z/OS® Workload Manager

Capping Technologies

INSIDE WLM

Issue No. 3

Günter Vater, vater@de.ibm.com
Robert Vaupel, vaupel@de.ibm.com
IBM Development Laboratory
Böblingen, Germany

September 8th, 2020

Contents

1. INTRODUCTION 6

2. RESOURCE GROUPS 8

2.1. GENERAL CONCEPTS 8
2.2. REQUIREMENTS 8
2.3. WHY USE RESOURCE GROUPS 9
2.4. EXAMPLE 9
2.5. THE DEFINITION OF RG LIMITS 10
2.6. CONTROLLING RESOURCE CONSUMPTION OF A RG 11
2.6.1. The maximum limit 11
2.6.2. The minimum limit 11
2.7. LIMITATIONS OF A RG 12
2.8. COMMON SCENARIOS FOR RGS 12

3. DEFINED CAPACITY LIMIT 13

3.1. GENERAL CONCEPTS 13
3.2. REQUIREMENTS 13
3.3. WHEN TO USE DEFINED CAPACITY 14
3.4. HOW DOES IT WORK 14
3.4.1. What are MSUs 14
3.4.2. What is the four hour rolling average 14
3.4.3. When the average consumption reaches the DC limit 14
3.4.4. Why can the 4HRAVG exceed the DC limit 16
3.4.5. Mechanisms to cap LPARs 17
3.4.6. Capping by Cap Pattern 17
3.4.7. Capping by Phantom Weight 19
3.5. PRICING ASPECT 20
3.6. OTHER CONSIDERATIONS 20
3.7. UNDESIRABLE CAPPING EFFECTS 20
3.7.1. LPARs with a small weight on a large CEC using cap pattern 20
3.8. HMC INTERFACE 22
3.8.1. Define a DC limit 22
3.8.2. Change a DC limit 22
3.8.3. Remove a DC limit 22
3.9. MINITORING DEFINED CAPACITY 22
3.9.1. RMF Partition Data Report 22
3.9.2. RMF Monitor III CPC Capacity Report 23
3.9.3. Useful SMF70 Fields 24
3.9.4. What about SMF72 Fields 25

4. GROUP CAPACITY 26

4.1. GENERAL CONCEPTS 26
4.2. REQUIREMENTS 27
4.3. ADVANTAGES OF A CAPACITY GROUP 27
4.4. HOW DOES WLM COMBINE LPARS TO A CAPACITY GROUP 28
4.4.1. Data Colletion 28
4.4.2. Maintaining the unused capacity 28
4.4.3. Capping of a Capacity Group 30
4.5. PRICING ASPECT 31

4.6. OTHER CONSIDERATIONS 32
4.7. SCENARIOS 32
4.7.1. *Increasing a Group Limit* *32*
4.7.2. *Decreasing a Group Limit* *32*
4.7.3. *IPL of a capacity group* *33*
4.7.4. *A group member leaves a capacity group* *34*
4.7.5. *A group member moves to another capacity group* *34*
4.7.6. *The limit of a capacity group gets removed (limit=0)* *34*
4.7.7. *An additional group member gets IPLed into a existing group* *35*
4.7.8. *Group Capacity adapting to workload changes* *36*
4.8. HMC INTERFACE 37
4.8.1. *Define a group* *37*
4.8.2. *Connect LPARs to a group* *37*
4.8.3. *Change a group limit* *37*
4.8.4. *Remove an LPAR from a existing group* *37*
4.8.5. *Add an LPAR to a existing group* *37*
4.9. INTERACTIONS WITH OTHER FUNCTIONS 37
4.10. INTERFACES THAT PROVIDE INFORMATION ABOUT GC 38
4.10.1. *REQLPDAT* *38*
4.10.2. *SMF99* *39*
4.11. HOW CAN GC BE MONITORED 39
4.11.1. *RMF Group Capacity Report* *39*
4.11.2. *Difference between ACT% and WLM%* *40*
4.11.3. *RMF Partition Data Report* *41*
4.11.4. *Remaining Time until Capping* *42*
4.11.5. *Useful SMF Type 70 fields* *43*
4.11.6. *Diagnoses* *43*
4.12. COMPATIBILITY BETWEEN LPAR CONTROLS 44
A *46*
B *46*
C *46*
D *46*
E *47*
F *47*
G *47*
H *47*
I *47*
J *48*
L *48*
M *48*
O *49*
P *49*
Q *49*
R *49*
S *49*
T *50*
U *50*
V *50*
W *50*
X *51*
Z *51*

List of Figures

1.1. Scope of WLM Capping Technologies 6

2.1. Resource Groups together with Other WLM Technologies 9
2.2. Resource Group Example 9
2.3. Service Class and Resource Group Definitions 10

3.1. Defined Capacity together with Other WLM Technologies 13
3.2. Example of the 4hr Rolling Average 15
3.3. Defined Capacity with constant high demand 15
3.4. Defined Capacity with declining demand 16
3.5. Why does the 4HRAVG exceed the DC limit 16
3.6. WLM Capping Methods 18
3.7. WLM Cap Pattern at 50% 18
3.8. WLM Phantom Weight Capping 20
3.9. WLM Cap Patterns 21
3.10. Cap Pattern for a LPAR with a small weight 21
3.11. RMF Partition Data Report 22
3.12. RMF Monitor III CPC Capacity Report 23

4.1. Scope of WLM Group Capping 26
4.2. Group Capacity together with Other WLM Technologies 27
4.3. Group Capacity Example with three partitions 28
4.4. Capacity Group and Unused Vector 29
4.5. Unused Vector and 4 Hour Rolling Average 29
4.6. Group Capacity IPL Bonus 33
4.7. LPAR joining a capacity group 34
4.8. Group Capacity Example with changing demands 35
4.9. Information provided by REQLPDAT 37
4.10. RMF Group Capacity Report 38
4.11. Difference between ACT% and WLM% 40
4.12. RMF Partition Data Report 40
4.13. RMF CPC Report on RMF Data Portal 41
4.14. Compatibility between LPAR controls 43

List of Tables

3.1. SMF Type 70 Fields for Defined Capacity 24

4.1. Group Capacity Example 31
4.2. SMF Type 70 Fields for Group Capacity 42

1.Introduction

This document provides information about WLM soft capping mechanisms to control the CPU consumption of workload running on z/OS. The intention of this chapter is to show the differences between them and to make you able to choose the one that fits to your needs.

WLM provides three methods with a different scope:

- WLM Resource Groups (RG)
  RG allow to control the minimum and the maximum CPU capacity provided to the service classes linked to a RG. The service classes which belong to the same RG have to reside on partitions in the same sysplex but the partitions don't have to be on the same CEC.
- WLM Defined Capacity (DC)
  DC is able to limit the CPU consumption of a single LPAR. The operating system and all workload running in it can be limited against an upper MSU limit.
- WLM Group Capacity (GC)
  GC is very similar to DC with the difference that a GC limit applies to a group of LPARs which have to reside on the same CEC but they don't have to be in the same Sysplex.

Figure 1.1 gives an overview of the different scopes of WLM capping technologies.

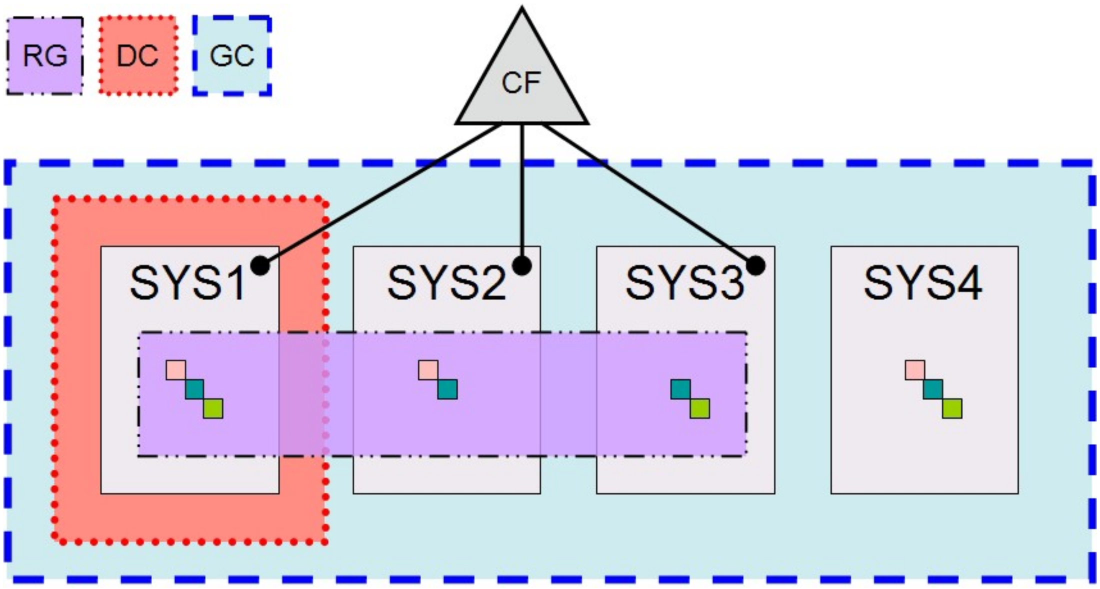


Figure 1.1.: Scope of WLM Capping Technologies

While a Resource Group (RG) is able to control the CPU consumption of a subset of the workload in a sysplex, a Defined Capacity limit (DC) limits the CPU consumption of an LPAR and a Group Capacity limit (GC) limits the CPU consumption of a group of LPARs which don't need to be in the same sysplex.

Be aware that the capping mechanisms mentioned above are for regular processors, only. The CPU capacity which is consumed on special processors like zAAP and zIIP is not evaluated by WLM when monitoring the workloads to ensure the limits.

If you need specific information which is not provided by this paper, you can find additional information in [11].

2. Resource Groups

WLM manages workload based on the goals defined in the WLM service definition. WLM tries to help workload which is not able to reach its goal by assigning more resources to it. If necessary, resources are taken away from workload that fulfills its goal to help other workload which misses its goal. If not all workload in an LPAR is able to reach its goal, WLM tries to help the more important workload first.

A resource group (RG) is a mechanism to set the bounds of the CPU goal management. A RG minimum can prevent WLM from taking away resources from workload which reaches its goal and a RG maximum reduces the CPU consumption of workload even when it misses its goal.

Note:
By using RGs, you may take away resources from the WLM goal management. Therefore try to adjust the service class goals first and use RG with caution.

2.1. General Concepts

WLM allows classifying the workload into Service Classes (SC) and giving them a goal and an importance. Furthermore a SC can be linked to a Resource Group (RG) which controls the CPU consumption of all service classes which are linked to it. A SC can only belong to one RG while one RG can be linked to more than one SCs but it is recommended to link only one SC to each RG. Up to 32 RGs can be defined in a service definition.

The sum of CPU consumption of all service classes belonging to a RG is controlled by the minimum and the maximum limit of the corresponding RG. Workload within a RG which is meeting its goal does not get additional CPU capacity even when the minimum of the RG is not reached.

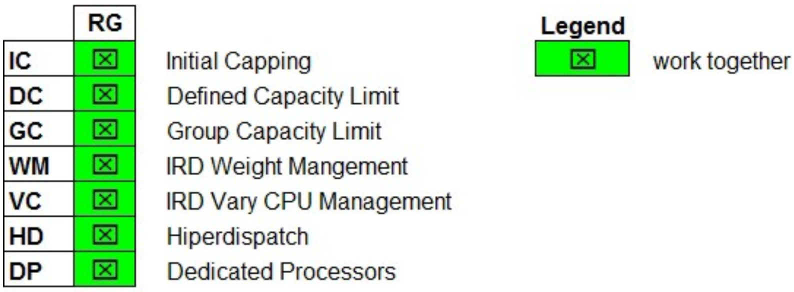
The scope of a RG is sysplex wide. All service classes belonging to the same RG are accounted to one entity, independent from the sysplex member where the workload is running. WLM evaluates RG goal achievement of the sysplex as well as the RG goal achievement of the local system. The RG minimum and maximum limits can be specified in sysplex wide measurements (RG type 1) or LPAR wide measurements (RG type 2 and 3).

WLM measures the consumption of a RG in service units per second which is averaged over a 60 second period. Since capping delay is one of the delay states recorded by WLM during sampling, you can monitor RGs together with other delay states in RMF reports.

2.2. Requirements

Figure 2.1 shows that RGs work independent from other WLM resource controlling mechanisms and therefore RGs can be used together with them.

Note:
Resource Groups work independent from the other listed functions, but they influence each other. For example a RG limit that is derived from the LPAR capacity, decreases when the LPAR capacity gets capped by DC or GC.



	RG	
IC	☒	Initial Capping
DC	☒	Defined Capacity Limit
GC	☒	Group Capacity Limit
WM	☒	IRD Weight Mangement
VC	☒	IRD Vary CPU Management
HD	☒	Hiperdispatch
DP	☒	Dedicated Processors

Legend

☒	work together

Figure 2.1.: Resource Groups together with Other WLM Technologies

2.3. Why use Resource Groups

In most configurations there should be no need for RGs. WLM manages each workload against its goal and distributes the available capacity in a way that the most important workload gets preferred access to it. As long as you are happy with WLM goal management, you should not add RGs because they might impact the WLM goal management.

For example if you would assign a lot of work to RGs with a minimum limit, WLM tries to help this workload to reach its goal even when more important workload misses its goal. An exception of this rule is discretionary workload because WLM will help higher important workload to reach its goal before it helps discretionary workload to reach its RG minimum.

If you want to help discretionary workload to get a minimum capacity or if you have the need to limit workload to a defined maximum limit, RGs can help you to implement this.

2.4. Example

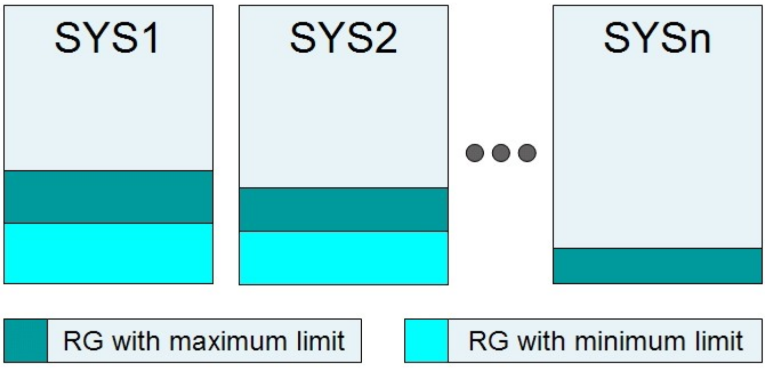
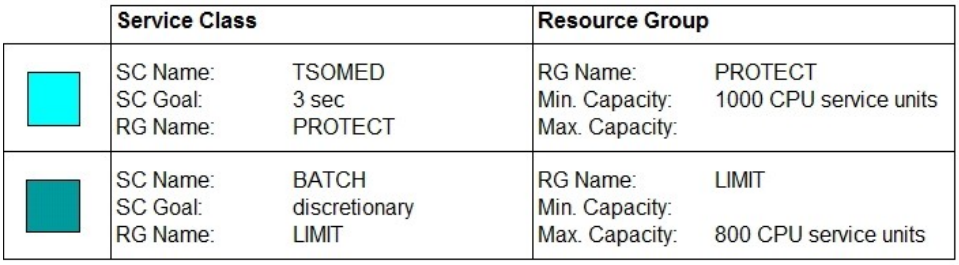


Figure 2.2.: Resource Group Example

Figure2.2 shows an example of a RG scenario. The LPARs are in the same sysplex and they have a RG with a minimum and a RG with a maximum defined. Each system of the sysplex can have workload in a service class that belongs to the RG, but it is also possible that a system has no workload which is connected to the RG.



	Service Class		Resource Group	
🟦	SC Name:	TSOMED	RG Name:	PROTECT
	SC Goal:	3 sec	Min. Capacity:	1000 CPU service units
	RG Name:	PROTECT	Max. Capacity:	
🟦	SC Name:	BATCH	RG Name:	LIMIT
	SC Goal:	discretionary	Min. Capacity:	
	RG Name:	LIMIT	Max. Capacity:	800 CPU service units

Figure 2.3.: Service Class and Resource Group Definitions

The definition of the corresponding Service Classes and Report Classes is shown in figure 2.3. Service classes are connected to resource groups by specifying a resource group name in the SC definition. An empty limit field in the RG definition means unlimited.

The service class TSOMED is linked to a RG with a minimum limit named PROTECT. If the goal of TSOMED is achieved, the RG will not assign additional resources to the service class (SC) even when the minimum limit is not reached. But if the SC misses its goal, WLM will try to reach the minimum limit of the RG by assigning more CPU resources to the SC.

To give the service class more resources, it would be better to give it a more aggressive goal, but if there is a need for a specified minimum, it can be done this way. In the example above the SC TSOMED would get at least 1000 CPU service units per second if it misses its goal and has the demand for this amount of capacity.

The service class BATCH is linked to a RG with a maximum limit named LIMIT. In most cases this limitation should not be necessary because discretionary workload gets only the capacity which is not needed by higher important workload. But if you think that this limitation is not strong enough or if there is another reason to limit this workload, this limit will allow to reduce the CPU capacity that is consumed by the entire RG to 800 service units per second.

2.5. The definition of RG limits

Resource groups are defined in the WLM service definition. There are three types of RGs. You can choose the one that is most convenient to you but be aware that the limits have a different scope.

Type1 in **Service Units** (Sysplex Scope)

A service unit limit has a sysplex scope. This means that the sum of the CPU service units of all service classes on all LPARs which belong to the RG are limited to the specified minimum and/or maximum limit. For example if you specify a maximum limit of 1000 service units per second, the combined RG consumption of all sysplex members are limited to consume 1000 SUs per second.

Type2 as **Percentage of the LPAR** share (System Scope)

Because this limit has system scope, the RG processing makes sure that each system gets the percentage of its LPAR share defined by the limits. Because each LPAR of the sysplex has its own size, each LPAR has its own limits, too. For example if you specify a maximum limit of 10% of the LPAR, on each sysplex member this RG group can consume 10% of the processor capacity of that LPAR.

