



IBM Systems and Technology Group

HiperDispatch Technical Overview

*Roger Fowler
Consulting IT Specialist
IBM United Kingdom Limited*



IBM Systems

Trademarks

The following are trademarks of the International Business Machines Corporation in the United States and/or other countries.

AIX*	ES/9000*	Language Environment*	S/390*
AnyNet*	FICON*	Lotus*	Sysplex Timer*
CICS*	FlashCopy*	Multiprise*	System Storage
DB2*	GDPS*	Notes*	System z
DB2 Connect	Geographically Dispersed Parallel Sysplex	OMEGAMON*	System z9
DB2 Universal Database	HiperSockets	On demand business logo	SystemPac*
developerWorks*	Hiperspace	OS/390*	Tivoli*
DFSMSdfp	HyperSwap	Parallel Sysplex*	TotalStorage*
DFSMSdss	IBM*	PR/SM	Virtualization Engine
DFSMSshsm	IBM eServer	Processor Resource/ Systems Manager	VTAM*
DFSMSrmm	IBM e(logo)server*	pSeries*	WebSphere*
DFSORT	IBM logo*	RACF*	z/Architecture
Domino*	IMS	RAMAC*	z/OS*
DRDA*	Infoprint*	Redbook	z/VM*
Enterprise Storage Server*	IP PrintWay	RMF	zSeries*
ESCON*	iSeries		

* Registered trademarks of IBM Corporation

The following are trademarks or registered trademarks of other companies.

Intel is a trademark of the Intel Corporation in the United States and other countries.

Linux is a trademark of Linux Torvalds in the United States, other countries, or both.

Java and all Java-related trademarks and logos are trademarks or registered trademarks of Sun Microsystems, Inc., in the United States and other countries.

Microsoft, Windows and Windows NT are registered trademarks of Microsoft Corporation.

UNIX is a registered trademark of The Open Group in the United States and other countries.

* 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.

This presentation and the claims outlined in it were reviewed for compliance with US law. Adaptations of these claims for use in other geographies must be reviewed by the local country counsel for compliance with local laws.

Acknowledgements

§ Original material by:

- ▶ **Bernie Pierce**
- ▶ **Alain Manneville**
- ▶ **Steve Grabarits**

AGENDA

§ HiperDispatch

- ▶ Vocabulary
- ▶ Background
- ▶ Double Dispatching
- ▶ Design objective
- ▶ HiperDispatch
- ▶ Planning
- ▶ Conclusion

VOCABULARY

§ LCP = Logical Central Processor

§ PCP = Physical Central Processor (aka CP)

§ LPAR = Logical Partition

- ▶ Contains the LCPs which are dispatched on the PCPs

§ WEIGHT = Partition weight (alias W)

- ▶ %SHARE = Percentage of the physical machine attributed to an LPAR
 - $\%SHARE_{LPARi} = WEIGHT_{LPARi} / WEIGHT_{total}$

§ DA / VCM

- ▶ Dispatcher Affinity / Vertical CPU Management

§ Marketing name:

- ▶ HiperDispatch
 - Refers to DA and VCM



IBM Systems and Technology Group

HiperDispatch

Background



IBM Systems

Background

§ HORIZONTAL MODE

- ▶ The PR/SM microcode distributes the WEIGHT of the LPAR uniformly across the ONLINE LCPs.
- ▶ This is what happens when HIPERDISPATCH=NO
- ▶ z/OS will utilize all the LCPs to obtain it's WEIGHT.

§ DISPATCHING

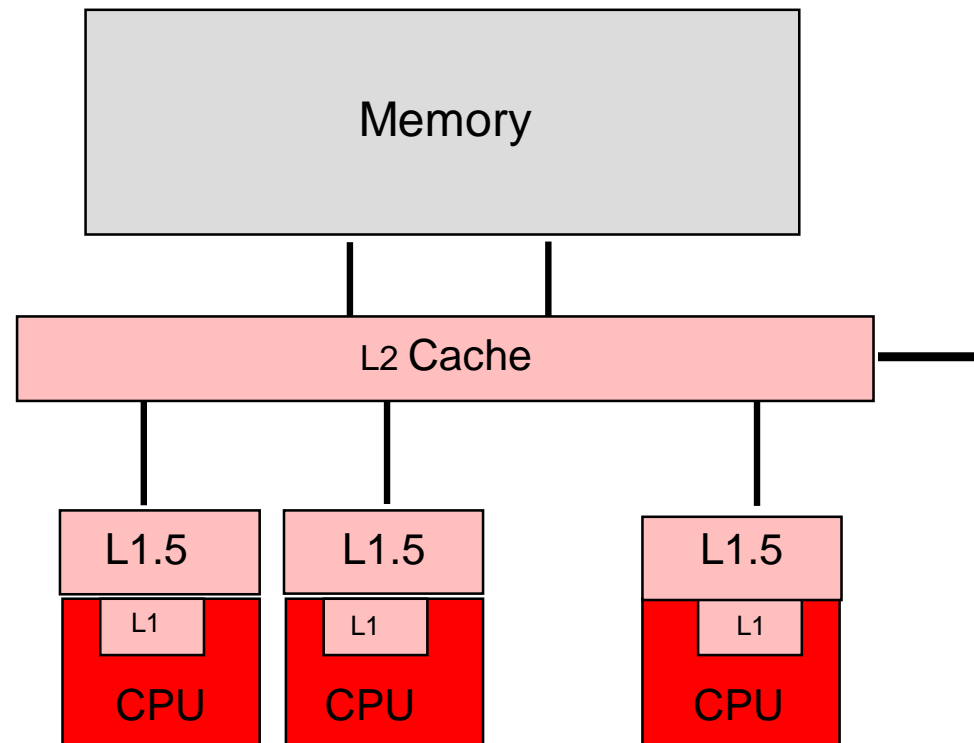
- ▶ Two levels of Dispatching
 - z/OS dispatches the TCBs & SRBs on the LCPs
 - PR/SM dispatches the LCPs on the PCPs
- ▶ HiperDispatch
 - PR/SM will create an LCP:PCP affinity to improve cache performance and utilization.
 - z/OS will create affinities for TCBs & SRBs to LCPs

