Consolidation/Partitioning
Hardware Partitioning

- CPU
- CPU
- CPU
- CPU
- CPU
- CPU
- RAM
- RAM
- RAM
- RAM
- RAM
- RAM
- I/O
- I/O
- I/O
- I/O
- I/O
- I/O
- HD
- HD
- HD
- HD
- HD
Resource Direction
Resource Sharing

Diagram showing the arrangement of CPUs, RAM, I/O, and HDs in a network.
Soft Partitioning

Virtualisation Layer

CPU  CPU  Ram  Ram  Ram
CPU  CPU  Ram  Ram  Ram
CPU  CPU  Ram  Ram  Ram  I/O  I/O

HD  HD  HD  HD  HD  HD
Hypervisor

OS Layer

Virtualisation Layer

Hardware Layer

iSeries Academic Initiative, FU-Berlin, 2010

Rolf M Dietze, rmd@dietze-consulting.de
Classifying Server Partitioning

Hardware Partitioning

Software Partitioning

Logical Partitioning

Applications

Apps

Apps

Apps

Linux

Linux

Linux

z/OS

Windows

Windows

Windows

VSE/ESA

CPU 1

CPU 2

CPU 3

CPU 4

CPU 1

CPU 2

CPU 3

CPU 4

Partitioning Software

Partitioning Firmware

CPU 1

CPU 2

CPU 3

CPU 4

Rolf M Dietze, rmd@dietze-consulting.de

iSeries Academic Initiative, FU-Berlin, 2010
Virtualisation for Partitioning

Virtualisation Layer

AIX | i5/OS | AIX | i5/OS | AIX
---|---|---|---|---
Kernel | SLIC | Kernel | SLIC | Kernel
OBP | OBP | OBP | OBP

Logical Partition Layer

CPU | CPU | Ram | Ram | HW–Layer
---|---|---|---|---
CPU | CPU | Ram | Ram | I/O | I/O

HD | HD | HD | HD | HD | HD
Advanced Virtualization Technologies

- IBM Director Multiplatform
- IBM Grid Toolbox
- IBM Enterprise Workload Manager
- Systems Provisioning
- Virtual I/O
- Virtual Ethernet
- Dynamic LPAR
- Capacity On Demand
- i5/OS
- Linux
- AIX 5L
- Windows
- Linux
- Others

© Copyright IBM Corporation 2005
with kind permission
Capacity on Demand

- **Permanent Capacity:** CUoD … pay when purchased (processors & memory)
- **Temporary Capacity:** On/Off CoD … pay after use (processors & memory)
- **Reserve Capacity:** CoD … pay before use (processors)
- **Trial Capacity:** CoD … no-charge for use (processors & memory)
i5/OS Dynamic Logical Partitioning

- New POWER Hypervisor™ for ~ i5 supports i5/OS, AIX 5L and Linux and up to 254 partitions
  - Up to 10 Partition per processor

- Increase server utilization rates across multiple workloads

- Dynamic resource movement

- Automatic processor balancing with uncapped partitions
Der Weg zur logischen CPU
Scheduling Processes

Diagram showing a process scheduling model with processes labeled Process1 to ProcessN, each connected by context switch arrows.
Spreading to Containers
MP Konzepte

- **SMP: Symmetric MultiProcessing**  
  - uniformer Zugriff aller CPUs auf RAM+I/O

- **NUMA: Non-Uniform Memory Access**  
  - variable RAM-Zugriffslatenz

- **CC-Numa: Cache Coherent NUMA**  
  - Hardwaregesteuerte Cache coherence

- **COMA: Cache Only Memory Architecture**  
  - Hardwaregesteuerte Replikation und Coherence

- **S-COMA: Simple-COMA**  
  - Software-Replikation, Hardware-Coherence
SMP Topology Restrictions

The bottleneck of the Von-Neumann Architecture
Die 80-20 Regel gilt auch für CPUs

Core Size vs. Die Usage

Relative Performance Per Core

1.0

0.5

10%

0.5X

100%

1X

© Copyright Sun Microsystems
with kind permission
Die Idee des Chip Multithreading

100% x 1 = 1x

50% x 10 = 5x

© Copyright Sun Microsystems
with kind permission

iSeries Academic Initiative, FU-Berlin, 2010
Rolf M Dietze, rmd@dietze-consulting.de
Compute- vs. Memorycycle
Thread CPU Model
UltraSPARC T1 Pipeline Diagramm

