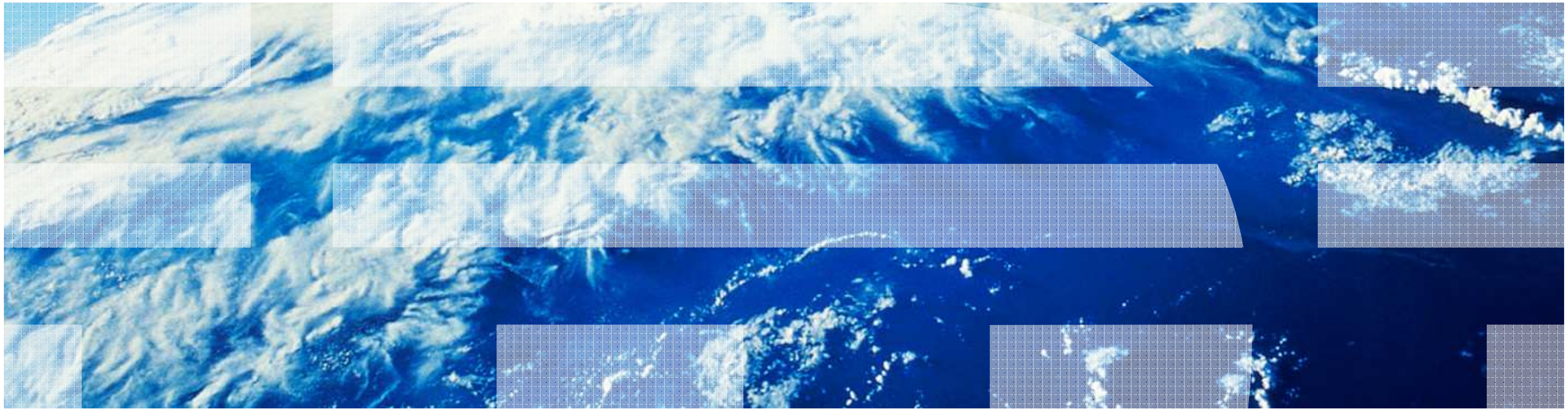
Session zVM03

Conquering Performance Gotchas for z/VM Mode LPARs (Mixed Engines)

Bill Bitner



Trademarks

Trademarks

The following are trademarks of the International Business Machines Corporation in the United States and/or other countries. For a complete list of IBM Trademarks, see www.ibm.com/legal/copytrade.shtml: AS/400, DBE, e-business logo, ESCO, eServer, FICON, IBM, IBM Logo, iSeries, MVS, OS/390, pSeries, RS/6000, S/30, VM/ESA, VSE/ESA, Websphere, xSeries, z/OS, zSeries, z/VM

The following are trademarks or registered trademarks of other companies

Lotus, Notes, and Domino are trademarks or registered trademarks of Lotus Development Corporation
Java and all Java-related trademarks and logos are trademarks of Sun Microsystems, Inc., in the United States and other countries
LINUX is a registered trademark of Linus Torvalds
UNIX is a registered trademark of The Open Group in the United States and other countries.
Microsoft, Windows and Windows NT are registered trademarks of Microsoft Corporation.
SET and Secure Electronic Transaction are trademarks owned by SET Secure Electronic Transaction LLC.
Intel is a registered trademark of Intel Corporation
* All other products may be trademarks or registered trademarks of their respective companies.

NOTES:
Performance is in Internal Throughput Rate (ITR) ratio based on measurements and projections using standard IBM benchmarks in a controlled environment. The actual throughput that any user will experience will vary depending upon considerations such as the amount of multiprogramming in the user's job stream, the I/O configuration, the storage configuration, and the workload processed. Therefore, no assurance can be given that an individual user will achieve throughput improvements equivalent to the performance ratios stated here.

IBM hardware products are manufactured from new parts, or new and serviceable used parts. Regardless, our warranty terms apply.

All customer examples cited or described in this presentation are presented as illustrations of the manner in which some customers have used IBM products and the results they may have achieved. Actual environmental costs and performance characteristics will vary depending on individual customer configurations and conditions.

This publication was produced in the United States. IBM may not offer the products, services or features discussed in this document in other countries, and the information may be subject to change without notice. Consult your local IBM business contact for information on the product or services available in your area.

All statements regarding IBM's future direction and intent are subject to change or withdrawal without notice, and represent goals and objectives only.

Information about non-IBM products is obtained from the manufacturers of those products or their published announcements. IBM has not tested those products and cannot confirm the performance, compatibility, or any other claims related to non-IBM products. Questions on the capabilities of non-IBM products should be addressed to the suppliers of those products.

Prices subject to change without notice. Contact your IBM representative or Business Partner for the most current pricing in your geography.

References in this document to IBM products or services do not imply that IBM intends to make them available in every country.

Any proposed use of claims in this presentation outside of the United States must be reviewed by local IBM country counsel prior to such use.

The information could include technical inaccuracies or typographical errors. Changes are periodically made to the information herein; these changes will be incorporated in new editions of the publication. IBM may make improvements and/or changes in the product(s) and/or the program(s) described in this publication at any time without notice.

Any references in this information to non-IBM Web sites are provided for convenience only and do not in any manner serve as an endorsement of those Web sites. The materials at those Web sites are not part of the materials for this IBM product and use of those Web sites is at your own risk.

Permission is hereby granted to SHARE to publish an exact copy of this paper in the SHARE proceedings. IBM retains the title to the copyright in this paper, as well as the copyright in all underlying works. IBM retains the right to make derivative works and to republish and distribute this paper to whomever it chooses in any way it chooses.

Agenda

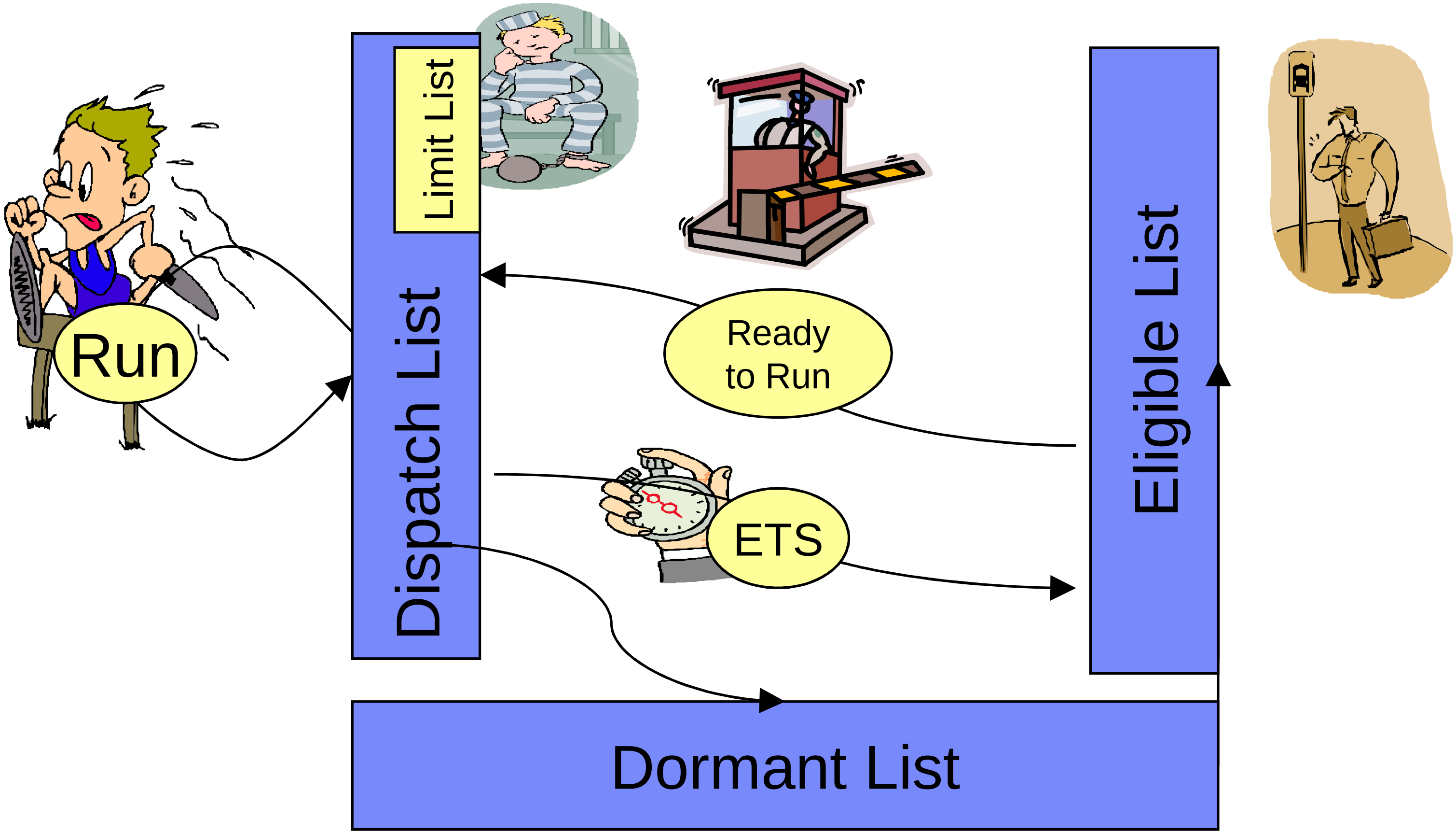
- **Some general z/VM Scheduling and Dispatching Discussion**
- **Some general z/OS Guest Tuning Discussion**
- **Background on Specialty Engine support in z/VM**
- **Configuring of Specialty Engines**
- **Measuring Specialty Engines**
- **Tuning of Specialty Engines**
- **Miscellaneous z/OS Tuning**

z/VM Scheduling & Dispatching at the High Level

- **Objectives of the z/VM Scheduler**
  - Protect the system from over committing resources to where the system thrashes
  - Prioritize access to system resources
- **Objectives of the z/VM Dispatcher**
  - Effectively run virtual processors based on their priorities

The Main Loops

- **Each virtual processor is in one of the following lists:**
  - Dispatch List – (D-List, in Q) users ready or near-ready to run
  - Eligible List – (E-list) Delayed here when cannot “fit” in D-List
  - Dormant List – users that are idle (from view of the scheduler)



Three Main Controls to Entering Dispatch List

```
cp q srm
IABIAS : INTENSITY=90%; DURATION=2
LDUBUF : Q1=100% Q2=75% Q3=60%
STORBUF: Q1=300% Q2=250% Q3=200%
DSPBUF : Q1=32767 Q2=32767 Q3=32767
DISPATCHING MINOR TIMESLICE = 5 MS
MAXWSS : LIMIT=9999%
...... : PAGES=999999
XSTORE : 0%
Ready;
```

LDUBUF: protects from thrashing DASD Paging

STORBUF: protects from general thrashing of real memory

DSPBUF: Absolute number allowed in dispatch list for each scheduling class

Comments on SRM Value for z/OS Systems

- **Defaults were determined based on traditional workload with mix of interactive CMS and Guest work.**
  - Defaults changed in z/VM 6.2 to be more appropriate for z/OS & Linux
- **Potential benefit from changing SRM values.**
  - If having problems, investigate STORBUF first
  - Second, look at LDUBUF
  - Keep your hands off DSPBUF unless you really know what you are doing.
  - Avoid temptation to increase/change several values at once
- **Increasing DSPSPLICE was considered clever at one time. The overhead from dispatching these days probably isn't worth the downside of increasing it. Leave it alone.**

Set Share Syntax

```
                        .-TYPE--ALL------.
>>--Set--SHARE--userid--+----------------+----------------------------------->
                        '-TYPE--.-ALL--.-'
                                |-CP---|
                                |-ZIIP-|
                                |-ZAAP-|
                                |-IFL--|
                                '-ICF--'

>--.-INITial-----------------------------------------------------.----------><
   |                        .-NOLimit---.                        |
   |-.-ABSolute--nnn%--.--.-+-----------+----------------------.-|
   | '-RELative--nnnnn-'  | |-LIMITSoft-|                      | |
   |                      | '-LIMITHard-'                      | |
   |                      |                     .-LIMITSoft-.  | |
   |                      |-.----------.--mmm%--+-----------+--| |
   |                      | '-ABSolute-'        '-LIMITHard-'  | |
   |                      |                      .-LIMITSoft-. | |
   |                      '-.----------.--mmmmm--+-----------+-' |
   |                        '-RELative-'         '-LIMITHard-'   |
   '-.-NOLimit---.-----------------------------------------------'
     |-LIMITSoft-|
     '-LIMITHard-'
```