Processor Design Basics

§ Processor Design

- CPU (core)
 - Cycle Time
 - Pipeline
 - Branch Prediction
 - Hardware vs Millicode
- Memory subsystem
 - High speed buffers (caches)
 - On chip / on Module
 - Private / Shared
 - Buses
 - Number, bandwidth
 - Latency
 - Distance
 - Speed of Light

§ Logical View of Single Book

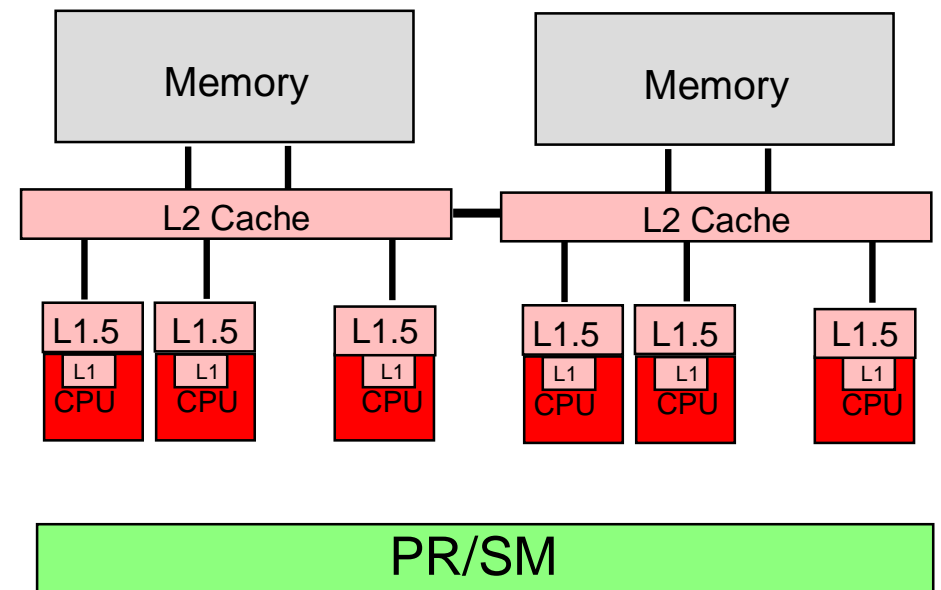


Hypervisor Overview

§ Hypervisor (PR/SM)

- Virtualization layer at OS level
- Distributes physical resources
 - Memory
 - Processors
 - Logical processors dispatched on physical processors
 - Dedicated / Shared
 - Affinities
 - Share distribution based on weights
 - Channels
 - EMIF

Logical View of 2 Book System



The challenge motivating HiperDispatch

§ Hardware cache can be optimized when a given unit of work is consistently dispatched on the same physical CPU (or related set of CPUs)

§ The physics of Non-Uniform-Memory-Access (NUMA) memory forces a paradigm change

- CPUs have Memory accesses can take less than 10 to several hundred cycles depending upon cache level / local or remote repository accessed
- Different distance-to-memory attributes
- Cache and memory latency on a hypothetical server in **Marble Arch**, London

– L1 Cache	1 machine cycle	Edgware Rd	< 1 mile
– L1.5 Cache	4 machine cycles	Cricklewood	4 miles
– Local L2 Cache	variable, 100+ cycles	Solihull	109 miles
– Remote L2 Cache	variable, 200+ cycles	Leeds	191 miles
– Real memory	~ 600 machine cycles	Inverness	553 miles



IBM Systems and Technology Group

DISPATCHING

z/OS
PR/SM



IBM Systems

DISPATCHING

§ z/OS:

- ▶ z/OS dispatches the TCB/SRB from the TRUE READY QUEUE on the logical processors (LCPs) in the LPAR.
- ▶ In LPAR MODE, the processors contained in an LPAR are LCPs.
 - If the LPAR engines are DEDICATED, there is an affinity between LCPs and PCPs.

§ PR/SM:

- ▶ PR/SM dispatches the LCPs in the dispatching queue onto the PCPs
- ▶ The choice of which LCP to dispatch depends on the actual utilization of it's %SHARE
 - An LCP with low weight which is 'behind' it's %SHARE may pre-empt an LCP with high weight which is 'ahead' of it's %SHARE
 - This is how PR/SM can deliver guaranteed %SHARE
- ▶ The dispatching happens at the LCP level, not at the LPAR level
 - All LCPs in an LPAR are not necessarily dispatched at the same time.

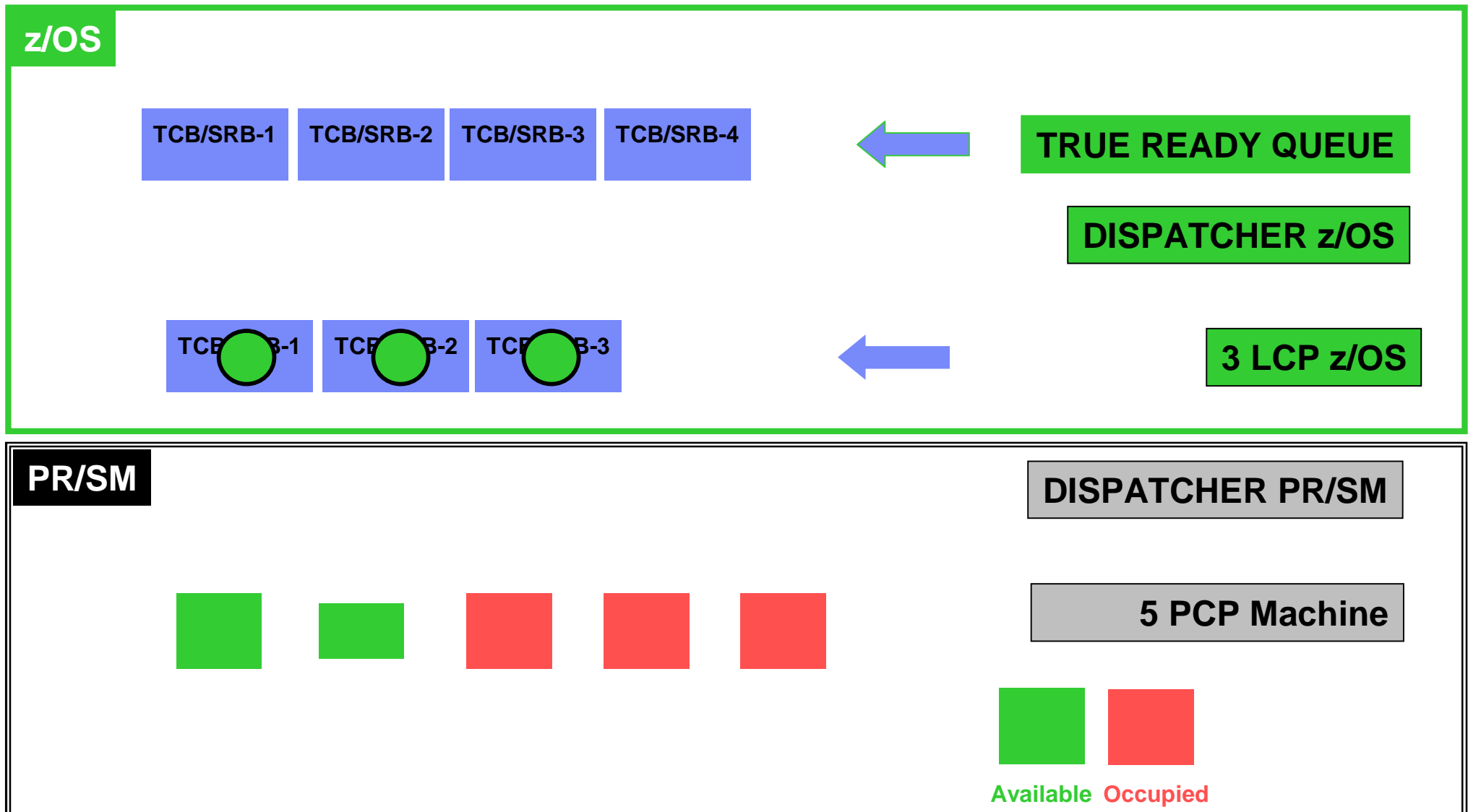
§ Distortion is possible:

- ▶ A TCB/SRB may be dispatched by z/OS on an LCP which is NOT dispatched by PR/SM
 - Distortion between LPAR BUSY% and MVS BUSY%

DISPATCHING

§ Double Dispatching

3 tasks are dispatched by z/OS on 3 LCPs
2 LCPs are dispatched on 2 PCPs by PR/SM





IBM Systems and Technology Group

HiperDispatch



Objective

IBM Systems

Design objective

§ HiperDispatch – Specific function for System z10 EC and z/OS V1R7+

▶ Interaction between z/OS and PR/SM

- **Dispatcher Affinity (DA) – new z/OS dispatcher function**
 - Limitation of the number of LCPs necessary to perform requested work
 - Based on the LPAR weight, the current demand and the available capacity
 - Knowledge of the topology of the server (multi-book).
 - Co-operation with PRSM to build LCP to PCP affinity
- **Vertical CPU Management (VCM) – new PR/SM function**
 - LCPs with 3 types of ‘polarity’ or ‘share’

▶ Optimization of the hardware caches

- **Re-utilization of the cache contents by re-dispatching the work to the same PCP where possible**



IBM Systems and Technology Group

HiperDispatch

PR/SM Vertical CPU Management (VCM)



IBM Systems

HiperDispatch – Vertical CPU Management

§ Re-distribute the logical processors to the minimum number of physical processors needed

▶ Based on #PCP guaranteed to the LPAR

- #PCP guaranteed = %SHARE x PCP
- Example – Wtotal=600, WLPAR=250, #PCP=6
 - %SHARE = $250/600 = 41.6\%$
 - #PCP Guaranteed = %SHARE x #PCP = $41.6\% \times 6 = 2.5$ PCP
 - This LPAR will have 2.5 PCPs worth of capacity guaranteed

HiperDispatch – Vertical CPU Management

§ #PCP guaranteed to the LPAR (Note)

▶ #PCP guaranteed = %SHARE x PCP

- The result of the calculation is in the form *n.m*
- n should be the number of « High-share » CP.
- m should be the number of « Medium-share » CP.
 - The next slide explains how the numbers are actually computed.

HiperDispatch – Vertical CPU Management

§ Re-distribution of engines – n.m

▶ IF $m \geq 0.5$ (i.e. $\geq 50\%$)

- Allocation of n PCPs at 100%
- Allocation of m% of 1 PCP
- Allocation of #LCP – n – 1 with ~0% (PARKED)

« HIGH SHARE »

« MEDIUM SHARE »

« LOW SHARE »

▶ If $m < 0.5$ (i.e. $< 50\%$)

- Allocation of n-1 PCPs at 100%
- Allocation of 2 PCPs at $[(1+m)/2]\%$
 - One PCP will be «stolen» from the «HIGH SHARE» pool
- Allocation of #LCP-(n-1)-2 at ~ 0% (PARKED)

« HIGH SHARE »

« MEDIUM SHARE »

« LOW SHARE »

HiperDispatch – Vertical CPU Management

§ Note on the number of « MEDIUM SHARE » LCPs

▶ Modification to the original design

- If the % of a PCP attributed to a « MEDIUM SHARE LCP » is less than 50%, a « HIGH SHARE LCP » will be «stolen» and converted to a « MEDIUM SHARE LCP ».
- So the rule of having only a single « MEDIUM SHARE LCP » is cancelled

▶ Example:

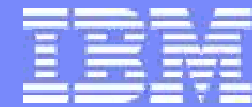
- The calculation gives 4 « HIGH SHARE » LCPs
- The calculation gives 1 « MEDIUM SHARE » at 0.4 (40%)
- HiperDispatch will do the following:
 - «Steal» 1 LCP from the « HIGH SHARE » pool
 - Calculate $(1+0.4)/2 = 0.7$ (i.e. 70%)
 - Configure 2 LCPs as « MEDIUM SHARE » at 70%
- The final configuration in HiperDispatch will then be:
 - 3 LCP as « HIGH SHARE »
 - 2 LCP as « MEDIUM SHARE » at 70%

▶ This change has been done to further improve z/OS performance

HiperDispatch – Vertical CPU Management

§ LCP PARKED ?

- ▶ **WAIT STATE** with no interrupts handled (I/O, Clock Comparator etc.)
- ▶ A parked LCP waits for an eventual call from WLM to indicate that resources not used by another partition can be utilized..



