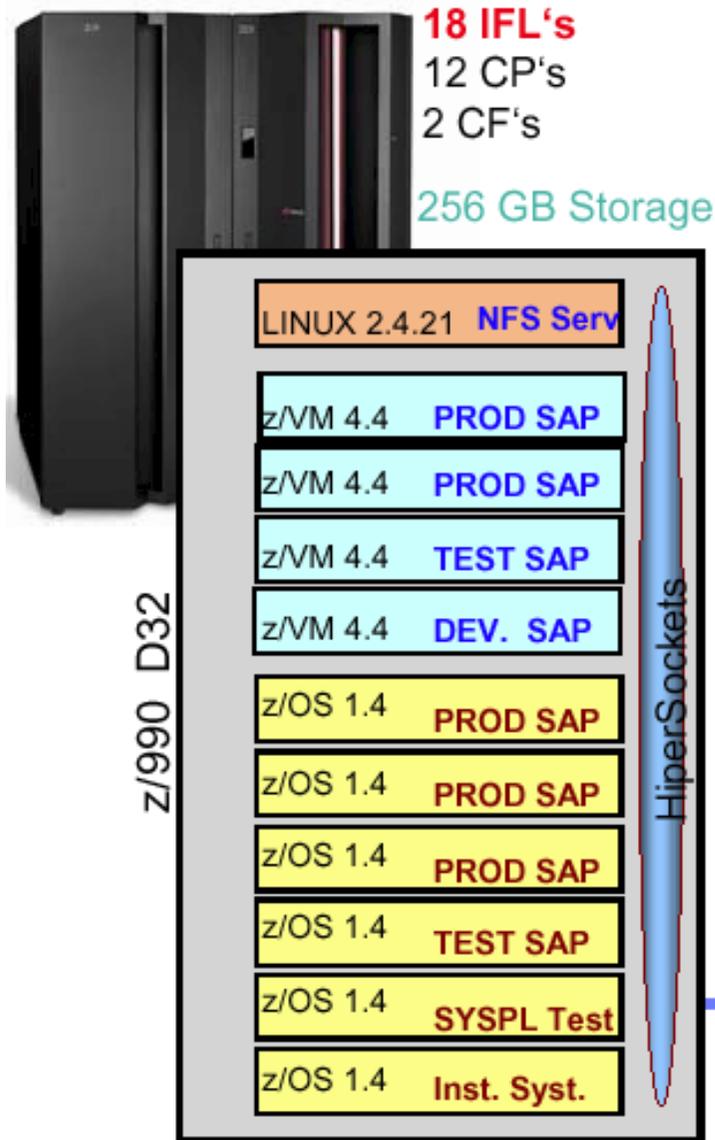


flocks of middle-tier  
Unix servers  
surrounding z/OS  
corporate data

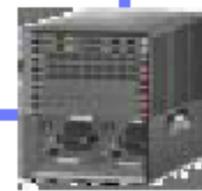
Integrated “logical” middle-tier  
servers access corporate data  
via high speed, low latency  
interpartition communication  
**“network in a box”**

The iQDIO Interface provides a high speed IP-Datacommunications between the LPARs of a zSeries system. Works like an IP-LAN interchange; data exchange via main storage Hipersockets.

## Actual LPAR structure



3.700 Benutzer,  
2500 simultan.  
0,5 seconds avg.  
response time  
für SAP



SWITCH

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

August 2004

35 subsidiaries,  
71 locations,

19 SAP R/3 Productionsysteme und 20 SAP R/3 Test-Systeme. verteilt auf 14 logische Partitionen (LPARs), laufen unter zLinux .

SAP-R/3 Datenbanken auf Basis von DB2 sind auf 6 logische Partitionen verteilt.

**Menü**

Home  
Lehre  
Tutorials OS/390  
Tutorials z/OS  
Diplomarbeiten  
Publikationen  
Rechner  
Anwendungen  
Zugriff  
Links  
Mitarbeiter

**Lehrbuch**

# Willkommen auf [jedi.informatik.uni-leipzig.de](http://jedi.informatik.uni-leipzig.de)

## Uni Leipzig OS/390 Web Application Server



Im dramatisch ansteigenden Internet-Computing erleben die vor ca. fünfzehn Jahren als Dinosaurier abgestempelten Mainframes eine unglaubliche Renaissance. Nach dem Motto "The mainframes are dead, long live the mainframes" erobern die Großrechner besonders auf dem Internet-Markt verlorengegläubtes Terrain zurück und verdrängen zunehmend PCs und Workstations in dem Bereich der Client/Server-Architekturen. Die Anzahl der bereits bestehenden und geplanten Mainframe-Installationen in den Geschäftsbereichen e-economy und e-business übertrifft weltweit inzwischen alle Erwartungen. Dieser Trend stützt sich vorrangig auf hohe Zuverlässigkeit, Sicherheit und enorme Verarbeitungsleistung.

Auf dieser Web-Seite werden die Bemühungen am Institut für Informatik der Universität Leipzig, den interessierten Studenten und Absolventen grundlegende Kenntnisse in der Hard- und Software-Architektur der IBM /390-Rechner einschließlich modernster Internet-Technologien zu vermitteln, vorgestellt.

# Rechner – Konfiguration

[jedi.informatik.uni-leipzig.de](http://jedi.informatik.uni-leipzig.de)

padme	lucas	kenob	
z/OS V 1.5 LPAR #1 2 Gbyte	OS/390 V 2.7 LPAR #2 1 Gbyte	zLinux experimental LPAR #3 2Gbyte	z/OS V 1.8 LPAR # 4 2Gbyte
PR/SM			
System z Hardware			

jedi besteht aus vier virtuellen Rechnern, die vollständig unabhängig voneinander arbeiten. Vom Standpunkt des Benutzers aus könnte es sich um vier getrennte Rechner handeln.

# **jedi.informatik.uni-leipzig.de**

**139.18.4.35:80**

## **z900 Rechner an den Universität Leipzig**

- **OS/390 V 2.7**
- **CICS**
- **IMS**
- **VSAM**
- **DB2**
- **Unix System Services**
- **JES**
- **Cobol**
- **Fortran**
- **C++**
- **Java**
- **WebSphere**
- **S/390-Linux**

**Benutzer loggen sich mit PC oder Unix Clienten in den System z Server ein.**

**Zugang mit einem gängigen 3270 Emulator, z.B. als Applet in einem Web Browser implementiert.**

**Verbindungsprotokoll : TCP/IP und Telnet.**

**Direkter Web Zugriff ist möglich**