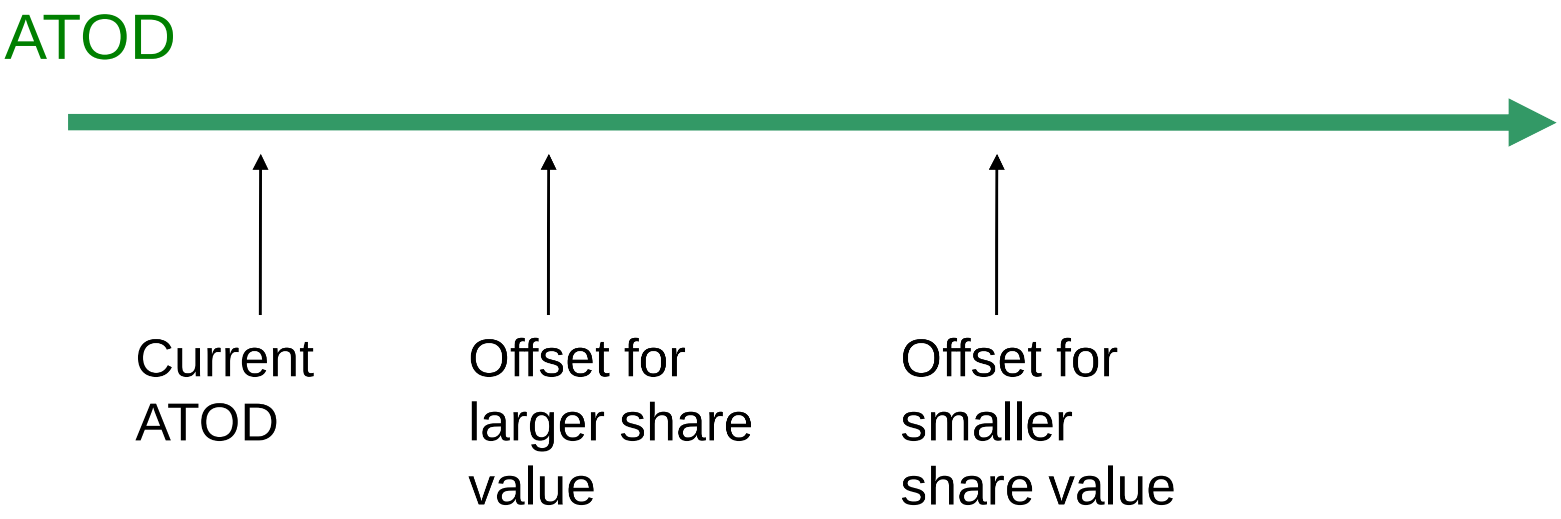
Set Share Options

- **vCPU Type: CP, IFL, ZAAP, ZIIP, ICF**
- **Target Minimum**
- **Target Maximum (Limit Share)**
- **For Min/Max:**
  - Absolute: Percent of System. Sum Normalized to 99%
  - Relative: 1 to 10,000
    - Relative to other virtual machines in dispatch list
    - Normalized to 100% - Sum of Absolutes

Deadline Scheduling – Prioritizing Work

- **Each virtual processor has a priority computed as a 'deadline' for when a unit of work should be completed.**
- **This 'deadline' is a time value on an artificial TOD often referred to as ATOD**
- **The 'deadline' is computed based on several factors, but the most significant is the normalized Share value**
- **Therefore the share setting is a big knob**
- **Virtual processors get ordered for dispatching based on their deadlines**
- **Note: with VM64721 and SET SRM LIMITHARD CONSUMPTION or z/VM 6.2, limit shares are controlled via a consumption scheduler instead of a deadline scheduler**

ATOD and Deadline



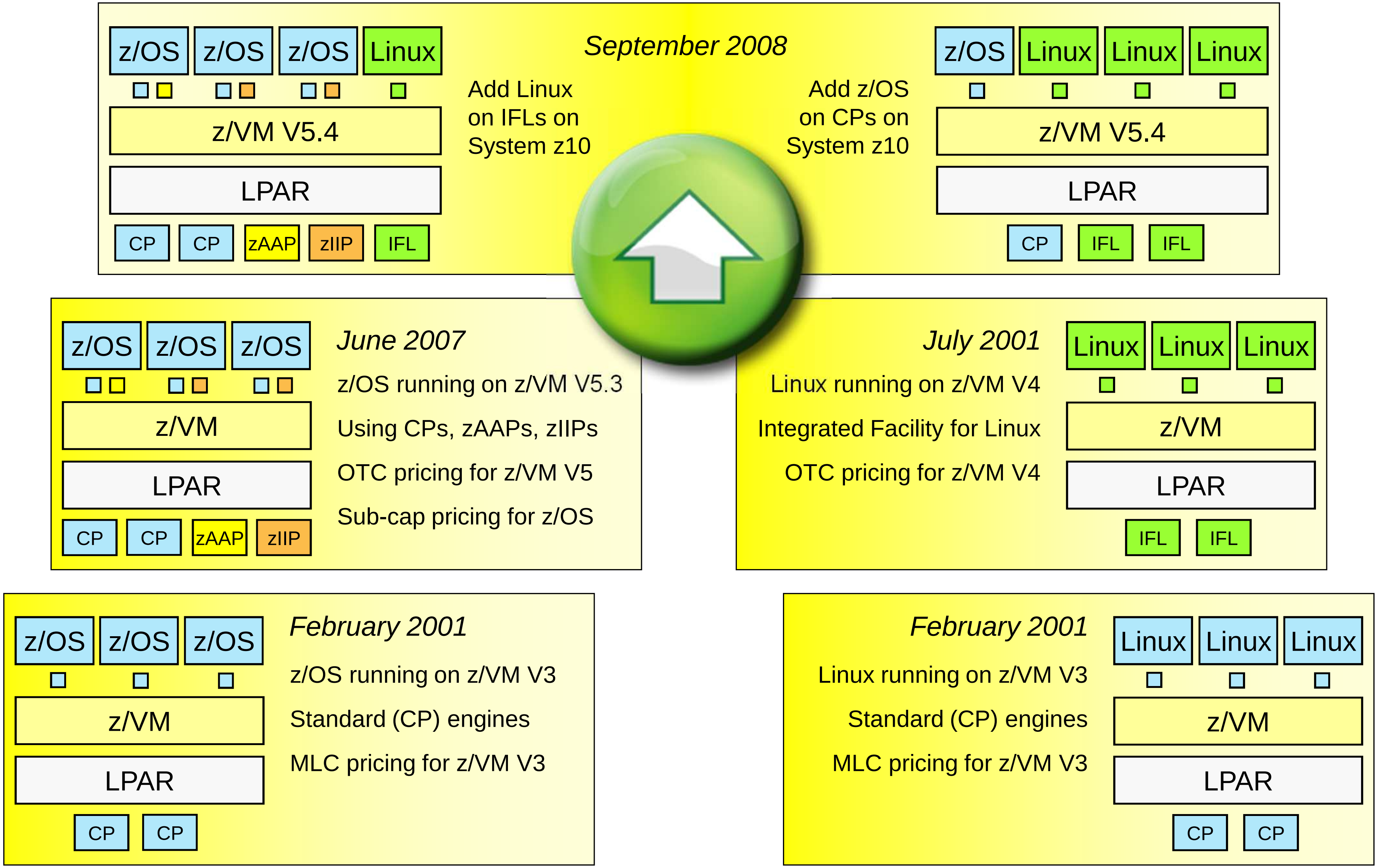
Simplified offset formula used to set deadline ‘offset’ from current ATOD:

$$\texttt{OFFSET} = \frac{\texttt{Minor\_TimeSlice + Previous\_TimeSlice Overrun}}{\texttt{Normalized\_Share x Number\_PUs}}$$

A Word About QUICKDSP

- **Quick Dispatch (SET QUICKDSP) for a virtual machine allows it to pass from eligible list to dispatch list without going through the system resource checks.**
- **Does NOT turn off the scheduler completely.**
- **Should be set on for:**
  - Mission Critical Virtual Machines
  - Virtual Machines that are extensions of Operating System (e.g. RACF, TCP/IP)
  - A virtual machine you have access to for tuning and problem determination.

z/VM and Specialty Engine Support



September 2008

z/OS | z/OS | z/OS | Linux

z/VM V5.4

LPAR

CP | CP | zAAP | zIIP | IFL

Add Linux on IFLs on System z10

Add z/OS on CPs on System z10

z/OS | Linux | Linux | Linux

z/VM V5.4

LPAR

CP | IFL | IFL

June 2007

z/OS | z/OS | z/OS

z/VM

LPAR

CP | CP | zAAP | zIIP

z/OS running on z/VM V5.3

Using CPs, zAAPs, zIIPs

OTC pricing for z/VM V5

Sub-cap pricing for z/OS

July 2001

Linux running on z/VM V4

Integrated Facility for Linux

OTC pricing for z/VM V4

Linux | Linux | Linux

z/VM

LPAR

IFL | IFL

February 2001

z/OS | z/OS | z/OS

z/VM

LPAR

CP | CP

z/OS running on z/VM V3

Standard (CP) engines

MLC pricing for z/VM V3

February 2001

Linux running on z/VM V3

Standard (CP) engines

MLC pricing for z/VM V3

Linux | Linux | Linux

z/VM

LPAR

CP | CP

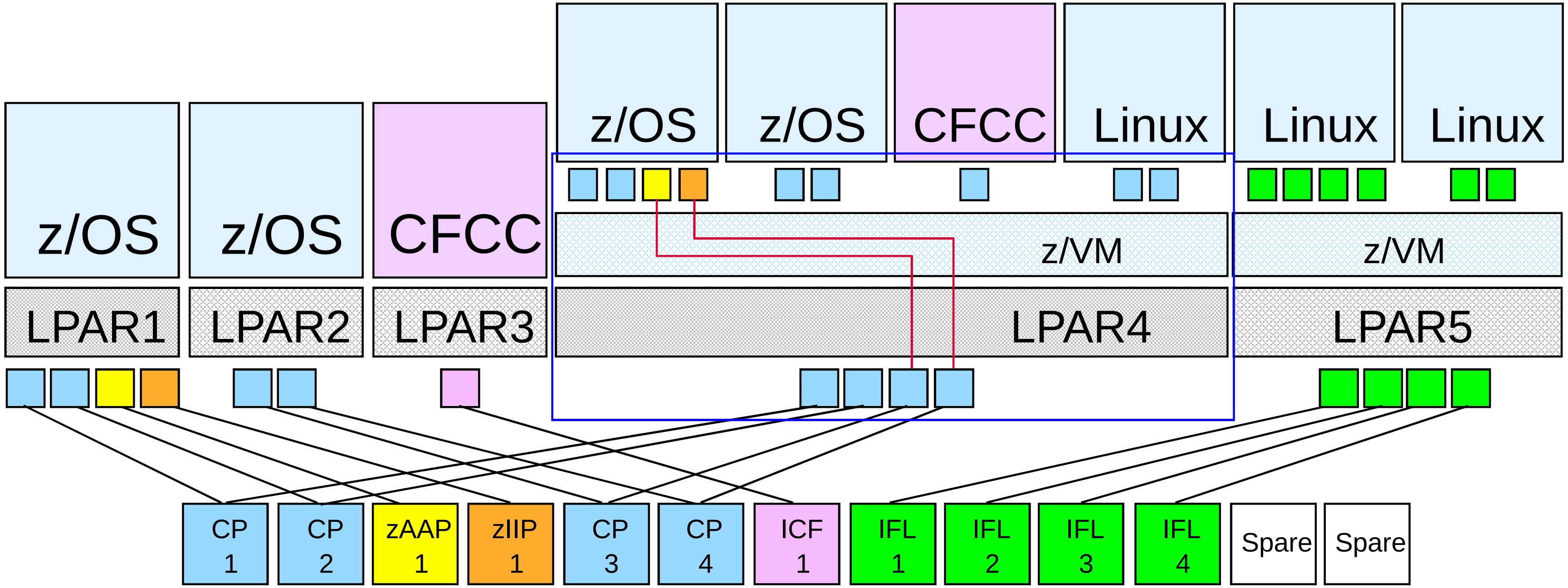
z/VM Support for Specialty Processors

- **z/VM V5.3 introduced support for zAAP and zIIP specialty processors**
  - System z Application Assist Processors (zAAPs) – provide an economical Java execution environment for z/OS and z/OS.e
  - System z9 Integrated Information Processors (zIIPs) – designed to help improve resource optimization and lower the cost for eligible z/OS and z/OS.e workloads by offloading software system overhead from standard Central Processors (CPs); this includes certain DB2 processing
- **z/VM support is provided for z/OS guest exploitation**
  - Offers additional hardware support for z/OS-on-z/VM development and test support
- **Two levels of z/VM support:**
  - *Simulation support*
    - z/VM dispatches virtual zAAPs and zIIPs on real CP engines
    - Only possible if the underlying hardware is capable of supporting the real engine type
    - Does not require activation of real specialty engine(s) within the mainframe server
  - *Virtualization support*
    - z/VM dispatches virtual zAAPs and zIIPs on corresponding real specialty engines
- **Consistent with z/OS, there are no z/VM license fees associated with real zAAP or zIIP processors**