IBM Systems and Technology Group

EXAMPLE #1



Number of guaranteed PCP is $n.m$ with $m \geq 0.5$

IBM Systems

HiperDispatch – PR/SM VCM – example 1

§ Visual example (HIPERDISPATCH=NO)

- ▶ 1 server with 5 PCPs
- ▶ 1 LPAR (LPAR1) – W=700 - #LCPs=5
- ▶ 1 LPAR (LPAR2) – W=300 – #LCPs=5

Example LPAR1
 3.50 PCP guaranteed
 So each LCP will have 3.50/5 of one PCP equals 70%

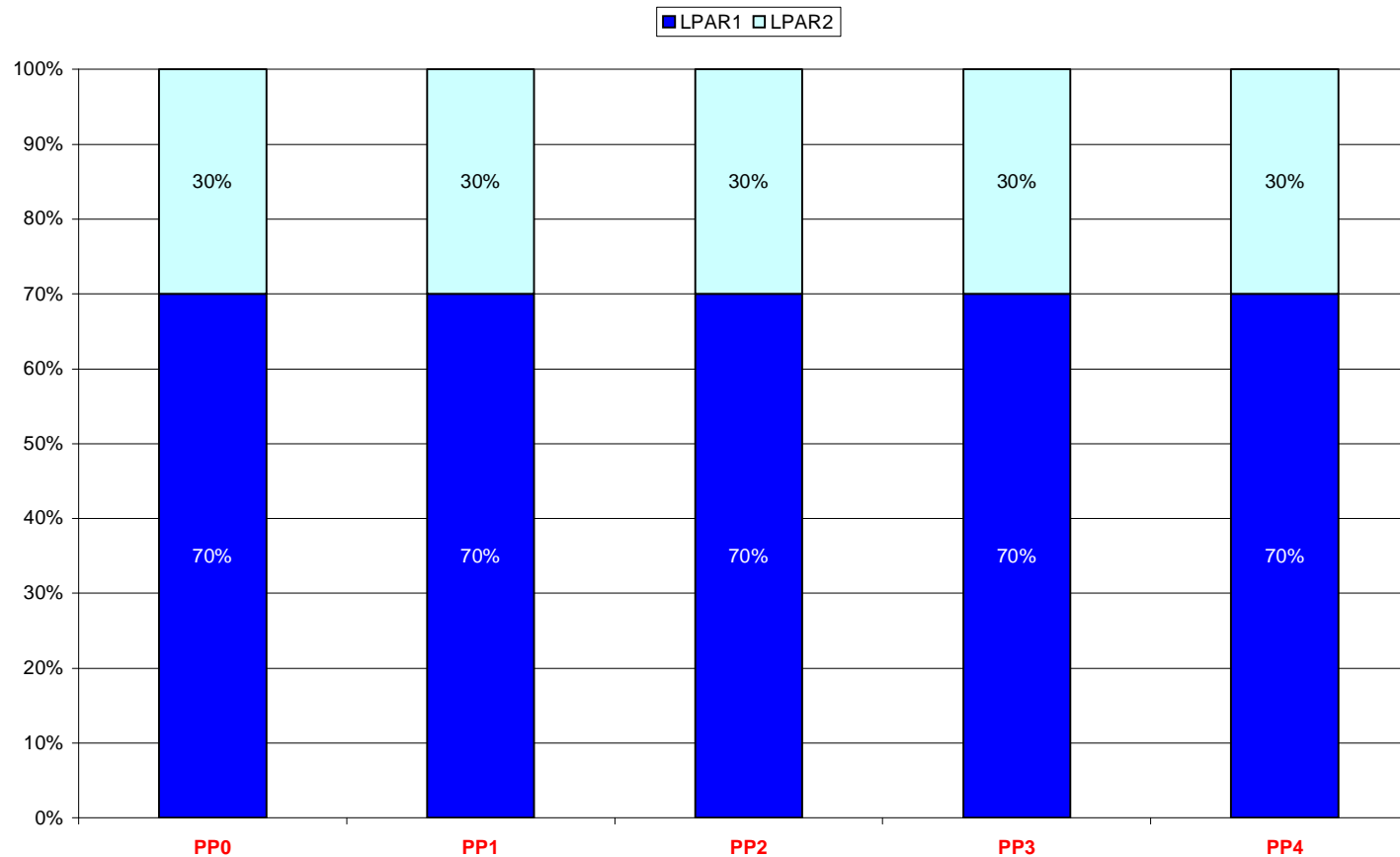
	#LP	Poids	%SHARE	#PP garantis
LPAR1	5	700	70.00%	3.5
LPAR2	5	300	30.00%	1.5
Poids-TOT		1000		
PP	5			

Example LPAR2
 1.5 PCP guaranteed
 So each LCP will have 1.5/5 of one PCP equals 30%

HiperDispatch – PR/SM VCM – example 1

§ Visual example (HIPERDISPATCH=NO)

- ▶ Graphical representation of the LCPs on the PCPs



- ▶ The dispatching is not necessarily optimal.....

HiperDispatch – PR/SM VCM – example 1

§ Visual example

- ▶ 1 server with 5 PCPs
- ▶ 1 LPAR (LPAR1) – W=700 - #LCPs=5
- ▶ 1 LPAR (LPAR2) – W=300 – #LCPs=5

	#LP	Poids	%SHARE	#PP garantis
LPAR1	5	700	70.00%	3.5
LPAR2	5	300	30.00%	1.5
Poids-TOT		1000		
PP	5			