Type3 as a **Number of CPs** times 100 (System Scope)

This limit has a system scope, also. Each system is measured against the number of processors defined by the limit. For example if you specify a maximum limit of 200 (2 processors), the RG can consume the capacity of two processors on each LPAR.

Note:
If you change the available CPU capacity, the RG types react in a different way.

- The limit of a Type1 RG is independent from the environmental conditions since service unitsare a hardware independent measurement.
- A Type2 RG limit changes with the size of the LPAR. If the capacity of the LPAR changes, the limit adapts to the new CPU capacity. Examples for capacity changes are: LPAR moves to another machine, weight changes, softcapping, etc.
- The limit of a Type3 RG adapts when you use faster (or slower) processors of another hardware model.

You may consider this when you choose a RG type and you may also reevaluate the limits when you change your available CPU capacity.

2.6. Controlling Resource Consumption of a RG

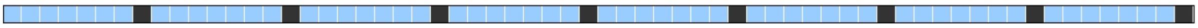
A RG can have a minimum or a maximum CPU limit or both. WLM uses totally different mechanisms to control these limits. The following two sections explain the different technics.

2.6.1. The maximum limit

When WLM recognizes that the workload of a RG exceeds its maximum limit, it will reduce the CPU access by capping the amount of service that can be consumed by the RG.

To do the capping, WLM calculates a pattern of intervals where the workload is set to non-dispatchable in some of the intervals. All service classes which belong to a RG are awake at the same time and they are also non-dispatchable at the same time. The RG cap pattern defines the time slices during them the workload is dispatchable or non-dispatchable.

The following example shows a pattern with 64 intervals[1]. During the awake slices (blue) the workload runs freely while it is non-dispatchable (dark grey) in some other slices.



The pattern is calculated in a way that the average of the consumed capacity over the pattern of intervals should match the maximum limit. Every ten seconds, the cap pattern will be recalculated and adapted to the new CPU consumption. Each time slice has the length of a half SRM second (hardware dependent), which is less than a millisecond on today's machines. Because the time slices are very short, the changes in dispatchability do not appear as abrupt service changes.

2.6.2. The minimum limit

The minimum limit does only help workload which doesn't reach its goal. A service class which reaches its goal does not get additional resources to fulfill the minimum limit of the RG. Discretionary workload (which has no goal) is an exception of this rule. WLM helps discretionary workload to reach the minimum when the goal achievement of higher important workload is not impacted by doing so.

WLM takes the RG goal achievement of the local system as well as the RG goal achievement of the sysplex into account. When WLM recognizes that the goal missing workload of a RG does not reach its

[1] The number of capping intervals has been extended to 256 with z/OS 2.1

minimum but it has additional demand for CPU, WLM will raise its dispatch priority to allow it to get dispatched more often.

WLM tries to help the workload in the following order:

1. Missing Goals & Below RG Minimum (from high to low importance)
2. Missing Goal (from high to low importance)
3. Discretionary & Below RG Minimum

This can lead to the fact that a lower importance workload can meet its goals even when a higher importance workload doesn't meet its goal.

2.7. Limitations of a RG

Be aware that when you use RGs you may impact other workloads. For example excessive use of RG minimum can take away resources from higher important workload. The workload which misses its goal and misses the minimum limit of its RG gets the most attention of WLM. A workload with a high RG minimum and an aggressive goal can dominate the system. Even higher important workload will suffer from the effect of this kind of contrary objectives.

The maximum limit of the RG can only be reached if no other limiting factor will prevent this. For example if there are not enough logical processors available, the maximum can not be reached. Also other limits like a Defined Capacity limit or a Group Capacity limit may disallow to get to the maximum of the RG.

2.8. Common Scenarios for RGs

When there is a good reason to limit specific workload in a system or sysplex to a defined limit, a maximum RG limit allows you to do so.

A minimum limit for discretionary workload may also be appropriate because it will not lead to the fact that higher important workload will miss its goal.

Even when you should use them rarely, a minimum limit for non-discretionary workload does allow the workload to keep a minimum capacity when missing the goal.

3. Defined Capacity Limit

Most systems have more than one LPAR running and the physical CPU capacity of the CEC is distributed across the partitions based on the weight defined for each LPAR on HMC. A partition gets all of its share and even beyond that when other LPARs don’t use their share.

Defined Capacity (DC) provides a way to limit the CPU usage of an LPAR by capping it to a user defined limit which is defined on HMC. In principle the DC limit can be specified from 1 MSU to the total capacity of the CEC while low single digit values are not recommended.

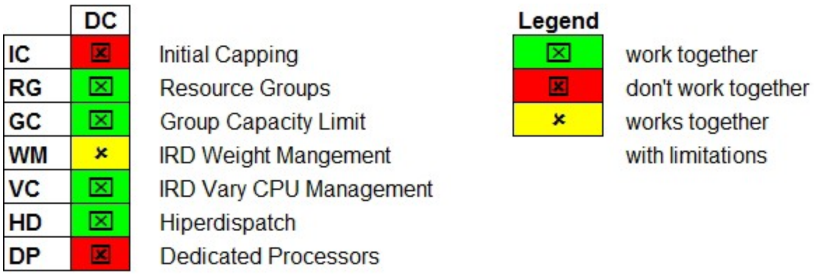
3.1. General Concepts

PR/SM is distributing the available physical capacity of a CEC by comparing the weight of all defined LPARs. If a partition is not using its share, this capacity can be consumed by other LPARs which demand more than their share would allow.

But PR/SM has also a mechanism to prevent an LPAR from exceeding its share which is called Initial Capping (IC) also known as Hard Capping and can be specified on HMC. A hard capped partition is allowed to consume the CPU capacity which is represented by the LPAR weight only. Even when a hard capped partition has the demand for more capacity and there is available capacity on the CEC, the IC capped partition can not go over its capacity represented by the partitions weight.

Similar to initial capping, WLM can tell PR/SM to limit the consumption of an LPAR to a limit which can also be defined on HMC and is called Defined Capacity (DC). But while a IC capped partition is not able to go over its weight capacity at any time, the DC limit is compared to the four hour rolling average of the partition. Therefore a DC capped partition can consume at one point in time more and at another point in time less than the DC limit as long as the 4hr rolling avg. does not exceed the DC limit.

3.2. Requirements



	DC	
IC	🟥 ☒	Initial Capping
RG	🟩 ☒	Resource Groups
GC	🟩 ☒	Group Capacity Limit
WM	🟨 ×	IRD Weight Mangement
VC	🟩 ☒	IRD Vary CPU Management
HD	🟩 ☒	Hiperdispatch
DP	🟥 ☒	Dedicated Processors

Legend	
🟩 ☒	work together
🟥 ☒	don't work together
🟨 ×	works together with limitations

Figure 3.1.: Defined Capacity together with Other WLM Technologies

Figure 3.1 shows that DC is not independent from all of the other WLM resource controlling mechanisms and therefore it can not be used together with all of them. IRD Weight Management only works as long as the defined capacity limit is not met and the partition is not capped. As soon as the partition is capped weight changes are not possible anymore.

Note:
When a partition uses dedicated processors or initial capping, WLM will not activate Defined Capacity on this partition. The DC limit will not be ensured by WLM in any way. If DC is desired, make sure to disable initial capping and dedicated processors for this partition on HMC.

3.3. When to use Defined Capacity

If a sub-capacity eligible WLC product runs on an LPAR, the maximum charge for this product can be limited by using DC. For details about pricing see LINK.

It is also possible to limit an unimportant LPAR which is used e.g. for testing or developing to a defined capacity to make sure that it will not take too much capacity from a production LPAR.

3.4. How does it work

To understand how WLM limits the consumption of an LPAR you need to know about two important metrics. One of them is MSU and the other one is the four hour rolling average.

3.4.1. What are MSUs

A service unit (SU) is a certain amount of service which can be consumed by the workload. MSU is a measurement for processor speed and means millions of service units per hour. Since faster machines can provide more service units per hour, they have a higher MSU rate.

WLM makes a difference between software MSUs and hardware MSUs. The DC algorithms measure the CPU consumption in software MSUs which are also called *software pricing capacity units*. Hardware MSUs or better service units are a measure of real hardware speed. Since System z9 hardware MSU and software MSU are no longer the same. Software MSU have a lower value named the technology dividend. Since System zEC12 the technology dividend doesn't change anymore and price adjustments are now included in the newest software pricing metrics. Additional information to software pricing can be found in [16].

3.4.2. What is the four hour rolling average

The DC limit is based on the average CPU service of an LPAR which has been consumed in the last four hours. WLM keeps track of this consumption and if it exceeds the amount that is allowed by the DC limit, WLM tells PR/SM to activate the capping feature.

Figure **??** shows an example of the 4hr rolling average (4HRAVG). As long as the 4HRAVG of the consumed CPU service units of the LPAR is below the defined capacity limit, there is no need for capping. The actual consumption can be sometimes over and sometimes below the limit. The WLM capping algorithms do not impact the workload in any way as long as the 4hr rolling avg. is below the defined DC limit.

3.4.3. When the average consumption reaches the DC limit

As soon as the 4hr rolling avg. exceeds the amount which is allowed by the DC limit, WLM needs to make sure, that the CPU consumption of the LPAR gets reduced. WLM monitors the LPAR consumption and detects the capping necessity. The capping itself is triggered by WLM but implemented by PR/SM.

When the average consumption of an LPAR exceeds the DC limit, it gets capped to its DC limit until the average consumption drops down below the limit. If the workload does not reduce its demand, the

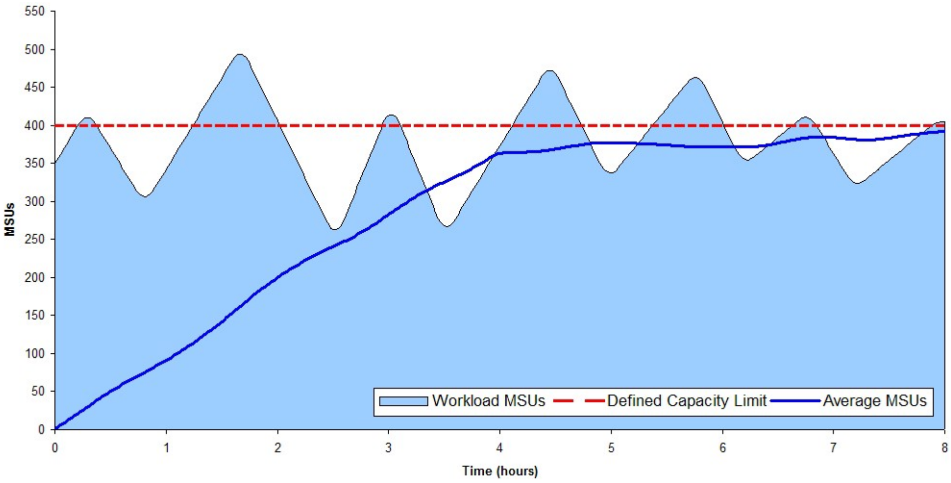


Figure 3.2.: Example of the 4hr Rolling Average

capping does not stop. Figure 3.3 shows this scenario with a constant high demand which is over the limit.

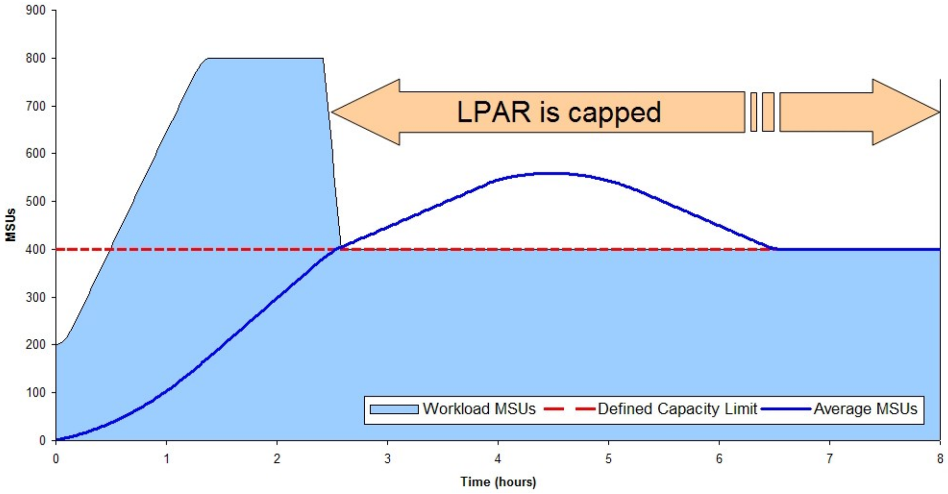


Figure 3.3.: Defined Capacity with constant high demand

When the workload of a capped LPAR reduces its demand and the 4hr rolling avg. drops below the DC limit, the capping ends and the workload is not impacted by the PR/SM capping anymore as you can see in Figure 3.4.

While the system is capped, it can not consume more than the defined limit in principle. As described later in this book, WLM uses two different methods for capping. While one of them does not allow to exceed the limit at all (like initial capping), the other one makes sure that a short term average (over about 5 minutes), will not exceed the limit.

As soon as the 4hr rolling avg. falls below the DC limit again, the LPAR can use all capacity it has access to. The LPAR stays uncapped until the 4hr rolling avg. goes above the DC limit again.

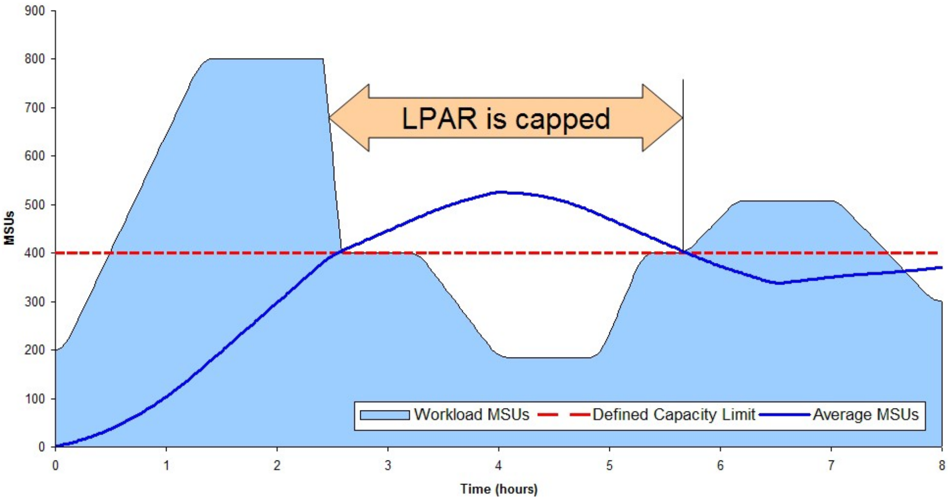


Figure 3.4.: Defined Capacity with declining demand

3.4.4. Why can the 4HRAVG exceed the DC limit

You may have wondered why in Figure 3.4 the average is still increasing even when the partition is capped.

Whenever the capacity which has been used 4 hours ago is smaller than the capacity that is used at the moment, the average will increase. This is because WLM is building the average over the 4 hour rolling window and if the capacity that is added to this window is greater than the capacity which disappears, the average increases.

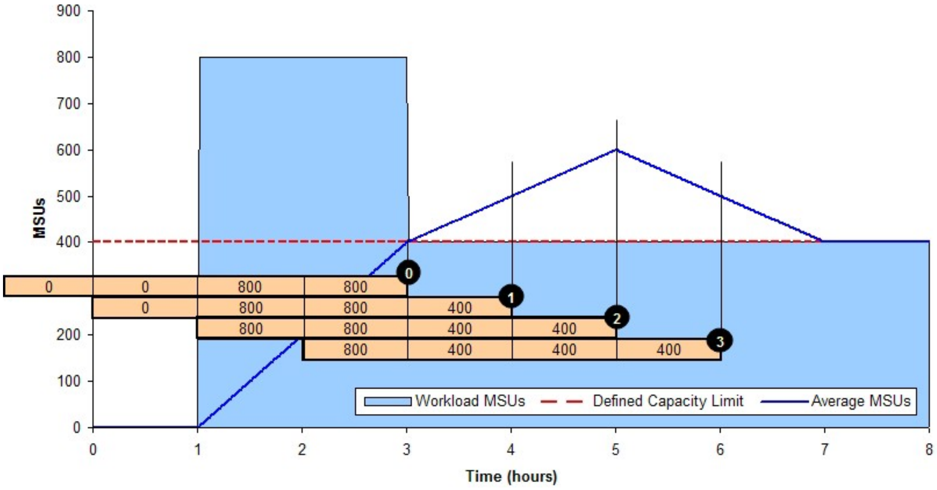


Figure 3.5.: Why does the 4HRAVG exceed the DC limit

This is especially visible at IPL time frame. Since the system was not IPLed four hours ago, all capacity which is added to the window is greater than zero. So if the system will hit its four hour average within the first 4 hours after IPL the average will always continue to increase.

In Figure 3.5 you can see an example where the 4hr rolling avg. still increases even when the partition is already capped.

The WLM four hour rolling average is empty when a partition gets IPLed. In the example above, the partition reaches the DC limit two hours after IPL at point 0 because it consumed 800 MSUs per hour and was IPLed for two hours (800 MSUs x 2 hours / 4 hours = 400 MSUs). The steps are:

0 At this point the partition is capped to 400 MSUs, but the average is still growing.

1 One hour later, the average consumption has increased to 500 MSUs: [800 + 800 + 400] / 4 = 500.

2 The average continues to increase to 600 MSUs: [800 + 800 + 400 + 400] / 4 = 600.

Beyond this point, the average decreases because the current consumption which is added to the window is smaller than the consumption four hours ago which gets subtracted from the average window.

3 While the average declines it reaches 500 MSUs: [800+400+400+400] / 4 = 500.

This explains why the average still increases whenever the current capacity is higher than the capacity which has been consumed four hours ago.

3.4.5. Mechanisms to cap LPARs

When WLM decides that it is required to cap a partition because the 4hr rolling avg. exceeds the DC limit, it has to choose one of two mechanisms to do the capping. WLM can not freely decide which method to use, but it depends on the relation between the DC limit and the MSU capacity which is represented by the weight of the partition.