z/VM Specialty Processor Support Example

Simulating Specialty Engines in Virtual Machines

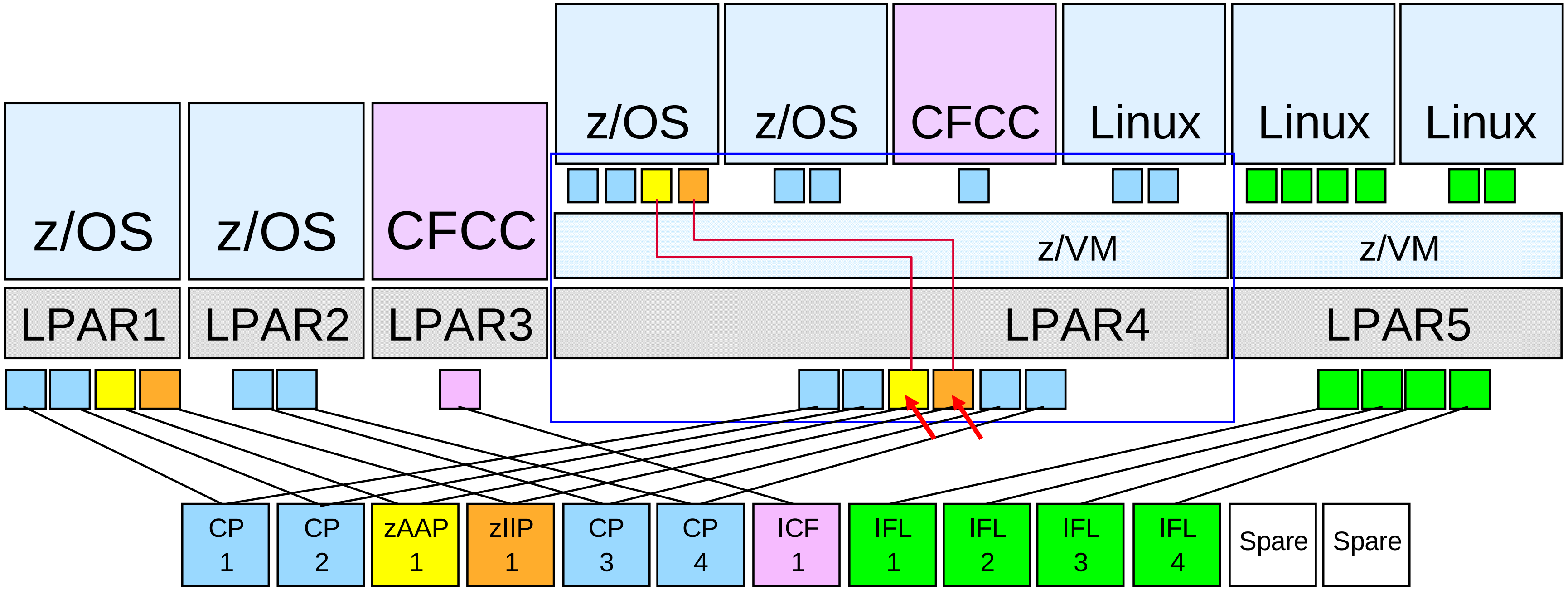
- Allows users to discover the operational aspects of using zAAPs and zIIPs in a z/OS environment without having to purchase real specialty processors
- May help users assess specialty-processor eligible workloads in a z/OS environment
- Provides a function test environment for z/OS workloads that use specialty processors
- Consumes CP processor capacity to host virtual zAAP and zIIP processor cycles



z/VM Specialty Processor Support Example

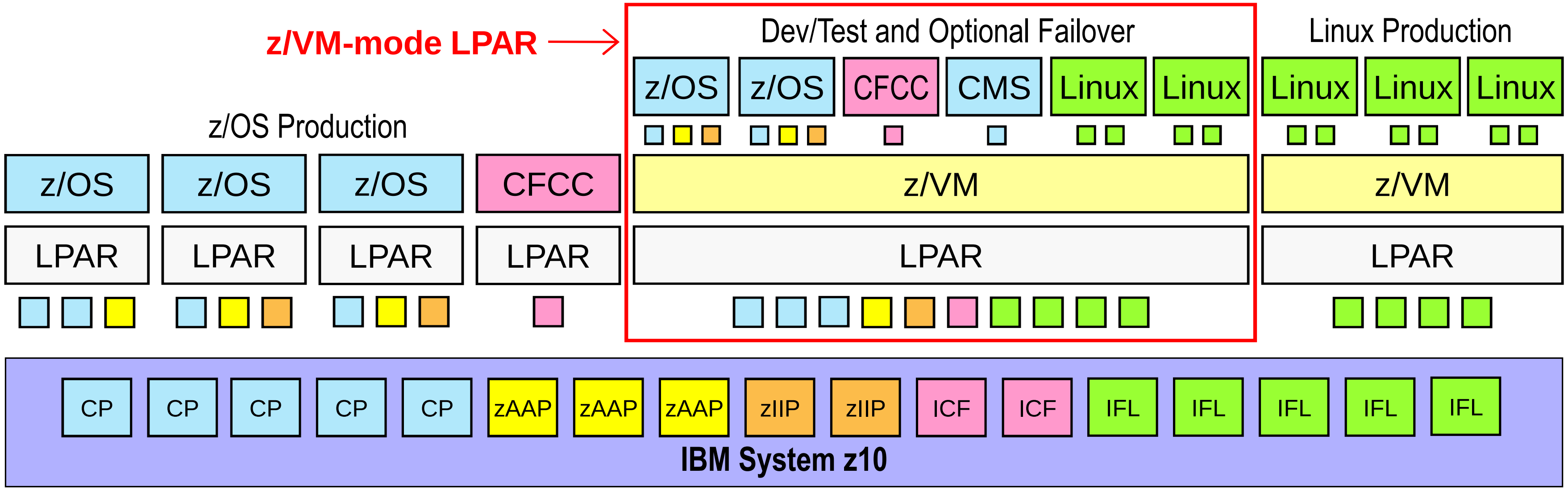
Using Real Specialty Engines in Virtual Machines

- Allows users to test and verify z/OS specialty processor support on the real hardware
- Users can maximize real specialty processor utilization by sharing processors among production and test LPARs
- Consumes specialty processor capacity to host virtual zAAP and zIIP processor cycles



z/VM-Mode LPAR Support for IBM zEnterprise Servers

- **New LPAR type for IBM System z10: *z/VM-mode***
  - Allows z/VM V5.4 users to configure all CPU types in a z10 LPAR
- **Offers added flexibility for hosting mainframe workloads**
  - Add *IFLs* to an existing standard-engine z/VM LPAR to host Linux workloads
  - Add *CPs* to an existing IFL z/VM LPAR to host z/OS, z/VSE, or traditional CMS workloads
  - Add *zAAPs* and *zIIPs* to host eligible z/OS specialty-engine processing
  - Test integrated Linux and z/OS solutions in the same LPAR
- **No change to software licensing**
  - Software continues to be licensed according to CPU type



Some Additional Background

- **CPU Affinity**
  - Setting to control whether virtualized Specialty Engines must be dispatched on real processors of that type
  - ON means virtual type must equal real type
  - Suppressed: you have asked for ON, but we don’t have processors of that type to use
- **Processor Type Pools**
  - Scheduling is done within a pool for CPUAFFINITY ON
  - Capacity Planning of each type
  - ATOD, ATOD2, etc. for each Processor Type Pool
- **Primary vs. Secondary Processor**
  - Primary: CP or IFL
  - Secondary: zAAP, zIIP, and sometimes IFL (secondary to CPs)
- **Different Speed Processors**
  - Specialty engines are full-speed on all System z Servers , while some general purpose run at a fractional speed.
- **The z/VM Scheduler is a deadline scheduler, not a consumption scheduler**

Considerations for z/VM-mode LPARs

- **Merging IFL only and CP only partitions in a z/VM-mode partition requires planning**
  - First step, make virtual machines on IFL LPAR have virtual IFLs
  - For duplicated work (RACF, TCP/IP, etc.), need to determine which to use or in some cases which to duplicate
  - Remember that in some environments, the IFLs may be faster than the CPs.
  - Determine any changes you want to make to the charge back model.

Configuration

- **See z/VM Running Guest Operating Systems book for more details**
  - CP Planning & Admin book is thin on this topic
- **To define virtual specialty engines use CP command:**
  - CP DEFINE CPU 01 TYPE ZIIP
  - For directory, can use COMMAND statement with above
  - Directory CPU statement cannot take TYPE option
- **Set virtual configuration mode of virtual machine via:**
  - CP SET VCONFIG MODE VM
  - CP SET VCONFIG MODE LINUX, etc.
- **z/OS must have all virtual CPUs defined prior to IPLing guest**

Default VCONFIG Settings

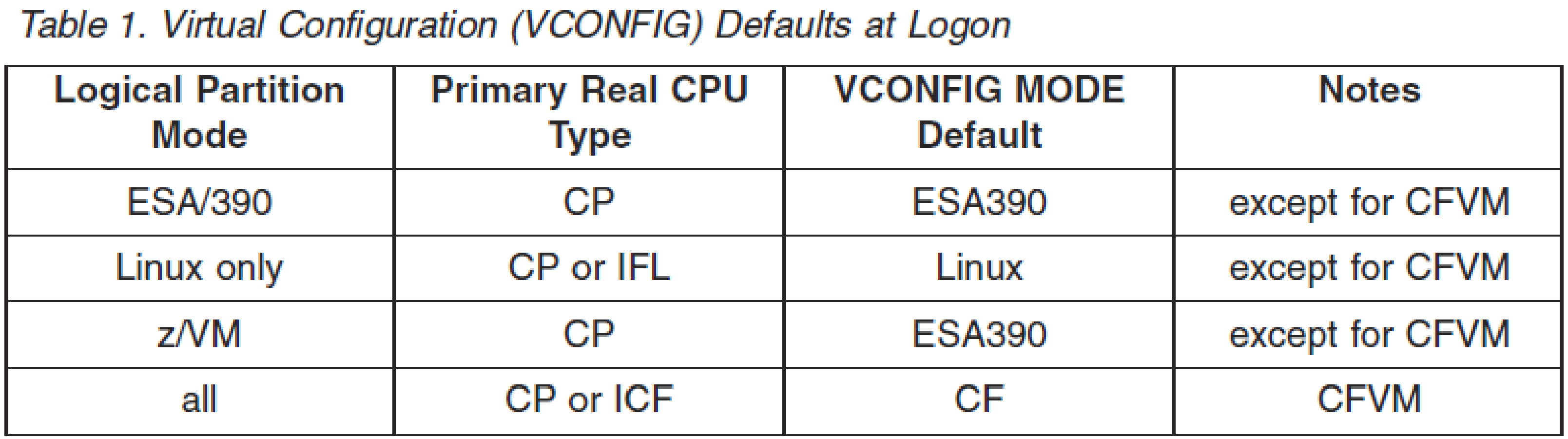


Table 1. Virtual Configuration (VCONFIG) Defaults at Logon

Logical Partition Mode	Primary Real CPU Type	VCONFIG MODE Default	Notes
ESA/390	CP	ESA390	except for CFVM
Linux only	CP or IFL	Linux	except for CFVM
z/VM	CP	ESA390	except for CFVM
all	CP or ICF	CF	CFVM

```
CP QUERY PROCESSORS EXPANDED
PROCESSOR 00 MASTER CP
PROCESSOR 01 ALTERNATE CP
PROCESSOR 02 ALTERNATE CP
PROCESSOR 03 ALTERNATE CP
PARTITION MODE ESA/390
```

Coupling Facility Virtual Machine are somewhat of an exception.
These must have **OPTION CFVM** in directory.