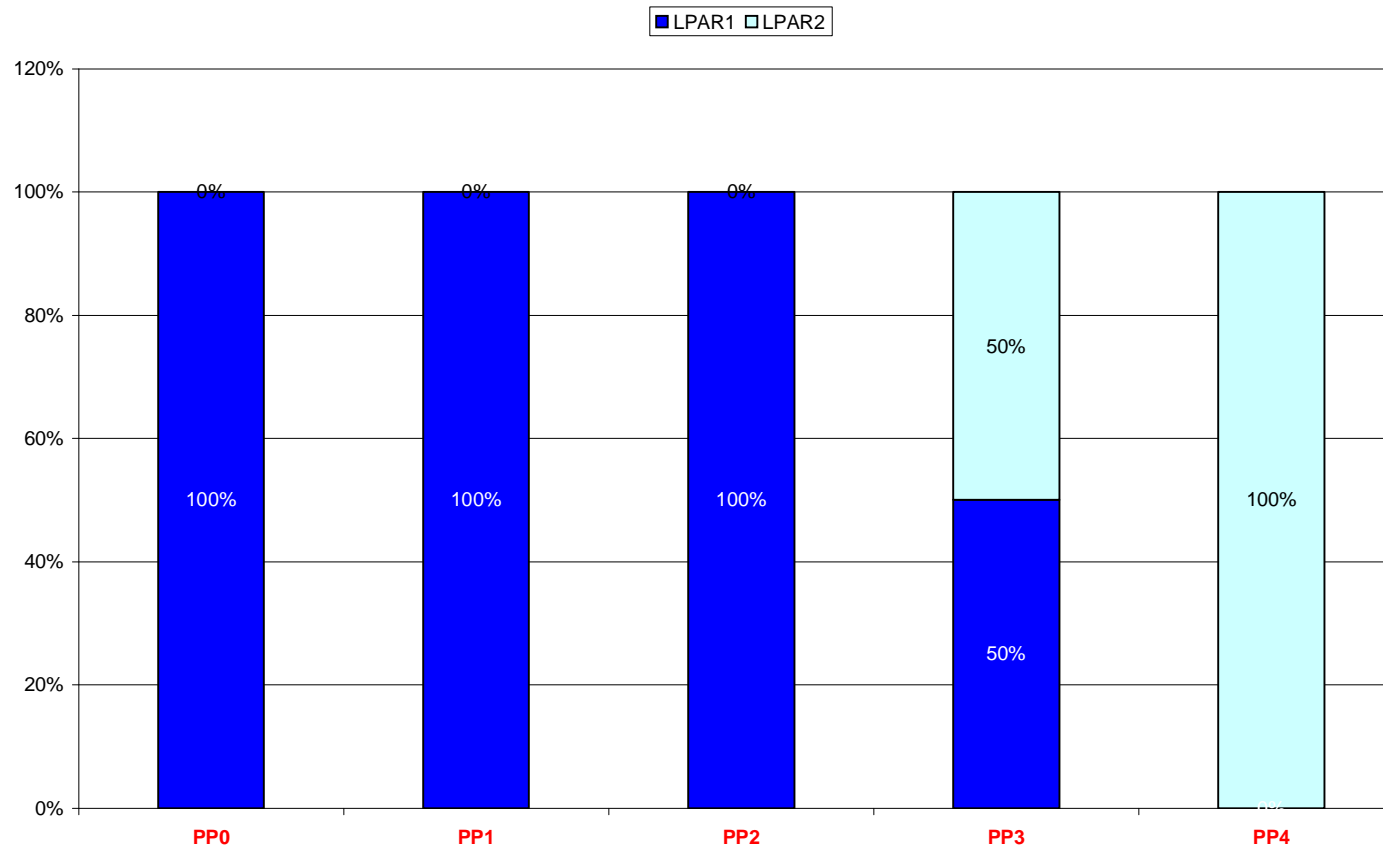
▶ HIPERDISPATCH=YES calculates as follows

- LPAR1 – The 3.50 PCP will be allocated as:
 - 3 LCP (at 100% of one PCP) « HIGH SHARE »
 - 1 LCP (at 50% of one PCP) « MEDIUM SHARE »
 - 1 LCP (at ~0% of one PCP) “LOW SHARE” - PARKED
- LPAR2 – The 1.50 PCP will be allocated as
 - 1 LCP (at 100% of one PCP)
 - 1 LCP (at 50% of one PCP)
 - 3 LCP(at ~0% of one PCP) - PARKED

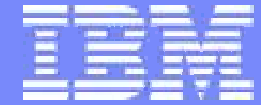
HiperDispatch – PR/SM Vertical CPU Management

§ Visual example (HIPERDISPATCH=YES)

- ▶ Graphical re-presentation of the LCPs on the PCPs



- ▶ The dispatching is optimized for best utilization of the PCPs
 - LPAR1 – One LCP is not utilized (PARKED)
 - LPAR2 – Three LCPs are not utilized (PARKED)



IBM Systems and Technology Group

EXAMPLE #2



Number of guaranteed PCP is $n.m$ with $m < 0.5$

IBM Systems

HiperDispatch – PR/SM VCM – example 2

§ Visual example (HIPERDISPATCH=YES)

- ▶ 1 machine with 5 PCP
- ▶ 1 LPAR (LPAR1) – W=750 - #LCP=5
- ▶ 1 LPAR (LPAR2) – W=400 – #LCP=5

	#LP	Poids	%SHARE	#PP garantis
LPAR1	5	750	65.22%	3.26
LPAR2	5	400	34.78%	1.74
Poids-TOT		1150		
PP	5			

Example LPAR1
3.26 PCP guaranteed

Example LPAR2
1.74 PCP guaranteed

HiperDispatch – PR/SM VCM – example 2

§ Visual example (HIPERDISPATCH=YES)

▶ Calculation for LPAR1 (n.m = 3.26)

- $m < 0.5$
- 'Steal' a « HIGH SHARE LCP » - going from 3 to 2
- Calculate $(1+0.26)/2 = 0.63$
- Distribution
 - 2 LCPs as « HIGH SHARE » at 100%
 - 2 LCPs as « MEDIUM SHARE » at 63%
 - $5-2-2= 1$ LCP as « LOW SHARE »

▶ For LPAR2 (n.m = 1.74)