While the DC limit is defined on HMC, the MSU@Weight is calculated by WLM using the equation (3.1).

$$MSUatWeight = \frac{LparWeight}{TotalWeight} \times CecCapacity \tag{3.1}$$

MSUatWeight equals to the current partition weight divided by the sum of all partition weights multiplied by the shareable CPU capacity of the CEC. For example if an LPAR has a current weight of 200 on a CEC with a total weight (sum of all LPAR weights) of 1000 and a total shareable capacity of2000 MSUs, the MSUatWeight will be 400 MSUs for this LPAR (see (3.1)).

$$MSUatWeight = \frac{200}{1000} \times 2000MSU = 400MSU \tag{3.2}$$

If MSUatWeight is below the defined capacity limit, WLM uses the cap pattern method to cap the partition. The cap pattern is required, because the partition needs to run uncapped part of the time, otherwise it would be capped down to the MSUatWeight level.

If an LPAR would get too much capacity even when it is capped all the time, a cap pattern wouldn't work because it can only cap to the MSU value represented by the weight. In this case, WLM caps the partition all the time and implements a phantom weight.

Figure 3.6 shows both methods side by side. Both capping methods cap to the DC limit but in a different way. You could influence the decision which method is used by changing the weight, but you should not do this just because you think one method is better than the other. The weight has to represent the importance of the LPAR, otherwise the capacity distribution may not be as expected.

3.4.6. Capping by Cap Pattern

As you already know, WLM is using PR/SM to do the actual capping. But WLM is using other metrics than PR/SM. While WLM is using service units (SU) for measuring the CPU consumption, PR/SM

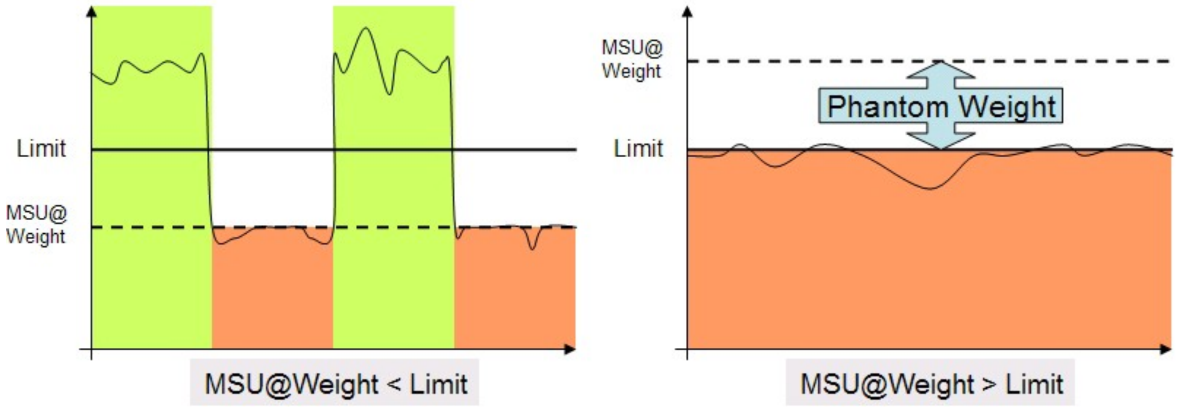


Figure 3.6.: WLM Capping Methods

distributes the CPU capacity by weight. Therefore WLM can not just tell PR/SM to cap at a specific MSU limit, it only can tell PR/SM to switch the capping ON or OFF.

WLM calculates a cap pattern with capped and uncapped intervals. When the LPAR runs uncapped, the consumption can exceed the limit and when the partition is capped, it can consume not more than MSUatWeight (as calculated above). To determine the correct mix of capped and uncapped intervals, WLM measures the consumption within the capped and uncapped intervals separately. This consumption is called short term average and spans about 5 minutes.

For example if the workload needs to be capped by 50%, the cap pattern will have the same amount of capped and uncapped intervals as you can see in figure 3.7.

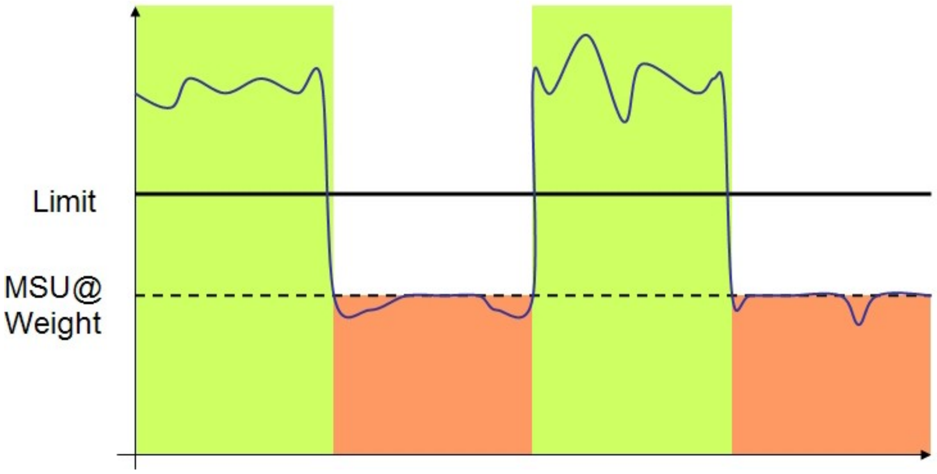


Figure 3.7.: WLM Cap Pattern at 50%

Every ten seconds, WLM verifies that the pattern is still appropriate or whether it needs to be adjusted. This is done by measuring the consumption during the capped and during the uncapped intervals. WLM calculates a new cap pattern based on this information if required.

3.4.7. Capping by Phantom Weight

When the MSUatWeight is higher than the defined capacity limit, capping to the capacity represented by the weight would not be enough. Therefore WLM has to tell PR/SM to do more than the *normal* capping down to MSUatWeight.

When PR/SM calculates whether an LPAR has already exceeded its capping limit, it calculates the share of the LPAR by comparing the weight of the LPAR with the total weight of all LPARs on the CEC and multiplies this ratio with the total shareable capacity.

Equation (3.3) depicts the formula which is identical to the calculation for MSUatWeight (see (3.1)). For example if an LPAR has a weight of 200 on a CEC with a total weight (sum of all LPARs) of 1000 and a total shareable capacity of 2000 MSUs, the MSUatWeight will be 400 MSUs for thie LPAR, see equations (3.3), (3.1), and (3.2).

$$LparCapacity = \frac{LparWeight}{TotalWeight} \times CecCapacity \tag{3.3}$$

Based on this calculation, PR/SM would cap a partition to its LparCapacity. When WLM needs to cap a partition below its weight, it has to tell PR/SM another value that PR/SM understands, the phantom weight. With the phantom weight added to equation (3.3) it looks like there would be anotherLPAR on the CEC which reduces the share of the LPAR see equation (3.4).

$$LparCapacity = \frac{LparWeight}{TotalWeight + PhantomWeight} \times CecCapacity \tag{3.4}$$

The *Phantom Weight* is only used to influence the LparCapacity calculation of PR/SM and will not impact the calculations of other LPARs. When WLM needs to cap the partition of the last example to 100 MSUs instead of 400 MSUs it will calculate a phantom weight by changing the formula of (3.4) to (3.5).

$$PhantomWeight = \frac{LparWeight \times CecCapacity}{LparCapacity} - TotalWeight \tag{3.5}$$

When we insert the values of the example above in equation (3.5), we can calculate the Phantom Weight. The bigger the phantom weight, the smaller the LparCapacity calculated by equation (3.3). This is because the phantom weight simulates an additional partition on the CEC which reduces the share of the partition for which we do the calculation. In our example the phantom partition needs a weight of 3000, to reduce the LparCapacity from 400 to 100, see equation (3.6).

$$PhantomWeight = \frac{200 \times 2000MSU}{100MSU} - 1000 = 3000 \tag{3.6}$$

This PhantomWeight of 3000 can be inserted in equation (3.4). Since the total weight of the CEC including the phantom partition increases from 1000 to 4000, the LparCapacity decreases from 400 to 100 MSUs (see equation (3.7)).

$$LparCapacity = \frac{200}{1000 + 3000} \times 2000MSU = 100MSU \tag{3.7}$$

When WLM uses the phantom weight, there is no need for a pattern and the LPAR is therefore capped 100% of the time as you can see in figure 3.8.

For a phantom phantom weight the partition is capped all the time compared to a cap pattern where WLM turns capping on and off at specified time intervals. As you will learn later in this book, this is the reason why RMF always shows a WLM% value of 100 when a partition is capped this way.

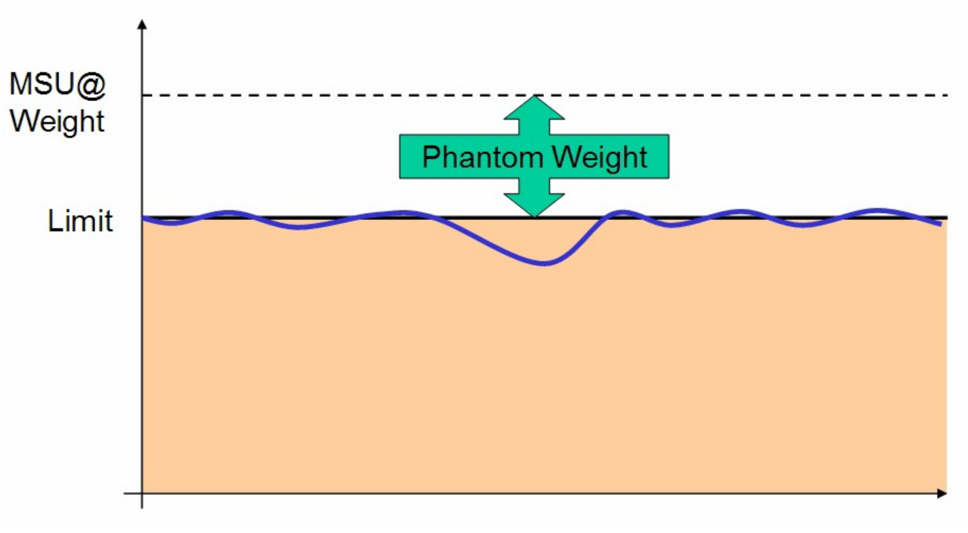


Figure 3.8.: WLM Phantom Weight Capping

3.5. Pricing aspect

If a sub-capacity eligible WLC product runs on an LPAR, the maximum charge for this product can be limited by using a defined capacity limit. For details about pricing see [16].

3.6. Other considerations

The defined capacity limit can only be reached if no other limiting factor prevents this. For example if there are not enough logical processors available, the limit can not be reached. Also other limits like a too small weight or a Group Capacity limit may disallow to reach the defined capacity limit.

HiperDispatch (HD) and DC work together with no restrictions. HD reacts on capped partitions for example by parking processors that are not used anymore due to the capped workload. If a partition is capped by phantom weight, the HD topology may change. For example the number of high polarized processors may decrease because the CPU access of a capped partition is restricted.

3.7. Undesirable Capping Effects

Phantom weight capping results in a homogeneous CPU consumption but the cap pattern method can be very inhomogeneous. The following section gives you some insights.

3.7.1. LPARs with a small weight on a large CEC using cap pattern

The WLM cap pattern algorithm calculates the number of capped and the number of uncapped intervals based on the CPU consumption which occur in these intervals. If the available capacity in the CEC is large compared to the defined capacity limit, the number of capped intervals can be very high. For example it is possible that the LPAR runs uncapped for 30 seconds in a 10 minute time frame only. Figure **??** shows how the cap pattern and its length depending on the capping percentage.

When the defined DC limit is small compared to the CPU capacity which is accessible for the LPAR, the capping percentage is high. In Figure 4-10 you can see that a high capping percentage means that

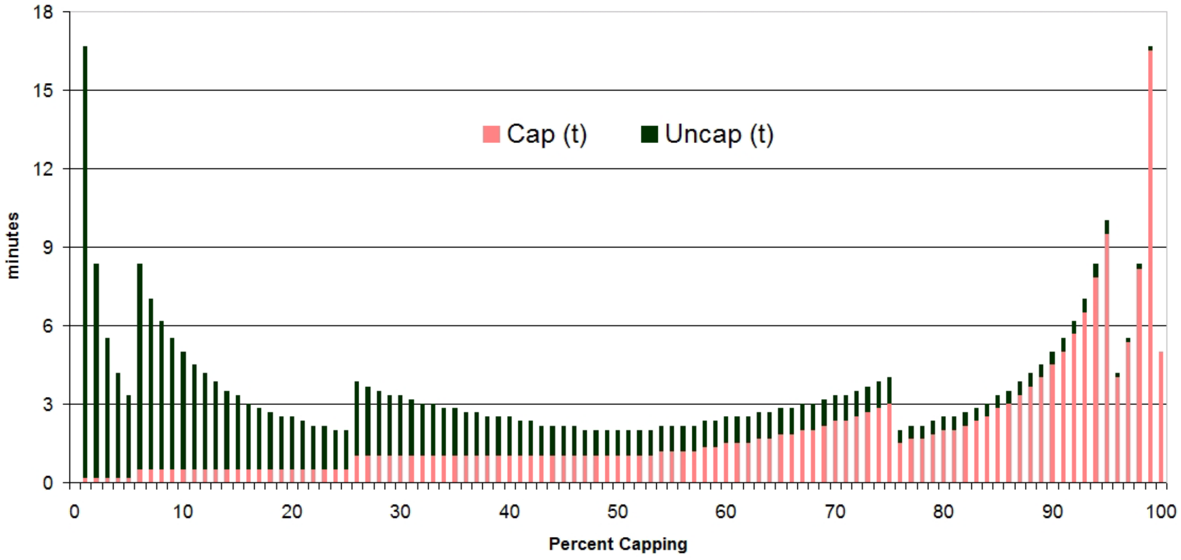


Figure 3.9.: WLM Cap Patterns

the partition is capped for a long time. Since cap pattern is only used when the MSU@Weight is below the defined capacity, we have now a situation where the LPAR is capped for a long time at a very low level compared to the uncapped intervals. You can see an example of this in figure 3.10.

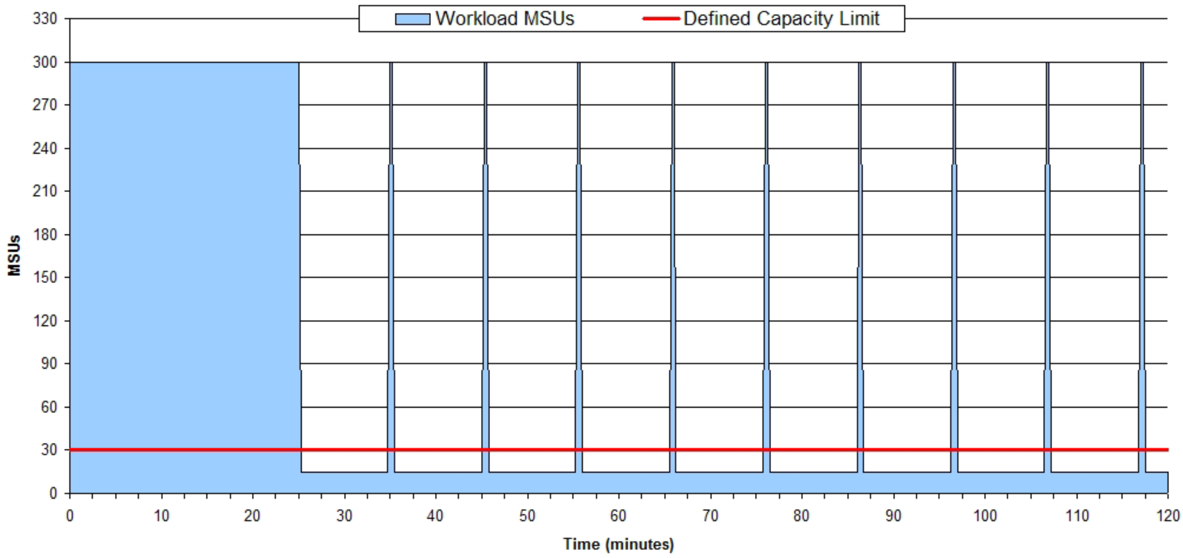


Figure 3.10.: Cap Pattern for a LPAR with a small weight

The result of this scenario is that the LPAR gets only 15 MSUs most of the time even when it is capped to a DC limit of 30 MSUs. From a DC perspective this is still correct because WLM ensures that the average consumption is at 30 MSUs as required by the DC limit. But be aware that this partition runs at a low speed in the capped intervals and if it is a member of a sysplex it may also impact other sysplex members because it may not react as fast as expected.

3.8. HMC Interface

The defined capacity limit is configured on Hardware Management Console (HMC). This section lists the options which are provided.

3.8.1. Define a DC limit

To activate WLM defined capacity you have to specify a DC limit. Each partition has its individual DC limit, which can be dynamically activated without IPL.

3.8.2. Change a DC limit

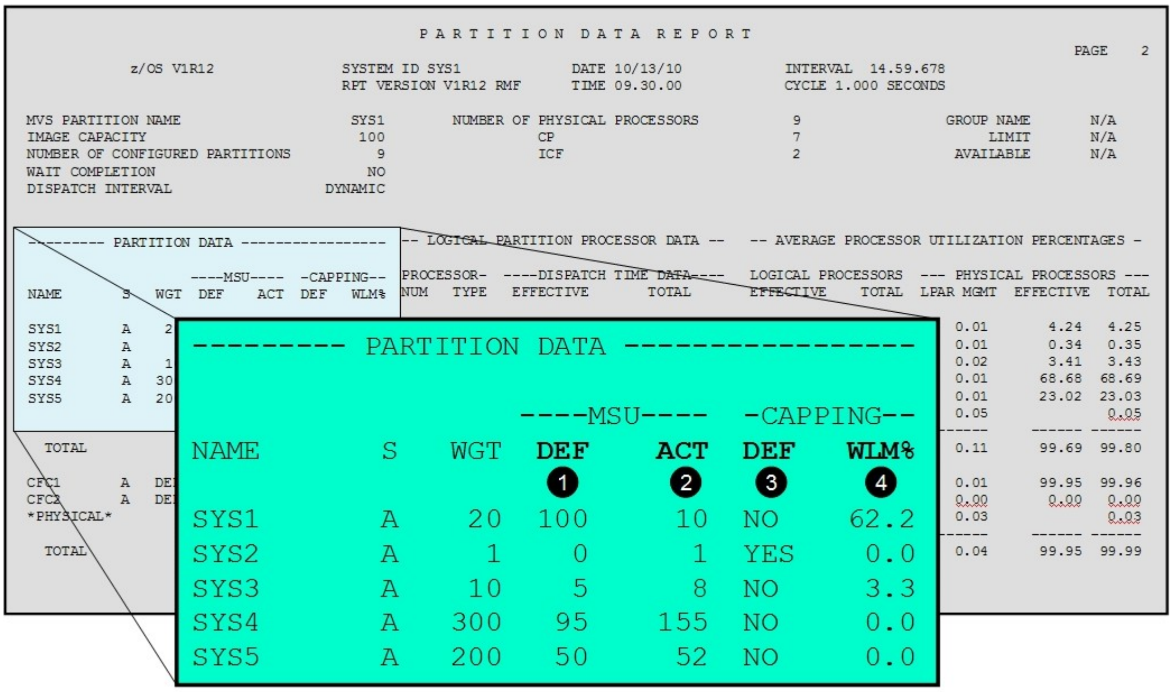
A DC limit can be changed at any time and it can be activated dynamically, IPL is not required. WLM will react to a changed limit in its next 10 second interval.