Output from INDICATE USER EXPanded

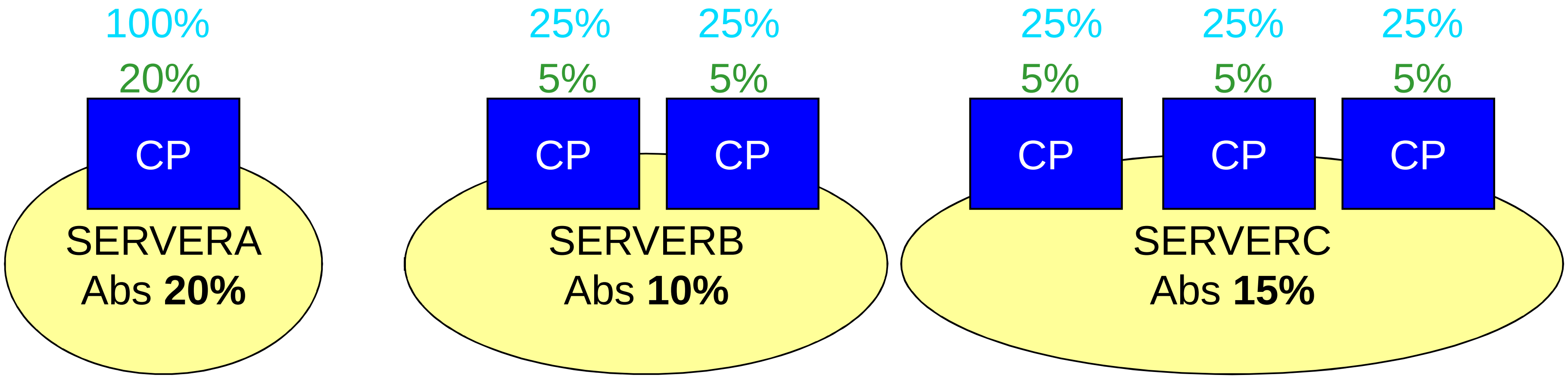
```
CPU 00: Ctime=0 00:00:53  Vtime=0 00:00:00  Ttime=0 00:00:00
        Rdr=0  Prt=0  Pch=0  IO=332
        Type=CP  CPUAffinity=ON
        VtimePrimary=0 00:00:00    TtimePrimary=0 00:00:00
        VtimeSecondary=0 00:00:00  TtimeSecondary=0 00:00:00
CPU 01: Ctime=0 00:00:30  Vtime=0 00:00:00  Ttime=0 00:00:00
        Rdr=0  Prt=0  Pch=0  IO=0
        Type=ZAAP  CPUAffinity=SUPP
        VtimePrimary=0 00:00:00    TtimePrimary=0 00:00:00
        VtimeSecondary=0 00:00:00  TtimeSecondary=0 00:00:00
CPU 02: Ctime=0 00:00:20  Vtime=0 00:00:00  Ttime=0 00:00:00
        Rdr=0  Prt=0  Pch=0  IO=0
        Type=ZIIP  CPUAffinity=SUPP
        VtimePrimary=0 00:00:00    TtimePrimary=0 00:00:00
        VtimeSecondary=0 00:00:00  TtimeSecondary=0 00:00:00
```

Specialty Engines and Share Settings

- **The Share setting for a virtual machine applies to each pool of the processor types**
  - CP, IFL, zIIP, zAAP, etc.
- **z/VM 5.4.0 added support to set a separate share setting for each processor type pool**
  - Default is TYPE ALL and results in one setting for all types
- **Normalized to the sum of shares of virtual machines in dispatch list for each pool of the processor types**
- **Absolute (and normalized) is percentage of resources of a given processor type.**

NN% = (IPW) In Perfect World percentage of real processor

NN% = split of share per virtual processor

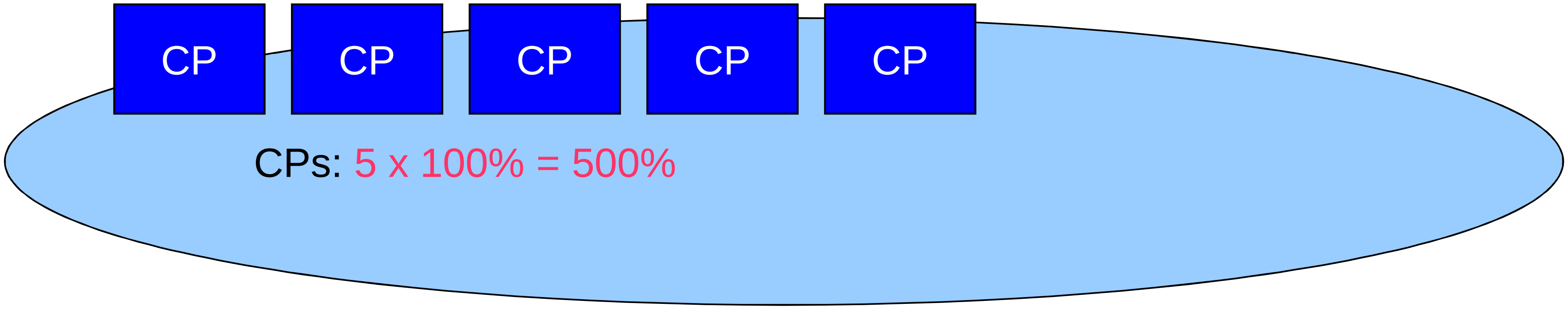


100%
20%
CP
SERVERA
Abs **20%**

25% 25%
5% 5%
CP CP
SERVERB
Abs **10%**

25% 25% 25%
5% 5% 5%
CP CP CP
SERVERC
Abs **15%**

(**20%+10%+15%**) x 500% = 225% = 100%+25%+25%+25%+25%+25%

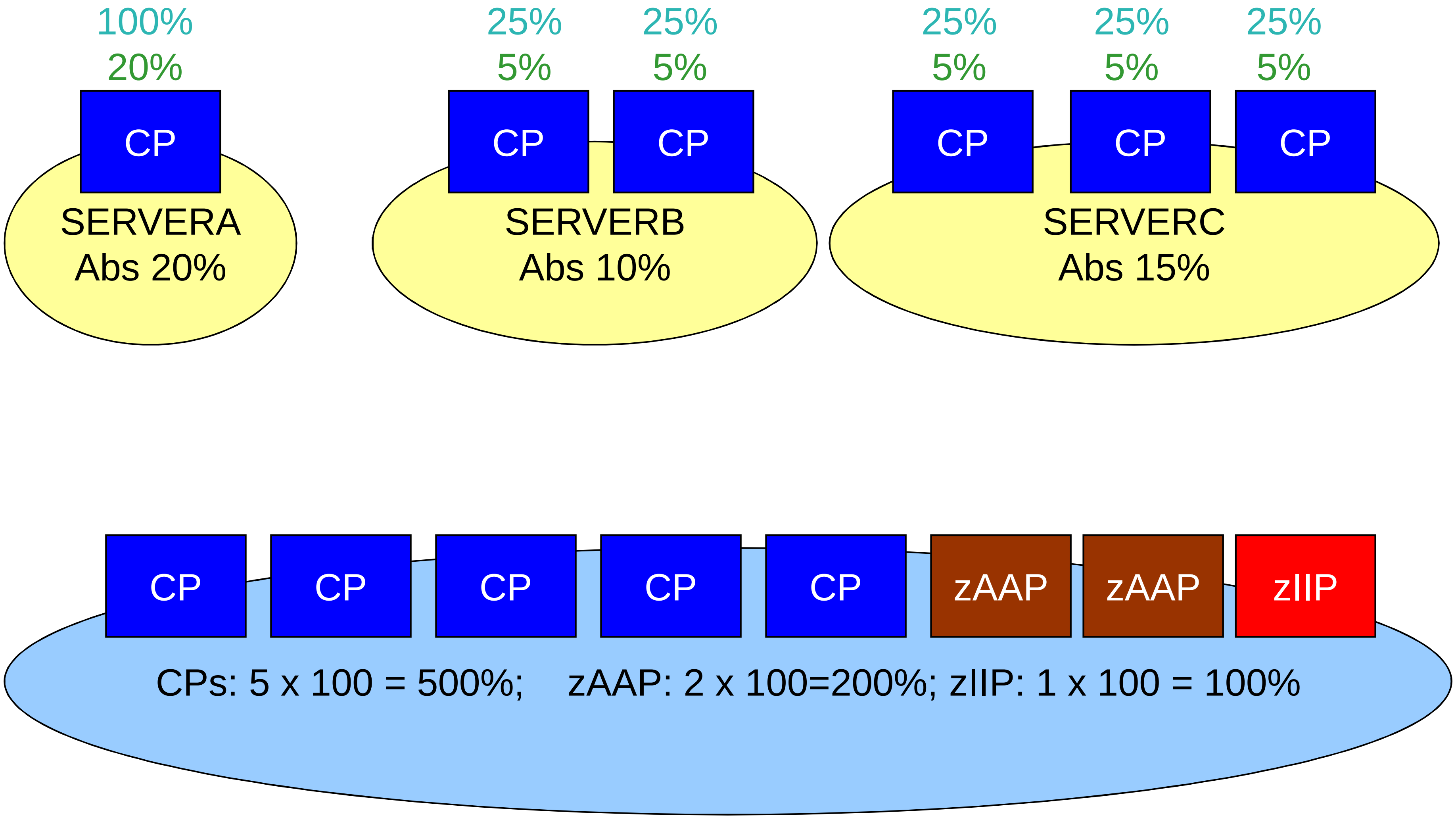


CP CP CP CP CP

CPs: 5 x 100% = 500%

NN% = IPW percentage of real processor

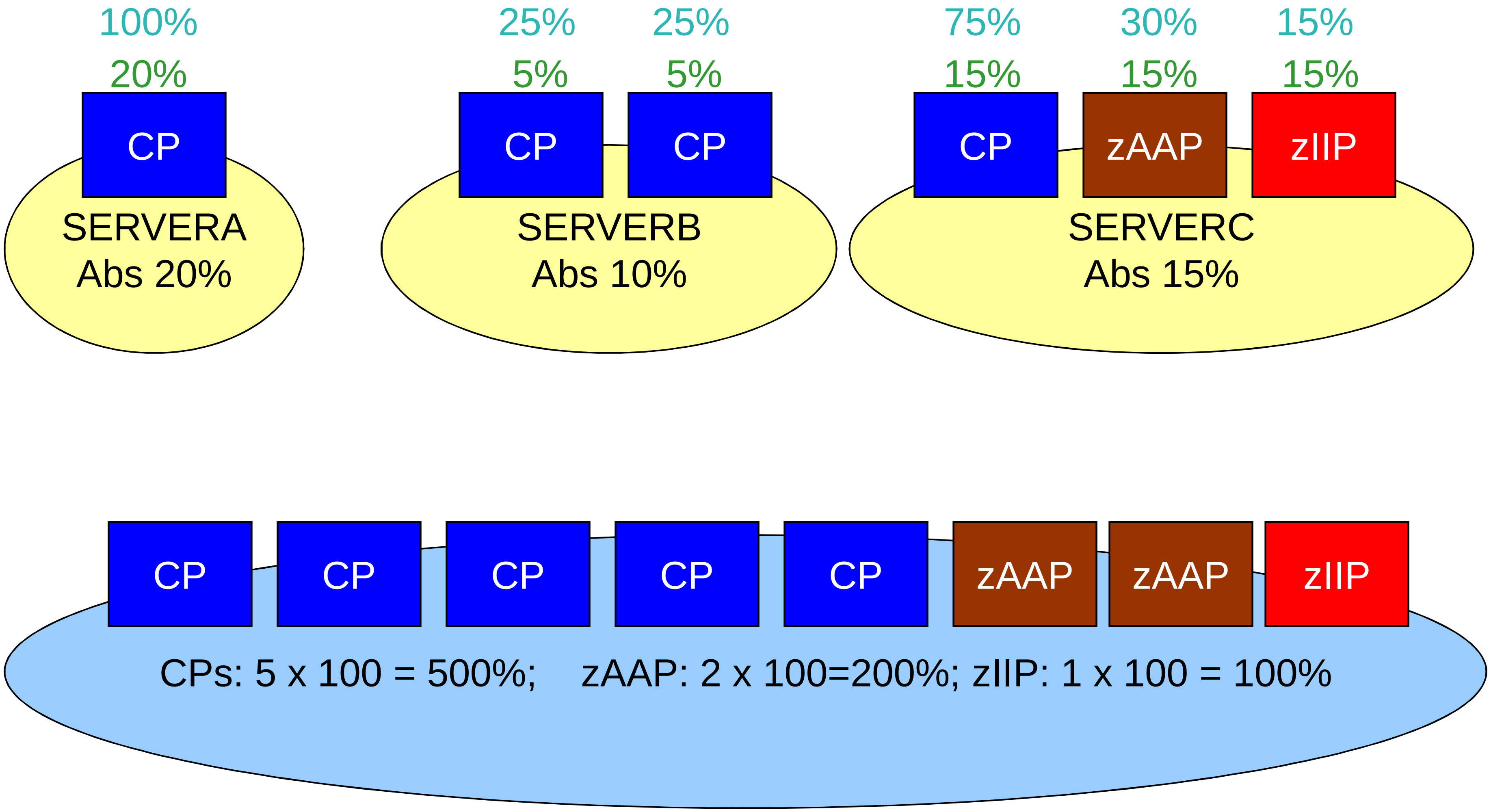
NN% = split of share per virtual processor