Fetch → Thrd Sel → Decode → Execute → Memory → WB

ICache Itlb

Inst buf x 4

Thrd Sel Mux

Decode

Regfile x4

Alu Mul Shft Div

DCache Dtlb Stbuf x 4

Crossbar Interface

Thread selects

PC logic x 4

Thread select logic

Instruction type misses
traps & interrupts
resource conflicts

© Copyright Sun Microsystems with kind permission

iSeries Academic Initiative, FU-Berlin, 2010

Rolf M Dietze, rmd@dietze-consulting.de
Timeslice CPU Model

Timeslice CPU Model
Multithreading Evolution

Single thread Out of Order

- FX0
- FX1
- FP0
- FP1
- LS0
- LS1
- BRX
- CRL

S80 Hardware Multi-thread

- FX0
- FX1
- FP0
- FP1
- LS0
- LS1
- BRX
- CRL

POWER5 2 Way SMT

- FX0
- FX1
- FP0
- FP1
- LS0
- LS1
- BRX
- CRL

POWER7 4 Way SMT

- FX0
- FX1
- FP0
- FP1
- LS0
- LS1
- BRX
- CRL

- No Thread Executing
- Thread 0 Executing
- Thread 1 Executing
- Thread 2 Executing
- Thread 3 Executing

© Copyright IBM
with kind permission
Understanding Shared Processors

To understand Processing Units – there are four main concepts

1. One single processor is equivalent to 1.00 Processing Units, 2 Processors = 2.00 Processing Units, etc. 0.5 processing units is NOT same as half a processor.

2. Shared processor pool. A processor must live in the shared processor pool (now the default) to become Processing units.

3. Virtual Processor – how many processors do you want the partition to be able to use (run jobs/threads on) simultaneously. It’s also the number of processors that the operating system thinks it has to use.

© Copyright IBM with kind permission
4. The iSeries processors run on 10 ms time slices

Each Processor use is allocated within a 10 ms Cycle

- A partition’s use of a processor is limited to its allocation during each 10 ms cycle.
- For example, 80% of a processor (.80 Processing Units) yields up to 8 ms of processing time out of this 10 ms time slice. It also yields .8 X CPW rating of the processor.
- Every 10 ms this cycle repeats itself.
How Does a Job Get Into the Processor?

For a job to run it has to get its data from DISK into memory, into cache, and finally into the processor.

- Slowest - disk
- Fastest - cache

Even if the data is already in memory it has to be loaded into cache to run – this takes time.

© Copyright IBM
with kind permission
Example of Two Partitions Sharing a Processor ("capped")

Partition **dog** jobs a,b,c allocated .6 Processing Units

Partition **cat** jobs 1,2,3 allocated .4 Processing Units

© Copyright IBM
with kind permission
Potential Shared Processor Penalty

• There is a potential for a performance penalty (from 0 to 10%) when using shared processors, due to:
  – Increasing the possibility that jobs won’t complete, and
  – Having to be redispacth and potentially reload cache, and
  – Increasing the chance of a cache miss

• Reduce the chance for processor and memory affinity

• The POWER Hypervisor overhead of:
  – Managing multiple processors
  – Tracking each partitions use of its allotted milliseconds
  – Managing time slices from multiple partitions

• All of the above are affected by how you allocate your virtual processors – next couple of foils
Desired Minimum/Maximum Processing Units

- How about 0.2 Processing Units
- Minimum of 0.1
- Maximum of 2.00
- Select Advanced
Capped Partitions

- First time through we select Capped
- You can’t have less than .10 processing units per virtual processor
- I’ll allocate two virtuals for my .2 PUs
- What’s a virtual processor?
Introduction to Virtual Processors

For every 10 milliseconds of wall clock time each processor in the shared pool is capable of 10 milliseconds of processing time

- If you give Partition x .5 processing units it could use (up to) 5 milliseconds of processing time – capped. (more on capped soon)

- But you have ABSOLUTELY no control over which processors your jobs/threads run on

- All you CAN control is how many of the processors in the pool, your jobs/threads do run on (potentially) simultaneously, via Virtual Processors

= 80 milliseconds

© Copyright IBM
with kind permission

iSeries Academic Initiative, FU-Berlin, 2010
Rolf M Dietze, rmd@dietze-consulting.de
P1 1.5 processing unit default of 2 virtual processors – max of 15 milliseconds – capped. Each job potentially could get 7.5 milliseconds (15/2 = 7.5)

P2 1.5 processing unit but using 6 virtual processors – max of 15 milliseconds – capped. But if all 6 Jobs ran at same time each may get no more than 2.5 milliseconds per job. (15/6 = 2.5)
As of IBM eServer i5/OS, and POWER5-based servers, it is now possible, by using the uncapped mode, to use more milliseconds than are allocated to a partition.

An uncapped partition can use excess pool Processing Units.

But even an uncapped partition could still be limited by setting the number of virtual processors too low. The number of processors it can use simultaneously is still limited by the number of virtual processors assigned.