3.8.3. Remove a DC limit

A DC limit can also be removed at any time. WLM reacts on a removed DC limit in its next interval and if an LPAR is capped, the capping ends after removing the DC limit.

3.9. Minitoring Defined Capacity

3.9.1. RMF Partition Data Report



```
                              P A R T I T I O N   D A T A   R E P O R T
                                                                                          PAGE    2
           z/OS V1R12            SYSTEM ID SYS1         DATE 10/13/10         INTERVAL  14.59.678
                                 RPT VERSION V1R12 RMF  TIME 09.30.00         CYCLE 1.000 SECONDS

MVS PARTITION NAME                     SYS1    NUMBER OF PHYSICAL PROCESSORS     9          GROUP NAME    N/A
IMAGE CAPACITY                          100                CP                    7               LIMIT    N/A
NUMBER OF CONFIGURED PARTITIONS           9                ICF                   2           AVAILABLE    N/A
WAIT COMPLETION                          NO
DISPATCH INTERVAL                   DYNAMIC

 --------- PARTITION DATA -----------------  -- LOGICAL PARTITION PROCESSOR DATA --   -- AVERAGE PROCESSOR UTILIZATION PERCENTAGES -
                       ----MSU----  -CAPPING--  PROCESSOR-  ----DISPATCH TIME DATA----   LOGICAL PROCESSORS  --- PHYSICAL PROCESSORS ---
NAME          S    WGT  DEF    ACT  DEF   WLM%  NUM   TYPE    EFFECTIVE       TOTAL       EFFECTIVE    TOTAL  LPAR MGMT  EFFECTIVE  TOTAL

SYS1          A     2                                                                                         0.01        4.24    4.25
SYS2          A                                                                                               0.01        0.34    0.35
SYS3          A     1                                                                                         0.02        3.41    3.43
SYS4          A    30                                                                                         0.01       68.68   68.69
SYS5          A    20                                                                                         0.01       23.02   23.03
                                                                                                              0.05                0.05
                                                                                                            ------      ------  ------
  TOTAL                                                                                                       0.11       99.69   99.80

CFC1          A    DE                                                                                         0.01       99.95   99.96
CFC2          A    DE                                                                                         0.00        0.00    0.00
*PHYSICAL*                                                                                                    0.03                0.03
                                                                                                            ------      ------  ------
  TOTAL                                                                                                       0.04       99.95   99.99
```

```
--------- PARTITION DATA -----------------

                     ----MSU----  -CAPPING--
NAME      S    WGT   DEF     ACT  DEF   WLM%
                      1       2    3     4
SYS1      A     20   100      10  NO    62.2
SYS2      A      1     0       1  YES    0.0
SYS3      A     10     5       8  NO     3.3
SYS4      A    300    95     155  NO     0.0
SYS5      A    200    50      52  NO     0.0
```

Figure 3.11.: RMF Partition Data Report

RMF reports some values that allow to monitor DC capping in the Partition Data Section of the Postprocessor CPU Activity Report. The Partition Data Report shown in figure 3.11 reports which partitions have a DC limit defined and if they have been capped in the last interval.

1 **MSU DEF**	DC limit for this partition in MSU as specified on HMC
2 **MSU ACT**	Actual avg. MSU consumption of this LPAR
3 **CAPPING DEF**	Indicates whether this partition uses initial capping
3 **CAPPING WLM%**	Portion of time the LPAR was capped during the RMF interval

MSU DEF shows the DC limit as specified on HMC. The system is capped when the 4hr rolling avg. of the workload CPU consumption exceeds this value.

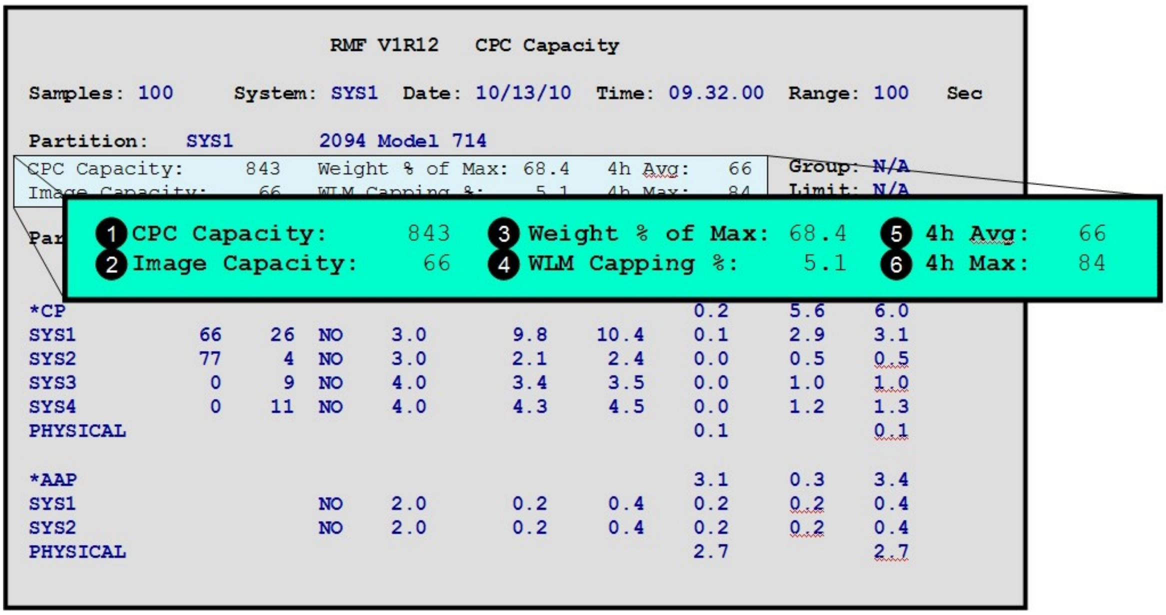
MSU ACT is the actual CPU consumption during this interval. Because this value is averaged over the RMF interval and not over the past four hours, this is not a reliable source to find out if the partition needs to be capped.

CAPPING DEF allows to find out if a partition is hard capped because initial capping is activated on HMC. If this is the case, the DC limit is ignored by WLM.

CAPPING WLM% tells the portion of time during the RMF interval, when the partition was capped by PR/SM. As described in section 3.4.7, the PR/SM capping is activated all the time when phantom weight is used, therefore the WLM% field is always 100%.

Note:
When you monitor a capped partition using a short monitor interval (e.g. 2 min.), you may recognize a pattern effect when the partition is capped by a cap pattern. This means that you will see some intervals with a high CPU consumption and others with a low CPU consumption (see Figure **??** for pattern length).

3.9.2. RMF Monitor III CPC Capacity Report



```
                         RMF V1R12    CPC Capacity

Samples: 100      System: SYS1   Date: 10/13/10  Time: 09.32.00  Range: 100    Sec

Partition:    SYS1          2094 Model 714
CPC Capacity:      843    Weight % of Max: 68.4    4h Avg:     66     Group: N/A
Image Capacity:     66    WLM Capping %:    5.1    4h Max:     84     Limit: N/A

Par

*CP                                                      0.2      5.6     6.0
SYS1            66     26  NO     3.0        9.8    10.4   0.1      2.9     3.1
SYS2            77      4  NO     3.0        2.1     2.4   0.0      0.5     0.5
SYS3             0      9  NO     4.0        3.4     3.5   0.0      1.0     1.0
SYS4             0     11  NO     4.0        4.3     4.5   0.0      1.2     1.3
PHYSICAL                                                 0.1              0.1

*AAP                                                     3.1      0.3     3.4
SYS1                       NO     2.0        0.2     0.4   0.2      0.2     0.4
SYS2                       NO     2.0        0.2     0.4   0.2      0.2     0.4
PHYSICAL                                                 2.7              2.7
```

```
1 CPC Capacity:       843   3 Weight % of Max: 68.4   5 4h Avg:     66
2 Image Capacity:      66   4 WLM Capping %:    5.1   6 4h Max:     84
```

Figure 3.12.: RMF Monitor III CPC Capacity Report

In the header of the RMF Monitor III CPC Capacity report [OVERVIEW(1), CPC(3)] there are also fields which are useful to monitor DC as shown in figure 3.12.

The report shows the following DC related fields:

1	**CPC Capacity**	Total capacity of the CPC in MSU/h
2	**Image Capacity**	Maximum capacity available to this partition
3	**Weight % of Max**	Percentage of time where this partition runs at its maximum
4	**WLM Capping %**	Percentage of time where this partition was capped by WLM
5	**4h Avg**	Average consumed MSU/h during the last 4 hours
6	**4h Max**	Maximum consumed MSUs during the last 4 hours

CPC Capacity is the total processor capacity of the CEC.

Image Capacity is the total capacity of the partition. If a partition has a DC limit, the Image Capacity matches the DC limit if the LPAR is not restricted by other metrics like the number of logical processors.

Weight % of Max shows the average weighting factor in relation to the maximum defined weighting factor of the partition.

WLM Capping% tells the portion of time during the RMF interval, when the partition was capped by PR/SM. As described in section 3.4.7, the PR/SM capping is activated all the time when phantom weight is used, therefore the WLM Capping % field is always 100% in this case.

4h Avg is the four hour rolling average of the workload consumption of the partition. When this value exceeds the DC limit, the partition gets capped by WLM.

4h Max shows the maximum five minute average consumption that occurred in the last four hours. WLM tracks the consumption of the last four hours by keeping 48 buckets of consumption data. Each bucket keeps the CPU consumption of 5 minutes. The 4h Max is the highest bucket value that have been consumed in the last four hours.

3.9.3. Useful SMF70 Fields

Table 3.1 lists some useful data for defined capacity located in the SMF Type 70 record.

SMF Field	Meaning	Description
SMF70MSU	DC limit	Defined capacity limit (in millions of service units).
SMF70LAC	4 hr rolling avg.	Long-term average of CPU service (millions of service units). Scope of the value depends on bit 3 of SMF70STF.
SMF70NSW	WLM%	Number of Diagnose samples where WLM considers to cap the set of logical CPUs of type SMF70CIX within the logical partition (see also SMF70NCA).
SMF70NCA	ACT%	Number of Diagnose samples where capping actually limited the usage of processor resources for the set of logical CPUs of type SMF70CIX within the logical partition.
SMF70PFG	Flags	Bit 6: Defined capacity limit has been changed.
SMF70VPF	capping is active	Bit 3: Partition capping is enabled.
SMF70PMA	phantom weight	Average adjustment weight for pricing management.

Table 3.1.: SMF Type 70 Fields for Defined Capacity

3.9.4. What about SMF72 Fields

The unit of the listed SMF70 records fields is in software service units, while the SMF72 record reports weighted hardware service units. You can not compare the values from SMF70 to the values of SMF72 for two reasons:

- The SMF72 service unit fields are weighted Service units reported in SMF72 are weighted by using the WLM CPU service coefficient but the software service units from SMF70 are not.
- Hardware and software service units are calculated the same way by using different machine dependent constants. On a System z196 the software service units (MSU) are about 10% smaller than the hardware service units.

4. Group Capacity

WLM Group Capacity (GC) is based on WLM Defined Capacity (DC) but it is not restricted to a limit for one LPAR. While DC manages an LPAR towards its own limit, GC allows to define a limit for a group of LPARs on HMC.

The consumption of the group members is combined and limited to one common group limit. As long as the 4hr rolling avg. of the group is below the group limit, the group runs uncapped and each group member can consume whatever it demands. When the 4hr rolling avg. of the group exceeds the limit, the group limit is subdivided into portions for the group members according to their weight.

The scope of GC is the CEC, this means some or all systems of a CEC can belong to the same group independent if they belong to the same sysplex or not, see figure 4.1.

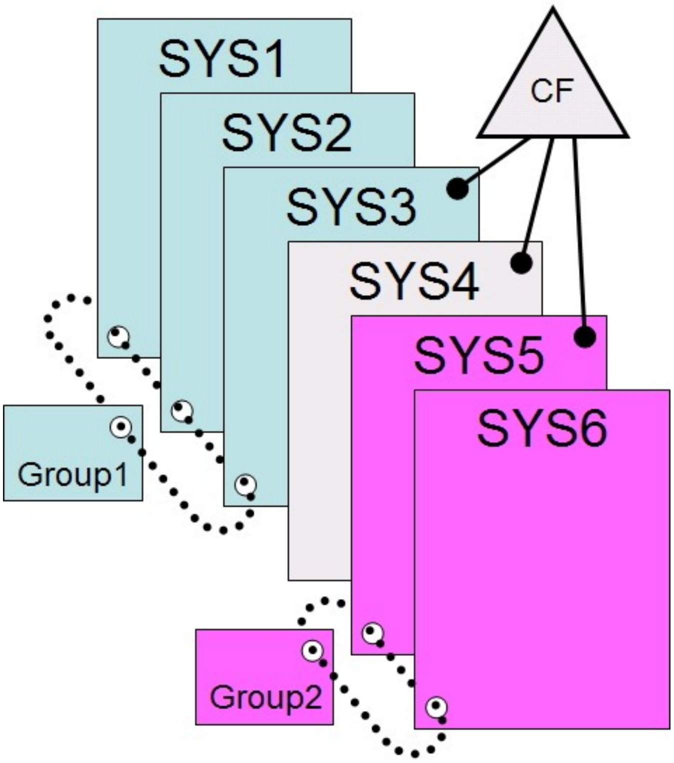
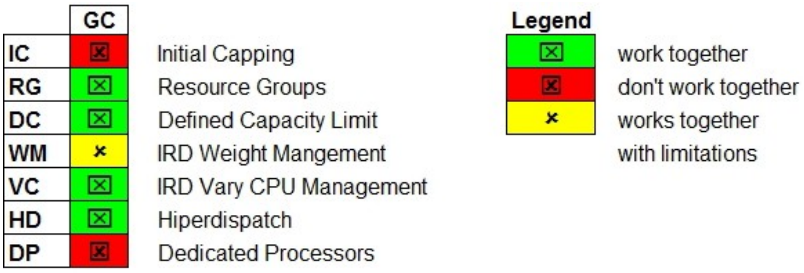


Figure 4.1.: Scope of WLM Group Capping

4.1. General Concepts

GC is using the same methods to cap an LPAR as DC but in addition to this WLM monitors the consumption of the other group members and adjusts the limit of its own partition. This allows WLM to make sure that the average consumption of the group does not exceed a common group limit which is defined on HMC.

4.2. Requirements



	GC	
IC	🟥 ☒	Initial Capping
RG	🟩 ☒	Resource Groups
DC	🟩 ☒	Defined Capacity Limit
WM	🟨 ×	IRD Weight Mangement
VC	🟩 ☒	IRD Vary CPU Management
HD	🟩 ☒	Hiperdispatch
DP	🟥 ☒	Dedicated Processors

Legend	
🟩 ☒	work together
🟥 ☒	don't work together
🟨 ×	works together with limitations

Figure 4.2.: Group Capacity together with Other WLM Technologies

Figure 4.2 shows that GC is not independent from all of the other WLM resource controlling mechanisms and therefore it can not be used together with all of them. IRD Weight Management only works as long as the defined capacity limit is not met and the partition is not capped. As soon as the partition is capped weight changes are not possible anymore.

Note:
When a partition uses dedicated processors, initial capping or WAITCOMPLETION=YES, WLM will not activate Group Capacity on this partition.The capacity group may exceed its group limit because the hard capped partition does not cap itself to its GC share. If GC is desired, make sure to disable these features for this partition on HMC.

4.3. Advantages of a Capacity Group

Because most LPARs are connected to other’s for several reasons, it’s often desirable to manage them against a common limit.

A group of LPARs with individual DC limits can not replace a GC limit because the individual limits do not allow an LPAR to consume the capacity of another LPAR which is below its limit. For example if there are three LPARs with a DC limit of 100 MSUs, each of the LPARs is restricted to 100 MSUs in average. When these three LPARs are connected to an LPAR group with a common limit of 300 MSUs, a group member could demand for example 200 MSUs in average when the other group members have a combined demand of 100 MSUs in average.

In figure 4.3 you can see an example of three partitions which are defined to be in the same group. They get the same kind of IPL bonus as partitions using a DC limit. When the average consumption exceeds the group limit, the group members are limited to the group limit which is 50 MSUs in the example. As you can notice at the left side of the chart, SYS1 and SYS2 have the biggest demand. After reaching the group limit, the group is capped to the 50 MSUs but SYS1 and SYS2 benefit from the fact that SYS3 is not using its share. But as soon as SYS3 is demanding its share, it gets all of it and the other LPARs are throttled back to their share which is based on their current weight.