CPs: 5 x 100 = 500%; zAAP: 2 x 100=200%; zIIP: 1 x 100 = 100%

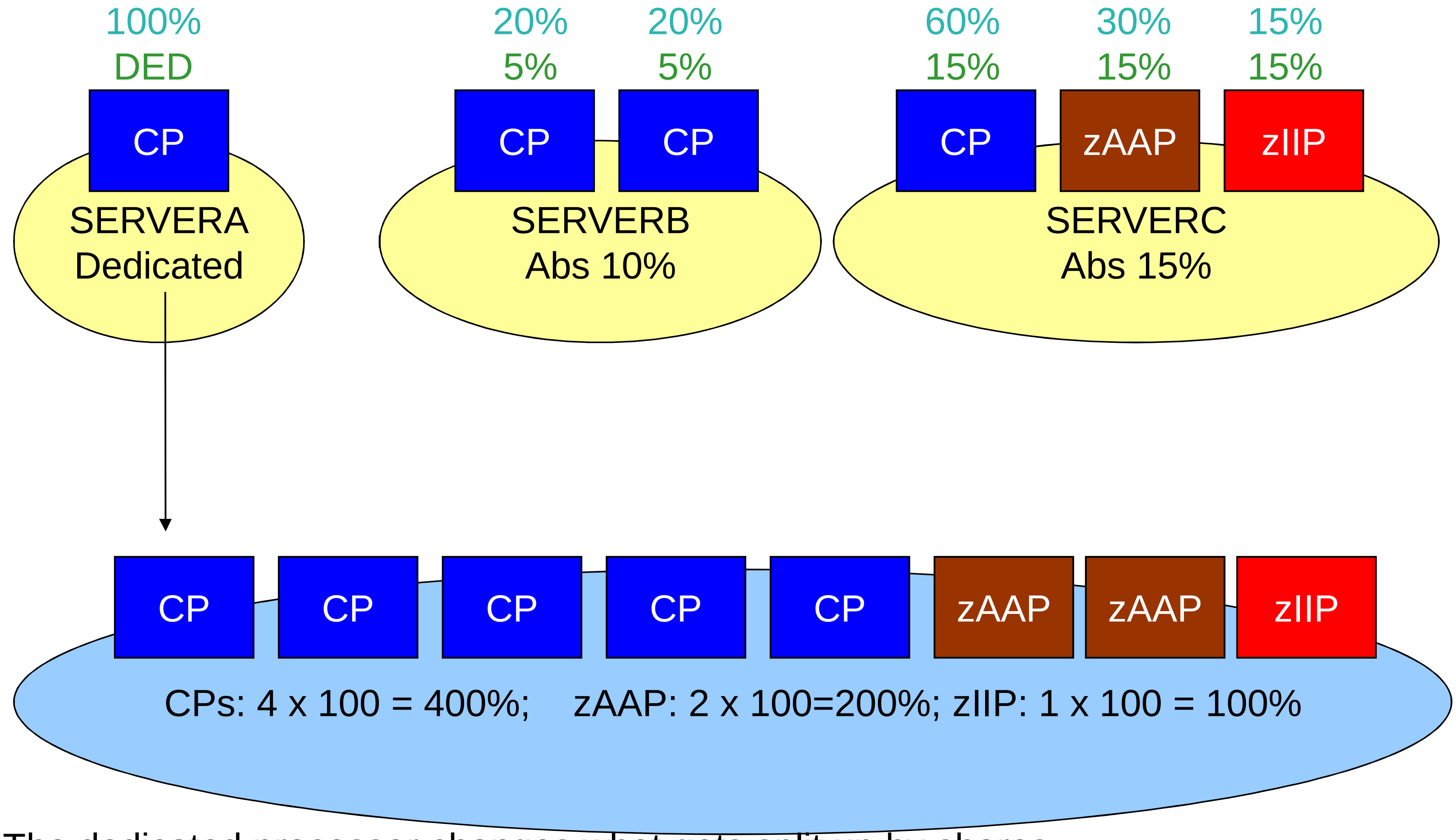
NN% = IPW percentage of real processor

NN% = split of share per virtual processor



NN% = IPW percentage of real processor

NN% = split of share per virtual processor

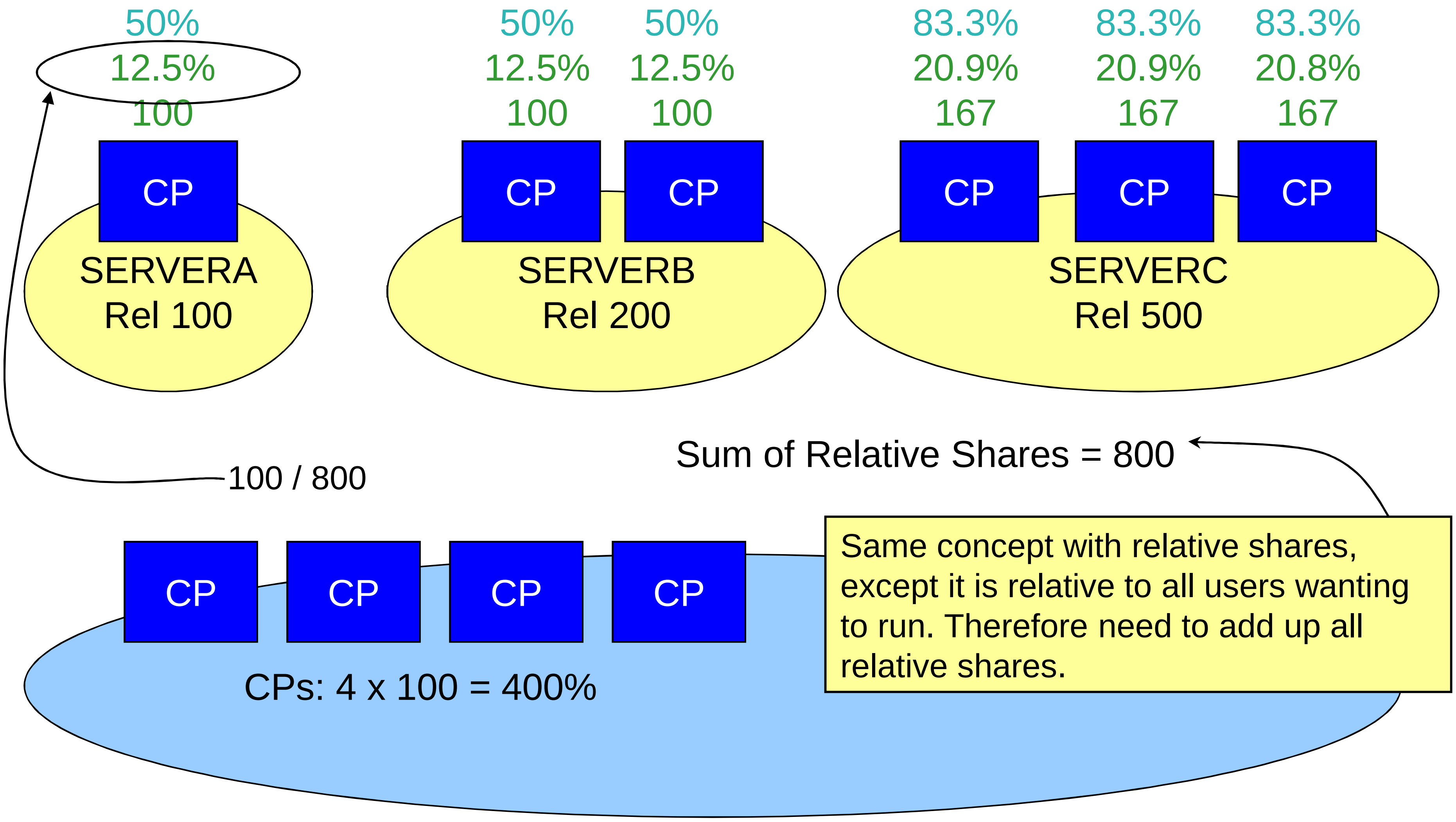


The dedicated processor changes what gets split up by shares.

NN% = IPW percentage of real processor

NN% = split of share per virtual processor

NN = relative split of share per virtual processor

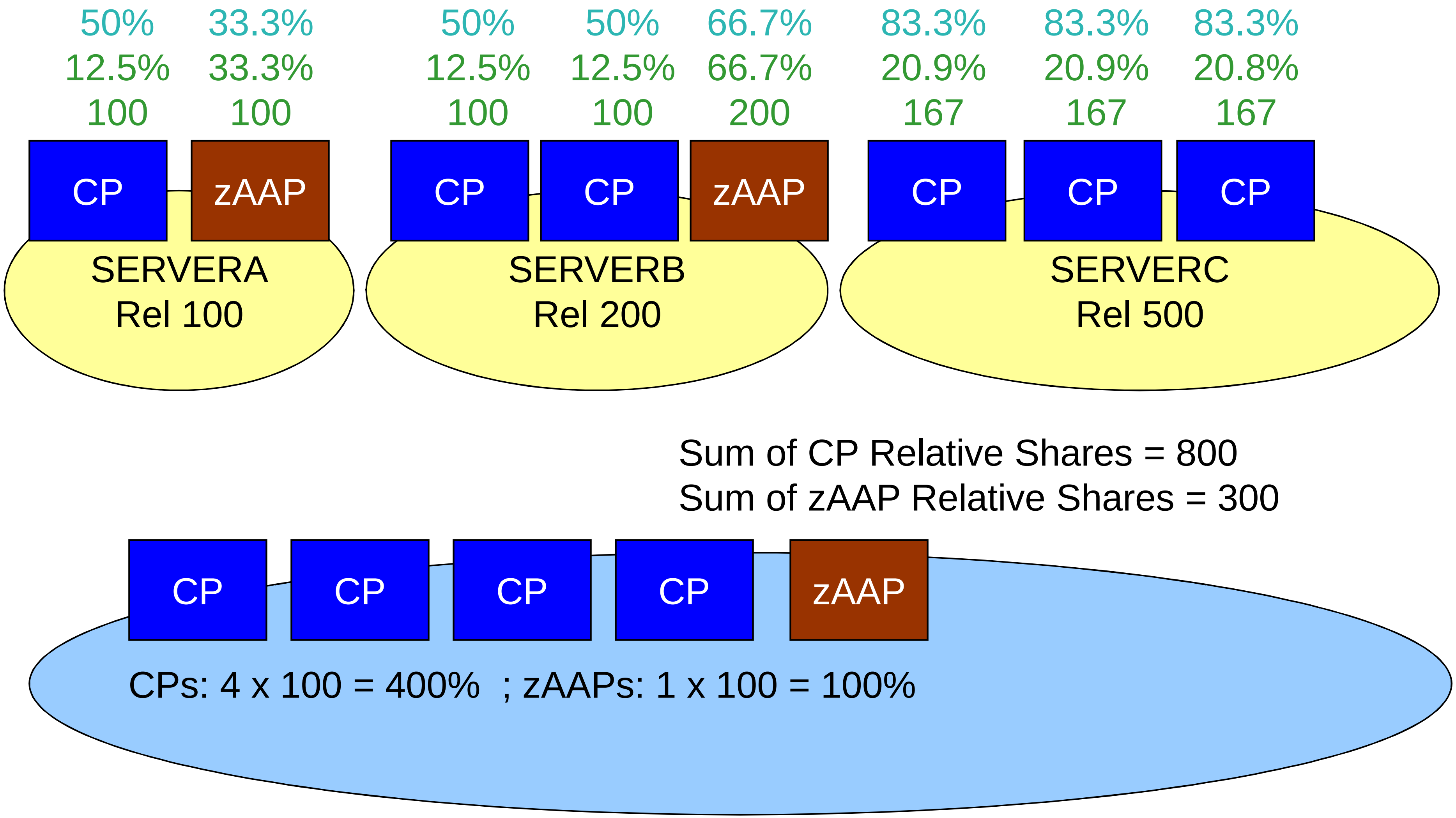


Same concept with relative shares, except it is relative to all users wanting to run. Therefore need to add up all relative shares.

