SEGUS

White Paper

Managing z/OS costs with capping: what's new with zEC12 GA2 and z/OS 2.1?

Fabio Massimo Ottaviani - EPV Technologies

1 Introduction

In the current volatile economic environment, companies want more IT support for their operation—but at a lower cost than before. Even after years of shrinking budgets, there is still a lot of pressure on IT managers to reduce personnel, hardware and software costs.

At any z/OS site, hardware costs are mostly driven by installed CPU capacity, whilst software costs are mostly driven by installed CPU capacity (OTC) and CPU used (MLC).
CPU utilization measurement and control is therefore the key factor in reducing costs.

This is the reason why every company collects a lot of different measurements related to CPU utilization; the mandatory input to tuning and capacity planning activities.

In addition, (and sometimes as an alternative), to tuning and capacity planning, the following control mechanisms can also be exploited to manage CPU utilization with the goal of reducing costs:

- hardware capping;
- software capping (defined capacity and group capacity).

In this paper we will discuss some updates to these control mechanisms, introduced with zEC12 GA2 machines and z/OS 2.1. (partially available through PTF to z/OS 1.13 and 1.12), designed to address important issues of the current hardware and software capping implementation.

2 Hardware capping

Hardware capping, also called initial capping, is the oldest available capping technique. It exploits the PR/SM capping function to limit the processing capacity used by one or more LPARs.
It applies to any processor pool (CPU, zAAP and zIIP) independently.

A capped LPAR, running at its cap, does not have access to the resources that are not utilized by other LPARs, while resources that are not used by a capped LPAR can also be used by other LPARs.

There are several reasons why you would want to use hardware capping; the most common ones are:

- to limit capacity (and performance) to that specified in an outsourcing service contract;
- to limit the effect of loops in development LPARs;
- to limit capacity used during stress tests.

Hardware capping is based on LPAR relative weight. Starting from the LPAR initial weight a relative weight (% Weight), corresponding to the machine share of each LPAR, is calculated.

Let's consider the simple LPAR configuration in Figure 1 where hardware capping is not in place.

- LPAA has a 60% machine share but a workload Demand corresponding to only 50%. So LPAA will use 50% (% Used) of the machine;
- LPAB has a 30% machine share and a workload Demand corresponding to 30%. So LPAB will use 30% (% Used) of the machine;
- LPAC has a 10% machine share but a workload Demand corresponding to 20%. So LPAC will use 20% (% Used) of the machine also getting the LPAA unused capacity.

LPAR	Active	Weight	%Weight	Demand	Hcap	%Used
LPAA	YES	600	60%	**50%**	NO	50%
LPAB	YES	300	30%	30%	NO	30%
LPAC	YES	100	10%	**20%**	NO	**20%**
TOTAL		1,000	100%	100%		100%

Figure 1

Now suppose that hardware capping is activated for LPAC.
You can see in Figure 2 on the next page, that nothing changes for LPAA and LPAB but LPAC can only use 10% of the machine because it is now capped at its % Weight.
One side effect is that the machine utilization is lower, because LPAA unused capacity remains unused.

LPAR	Active	Weight	%Weight	Demand	Hcap	%Used
LPAA	YES	600	60%	**50%**	NO	50%
LPAB	YES	300	30%	30%	NO	30%
LPAC	YES	100	10%	**20%**	**YES**	**10%**
TOTAL		1,000	100%	100%		**90%**

Figure 2

2.1 Problem

The weakness of this mechanism is that, if for any reason you deactivate an LPAR, all remaining relative weights (% Weight) increase, and capped LPARs can get more capacity than desired.

As you can see in Figure 3 below, LPAB has been deactivated. The consequence is that LPAC is allowed to use 14% of the machine instead of 10%.