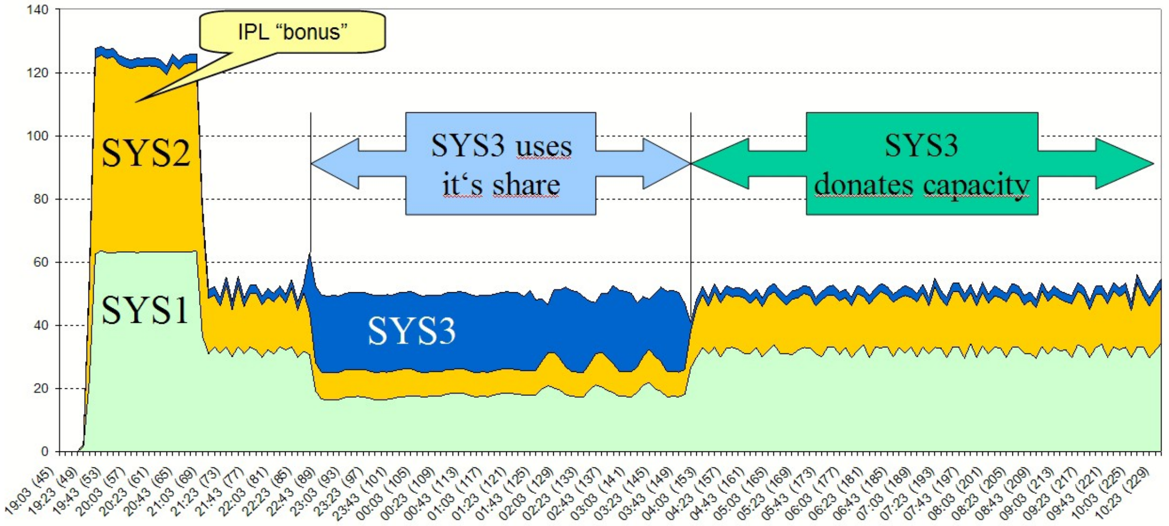


Figure 4.3.: Group Capacity Example with three partitions

4.4. How does WLM combine LPARs to a capacity group

WLM Group Capacity (GC) adds an additional layer to WLM Defined Capacity (DC). If an LPAR is belonging to a capacity group, it determines the other group members and collects data about their CPU consumption. As soon as an LPAR recognizes that the group exceeds its group limit, it calculates its own share and uses the same capping mechanisms as DC to cap itself to this share. If there is an additional DC limit defined, the minimum of the calculated group share and the DC limit is used to cap the partition.

4.4.1. Data Colletion

The hardware tells WLM about the other LPARs, their group connection and their CPU consumption. This is a *one way* interface, so each LPAR collects its own data and does not communicate with the other LPARs.

Note:
As you will see later, this has some disadvantages but allows to build groups across sysplex borders.

4.4.2. Maintaining the unused capacity

Every ten seconds, when WLM collects data about the capacity group, it calculates what the group consumed in the previous interval and subtracts this consumption from the capacity which is allowed by the group limit. This unused capacity is stored in the so called *unused vector*.

Figure 4.4 shows the stacked consumption of the individual group members and the line which is sometimes below zero is the unused vector. When the unused vector is negative, the group consumes more than the group limit and when the unused vector is positive, the group consumes less than the group limit (400 MSUs in the example).

The unused vector is an important metric for group capacity because its 4 hour rolling average decides if a group needs capping. When a group needs capping, at least one member gets capped. You can understand the average Unused Capacity (aUC) as an inverse 4hr rolling avg. If the aUC is high, the group has plenty of capacity to consume before capping occurs. If the aUC turns to be negative, the

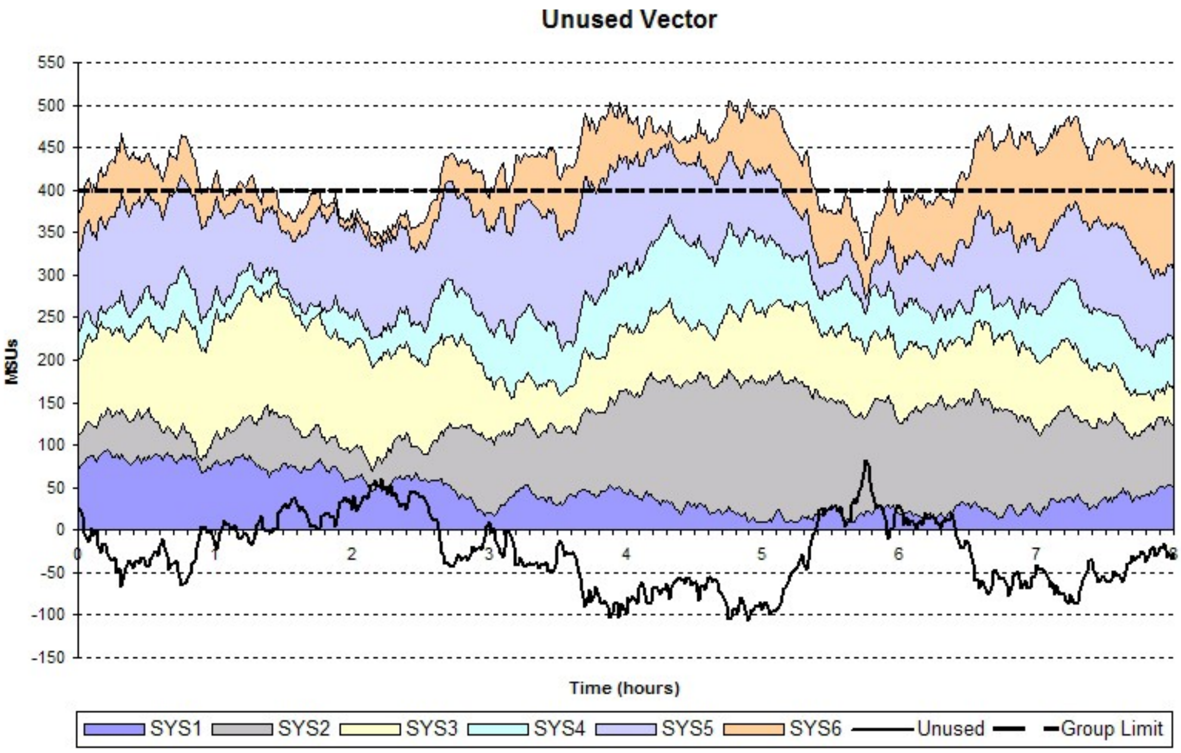


Figure 4.4.: Capacity Group and Unused Vector

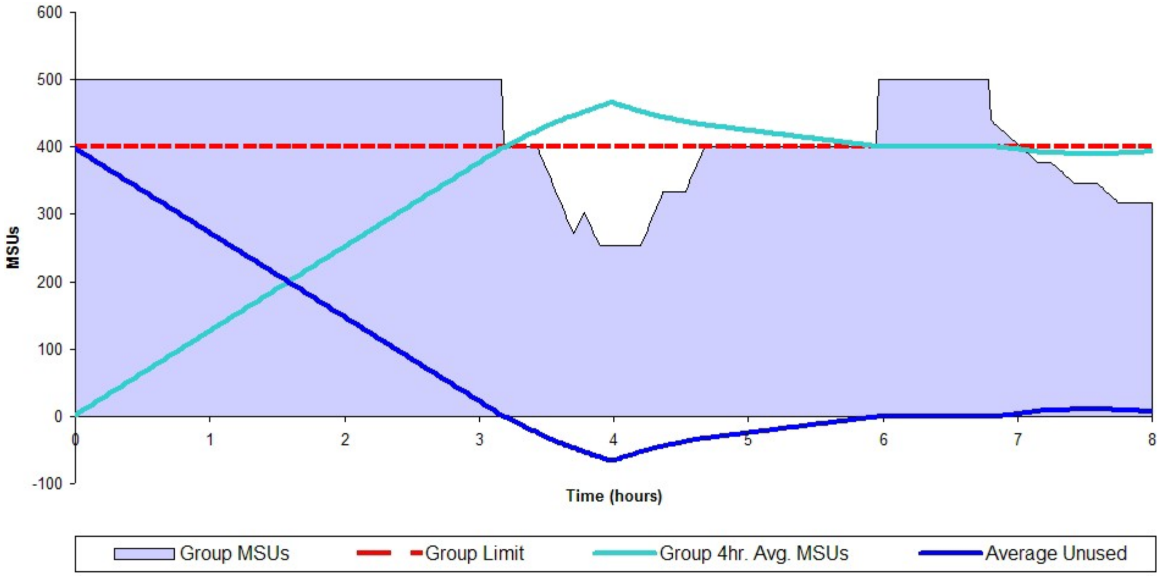


Figure 4.5.: Unused Vector and 4 Hour Rolling Average

group is capped and each group member calculates its individual capping limit. As long as the aUC is negative the group stays capped and each group member recalculates its limit every 10 seconds.

Figure 4.5 depicts the relationship between the aUC and the 4 hour rolling average. The unused vector is a mirrored 4 hour rolling average. When the unused vector becomes negative it is the same as if the 4 hour rolling average exceeds a defined capacity limit.

Since the group members calculate the aUC independent from each other, each group members has a slightly different aUC. WLM recalculates the aUC in each group member every ten seconds but the partitions don't do it at the exactly same time. Therefore the group members may react with some time offset.

4.4.3. Capping of a Capacity Group

As long as the aUC is positive, there is no capping required for the group and the workload is not impacted by group capacity. But as soon as the aUC turns to be negative, the capacity group is exceeding its group limit and at least one member needs to be capped.

When the group needs to be capped, each member calculates its portion of the group capacity as shown in the next section. If a group member detects that it has to be capped, it calculates an appropriate limit and caps itself by activating PR/SM capping.

Calculating the share of the group members

Each group member calculates its limit individually and decides if capping is required for itself. The first step is to calculate the MemberCapacity which is the minimum entitlement of the group member.

The group limit is distributed proportionally to the weight of the group members. Each member calculates as shown in equation (4.1).

$$MemberCapacity=\frac{LparWeight}{GroupWeight}\times GroupLimit \tag{4.1}$$

The MemberCapacity is calculated by dividing the current weight of the partition by the sum of the current weights of the group members and multiplying the result by the group limit. For example if a group member has a current weight of 200 and the group has total weight (sum of the weights of all group members) of 600 and a group limit of 900 MSUs, the MemberCapacity calculates to be 300 MSUs, see example in equation (4.2).

$$MemberCapacity=\frac{200}{600}\times 900MSU=300MSU \tag{4.2}$$

A group member should get at least its MemberCapacity in average if it has the demand for it. When all group members demanding their member capacity, the result of equation (4.1) is the capping limit that applies to the group members.

Redistributing available capacity

Most of the time at least one group member will not take its share and therefore WLM has to redistribute donated capacity amongst the group members which demand more than their share. To do this, WLM needs to differentiate between donors and receivers. Every ten seconds when WLM collects the consumption data of the group, it will also verify if a partition takes its share or not.

All group members which use their share are treated as receivers and all group members which don't use all of their share are treated as a donor. WLM summarizes the donated service units of the donors and the weight of the receivers.

All receivers get in addition to their member share a DonatedCapacity which is the share of the donated service units, see equation (4.3).

$$DonatedCapacity = \frac{LparWeight}{ReceiverWeightSum} \times DonatedSUs \tag{4.3}$$

The DonatedCapacity is calculated by dividing the current weight of a receiver by the total weight of all receivers and multiplied by the donated service units of the donors. For example a receiver with a weight of 200 and a total receiver weight of 400 in a group with 20 MSUs donated service units leads to the following a DonatedCapacity of 10 MSUs, , see equation (4.4).

$$DonatedCapacity = \frac{200}{400} \times 20MSU = 10MSU \tag{4.4}$$

Each partition does this calculation independent from the others. In the example above the calculating LPAR would get in addition to its member share of 300 MSUs a donated share of 10 MSUs. The total group share of the group member is now calculated to be 310 MSUs.

Capping a group member

Now that WLM knows the total share of a group member, it will convert this share to a soft cap. If there is also a Defined Capacity limit defined for this partition, WLM will take the minimum from DC and the GC share to limit the partition. When a partition is not allowed to consume its GC share because of a DC limit, the part of the GC share that is not consumed is redistributed as DonatedCapacity in the next interval.

The example in table **??** consists of a capacity group with three members. One of them (SYS3) has a DC limit also.

System	Weight	DC limit	GC limit	MemCapacity	Donates	Limit
SYS1	600		400	200		240
SYS2	300			100		120
SYS3	300	40		100	60	40

Table 4.1.: Group Capacity Example

Let's assume that all three systems have a high CPU demand. The group limit of 400 MSUs would allow SYS3 to consume 100 MSUs in average, but because it's limited by DC to 40 MSUs, it will donate 60 MSUs to the other group members. SYS1 gets in addition to its MemberCapacity of 200 MSUs a DonatedShare of 40 MSUs and SYS2 gets additional 20 MSUs. SYS1 gets twice as much DonatedShare as SYS2 because its weight is twice as high.

Keep in mind that each group member does its own calculation and also each member caps itself by telling PR/SM to do the capping. A group member does not decide for another group member and they don't communicate with each other (they are not necessarily in the same sysplex).

This limit will be enforced by the same algorithms that are used for Defined Capacity which are explained in section 3.4.5.

4.5. Pricing aspect

If a sub-capacity eligible WLC product runs on an LPAR, the maximum charge for this product can be limited by using GC. For details about pricing see [16].

4.6. Other considerations

The group capacity limit can only be reached if no other limiting factor will prevent this. For example if there are not enough logical processors available, the limit can not be reached. Also other limits like too small weights or Defined Capacity limits may disallow to reach the group capacity limit.

All members of a capacity group need to reside on the same CEC but they have not to be in the same sysplex. An LPAR can be connected to one capacity group only, but multiple groups can exist on a CEC.

As described before, there is no communication between the LPARs about group capacity. Each group member verifies its calculations every ten seconds, but the members don't do this at exactly the same time, therefore the group members may react on workload changes with a short offset.

The unused vector, which determines if the group needs to be capped is calculated on each system individually. During normal processing this vector should be almost the same on all group members, but when the group composition changes because a member gets IPLed into the group or a member joins the group, it can take up to 4 hours until all group members are synchronized.

4.7. Scenarios

4.7.1. Increasing a Group Limit

When the group limit gets increased, WLM will react on the changed limit in the next policy adjustment interval (every 10sec). If you monitor the avg. unused vector in RMF, you can notice that it will increase. The consequences for the partitions depend on the 4hr rolling average of the group compared to the group limit.

Capacity group average is below its limit

If the capacity group is below its limit (positive average unused vector), WLM compares the average of the group to the new limit, but since the group was not capped at the lower limit there is even less need to cap the group with the higher limit.

Capacity group average exceeds its limit

WLM compares the average of the group to the new limit. If the new group limit is above the 4hr rolling avg. of the group, the capping will end for all group members. But even when the average of the group is still above the new limit, the group members profit of the higher limit. WLM is calculating a new individual capping limit for each group member which is higher than the old one and the capping may end for some group members. The group members which are still capped are capped at a higher level and get more capacity.

4.7.2. Decreasing a Group Limit

When the group limit gets decreased, WLM will react on the changed limit in the next policy adjustment interval (every 10sec). If you monitor the avg. unused vector in RMF, you can notice that it will decrease. The consequences for the partitions depend on the 4hr rolling average of the group compared to the group limit.

Capacity group average is below its limit

If the capacity group is above its group limit (negative avg. unused vector) already, WLM continues to cap the members of the group. If not all members of the group were capped with the higher limit, it is possible that additional members get capped now. If the CPU consumption of the workload drops down, it will take more time than it would have taken with the higher limit for the capping to end.

Capacity group average exceeds its limit

WLM compares the average of the group to the new limit. If the new group limit is below the 4hr rolling average of the group, the capping will start for at least one group member. WLM calculates an individual capping limit for each group member and compares the 4hr rolling average of each group member with its new individual limit.

4.7.3. IPL of a capacity group

Similar to Defined Capacity, Group Capacity does allow to exceed the 4hr rolling avg. at IPL time frame. You can find a technical description in section 3.4.4.

How this applies to a capacity group can be seen in figure 4.6. This figure shows the stacked 4hr rolling avg. of three capacity group members.

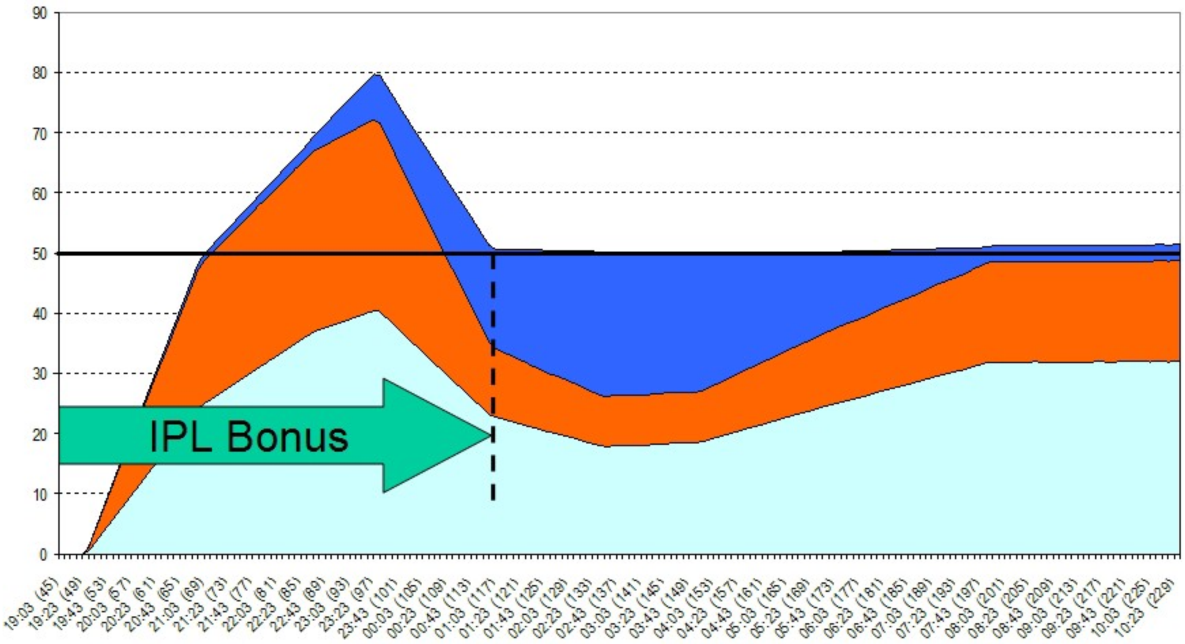


Figure 4.6.: Group Capacity IPL Bonus

In this example all three partitions of the group got IPLed at the same time. Due to the IPL bonus, the 4hr rolling average of the group exceeds the group limit.