NN% = IPW percentage of real processor

NN% = split of share per virtual processor

NN = relative split of share per virtual processor



CP	zAAP	CP	CP	zAAP	CP	CP	CP
50%	33.3%	50%	50%	66.7%	83.3%	83.3%	83.3%
12.5%	33.3%	12.5%	12.5%	66.7%	20.9%	20.9%	20.8%
100	100	100	100	200	167	167	167

SERVERA
Rel 100

SERVERB
Rel 200

SERVERC
Rel 500

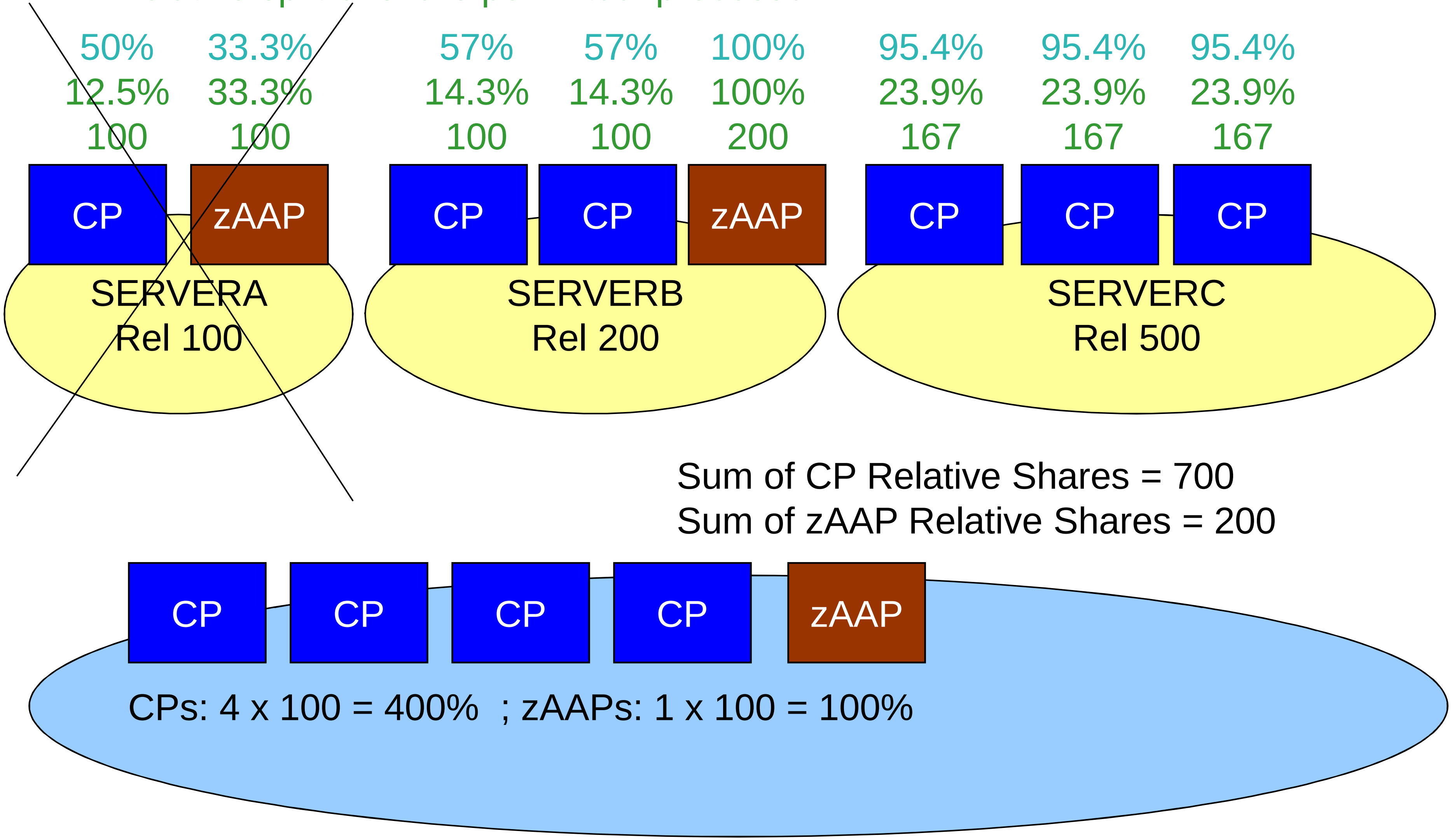
Sum of CP Relative Shares = 800
Sum of zAAP Relative Shares = 300

CPs: 4 x 100 = 400% ; zAAPs: 1 x 100 = 100%

NN% = IPW percentage of real processor

NN% = split of share per virtual processor

NN = relative split of share per virtual processor



SERVERA Rel 100		SERVERB Rel 200			SERVERC Rel 500		
50%	33.3%	57%	57%	100%	95.4%	95.4%	95.4%
12.5%	33.3%	14.3%	14.3%	100%	23.9%	23.9%	23.9%
100	100	100	100	200	167	167	167
CP	zAAP	CP	CP	zAAP	CP	CP	CP

Sum of CP Relative Shares = 700

Sum of zAAP Relative Shares = 200

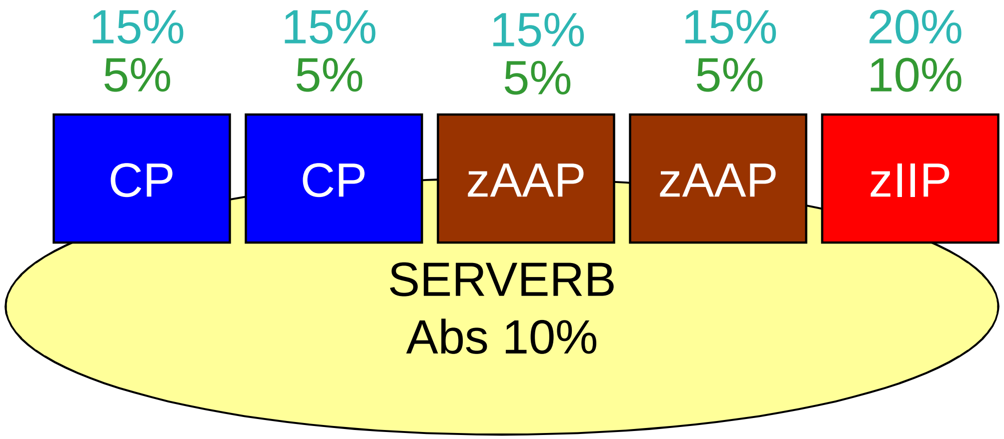
CP CP CP CP zAAP

CPs: 4 x 100 = 400% ; zAAPs: 1 x 100 = 100%

NN% = IPW percentage of real processor

NN% = split of share per virtual processor

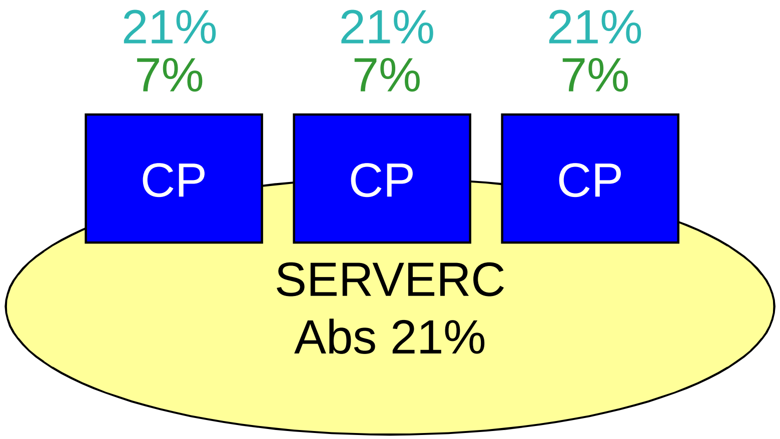
Tricky Scenario



15% 5% CP | 15% 5% CP | 15% 5% zAAP | 15% 5% zAAP | 20% 10% zIIP

SERVERB

Abs 10%



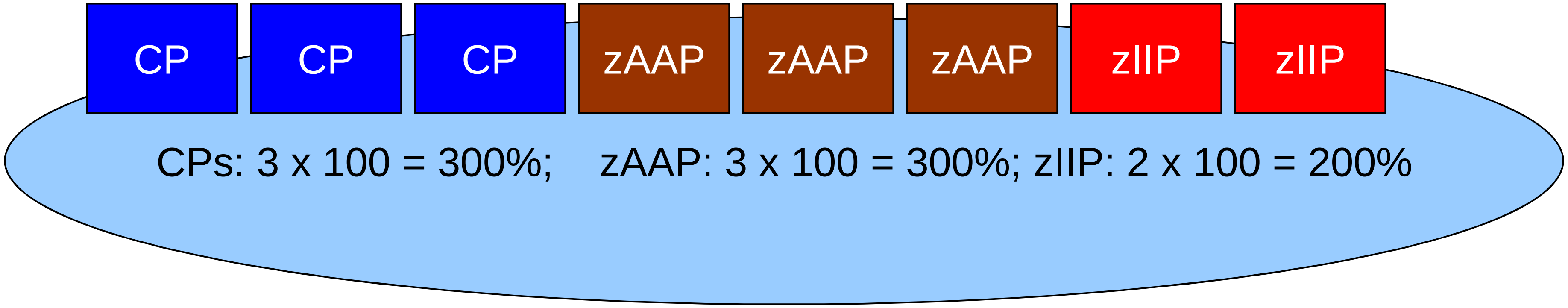
21% 7% CP | 21% 7% CP | 21% 7% CP

SERVERC

Abs 21%

Total Processor for SERVERB
is 15+15+15+15+20 = 80%

Total Processor for SERVERC
is 21+21+21 = 63%



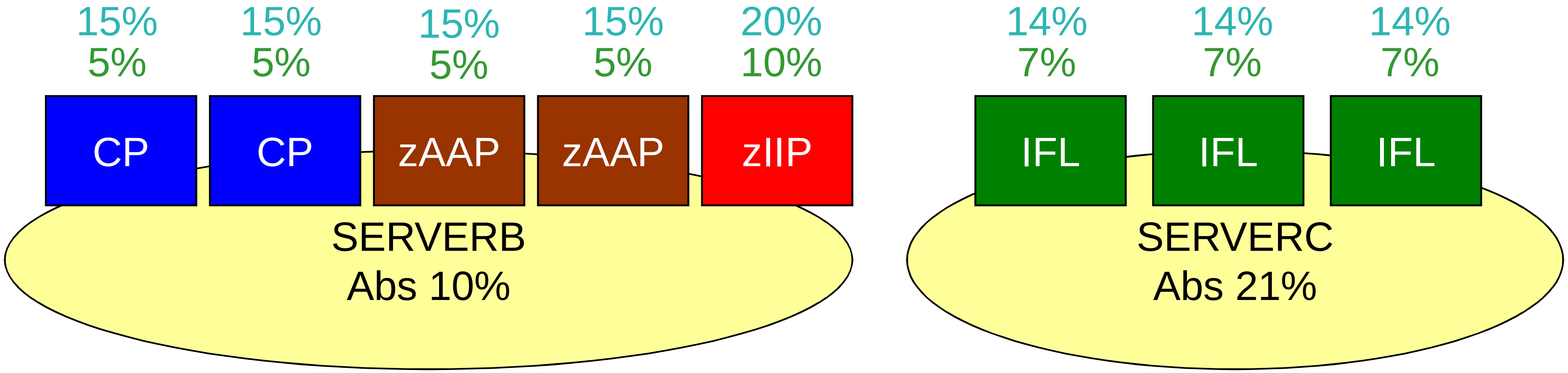
CP | CP | CP | zAAP | zAAP | zAAP | zIIP | zIIP