As of Power5 it’s also possible to allocate more virtual processors than there are processors in the pool, since the actual number of processors in the pool is a ‘floating’ number. However, you still cannot allocate less than 1 ms (.10 PUs) per processor per job (virtual processor). For example, .5 PUs and 6 virtuals is a dog that doesn’t hunt. 5 (milliseconds)/6 (jobs) < 1 milliseconds per job.
• This time we deal with uncapped

• You can’t have any less than .10 processing units per virtual processor

• Allocate two virtuals for my .2 PUs

• Select OK

© Copyright IBM with kind permission
Virtual Processors (Limited) – Uncapped

P1 1.5 processing unit default of 2 virtual processors – max of 20 milliseconds – uncapped because you are limited to only use 2 Processors simultaneously

P2 1.5 processing unit 6 virtual processors – max of 60 milliseconds – uncapped

© Copyright IBM with kind permission
Example of Two Partitions Sharing a Processor (“uncapped”)

Partition **dog** jobs a,b,c allocated .6 Processing Units

Partition **cat** jobs 1,2,3 allocated .4 Processing Units

(iSeries Academic Initiative, FU-Berlin, 2010)

Rolf M Dietze, rmd@dietze-consulting.de
iSeries “uncapped shared pool” with CoD

LPAR 1 (uncapped)
Optimized capacity of 0.4 processors for LPAR1

LPAR 2 (uncapped)
Optimized capacity of an entire processor for LPAR 2

LPAR 3 (uncapped)
Optimized capacity of 0.6 processors for LPAR3

- Resources can be requested by any partition
- Unused resources can be released
- Priorities can be assigned
- Unused resources CPUs/MEM will automatically be used to solve failures in a running operating environment

© Copyright IBM with kind permission
Virtual Processors (Unlimited) - Uncapped

P2 1.5 processing unit with 15 Virtual Processors (maximum allowed) gives Partition – potentially – **ALL 80 milliseconds** of Processing time – for every 10 physical milliseconds BUT only as long as the other shared processor partitions DON’T have jobs ready to run!
Floating Processors

• You have eight processors on your system. Seven are in the pool and one partition uses a dedicated processor.

• Dedicated partitions can allow its processors to be used for uncapped capacity (returned to the shared pool) when the partition is powered off, or a processor is removed from the partition. This is the default.
Dedicated or Shared / Capped or Uncapped?

- The best performance may well be achieved by using dedicated processors. However, dedicated processors cannot utilize excess capacity.

- For both capped and uncapped partitions, setting the virtual processor number too high may degrade performance.

- Shared uncapped allows use of excess capacity of the processors in the shared pool. Setting virtual processors too low limits the amount of excess you can use. Setting too high may negatively impact performance.

- Also be aware for uncapped partitions the operating system sees the number of desired virtual processors as equal to the number of physical processors, you need an OS license (i5/OS, Linux and AIX 5L) for the lesser of the number of virtual processors or the number of processors in the shared pool.

- So what could be recommended? The right answer depends on workload.
Partitioning Computers

Virtualization Engine
- IBM Director Multiplatform
- IBM Grid Toolbox
- IBM Enterprise Workload Manager
- Systems Provisioning

Virtualization Technologies
- Virtual I/O
- Virtual Ethernet
- Dynamic LPAR
- Capacity On Demand

Systems Services
- i5/OS
- Linux
- AIX 5L
- Windows
- Linux
- Others

POWER Hypervisor
- POWER5

 сервер i5

© Copyright IBM
with kind permission

iSeries Academic Initiative, FU-Berlin, 2010
Rolf M Dietze, rmd@dietze-consulting.de
Virtual I/O Example

```
“Client” Partition

Virtual Ethernet

Virtual Ethernet

Client Adapter

DMA Buffer

POWER Hypervisor

Virtual Switch

Virtual SCSI

“Server” Partition

Virtual Ethernet

Proxy ARP

Physical Ethernet

Virtual Ethernet

NBSD

Device Mapping

Server Adapter

SCSI, SSA, FC

Physical or Logical Disks

© Copyright IBM Corporation 2005
Material may not be reproduced in whole or in part without the prior written permission of IBM.
```
Virtual Serial

- First 2 virtual slots in every partition reserved for virtual serial server adapters for system console in HMC

- For i5/OS, virtual serial adapters provide 5250 console
- For Linux and AIX 5L, they provide character console
Virtual Serial

Detailed below are the virtual adapters created in this partition.

Virtual adapters

Number of virtual adapters: 10

<table>
<thead>
<tr>
<th>Slot Number</th>
<th>Type</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>Server Serial</td>
</tr>
<tr>
<td>1</td>
<td>Server Serial</td>
</tr>
<tr>
<td>3</td>
<td>Server SCSI</td>
</tr>
<tr>
<td>4</td>
<td>Server SCSI</td>
</tr>
</tbody>
</table>

Adapter Type

- Client
- Server

Connection Information

- HMC and any remote partition and slot can connect
- Any remote partition and slot can connect
- Only selected remote partition and slot can connect

Remote partition: Milwaukee (3)
Remote partition virtual slot number: 5

(Check to create the selected adapter type)
Adding SCSI Adapter via DLPAR

- Create a virtual SCSI client adapter via the partition creation wizard
- Create a virtual SCSI server adapter via Dynamic LPAR on the i5/OS partition – Does not require any restart
Virtual SCSI

Individual slot numbers do no matter, as long as they are configured in pairs.

© Copyright IBM
with kind permission