Since WLM has a precision of about 1 MSU on each system and PR/SM has a precision of about 3% of the limit, the 4hr rolling average can deviate from the limit slightly as you can see at the end of the chart.

4.7.4. A group member leaves a capacity group

When a group member leaves a capacity group, the remaining group members are managed against the group limit. Since the total group weight is reduced, the MemberCapacity of the remaining group members is increased, see section 4.4.3. When a group member leaves a group does not impact the group limit.

4.7.5. A group member moves to another capacity group

When a partition moves to another group, its aUC gets cleared, meaning that it assumes that the new group has no unused capacity from the past four hours. It takes four hours until all group members use the same aUC.

If the group has a high demand at this moment, the new group member may get a negative aUC very fast and it caps itself. The other group members of the new group may continue to run uncapped because they use unused capacity they collected the last four hours which is not known by the new group member.

4.7.6. The limit of a capacity group gets removed (limit=0)

A capacity group needs a name and a limit, otherwise WLM ignores it.
This means that lowering the limit to zero suspends the group and cleares the aUC. After adding a new group limit to this group, it starts with a cleared aUC. If the group consumes more than its group limit, the aUC can get negative very soon, and the group gets capped.

4.7.7. An additional group member gets IPLed into a existing group

When an LPAR gets IPLed into a existing group, it will not know what the other group members consumed in the past four hours. Since it has just been IPLed, it gets its IPL bonus and stays uncapped until the group exceeds the group limit based on its own aUC.

The system that has just IPLed has a different view of the group in the first four hours, so it may recognize the exceeded limit later than the other group members. Therefore it may run uncapped even when the other systems are capped already. An example of this can be seen in figure 4.7.

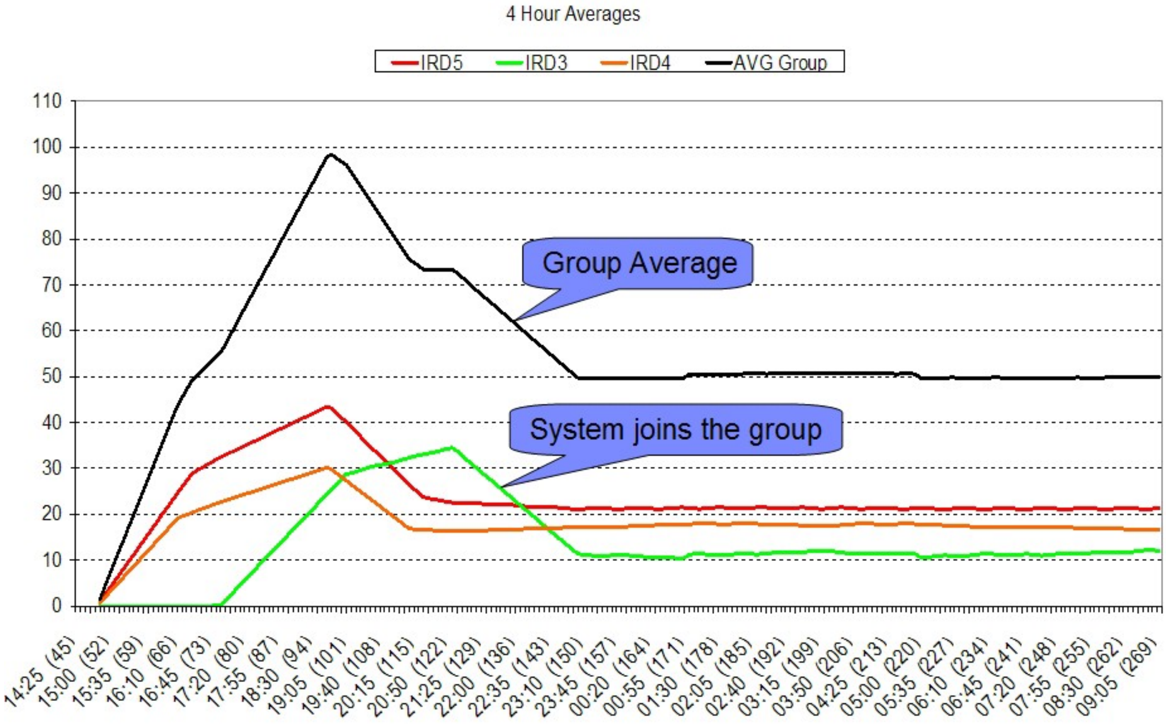


Figure 4.7.: LPAR joining a capacity group

Two of the LPARs have been IPLed at the same time and as soon as the group average is above the group limit (of 50 MSUs), these two LPARs get capped. While the two LPARs are already capped a third partition gets IPLed. Since the third LPAR does not know the past consumption of the other two LPARs, it takes its IPL bonus. Later on, the third partition will also get a negative unused vector and will cap itself.

Note:
If you IPL a new group member into an existing group, the new member may impact the group even if it has a low weight because it may get capped later than the existing group members.

4.7.8. Group Capacity adapting to workload changes

When the workload changes within a capacity group, WLM has to adapt the share of the group members. Since WLM uses short term averages (over a 5 min. time span) to measure the current consumption, the capping limits need some time to adapt to the new situation. An example of this can be seen infigure 4.8.

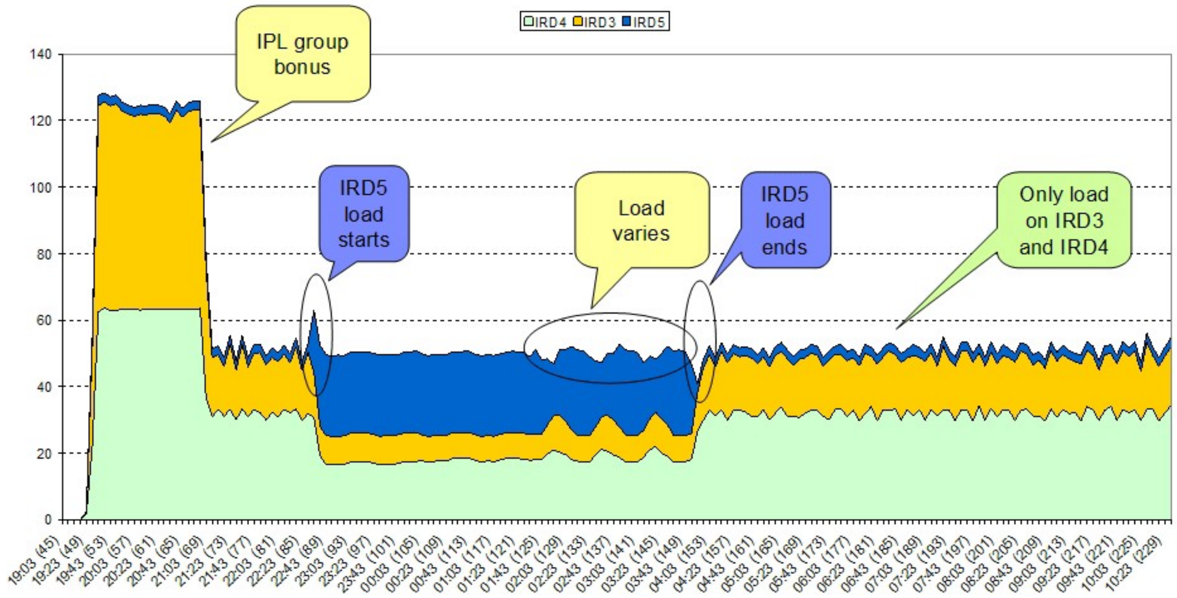


Figure 4.8.: Group Capacity Example with changing demands

The example shows a capacity group consisting of three members named IRD3, IRD4 and IRD5. While IRD3 and IRD4 are heavy consumers right from the start, IRD5 has no workload running after IPL. When the 4hr rolling avg. of the group exceeds the limit of 50 MSUs, the group is capped to a total consumption of 50 MSUs. Since IRD5 has no workload, IRD3 and IRD4 consume the capacity donated by the idle partition IRD5.

When IRD5 starts to take its share, the other LPARs will lower their capping limit and PR/SM reduces their consumption. When the workload of IRD5 drops down, the other LPARs increase their share and consume the donated capacity. As soon as the workload of IRD5 ends, the other LPARs compensate this by increasing their limit.

Workload changes can be seen as spikes in the consumption chart above. They are caused by the reaction time of WLM.

4.8. HMC Interface

The definition of WLM capacity groups and their members are done on HMC. The groups can be modified at any time without the need to reIPL the group members.

4.8.1. Define a group

A WLM capacity group consists of a group name and a group limit. None of them is optional. If WLM recognizes the name to be blank or the limit to be zero, it ignores this group.

4.8.2. Connect LPARs to a group

Some or all LPARs of a CEC can be added to a WLM capacity group. The group members don't need to be in the same sysplex. There can exist multiple groups on a CEC, but each LPAR can only belong to one group. The group members can not use initial capping or dedicated processors together with GC. WLM weight management can coexist with GC with the limitation that WM will not change the weight of capped LPARs.

4.8.3. Change a group limit

A group limit can be changed at any time. The group will adapt to the new limit in the next WLM policy adjustment interval (every 10 seconds).

4.8.4. Remove an LPAR from a existing group

When an LPAR gets removed from a existing group, the remaining members are managed against the group limit. If you want to take away the capacity of this member also, you have to lower the group limit.

4.8.5. Add an LPAR to a existing group

After an LPAR gets linked to another group, it will not know what the other group members consumed in the last four hours. Therefore it takes up to four hours until the group is synchronized again. It may occur that the new group member gets capped earlier or later than the other group members depending of their cpu consumption of the last four hours.

4.9. Interactions with other functions

As described earlier, WLM Group Capacity (GC) and WLM Defined Capacity (DC) can be used together. If a group member of GC has also an DC limit, the minimum of the DC limit and the GC share will be used to cap the partition.

GC and WLM IRD Weight Management (WM) can be active at the same time. This means that WM can change the weight of group members according to the rules of WM. WLM may redistribute the capacity provided by the group limit because the share of the group members will change when their weight changes. Because GC does react on the changed weight, GC and WM work together in principle. But there is a restriction for WM because it does not change the weight of capped partitions. As soon as the capping ends, WM is able to change the weight of the partition if needed.

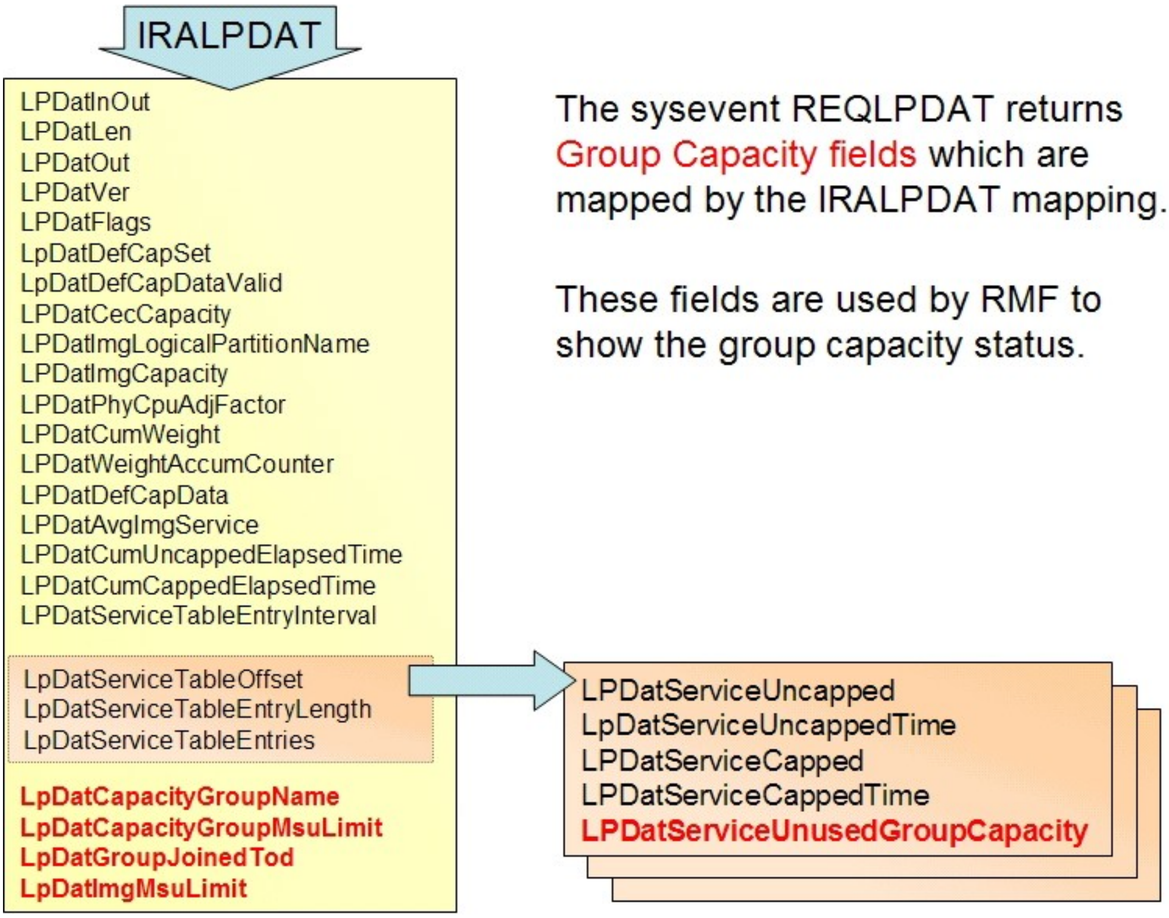
WLM IRD VARY CPU (VaryCPU) management does not vary CPUs online or offline for capped partitions. When a partition is capped by cap pattern, VaryCPU is not allowed to reduce the number of CPUs because they will be needed a few seconds later when the partition is uncapped. HiperDispatch (HD) and GC work together with no restrictions. HD reacts on capped partitions for example by parking processors that are not used anymore due to the capped workload. If a partition is capped by phantom weight, the HD topology may change. For example the number of high polarized processors may decrease because the CPU access of a capped system is restricted.

4.10. Interfaces that provide information about GC

The WLM interfaces allow other programs to get information about group capacity. While REQLPDAT is used by monitors like RMF, the SMF99 data is meant for debugging purposes, only.

4.10.1. REQLPDAT

WLM provides the REQLPDAT interface for monitors like RMF. Figure4.9 shows the fields that have been added for GC.



The sysevent REQLPDAT returns Group Capacity fields which are mapped by the IRALPDAT mapping.

These fields are used by RMF to show the group capacity status.

Figure 4.9.: Information provided by REQLPDAT

While the content of GroupName and GroupMsuLimit is quite obvious, the others may need some explanation.

The GroupJoinedTod value allows RMF to recognize since when a partition is a member of its group. This is necessary to highlight a partition which joined its group less than four hours ago and may has therefore not the same data as the other group members.

ImgMsuLimit is the dynamically calculated MSU limit of the LPAR. This includes the MemberCapacity and also the DonatedCapacity which are explained in section 4.4.3. If a partition has a DC limit defined, the ImgMsuLimit keeps the minimum of the DC limit and the share which has been calculated by GC.

The ServiceUnusedGroupCapacity field is part of the service table which has 48 entries. Each entry keeps the information of five minutes (48 x 5min = 4 hours). These values are the so called “unused vector” and are the base for the average Unused Capacity (aUC) which decides if a group needs to be capped.

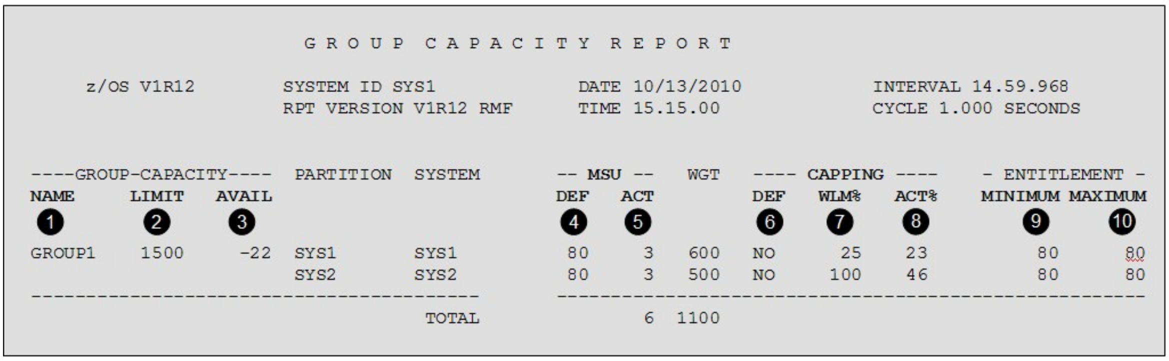
4.10.2. SMF99

The SMF99 Subtype 11 records keep group capacity information which is intended for use of IBM service personnel, only.

4.11. How can GC be monitored

4.11.1. RMF Group Capacity Report

RMF provides a Group Capacity Section in the Postprocessor CPU Activity Report. This Section covers all important information about GC as can be seen in figure 4.10.



```
                        G R O U P   C A P A C I T Y   R E P O R T

     z/OS V1R12         SYSTEM ID SYS1             DATE 10/13/2010            INTERVAL 14.59.968
                        RPT VERSION V1R12 RMF      TIME 15.15.00              CYCLE 1.000 SECONDS


 ----GROUP-CAPACITY----  PARTITION  SYSTEM       -- MSU --   WGT   ---- CAPPING ----    - ENTITLEMENT -
 NAME      LIMIT   AVAIL                         DEF   ACT         DEF   WLM%   ACT%    MINIMUM MAXIMUM
  ❶         ❷        ❸                            ❹     ❺           ❻     ❼      ❽        ❾       ❿
 GROUP1    1500     -22  SYS1       SYS1          80     3   600   NO      25     23        80      80
                         SYS2       SYS2          80     3   500   NO     100     46        80      80
 -----------------------------------------       -----------------------------------------------------
                                    TOTAL                6  1100
```

Figure 4.10.: RMF Group Capacity Report

The columns shown in the report are:

① **NAME** Name of the WLM capacity group

② **LIMIT** Capacity group limit

③ **AVAIL** Average unused capacity in MSU (average unused vector)

④ **MSU DEF** Defined capacity limit

⑤ **MSU ACT** Actual used capacity (on average for the RMF Interval)

⑥ **CAPPING DEF** Indicates whether this partition uses initial capping

⑦ **CAPPING WLM%** Percentage of time the LPAR was capped during the RMF interval

⑧ **CAPPING ACT%** Percentage of time when capping actually limited the usage of processor resources for the partition

⑨ **MINIMUM ENT.** Minimum of the GC member share and the DC limit

⑨ **MAXIMUM ENT.** Maximum of the GC limit and the DC limit

The field AVAIL shows the average Unused Capacity. If this field is negative, the group has the capping status. This means that each member of the group calculates its own capping limit and decides if it has to cap itself.