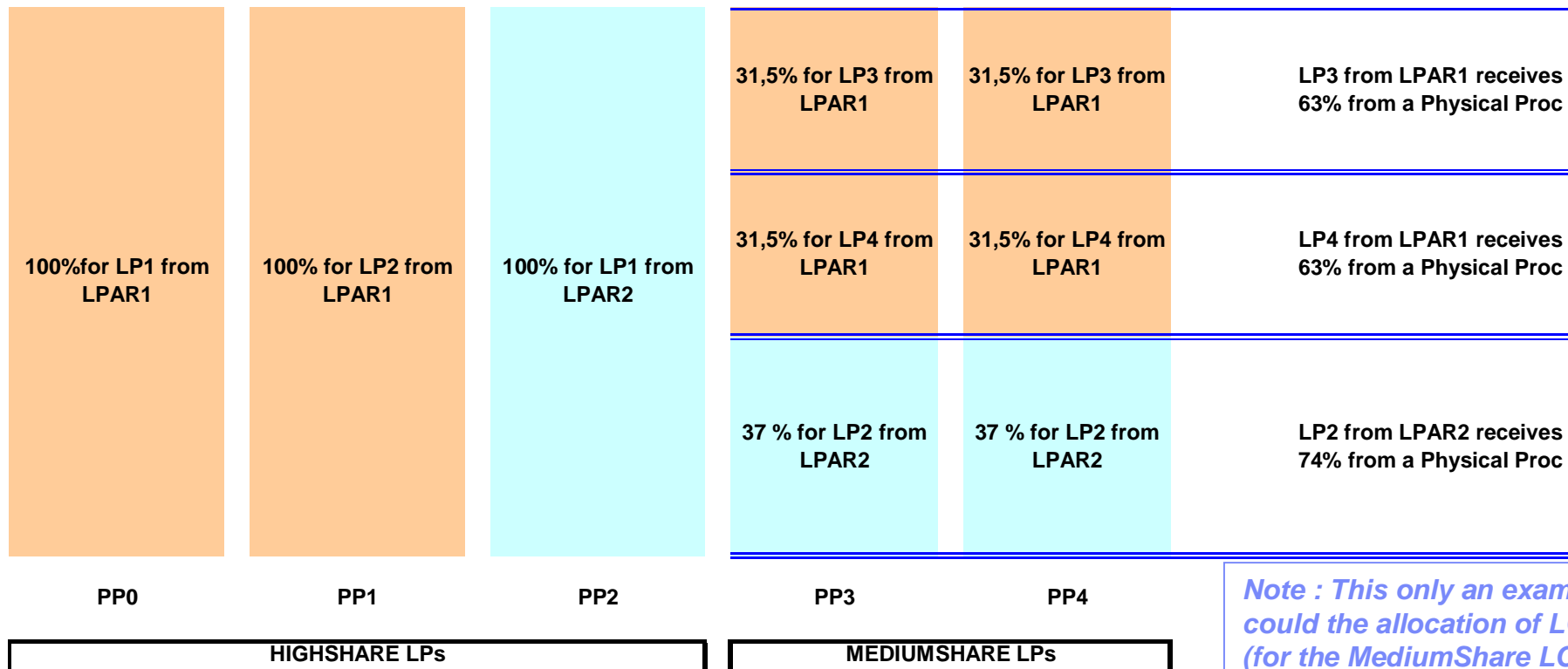
- $m \geq 0.5$
- Distribution
 - 1 LCP as « HIGH SHARE » at 100%
 - 1 LCP as « MEDIUM SHARE » at 74%
 - $5-1-1= 3$ LCP as « LOW SHARE »

	#LP	Poids	%SHARE	#PP garantis
LPAR1	5	750	65.22%	3.26
LPAR2	5	400	34.78%	1.74
Poids-TOT		1150		
PP	5			

HiperDispatch – PR/SM VCM – example 2

§ Visual example (HIPERDISPATCH=YES)

- ▶ The 74% of LCP from LPAR2 and the 63% of the 2 LCPs from LPAR1 will «float» on the two PCPs distributed as « MEDIUM SHARE » as a function of the current demand
- ▶ The %SHARE of the two LPARs are delivered in this way.
- ▶ 2 PCP = 200%, « MEDIUM SHARE » for LPAR1+LPAR2 = 63%+63%+74%=200%



Note : This only an example of what could the allocation of LCP/PCP be (for the MediumShare LCPs). Depending of the LPAR utilization the allocation may vary.

HiperDispatch – Note on MediumShare LCP and LCP dispatched from non eligible LPARs

§ The fundamental PR/SM LPAR dispatching has not changed with Hiperdispatch

- ▶ **Logical processors are dispatched in accordance to their target share**
 - The medium processor "behind" in its fair share will get priority over the medium processors "ahead" of its share at instances of time when there are more logicals with ready work to dispatch than available physical processors.
 - And while PR/SM has always tried to be smart in dispatching logical processors to the same physical processor when possible (i.e. reuse if possible):
 - much depends on the actual load of the logical processors and the pattern of the arrival of work for each.
 - A Mediumshare LCP could be dispatched at times on a PCP (affected to a Highshare LCP):
 - if the high logicals do not have the load to consume their target 100%.
 - And the low processors, if “unparked”, may be dispatched on Mediumshare PCP (if cycles are available)



IBM Systems and Technology Group

HiperDispatch

z/OS Dispatcher Affinity



IBM Systems

HiperDispatch – z/OS Dispatcher Affinity

§ New z/OS dispatcher (DA = Dispatcher Affinity)

- ▶ **New dispatcher queue(s)**

- ▶ **Dispatching ‘nodes’**
 - LCPs are attached to a dispatching ‘node’
 - 1 to 4 LCPs for each ‘node’ (design objective)
 - Considers the physical topology of the server

- ▶ **The TCBs & SRBs are distributed by priority across the ‘nodes’**

- ▶ **There is a periodical rebalancing of the task distribution**
 - **SRM : Work distribution by priority across ‘nodes’**
 - **Considers the types of LCP:**
 - High Share
 - Medium Share
 - Low Share (add / delete) – White Space



IBM Systems and Technology Group

HiperDispatch



Technical information

IBM Systems

HiperDispatch - TECHNICAL

§ Parameter IEAOPTxx HIPERDISPATCH=YES/NO

▶ HIPERDISPATCH=YES|NO IEAOPTxx

- **YES** Specifies that SRM should switch to HiperDispatch mode when the total physical processor equivalent of the defined weight of a partition is **1.5 standard processor CPUs or higher**. However, if the total physical processor equivalent of the defined weight of a partition is less than 1.5 standard processor CPUs, a console message is issued as a warning that the system is not configured as recommended.
- **zAAP and zIIP processors will not be considered when deciding the weight of a partition.**
 - **zAAP and zIIP capacity of the partition do not contribute in the determination of whether the partition is greater or less than 1.5 standard CPs.**
- **NO** Specifies that SRM should not switch to HiperDispatch mode. Default Value: NO