LPAR	Active	Weight	%Weight	Demand	Hcap	%Used
LPAA	YES	600	60%	**50%**	NO	50%
LPAB	NO					
LPAC	YES	100	10%	**20%**	**YES**	**14%**
TOTAL		700	100%	100%		**64%**

Figure 3

2.2 Solution

zEC12 GA2 introduces an "absolute capping limit" in 1/100ths of a processor which can be specified independently from the LPAR weight and, therefore, it is insensitive to capacity changes or LPAR (de)activations. The absolute capping limit used is provided in the SMF70HW_Cap_Limit field of the SMF 70 record.

Another advantage of this new capping option is that, unlike initial capping, it may be used concurrently with defined and group capacity management[1].

Absolute capping is available on z/OS 2.1 natively; it is also available on z/OS 1.12 and 1.13 after applying the OA41125 maintenance.

1 WLM will use the minimum among absolute capping, defined capacity and group capacity limits when all capping types are in effect. The value used will be provided in the SMF70WLA field of the SMF 70 record.

3 Defined capacity

Defined capacity is a limit set to the capacity, measured by the MSU 4-hour rolling average, which can be used by an LPAR.
Defined capacity is only applied to CPU (not zAAP, or zIIP).

Probably the only reason you would want to use software capping, is to reduce the monthly software bill.

CPU usage can spike above the defined and group capacity limit as long as the MSU used in the 4-hour rolling average remains below it. When the 4-hour rolling average exceeds the defined capacity limit, WLM will signal PR/SM to cap the LPAR by using the hardware capping function—which is based on the relative weight.

Different techniques are used to cap an LPAR partition depending on the value of MSU at WEIGHT (%Weight * CEC MSU) and DEF MSU (defined capacity MSU limit):

1) If MSU at WEIGHT = DEF MSU, the standard hardware capping technique is used;
2) If MSU at WEIGHT > DEF MSU, a phantom weight is automatically created and used; (a phantom weight is needed to reduce MSU at WEIGHT in order to avoid capping the LPAR at a higher value than DEF MSU);
3) If MSU at WEIGHT < DEF MSU, WLM defines a cap pattern that repeatedly applies and removes the cap at LPAR weight (a cap pattern is needed in order to avoid capping the LPAR at a lower value than DEF MSU).

3.1 Problem

The third case, based on a cap pattern, can be an issue for application performance—especially if MSU at weight is much smaller than the defined capacity limit. An example is provided in Figure 4 on the next page.

In this case the MSU at weight is so much smaller than the defined capacity limit, that the LPAR will be practically stopped in some intervals, causing a pulsing effect and making application performance erratic.

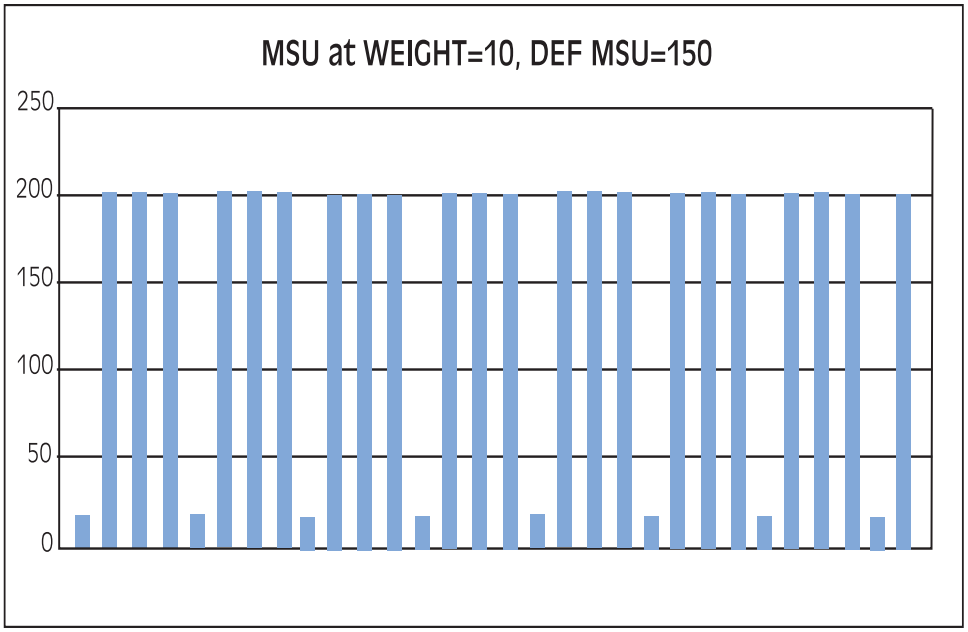

Figure 4

3.2 Solution

zEC12 GA2 introduces the possibility for WLM to automatically use a negative phantom weight instead of a cap pattern when MSU at WEIGHT < DEF MSU.