MSU DEF shows if a partition has a defined capacity limit in addition to its group limit. WLM caps always to the minimum of the DC limit and the calculated share of the group.

MSU ACT is the actual CPU consumption during this interval. Because this value is averaged over the RMF interval and not over the past four hours, this is not a reliable source to find out if the partition needs to be capped.

CAPPING DEF shows if initial capping is activated. If it is activated, the GC definitions of this partition on HMC are ignored by WLM, because GC can not coexist with initial capping.

WLM% shows the percentage of time, WLM activated the PR/SM capping. As described earlier in this book, WLM% is always 100% when phantom weight capping is used in the whole RMF interval. Be aware that WLM% tells you the percentage of time a partition was capped and it can not tell the reduction of capacity occurred because of the capping.

ACT% shows the percentage of time, the partition was impacted from the capping limit. When a partition is capped by cap pattern, ACT% should be close to WLM%. But if phantom weight capping is used, ACT% can have any value from 0 to 100 while WLM% is always 100 in this case. Figure 4.11 explains this metric.

MINIMUM ENTITLEMENT is the minimum of the weight based share of the group (MemberCapacity) and the DC limit. A group member should get at least this amount of capacity in average (given that it has the CPU demand). When all group members are demanding their share, they get the MINIMUM ENTITLEMENT.

MAXIMUM ENTITLEMENT is the minimum from the DC limit and the GC limit. A partition can not get more than this capacity in average. A partition gets the max. entitlement, when all other group members are idle.

4.11.2. Difference between ACT% and WLM%

Since the WLM% field caused some confusion to customers, RMF added an additional field to the reports which is named ACT%. This new field is especially useful when a partition is capped by WLM using the phantom weight method, since the WLM% field is always 100% in this case.

When WLM uses phantom weight to cap a partition, PR/SM capping is active all the time. Therefore WLM% is always 100% in this case. Since ACT% shows the percentage of time when the partition was impacted by capping, it is less than 100 when the partition drops below the limit sometimes.

In the example above, the partition is impacted by capping 60% of the time. The remaining 40% of the time the demand of the workload is below the limit and the partition therefore does not suffer from the capping. So if you see a small number in the ACT% field (or even zero), this partition is not impacted by the capping highly.

Note:
Even when ACT% is more meaningful for phantom weight than WLM%, it is still a percentage of time and it can not tell you the reduction of the CPU capacity occurred because of the capping.

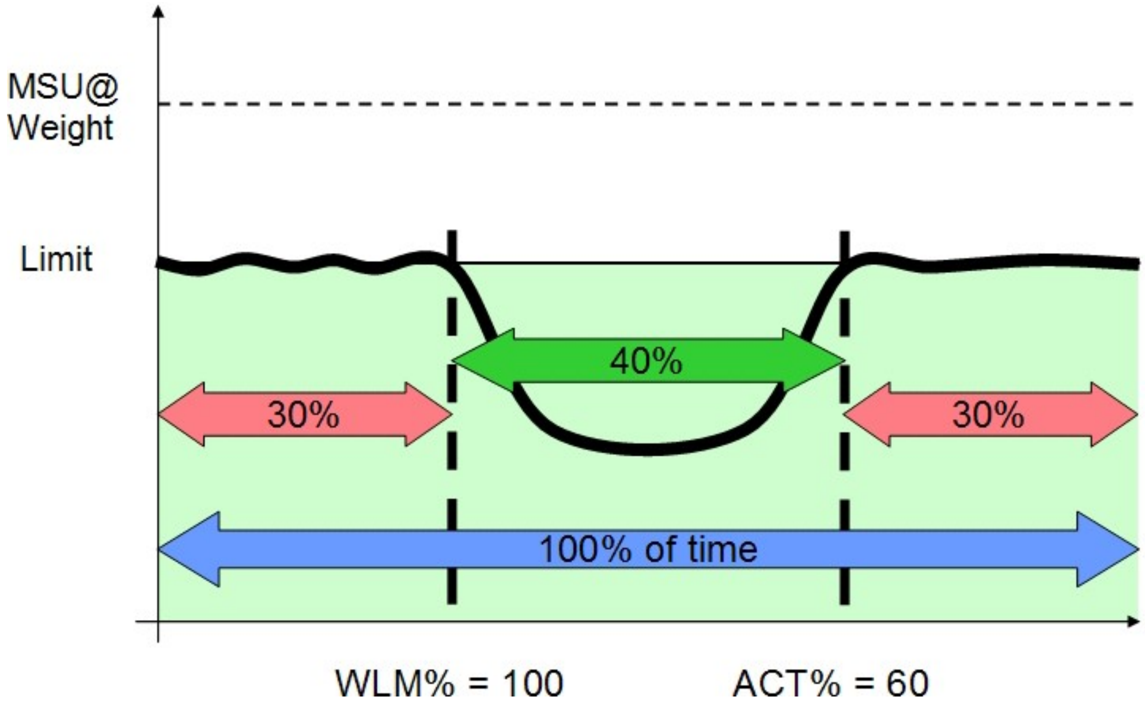
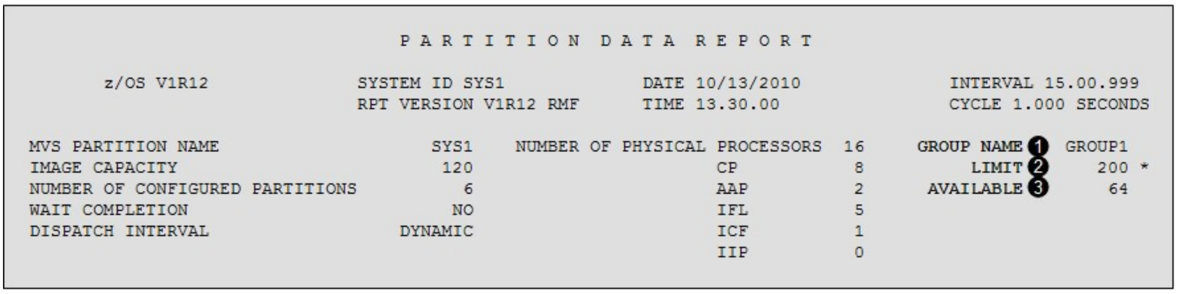


Figure 4.11.: Difference between ACT% and WLM%

4.11.3. RMF Partition Data Report

The header section of the Partition Data Section of the RMF Postprocessor CPU Activity Report has also group capacity information if the LPAR belongs to a capacity group. It is located on the right side of the report as can be seen in figure 4.12.



```
                                  P A R T I T I O N   D A T A   R E P O R T

        z/OS V1R12              SYSTEM ID SYS1            DATE 10/13/2010              INTERVAL 15.00.999
                                RPT VERSION V1R12 RMF     TIME 13.30.00                CYCLE 1.000 SECONDS

 MVS PARTITION NAME                     SYS1   NUMBER OF PHYSICAL PROCESSORS  16     GROUP NAME ❶  GROUP1
 IMAGE CAPACITY                          120                             CP    8          LIMIT ❷    200 *
 NUMBER OF CONFIGURED PARTITIONS           6                            AAP    2      AVAILABLE ❸     64
 WAIT COMPLETION                          NO                            IFL    5
 DISPATCH INTERVAL                   DYNAMIC                            ICF    1
                                                                        IIP    0
```

Figure 4.12.: RMF Partition Data Report

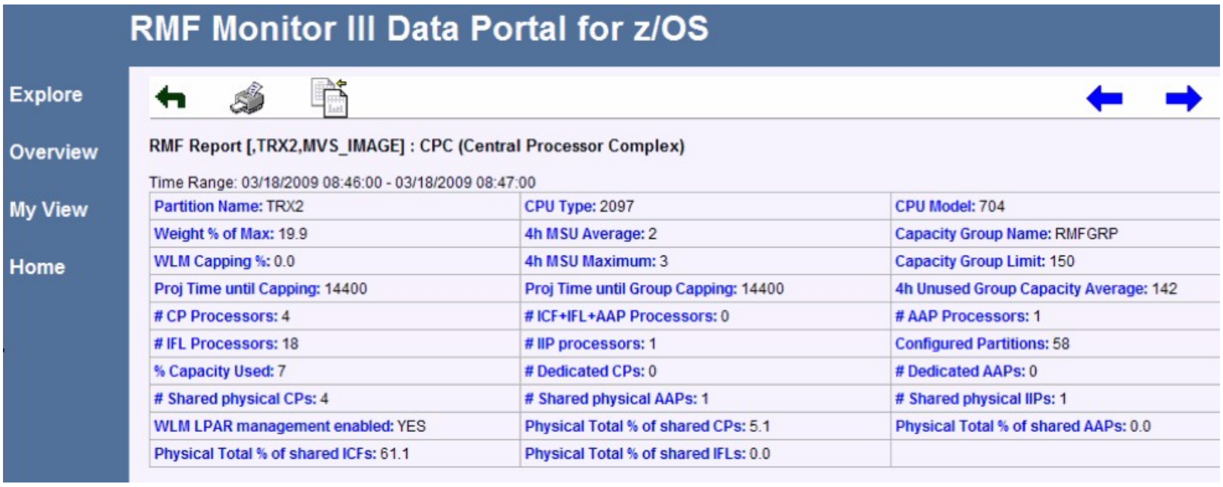
Beside the GROUP NAME ① and the group LIMIT ② you may see an asterisk behind the group limit. As explained before, an LPAR which is a group member for less than four hours, may not have the same group information than the other group members. If RMF recognizes that an LPAR is in its group for less than four hours, it will mark it with an asterisk.

The AVAILABLE ③ field is the average Unused Capacity which is described in section 4.4.2. As long as this field is positive the capacity group is not capped.

4.11.4. Remaining Time until Capping

The RMF Distributed Data Server (DDS) has performance metrics to forecast when capping will take place (See note below for additional information about DDS). This forecast is calculated by extrapolating the cpu consumption of the immediate past and verifying if this consumption would lead to capping within the next four hours.

Figure 4.13 shows the RMF Monitor III Data Portal. You can see the forecast fields and other useful fields related to group capping.



RMF Report [,TRX2,MVS_IMAGE] : CPC (Central Processor Complex)

Time Range: 03/18/2009 08:46:00 - 03/18/2009 08:47:00

Partition Name: TRX2	CPU Type: 2097	CPU Model: 704
Weight % of Max: 19.9	4h MSU Average: 2	Capacity Group Name: RMFGRP
WLM Capping %: 0.0	4h MSU Maximum: 3	Capacity Group Limit: 150
Proj Time until Capping: 14400	Proj Time until Group Capping: 14400	4h Unused Group Capacity Average: 142
# CP Processors: 4	# ICF+IFL+AAP Processors: 0	# AAP Processors: 1
# IFL Processors: 18	# IIP processors: 1	Configured Partitions: 58
% Capacity Used: 7	# Dedicated CPs: 0	# Dedicated AAPs: 0
# Shared physical CPs: 4	# Shared physical AAPs: 1	# Shared physical IIPs: 1
WLM LPAR management enabled: YES	Physical Total % of shared CPs: 5.1	Physical Total % of shared AAPs: 0.0
Physical Total % of shared ICFs: 61.1	Physical Total % of shared IFLs: 0.0	

Figure 4.13.: RMF CPC Report on RMF Data Portal

- Proj. Time until Group Capping:

  When the 4hr rolling avg. exceeds the group limit, the group gets into the capping status. This means that each group member calculates its own limit and compares it against its 4hr rolling avg. The field Proj. Time until Group Capping shows the time in seconds when RMF forecasts the group to reach its limit. When this field is 14400 (4h), RMF forecasts that the group will not reach its limit in the next four hours.

- Proj. Time until Capping:

  As soon as a capacity group is in capping status, each member calculates its group share and turns this share into a limit. The field Proj. Time until Capping shows the time in seconds, when RMF forecasts the LPAR will reach its limit. If an LPAR reaches its limit, it gets capped by cap pattern or by phantom weight.

Using the RMF Data Portal

RMF Monitor III Data Portal allows you to view RMF Monitor III performance data through your Web browser. To use this tool, you don't need to perform any special RMF setup or customization tasks. The core component of RMF Monitor III Data Portal is the RMF Distributed Data Server (DDS), which runs as a single server address space with sysplex wide scope. Once started, this server can retrieve performance data from any system within your sysplex. To begin using RMF Monitor III Data Portal, just follow these steps:

1. Start the DDS on one system in your sysplex: From a z/OS console, enter the START GPMSERVE command.
2. Display the RMF Monitor III Data Portal welcome panel: In the address bar of your Web browser, enter the host name or IP address of your sysplex, and port number 8803. DDS listens for incoming connections on this port. For more information about the Data Portal see [20].

4.11.5. Useful SMF Type 70 fields

SMF Field	Meaning	Description
SMF70GNM	Group Name	Name of the capacity group to which this partition belongs. Valid if bit 1 of SMF70PFL is set.
SMF70GMU	Group Limit	Maximum licensing units of a group. The maximum number of processor licensing units for the group of logical partitions of which this partition is a member, and which may be consumed per unit of time, on average. Valid if bit 1 of SMF70PFL is set.
SMF70LAC	4 HRAVG	Long-term average of CPU service (millions of service units). Scope of the value depends on bit 3 of SMF70STF.
SMF70GAU	Avg. Unused GC	Long-term average of CPU service in millions of service units which would be allowed by the limit of the capacity group but is not used by its members. If the value is negative, the capacity group is subject to capping. Valid if bit 7 of SMF70STF is set.
SMF70GJT	Group Joined	Time in STCK format when the partition that wrote this record has joined or left a capacity group (last change of group name). Also set at IPL time, when the partition is not a member of a capacity group.
SMF70VPF	capping is active	Bit 3: Partition capping is enabled.
SMF70NSW	WLM%	Number of Diagnose samples where WLM considers to cap the set of logical CPUs of type SMF70CIX within the logical partition (see also SMF70NCA).
SMF70NCA	ACT%	Number of Diagnose samples where capping actually limited the usage of processor resources for the set of logical CPUs of type SMF70CIX within the logical partition.
SMF70PFL	Flags	Bit 1: This partition is member of a capacity group.
SMF70VPF	Capping active	Bit 3: Partition capping is enabled.
SMF70PMA	phantom weight	Average adjustment weight for pricing management.

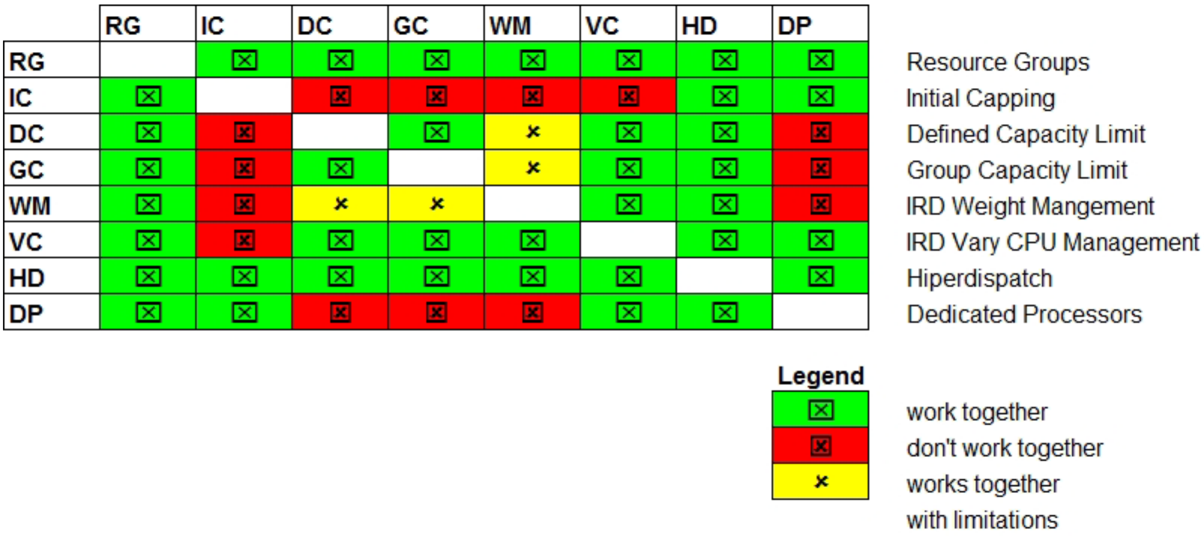
Table 4.2.: SMF Type 70 Fields for Group Capacity

For SMF 72 data please refer to section 3.9.4.

4.11.6. Diagnoses

The WLM group capacity algorithm is writing SMF99 subtype 11 records which are useful for debugging group capacity problems. IBM support personnel may ask you to collect them for analysis. Also storage dumps including WLM address space and common storage areas may be needed to analyze group capacity related problems.

4.12. Compatibility between LPAR controls



	RG	IC	DC	GC	WM	VC	HD	DP	
RG		🟩	🟩	🟩	🟩	🟩	🟩	🟩	Resource Groups
IC	🟩		🟥	🟥	🟥	🟥	🟩	🟩	Initial Capping
DC	🟩	🟥		🟩	🟨	🟩	🟩	🟥	Defined Capacity Limit
GC	🟩	🟥	🟩		🟨	🟩	🟩	🟥	Group Capacity Limit
WM	🟩	🟥	🟨	🟨		🟩	🟩	🟥	IRD Weight Mangement
VC	🟩	🟥	🟩	🟩	🟩		🟩	🟩	IRD Vary CPU Management
HD	🟩	🟩	🟩	🟩	🟩	🟩		🟩	Hiperdispatch
DP	🟩	🟩	🟥	🟥	🟥	🟩	🟩		Dedicated Processors

Legend

🟩	work together
🟥	don't work together
🟨	works together with limitations

Figure 4.14.: Compatibility between LPAR controls

Figure 4.14 summarizes the possibilities to combine WLM and LPAR capping functions. While resource groups which are a z/OS internal control can be combined with all other mechanisms, some of the capping controls which require PR/SM involvement exclude each other.