▶ MESSAGES

- **IRA862I** IEAOPT PARAMETER HIPERDISPATCH IS IN EFFECT, BUT THE PHYSICAL PROCESSOR SHARE IS TOO SMALL TO BE EFFICIENT
 - Explanation: The IEAOPT parameter HIPERDISPATCH =YES has turned on HiperDispatch mode . However, the system has determined that the physical processor share for this LPAR is too small to be efficient.
 - This message can also be issued if the physical **processor share drops below 1.5 regular CPUs** due to partition weight changes while operating in HiperDispatch mode. The message will only be issued once. Only general CPUs are considered when determining the physical processor share. Other CPU types, such as zAAP, are not considered

HiperDispatch - TECHNICAL

§ HiperDispatch

- ▶ **Minimizing PR/SM overhead**
 - Utilization of the VERTICAL capacity of the processors
 - Dispatching affinity of the LCPs on to the PCPs
- ▶ **Exclusively for LPARs which have a minimum of 1.5 PCPs guaranteed**
 - #PCP guaranteed = %SHARE x #PCP
 - Warning Message if the LPAR is in HIPERDISPATCH=YES and goes below 1.5 CP
- ▶ **Depending on the number of PCPs guaranteed, processors allocation falls into 3 levels:**
 - *The polarity describes the quantity of VERTICAL processors authorised for the LCP*
 - <n> LCPs with high polarity Close to 100% CP SHARE
 - 1 or 2 LCPs with medium polarity (0% < %share < 99%)
 - <m> LCPs with low polarity (0% share or close)

§ Prerequisites

- ▶ **Software : z/OS V1R7 with zIIP Web SuPCPort deliverables or higher**
- ▶ **Hardware : System z10 EC**
- ▶ **IRD processor management is automatically switched off if HIPERDISPATCH=YES**
- ▶ **zAAP and zIIP capacity of the partition do not contribute in the determination of whether the partition is greater or less than 1.5 standard CPs.**



IBM Systems and Technology Group

HiperDispatch

RMF/SMF/POO



IBM Systems

HiperDispatch - TECHNICAL - RMF

§ HiperDispatch

▶ EXAMPLE RMF report / 1

C P U A C T I V I T Y														
CPU	2097	MODEL	ABC	H/W	MODEL	XXX	SERQUENCE	CODE	00000000000D6AAD					
---CPU---		----- TIME % -----				LOG PROC		----I/O INTERRUPTS----						
TYPE	NUM	ONLINE	LPAR	BUSY	MVS	BUSY	PARKED	SHARE	%	TOTAL	RATE	%	VIA	TPI
CP	0	100.00		69.41		69.41	0.00	100.0		58.67		0.00		
	1	100.00		70.75		70.75	0.00	100.0		233.6		0.00		
	2	100.00		68.40		68.40	0.00	100.0		254.2		0.00		
	3	100.00		63.64		63.64	0.00	45.2		63.49		0.00		
	4	100.00		67.74		67.74	20.00	0.0		1380		0.01		
TOTAL/AVERAGE				67.99		67.99	0.00	345.2		1990		0.01		
AAP	8	100.00		39.41		39.41	0.00	100.0						
	9	100.00		40.75		40.75	0.00	75.0						
TOTAL/AVERAGE				40.08		40.08	0.00	175.0						

New field	Description
PARKED TIME %	§ The percentage of time where the CPU is not utilized in the interval. The CPUs are only non-utilized when they are in « low share » in Vertical Mode. <i>Parked Time</i> is not included in <i>Wait Time</i> which is used to calculate <i>MVS BUSY TIME %</i> . In horizontal mode (HIPERDISPATCH=NO), N/A is specified.
LOG PROC SHARE %	Logical processor share for the standard CPs and Special Purpose Processors. § Vertical mode (HIPERDISPATCH=YES): The logical processors have high, medium or low share § Horizontal mode (HIPERDISPATCH=NO): The LPAR weights are equally distributed across all ONLINE processors. So all processors of the same type will have equal utilizations. § DEDICATED CPU: Share=100% because one logical processors runs on only one physical processor

HiperDispatch - TECHNICAL - RMF

§ HiperDispatch

▶ EXAMPLE RMF report / 2

C P U A C T I V I T Y									
z/OS V1R8				SYSTEM ID UNKN			DATE 11/26/2007		
				RPT VERSION V1R8 RMF			TIME 22.33.43		
CPU 2097				E40 SEQUENCE CODE 0000000000DC6CE					
---	---	---	---	---	---	---	---	---	---
NUM	TYPE	ONLINE	LPAR BUSY	MVS BUSY	PARKED	LOG PROC SHARE %	RATE	INTERRUPTS- % VIA TP	
0	CP	100.00	96.33	97.34	0.00	100.0	5.80	48.75	
1	CP	100.00	95.96	97.07	0.00	100.0	4.59	55.30	
2	CP	100.00	95.79	96.84	0.00	100.0	5.10	55.18	
3	CP	100.00	95.46	96.68	0.00	100.0	2.40	53.75	
4	CP	100.00	95.08	96.41	0.00	100.0	8435	10.05	
5	CP	100.00	73.92	96.86	0.00	70.0	20.74	4.95	
6	CP	100.00	74.33	97.13	0.00	70.0	14.15	19.39	
7	CP	100.00	13.84	14.06	85.78	0.0	0.00	0.00	
TOTAL/AVERAGE			80.09	86.55		640.0	8488	10.14	

- ▶ A value > 0 in « PARKED » or non-uniform values in « LOG PROC SHARE » indicates that HiperDispatch is being used.
- ▶ In this example, the 640% of Logical Share are distributed in:
 - 5 LCP at 100% (High Share)
 - 2 LCP at 70% (Medium Share) –
 - 1 LCP at 0% (Low Share) which was « PARKED » in 85.78% of the interval

HiperDispatch - TECHNICAL - SMF

§ HiperDispatch

▶ SMF record 70 – new fields

SMF Record Type 70 Subtype 1 (CPU Activity)				
Offsets	Name	Len	Format	Description
CPU Data Section				
...				
32 x20	SMF70PAT	8	Binary	CPU parked time
Partition Data Section				
...				
56 x38	SMF70PFL	2	Binary	Additional partition flags. Bit Meaning When Set 0 Content of SMF70UPI is valid. 1 Group flag. This partition is member of a capacity group. 2 Polarization flag. This partition is vertically polarized and the polar weight fields in the logical processor data section are valid for CPUs of this partition. 3-15 Reserved
Logical Processor Data Section				
...				
64 x40	SMF70POW	4	Binary	Polarization weight. Current weight for logical CPU when polarization weighting applies.
68 x44		12		Reserved

▶ New fields in red

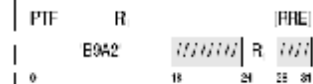
HiperDispatch - TECHNICAL - POO

§ HiperDispatch

▶ Hardware instruction PERFORM TOPOLOGY FUNCTION (PTF)

- X'B9A2' –

PERFORM TOPOLOGY FUNCTION



The contents of general register R₁ specify a function code in bit positions 56-63, as illustrated in Figure 10-47.