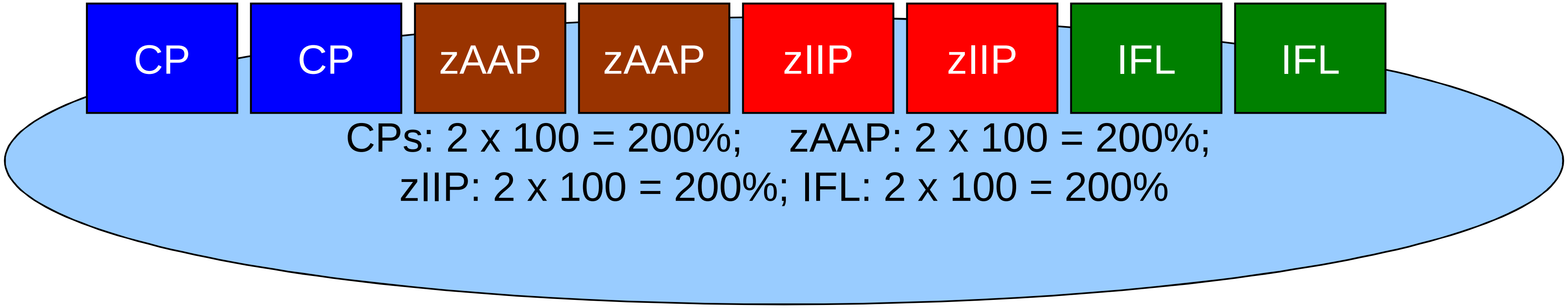
CPs: 3 x 100 = 300%; zAAP: 3 x 100 = 300%; zIIP: 2 x 100 = 200%

NN% = IPW percentage of real processor

NN% = split of share per virtual processor

z/VM-Mode LPAR



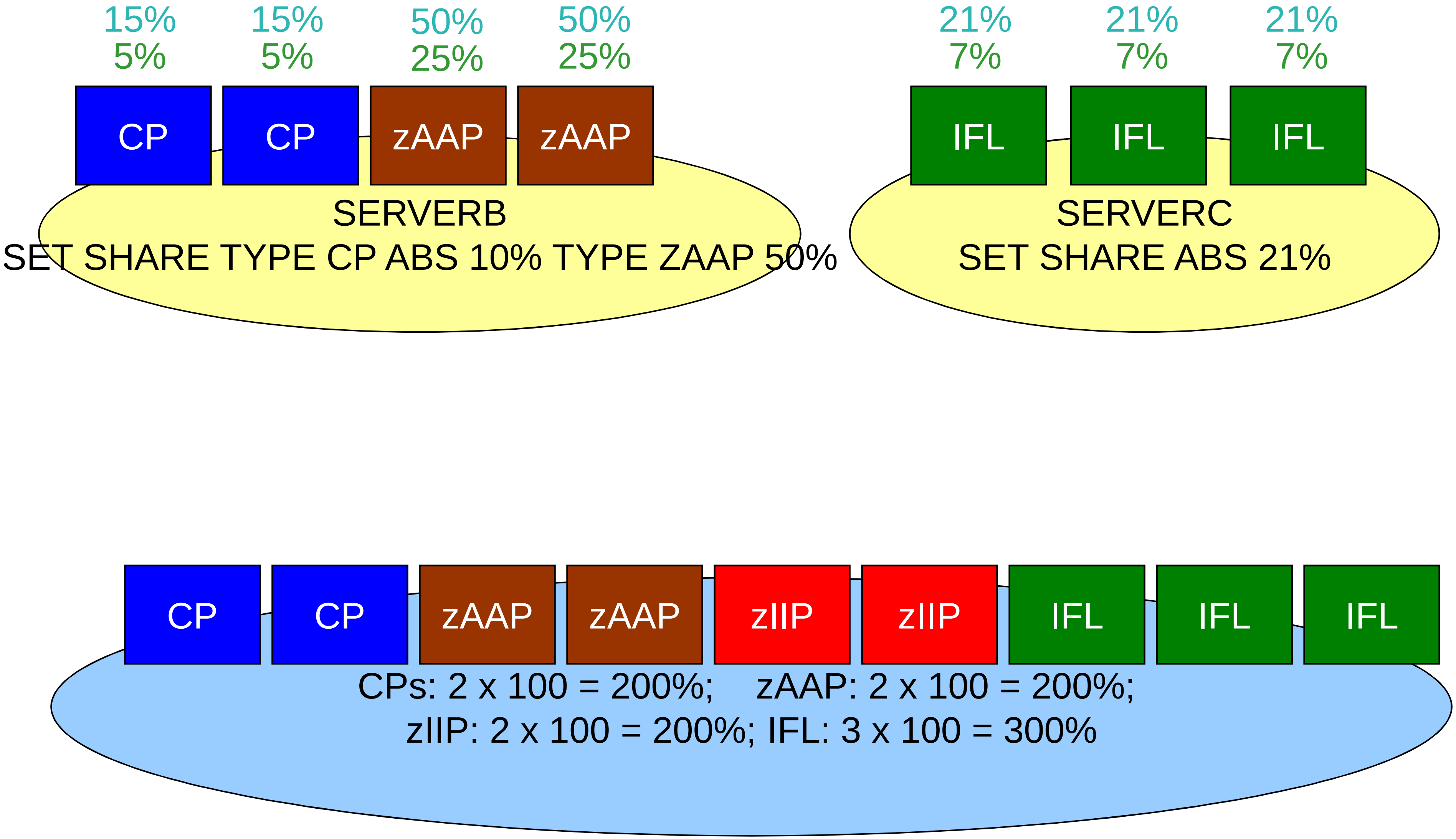


CPs: 2 x 100 = 200%; zAAP: 2 x 100 = 200%;
zIIP: 2 x 100 = 200%; IFL: 2 x 100 = 200%

NN% = IPW percentage of real processor

NN% = split of share per virtual processor

z/VM-Mode LPAR



15% 5% CP · 15% 5% CP · 50% 25% zAAP · 50% 25% zAAP

SERVERB

SET SHARE TYPE CP ABS 10% TYPE ZAAP 50%

21% 7% IFL · 21% 7% IFL · 21% 7% IFL

SERVERC

SET SHARE ABS 21%

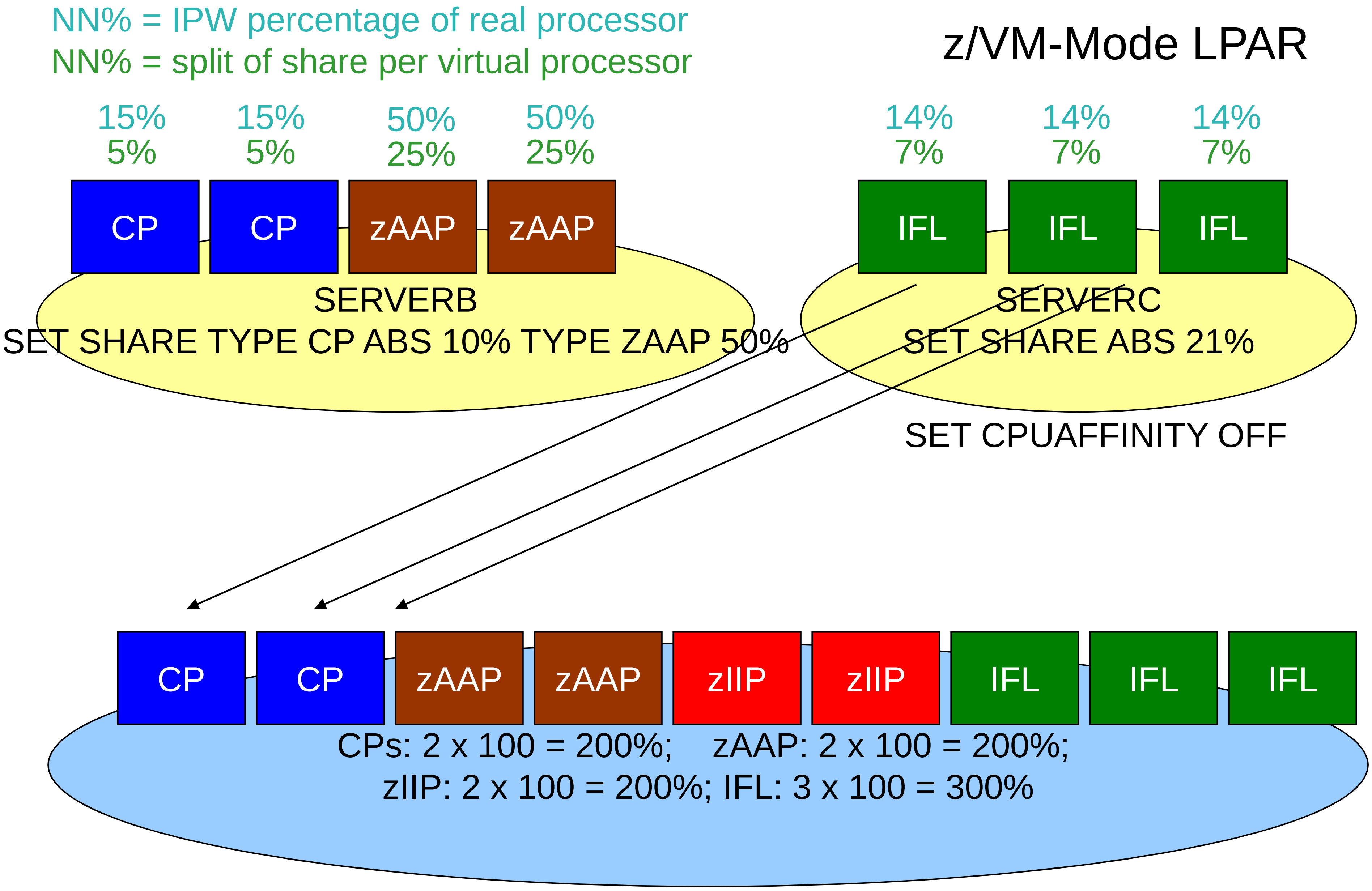
CP · CP · zAAP · zAAP · zIIP · zIIP · IFL · IFL · IFL

CPs: 2 x 100 = 200%; zAAP: 2 x 100 = 200%;