Negative phantom weight is available only on z/OS 2.1.

4 Group capacity

The monthly software bill is calculated by taking into account all the LPARs of a CEC, so setting defined capacity limits on one LPAR will not guarantee money savings.
However, if the sum of the limits is less than the total CEC capacity, then setting defined capacity limits on all the LPARs in a CEC will save money—but it could also greatly reduce CPU sharing flexibility.
This is precisely the reason why IBM introduced group capacity to limit the CPU globally used in a CEC.
Its goal is to guarantee the expected savings—while maintaining the required flexibility to maximize resource usage.

Based on the current weight, WLM assigns minimum and maximum entitlement values to each LPAR (see Figure 5 below[2]). It is worth noting that, because group capacity and defined capacity can be used together, the LPAR entitlement calculation takes into account the existence of a defined capacity limit.
If the group has to be capped because the group capacity limit has been reached, the LPARs above their minimum entitlement will also be capped.

Minimum Entitlement; it is the guaranteed MSU share the LPAR can get when in contention; it is calculated as follows: MIN((WGT x GROUP MSU / SUM(WGT)), DEF MSU) if DEF MSU GT 0
Maximum Entitlement; it is the maximum MSU share the LPAR can get; it is calculated as follows: MIN(DEF MSU, GROUP MSU) if DEF MSU GT 0

Figure 5[2]

4.1 Problem

IRD (Intelligent Resource Director) weight management, starting from the initial LPAR weights, manages the current weight in order to shift capacity within an LPAR cluster.
IRD weight management can be combined with group capacity, but it gets suspended when capping is in effect to avoid conflicts between the two algorithms.

The problem is that the LPAR entitlement within a capacity group is currently derived from the current weight, which could have been lowered by IRD. So when IRD is suspended, the LPAR might get "stuck" at a lower weight and, consequently, at a lower entitlement for a long time.

4.2 Solution

On zEC12 GA2, the initial LPAR weight will be used for group capacity entitlement instead of the current weight. It will result in more predictable and more controllable LPAR entitlement within a capacity group when IRD weight management is also used.

This enhancement is available only if all the systems in a capacity group are z/OS 2.1, or z/OS 1.12 and 1.13 with OA41125 maintenance applied.

2 This information comes from the *EPV for z/OS* Help System

5 Conclusions

Hardware and software capping are relevant components of the strategy implemented at many sites to contain or even reduce z/OS costs.

In this paper we discussed some updates to these control mechanisms, introduced with zEC12 GA2 machines and z/OS 2.1 (partially available through PTF to z/OS 1.13 and 1.12), which could make that strategy more effective.

EPV for z/OS

EPV for z/OS Resource provides a complete view of the "health" of all critical resources—especially those shared amongst different z/OS systems, such as processors (including zAAP, zIIP and crypto), disks, physical control units, coupling facilities and channels.

EPV for z/OS WLC supports the IBM software costs policy, providing all metrics necessary to manage the MSU utilisation—both at system and sublevel.

EPV for z/OS Workload provides a complete view of all of the workloads running on the z/OS systems.

EPV for z/OS Trend provides daily and monthly productivity and resource consumption trends at the system and workload levels.

EPV for z/OS Configuration provides detailed reports of the hardware and software configuration, including the total DASD space by provider and physical control unit.

SEGUS Inc is the North American distributor for EPV products

For more information regarding EPV for z/OS, please visit www.segus.com or call (800) 327-9650