A. TRADEMARKS

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

AIX®, AS/400®, BatchPipes®, C++/MVS, CICS®, CICS/MVS®, CICSPlex®, COBOL/370, DB2®, DB2 Connect, DB2 Universal Database, DFSMS/MVS®, DFSMSdfp, DFSMSdss, DFSMShsm, DFSORT, e (logo)®, ECKD, ES/3090, ES/9000®, ES/9370, ESCON®, FICON, GDPS, Geographically Dispersed Parallel Sysplex, HACMP/6000, Hiperbatch, Hiperspace, IBM®, IBM (logo)®, IMS, IMS/ESA®, Language Environment®, Lotus®, OpenEdition®, OS/390®, Parallel Sysplex®, PR/SM, pSeries, RACF®, Redbooks, RISC System/6000®, RMF, RS/6000®, S/370, S/390®, S/390 Parallel Enterprise Server, System/360, System/370, System/390®, System z, ThinkPad®, UNIX System Services, VM/ESA®, VSE/ESA, VTAM®, WebSphere®, xSeries, z/Architecture, z/OS, z/VM, zSeries

The following are trademarks or registered trademarks of other companies:

- 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 in the United States, other countries, or both.
- 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.
- Red Hat, the Red Hat ”Shadow Man” logo, and all Red Hat-based trademarks and logos are trademarks or registered trademarks of Red Hat, Inc., in the United States and other countries.
- SET and Secure Electronic Transaction are trademarks owned by SET Secure Electronic Transaction LLC.

B. GLOSSARY

A

APPC *Advanced Program to Program Communication* **-** Protocol for program to program communication between multiple systems

APPL% *Percent of Application Utilization* **-**Converted application time into an overall percent of a single CPU used for the workload

ASCB *Address Space Control Block* **-** z/OS control block which represents a virtual storage entity tight to an end user or set of programs to execute

B

BCP *Basic Control Program* **-** z/OS or MVS kernel routines

C

CCW *Channel Command Word* **-** Defines an I/O operation (read, write, control) to be performed on an I/O device

CDS *Couple Dataset* **-** Dataset which contains control information to setup a parallel sysplex environment

CEC *Central Electronic Complex* **-** The system (processors, memory, I/O adapters), not including I/O devices

CFCC *Coupling Facility Control Code* **-** Operating System of the coupling facility

CHPID *Channel Path Identifier* **-** Identification of the channel path, typically a number

CICS *Customer Information Control Server* **-** A transaction monitor that runs primarily on z/OS

CISC *Complex Instruction Set Computing* **-** Processing architecture which contains many complex instructions which perform functions like small programs

CKD *Count Key Data* **-** System z disk architecture

CP *Central Processor* **-** Standard processor of a System z

CPU *Central Processing Unit* **-** see CP

CSS *Channel Subsystem* **-** The heart of moving data in and out of of a mainframe

CSSID *Channel Subsystem Identifier* **-** Number which identifies the Channel Subsystem in case multiple exist

D

DASD *Direct Access Storage Device* **-** A storage device which supports direct access (typically a disk)

DB2 *Database* **-** Relational database based on E. F. Codd's theory of relational databases

DDF *Distributed Data Facility* **-** Component of DB2 to exchange information with external clients

DEDB *Data Entry Database* **-** Fast path database for IMS

DL/I *Data Language Interface* **-** Database of IMS

DRDA *Distributed Relational Database Architecture* **-** Distributed database architecture of the open group standard

E

ECKD *Extended Count Key Data* **-** incorporates fixed-block and CKD architecture

EMIF *ESCON Multiple Image Facility* **-** Feature which allows to use ESCON channels from multiple partitions

ESA/390 *Enterprise Systems Architecture/390* **-** 32-bit predecessor of System z architecture

ESCON *Enterprise System Connection* **-** Half-duplex optical fiber serial channel

ESPIE *Extended Specify Program Interruption Exit* **-** Interrupt exit routine

ESTAE *Extended Specified Task Abnormal Exit* **-** Recovery routine for z/OS user or problem state programs

ESTI *Enhanced Self-Timed Interface* **-**

ETR *External Time Reference* **-** Device to synchronize all TOD (time-of-day) clocks in a cluster environment (Parallel Sysplex)

EXCP *Execute Channel Program* **-** z/OS macro to execute an I/O operation

F

FCP *Fibre Channel Protocol* **-** Transport protocol for transporting SCSI commands on Fibre Channel networks

FICON *Fibre Channel Connection* **-** Full-duplex fibre optical serial channel

FIFO *First In, First Out* **-** Queuing mechanism

FLIH *First Level Interrupt Handler* **-** Interrupt handler that gets immediate control when the interrupt occurs (where the new Program Status Word points to)

FRR *Functional Recovery Routine* **-** Recovery routine for z/OS system programs

G

GDPS *Global Dispersed Parallel Sysplex* **-** Parallel Sysplex which is spatially distributed to ensure high availability

GRS *Global Resource Serialization* **-** z/OS subsystem which supports global lock management

H

HCD *Hardware Configuration Dialog* **-** z/OS component to define I/O devices to the system

HFS *Hierarchical File System* **-** UNIX file system on z/OS

HMC *Hardware Management Console* **-** Console to access and manage hardware components of System z

HWA *Hardware Address* **-**

I

I/O *Input/Output* **-** Abbreviation for all parts which send data to and from an electronic complex

ICB *Integrated Cluster Bus* **-** Bus for connecting system in a parallel sysplex for short distance. The bus relies on few parts and provides very high speed and reliable connectivity

ICF *Integrated Coupling Facility* **-** Processor on system z which allows to run coupling facility control code

IFL *Integrated Facility for Linux* **-** Processor on system z which allows to execute z/VM and Linux operating systems

IML *Initial Microcode Load* **-** Initialization process of System z hardware. At its completion, operating systems can be booted (IPLed).

IMS *Information Management System* - A transaction monitor and database for z/OS (introduced 1968 for the Apollo space program)

IOCDS *Input/Output Configuration Data Set* - Data set which contains hardware I/O definitions (related to IODF; also should be consistent with IODF)

IODF *I/O Definition File* - Data file which contains the I/O definitions for System z (created by HCD)

IPC *Inter Process Communication* - Protocol for system processes to interact which each other

IPL *Initial Program Load* - Process to start the z/OS operating system

IRB *Interrupt Request Block* - z/OS Control Structure to start an I/O routine

IRD *Intelligent Resource Director* - A combination of multiple z technologies to enhance the autonomic capabilities of PR/SM, z/OS and the I/O subsystem

IRLM *IBM Resource Lock Manager* - Lock manager for DB2 and IMS

ISPF *Interactive System Productivity Facility* - End user interface for TSO users

J

JES *Job Entry System* - z/OS subsystems which support the execution of scheduled programs

L

LCP *Logical Processor* - Representation of a processor to the virtual system or logical partition

LCSS *Logical Channel Subsystem* - A system may use multiple logical channel subsystems (currently up to 4) to increase connectivity

LDAP *Lightweight Directory Access Protocol* - Application protocol for accessing and maintaining distributed directory information services over an IP network

LIC *Licensed Internal Code* - System z microcode or firmware

LICCC *Licensed Internal Code Configuration Control* -

Ln *Level n Cache* - L1 is closest to the processor, the highest number is used to describe memory (”storage” in System z terminology)

LPAR *Logical Partition* - Container which hosts an operating system to execute on System z virtualization layer. Up to 60 LPARs are supported

M

MBA *Memory Bus Adapter* - I/O hub chip used on z10 and earlier machines. No longer used on z196.

MCM *Multi-Chip Module* - contains processor and storage controller chips

MCSS *Multiple-logical-Channel Subsystem* - Restructuring of the physical CSS into multiple logical instances in order to enable more external devices to be attached to the CEC

MLCS *Multiple Logical Channel Subsystems* - see MCSS

MMU *Memory Management Unit* - Hardware component which handles virtual memory

MPL *Multi Programming Level* - Term which expresses the ability of workload to access system resources

MSU *Million of Service Units per Hour* - Unit to measure CPU capacity on System z

MTTW *Mean Time To Wait* - Algorithm which gives access to units of work based on their deliberate wait time

MVS *Multiple Virtual Storage* - Original name of z/OS based on the ability to support multiple applications in virtual storage

O

OLTP *Online Transaction Processing* - Umbrella term for transaction processing

OSA *Open System Adapter* - Networking adapter

P

PAV *Parallel Access Volume* - Protocol which supports parallel access to the same I/O device

PCHID *Physical Channel Identifier* - identifies the plug position of a channel adapter

PCP *Physical Processor* - see CP

PPRC *Peer to Peer Remote Copy* - A protocol to replicate a storage volume to a remote site

PR/SM *Process Resource and System Manager* - Management component of the logical partition technology of System z (alias for LPAR hypervisor)

PSW *Program Status Word* - Central register to control all program execution

PU *Processing Unit* - Physical processor

Q

QDIO *Queued Direct I/O* - memory to Memory I/O mechanism between LPARs on System z

R

RACF *Resource Access Control Facility* - z/OS subsystem which supports access control

RAS *Reliability, Availability, Serviceability* - Terminology to depict the robustness of information technology systems (originated from IBM mainframe)

RETAIN *Remote Technical Assistance Information Network* - IBM network to handle service requests for end users

REXX *Restructured Extended Executor* - Interpretive Execution Language from IBM

RISC *Reduced Instruction Set Computing* - Processing architecture which only contains elementary instructions like LOAD, STORE, and register-to-register operations

RLS *Record Level Sharing* - VSAM access method which introduces record sharing and serialization

RMF *Resource Measurement Facility* - z/OS Performance Monitor

RRMS *Resource Recovery Management Services* - z/OS component to synchronize the activities of various syncpoint managers

RSF *Remote Support Facility* - Part of HMC to report and repair hardware and firmware components

S

S/360 *IBM System/360* - Is a mainframe computer system family announced by IBM on April 7, 1964. It is the computer architecture of which System z is the current incarnation.

SAP *System Assist Processor* - System z I/O processor

SCC *Storage Controller Control* - Storage controller chip

SCD *Storage Controller Data* - Cache chip

SCE *Storage Control Element* - Controls access to main storage data by processor unit

SDWA *System Diagnostic Work Area* - Control structure to capture information in case of an abnormal program termination

SE *Support Element* - Laptop that acts as user interface to System z machine

SIE *Start Interpretive Execution* - Instruction to drive a processor in a logical partition (LPAR) or virtual machine (z/VM)

SIGP *Signal Processor* - Instruction to inform a processor about status change

SLE *Session Level Encryption* - Encryption between originator and receiver of data across all network elements

SLIH *Second Level Interrupt Handler* - Term which encompasses a set of specialized interrupt handling routines

SMF *Systems Management Facility* - z/OS component which supports performance and status logging

SMP *Symmetric Multiprocessing* - A computer system with all physical processors accessing the same storage and I/O subsystems

SRB *Service Request Block* - Control structure to execute a z/OS system program

SRM *System Resource Manager* - Component of z/OS for resource management (introduced 1974, now part of WLM)

STP *Server Time Protocol* - Follow-on to ETR

SU/sec *Service Unit per second* - Capability of a System z processor to execute instructions

SVC *Supervisor Call* - Interface to invoke a z/OS system program

Sysplex *System Complex* - A single logical system running on one or more physical systems

System z *IBM mainframe computer brand* - Current 64-bit incarnation of the S/360 architecture

T

TCB *Task Control Block* - Control Structure to execute user or problem state programs on z/OS

TSO *Time Sharing Option* - z/OS component which supports the parallel execution of multiple end users on MVS

U

UCB *Unit Control Block* - z/OS control structure which represents an I/O device

UoW *Unit of Work* - An execution unit on z/OS

USS *Unix System Services* - z/OS component which supports a full functioning UNIX environment on z/OS

V

VCPU *Virtual CPU* - see LCP

VMM *Virtual Machine Monitor* - Hypervisor or control program to run multiple virtual machines

VSAM *Virtual Storage Access Method* - A set of access methods for System z I/O devices

VTAM *Virtual Terminal Access Method* - Access method for communications devices (now part of z/OS TCPIP subsystem)

VTOC *Volume Table of Content* - Index of a DASD device

W

WLM *Workload Manager* - Central z/OS component for resource management (introduced 1995)

X

XCF *Cross System Coupling Services* **-** z/OS Services which support the exploitation of a z/OS sysplex

XES *Cross System Extended Services* **-** z/OS services which support the access to the coupling facility

XRC *Extended Remote Copy* **-** System z protocol for data replication

Z

z114 *zEnterprise 114* **-** Mid-range end model of System z processor family (2011)

z196 *zEnterprise 196* **-** High end model of System z processor family (2010)

zEC12 *zEnterprise EC12* **-** High end model of System z processor family (2012)

zAAP *System z Application Assist Processor* **-** System z processor to execute Java code. This processor type can only be used by z/OS and only for instrumented software like the Java Virtual Machine. A special instruction tells the dispatcher when Java execute starts and ends.

zFS *System z File System* **-** UNIX file system on z/OS

zIIP *System z Integrated Information Processor* **-** System z processor to execute code which is subject to get offloaded from regular processors. The offload capability is described by the middleware through an interface to WLM and the z/OS dispatcher. Exploiters are middleware like DB2 and TCPIP.D5

Bibliography

[1] *TSO Time Sharing Option im Betriebssystem z/OS*, Dr. Michael Teuffel, Oldenbourg, 2002, ISBN-13: 978-3486255607

[2] *Das Betriebssystem z/OS und die zSeries: Die Darstellung eines modernen Großrechnersystems*, Dr. Michael Teuffel, Robert Vaupel, Oldenbourg, 2004, ISBN-13: 978-3486275285

[3] *High Availability and Scalability of Mainframe Environments*, Robert Vaupel, KIT Scientific Publishing, 2013, ISBN-13: 978-3-7315-0022-3

[4] *In Search Of Clusters, The Ongoing Battle in Lowly Parallel Computing*, Gregory F. Pfister, Prentice Hall, 1998, ISBN 0-13-899709-8

[5] *Adaptive Algorithms for Managing A Distributed Data Processing Workload*, J. Aman, C.K. Eilert, D. Emmes, P. Yocom, D. Dillenberger, IBM Systems Journal, Vol. 36, No. 2, 1997, Seiten 242-283

[6] *MVS Performance Management (ESA/390 Edition)*, Steve Samson, J. Ranade IBM Series, Printed and bound by R.R.Donnelley and Sons Company, ISBN 0-07-054529-4, 1992

[7] *z/OS Workload Manager - How it works and how to use it* Robert Vaupel, March 2014, 3rd edition http://www-03.ibm.com/systems/z/os/zos/features/wlm/WLM_Further_Info.html

[8] *Resource Groups and how they work* Dieter Wellerdiek, 2008, http://www-03.ibm.com/systems/z/os/zos/features/wlm/WLM_Further_Info.html

[9] *ABC of z/OS System Programming*, IBM Redbooks, Volume 11, SG24-6327-xx

[10] *OS/390 MVS Parallel Sysplex Capacity Planning*, IBM Redbook, SG24-4680-01, January 1998

[11] *z/OS MVS Planning: Workload Management*, z/OS Literatur, SA22-7602-xx

[12] *System's Programmer Guide to: Workload Management*, IBM Redbook, SG24-6472-xx

[13] *z/OS MVS Programming: Workload Management Services*, z/OS Literatur, SA22-7619-xx

[14] *z/OS Resource Measurement Facility: Performance Management Guide*, z/OS Literatur, SC28-1951-xx

[15] *z/OS Basic Skills Center*, http://publib.boulder.ibm.com/infocenter/zos/basics/index.jsp

[16] *IBM System z Software Pricing*, http://www-03.ibm.com/systems/z/resources/swprice/

[17] *CICS Transaction Server for z/OS: Performance Guide*, IBM Literatur, Various numbers please check your CICS version and release

[18] *CICS Transaction Server for z/OS: Customization Guide*, IBM Literatur, Various numbers please check your CICS version and release

[19] *IMS System Administration Guide*, IBM Literatur, Various numbers please check your IMS version and release

[20] *z/OS Hot Topics Newsletter #16*, http://publibz.boulder.ibm.com/epubs/pdf/e0z2n171.pdf

Index

CEC, 24
CPC, 24

DC, 13
Defined Capacity
- 4HRAVG, 16
- Cap Pattern, 17
- Capping, 14
- Capping Effects, 20
- Capping Methods, 17
- HMC, 22
- How does it work, 14
- Monitoring, 22
- Phantom Weight, 19
- Requirements, 13
- Small LPARs, 20

Defined Capacity Limit, 13
- Concepts, 13

Defined Capacitys
- How does it work, 14

Group Capacity, 26
- Advantages, 27
- Available Capacity, 30
- Capping, 30
- Capping a Group Member, 31
- Concepts, 26
- Data Collection, 28
- Groups, 28
- HMC, 36
- IPL Bonus, 33
- Monitoring, 38
- Requirements, 27
- Scenarios, 32
- Scope, 26
- Share, 30
- Unused Capacity, 28
- Unused Vector, 28

Group Capping, 26

MSU, 14
- hardware, 14
- software, 14

REQLPDAT, 37
Resource Groups, 8
- Concepts, 8
- Example, 9
- Limitations, 12
- Limits, 10
- Maximum Limit, 11
- Minimum Limit, 11
- Requirements, 8
- Resource Consumption, 11
- Scenarios, 12
- When to use, 14
- Why use them, 9

RG, 8
RMF
- Data Portal, 41
- Group Capacity Report, 38
- Partition Data Report, 22
- Partition data Report, 40
- SMF Type 70, 24
- SMF Type 72, 25

RMF Monitor III
- CPC Capacity Data Report, 23

SMF, 24, 25
Software Pricing, 20, 31

Trademarks, 44

WLM Capping Technologies, 6, 8, 13–15