zIIP: 2 x 100 = 200%; IFL: 3 x 100 = 300%

NN% = IPW percentage of real processor

NN% = split of share per virtual processor



z/VM-Mode LPAR

15%
5%

15%
5%

50%
25%

50%
25%

CP CP zAAP zAAP

SERVERB

SET SHARE TYPE CP ABS 10% TYPE ZAAP 50%

14%
7%

14%
7%

14%
7%

IFL IFL IFL

SERVERC

SET SHARE ABS 21%

SET CPUAFFINITY OFF

CP CP zAAP zAAP zIIP zIIP IFL IFL IFL

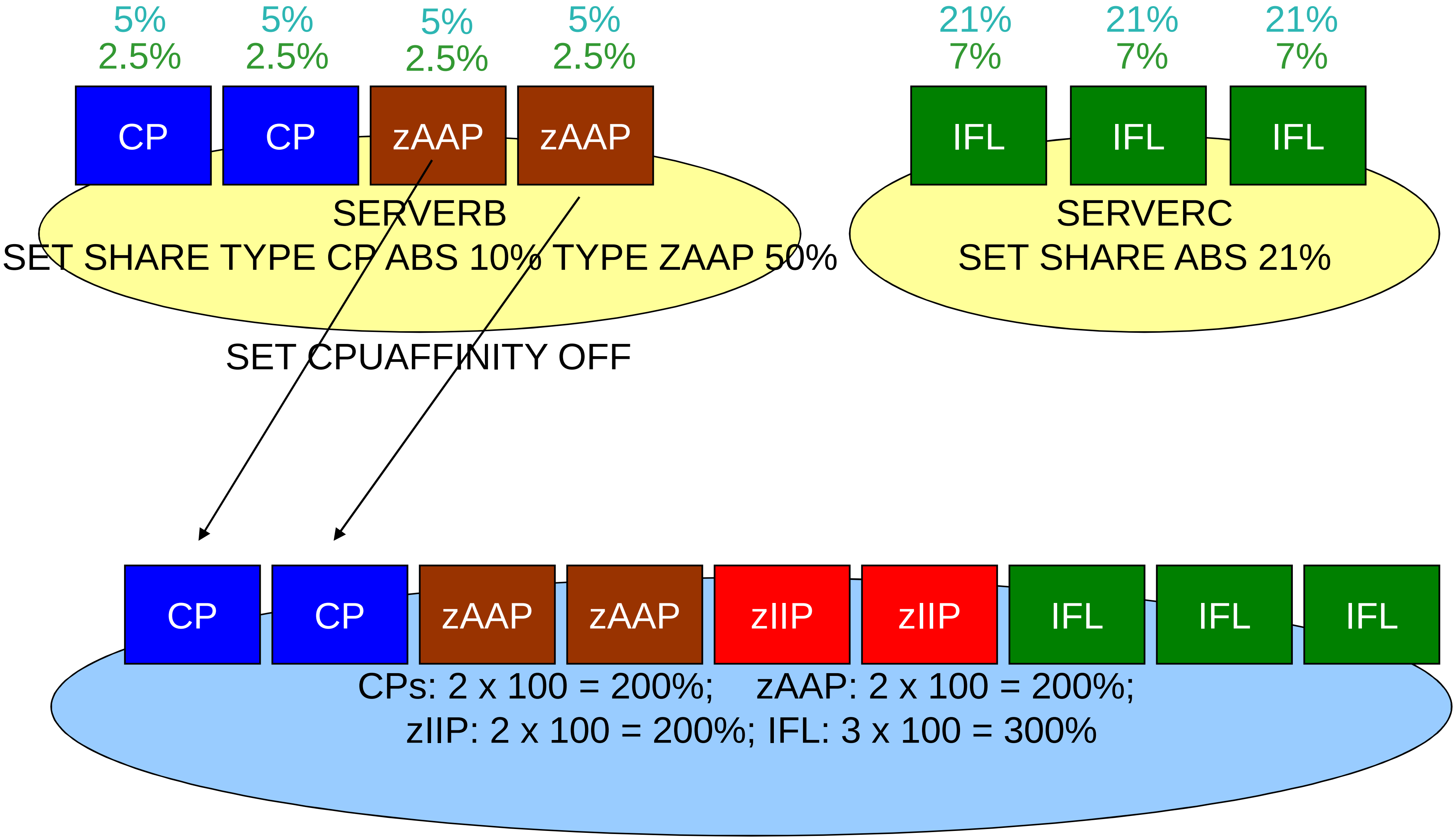
CPs: 2 x 100 = 200%; zAAP: 2 x 100 = 200%;

zIIP: 2 x 100 = 200%; IFL: 3 x 100 = 300%

NN% = IPW percentage of real processor

NN% = split of share per virtual processor

z/VM-Mode LPAR



5% 2.5% | 5% 2.5% | 5% 2.5% | 5% 2.5%

CP | CP | zAAP | zAAP

SERVERB

SET SHARE TYPE CP ABS 10% TYPE ZAAP 50%

SET CPUAFFINITY OFF

21% 7% | 21% 7% | 21% 7%

IFL | IFL | IFL

SERVERC

SET SHARE ABS 21%

CP | CP | zAAP | zAAP | zIIP | zIIP | IFL | IFL | IFL

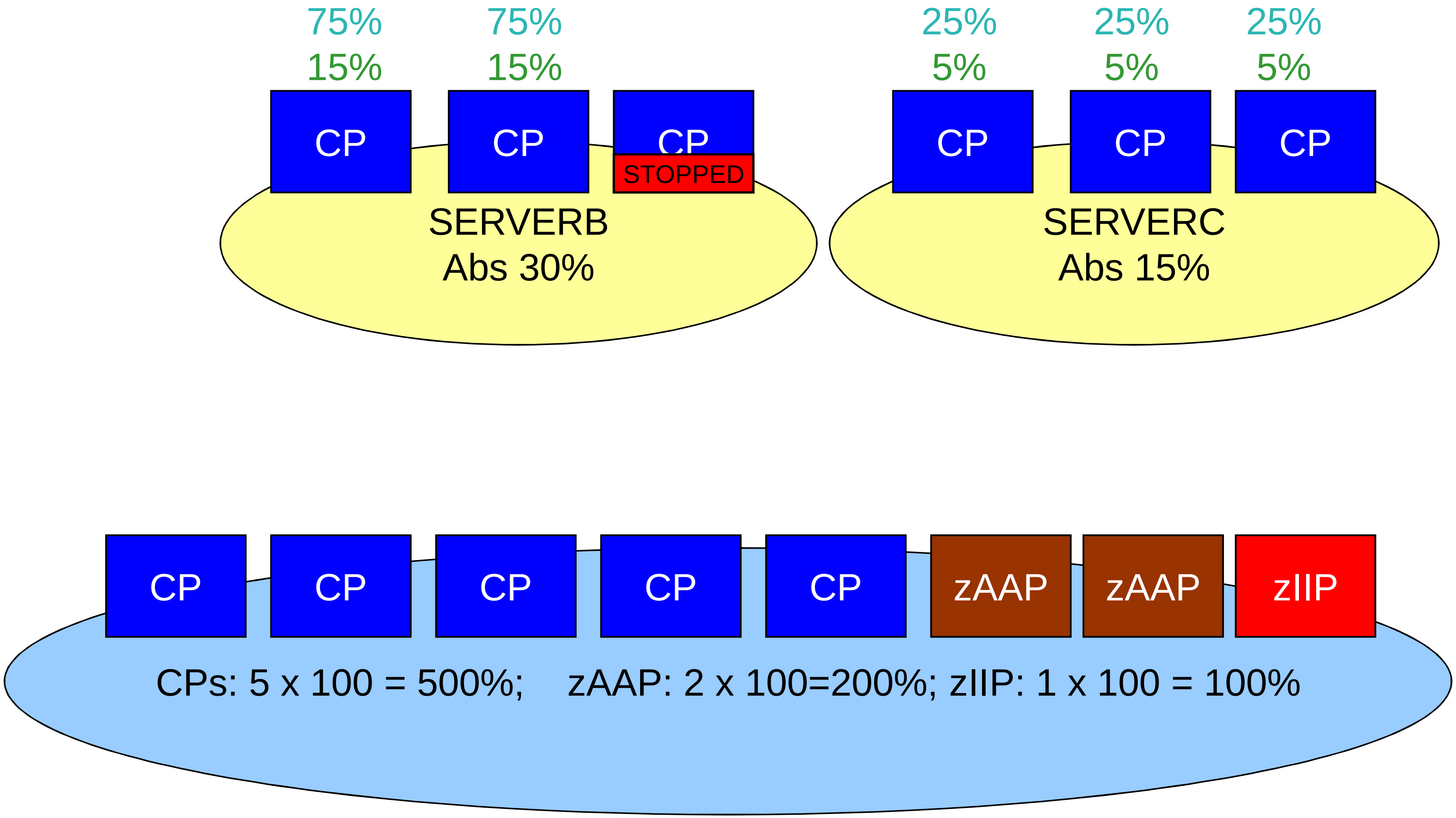
CPs: 2 x 100 = 200%; zAAP: 2 x 100 = 200%;

zIIP: 2 x 100 = 200%; IFL: 3 x 100 = 300%

NN% = IPW percentage of real processor

NN% = split of share per virtual processor

Share Redistribution



FCX126 LPAR – updated

```
FCX126  Run 2007/06/20 09:55:12         LPAR
                                        Logical Partition Activity

 Partition Nr.  Upid #Proc Weight Wait-C Cap %Load CPU %Busy %Ovhd %Susp %VMld %Logld Type
 VMT1        4    04     5     20     NO  NO  51.5   0  99.9    .0    .1  99.9   99.9 CP
                               20         NO         1  99.9    .0    .1  99.8   99.9 CP
                               20         NO         2  99.9    .1    .1  99.8   99.9 CP
                               80         NO         3    .0    .0    .4    .0     .0 ZAAP
                               80         NO         4   9.3    .2    .3   9.0    9.0 ZIIP

 Summary of physical processors:
 Type  Number  Weight  Dedicated  %LPBUSY  %LPOVHD %NCOVHD  %BUSY
 CP         3     100          0      299        1       0    300
 ZAAP       1     100          0        0        0       0      0
 IFL        1     200          0      151        4       2    157
 ZIIP       1     100          0        9        0       0      9
```

FCX202 LPARLOG

```
FCX202  Run 2007/06/20 09:55:12         LPARLOG
                                       Logical Partition Activity Log

Interval <Partition->                                      <- Load per Log. Processor -->
End Time Name    Nr.  Upid #Proc Weight Wait-C Cap %Load  %Busy %Ovhd %Susp %VMld %Logld Type
>>Mean>> VMT1      4    04     5     80     NO  NO   ...   61.8    .1    .2  61.7   61.7 MIX
>>Mean>> Total    ..    ..     6    100     ..  ..    .1   30.9    .0   ...   ...    ... ..
```

FCX180 SYSCONF

```
FCX180  Run 2007/06/20 09:53:49          SYSCONF
                                         System Configuration

 Initial Status on 2007/03/07 at 21:30, Processor 2096-X03
 Real Proc: Cap  2224, Total  7, Conf  3, Stby  0, Resvd  4
 Sec. Proc: Cap  1760, Total  3, Conf  3, Stby  0, Resvd  2
```

- With new Specialty Engine support, your z/VM system may include processors that are different speeds.
- Smaller “Cap” number indicates faster Processor.

FCX144 PROCLOG – updated

```
FCX144   Run 2007/06/20 09:53:49         PROCLOG
                                          Processor Activity, by T ime
________________________________________________________________  ________

                  <------ Percent Busy -------> <--- Rates per Sec.--->
           C
 Interval  P                                          Inst
 End Time  U Type Total  User   Syst  Emul  Vect  Siml  DIAG  SIGP  SSCH
 >>Mean>>  0 CP    99.8  99.5     .3  97.9  .... 125.0  12.6   . 7  71.6
 >>Mean>>  1 CP    99.8  99.5     .2  98.0  .... 120.9   4.5   . 8  58.4
 >>Mean>>  2 CP    99.8  99.5     .3  98.0  .... 123.4   3.2   . 7  59.5
 >>Mean>>  3 ZAAP  96.0  96.0     .1  95.8  ....   1.1    .0  36.6   1.4
 >>Mean>>  4 ZIIP   8.8   8.4     .4   8.1  ....   1.0    .0 289. 9   7.5

 >>Mean>>  . CP    99.7  99.5     .2  98.0  .... 123.0   6.7   . 7  63.1
 >>Mean>>  . ZAAP  96.0  96.0     .1  95.8  ....   1.1    .0  36.6   1.4
 >>Mean>>  . ZIIP   8.8   8.4     .4   8.1  ....   1.0    .0 289. 9   7.5
```

Virtual Sysplex Environments

- **Key is tuning effectively for the virtual coupling machines**
  - QUICKDSP ON
  - Sufficiently High Share setting
  - Using real ICFs in z/VM-mode LPARs in z/VM 5.4.0 where appropriate
- **Beware of scenarios with both a large number of systems in a virtual sysplex and the systems join and leave the sysplex frequently.**
  - High CP CPU overhead as the z/OS systems that are not changing state issue large number of messages while the coupling machine is busy making updates for the system that is leaving/joining.
  - Privileged Operations count will be very high (>10,000s/second)

Miscellaneous Tuning Thoughts

- **Disable IRD for z/OS virtual machines**
- **PAV Usage & Trade-offs**
  - Dedicating volumes to z/OS guests and letting them use PAV can be the best performance
  - More flexibility in sharing volumes and using PAV volumes through the minidisk support
  - Analysis of where the I/Os are queued up may require looking at both z/VM data and z/OS data
- **HiperDispatch**
  - Does not apply in z/VM guests
- **In z/VM Mode LPAR, z/VM IPLs and uses CPs for initialization**
  - Be careful of short-changing yourself on CPs.

Summary

- **Specialty Engines enhance z/VM's virtualization capabilities**
- **A few things to keep in mind…**
  - Share for virtual machine applies to each processor type pool
  - There are scenarios where processors can be different speeds
  - Looking at averages takes on new meaning
  - CPU Affinity Setting is important
- **Monitor and Accounting records updated to provide needed information**
- **For more on scheduling, see z/VM Scheduler Overview**
  - http://www.vm.ibm.com/devpages/bitner/presents/zvmsched.pdf
- **For more on Specialty Engine Performance, see Performance Report**
  - http://www.vm.ibm.com/perf/reports/zvm/html/530se.html

Contact Information

Contact Info:

Bill Bitner
z/VM Customer Focus and Care
z/VM Development Lab – Endicott, NY
bitnerb@us.ibm.com
+1 607-429-3286