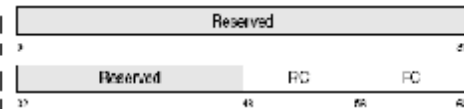
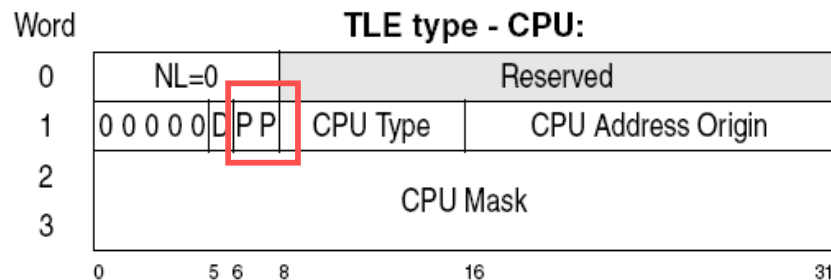


Figure 10-47. General-Register R₁ Format

FC = Function Code

- 0 Request horizontal polarization.
- 1 Request vertical polarization.
- 2 Check topology-change status.

▶ TOPOLOGY LIST ENTRY (TLE)



PP Meaning

- 0 The one or more CPUs represented by the TLE are horizontally polarized.
- 1 The one or more CPUs represented by the TLE are vertically polarized. Entitlement is low.
- 2 The one or more CPUs represented by the TLE are vertically polarized. Entitlement is medium.
- 3 The one or more CPUs represented by the TLE are vertically polarized. Entitlement is high.

Notes on: Dedicated processors

§ LPAR with dedicated processors:

- ▶ **Half of the work is already done:**
 - **The HighShare CP allocation is set.**
- ▶ **HiperDispatch is efficient too in this case:**
 - **z/OS part - redispach tasks on the same PCP.**

Notes on: zAAP and zIIP

§ zAAP and zIIP:

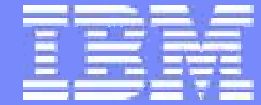
- ▶ **zAAP and zIIP do not contribute in the determination of whether the partition is greater or less than 1.5 standard CPs.**

- ▶ **If the configuration contains zAAPs, zIIPs and HD=YES:**
 - **zAAP and zIIP are managed vertically.**
 - **Example, for the z/OS part, if there are more than one zAAP, z/OS will try to dispatch the JVM TCBs/SRBs on the zAAP engine used in the previous dispatch.**

Notes on: Defined Capacity (VWLC)

§ SOFT CAPPING does apply to HD Mode:

- ▶ If a defined capacity weight is supplied and capping is turned on, the vertical configuration is recalculated and if it results in a change to the entitlement numbers (numbers of online highs, mediums, or lows), z/OS is notified of the change and that configuration is capped.
- ▶ When capping is turned off (removing a defined capacity weight), this is undone and similarly notified.



IBM Systems and Technology Group

HiperDispatch

Planning



IBM Systems

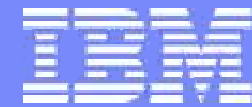
HiperDispatch - Planning

§ PR/SM

- ▶ **Make sure that the configuration has:**
 - **A number of LPARs with guaranteed #PCP ≥ 1.5**
 - Calculate for each LPAR, %SHARE x #PCP

§ z/OS

- ▶ **Verify workloads attributed to SYSSTC.**
 - Ex: VTAM, IRLM, TCP/IP (a lot of dispatching of short activities)
- ▶ **Verify %VELOCITY for configurations with ONE « High Share CP ».**
 - **At constant %SHARE, the PI for Service_Class with %VELOCITY will descend if the number of processors descend**
 - Refer to RedBook WLM SG24-6472 Chapter 6, « Impact of the number of engines on velocity »



IBM Systems and Technology Group

HiperDispatch



Conclusion

IBM Systems

CONCLUSION

§ HiperDispatch

▶ Objective

- Allow better utilization of the Logical Processor. In our example, LPAR1 in HIPERDISPATCH=NO, will dispatch 3.5 processor's worth on 5 CPs, using the CPs at only 70%
- With HIPERDISPATCH=YES, LPAR1 will use 3 LCPs at 100% and 1 LCP at 50%, the 5th LCP is not allocated. So the LPAR will run with 4 processors giving an average utilization of 87.5%
- With the reduction of active LCPs the server will see less PR/SM overhead and a better MP factor

CONCLUSION

§ Hiperdispatch – Considerations – optimum efficiency

▶ When is it efficient ?

- **Definitely in a configuration with a high number of « High Share CPs » and with LPARs having sufficient weights to create « High Share CPs »**
 - On configurations with many CPs and large LPARs
- **With a number of « High Share CP » of 3 or more, the benefit is evident and very few changes are needed to the WLM policy**
- **For workloads with a stable Working Set and a high number of dispatches**
 - Type CICS
- **In multi book configurations**
 - z/OS understands the topology of the server

CONCLUSION

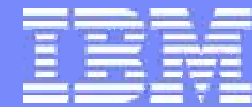
§ Hiperdispatch – Considerations – less efficient

▶ « Small » LPARs

- When the result is 1 « High-Share CP », we are in a ‘single-engine’ environment with all its inconveniences, so we must have a very well adapted Service Policy:
 - Short, important tasks very high up on the DP list
 - CPU intensive at the bottom of the list
 - ♦ Mean Time To Wait algorithm
 - So – it will be necessary to adjust goals for VELOCITY workloads

▶ CPU intensive workloads

- Long BATCH workloads



IBM Systems and Technology Group

HiperDispatch



End of Presentation

IBM Systems