VWLC Evolution: Soft-Capping, Group Capacity Limit & AutoSoftCapping

Jacky Hofbauer – zCost Management – jhof@zcostmanagement.com

White Paper

VWLC Introduction

VWLC (Variable Workload License Charges) is primarily an invoicing system for IBM z/OS based on software consumption, or "Pay for what you use". It is in line with IBM's Mainframe Charter (started at the beginning of the 2000's). The goal was to increase the attractiveness of the System z platform.

VWLC is based on the CPU usage in MSU instead of the CPC full capacity. In order to do so, IBM created the 4-hour rolling average: R4HA – WLM computes the average of the 48 last measures of instantaneous MSU (IMSU or ACTMSU, the real consumption).

To allow more software billing management, IBM also created the Soft-Capping feature by creating a resource limit named "Defined Capacity" (DC).

The rule of Soft-Capping is: "When the Rolling 4 Hours (R4HA) becomes superior or equal to the DC then the LPAR is capped". This means that the IMSU consumption will not be able to exceed the DC until the R4HA becomes inferior to the DC again. This action is performed by PR/SM and is shown by the WLM Capping% metric.

How does Soft-Capping work?

The WLM Capping% represents the percentage of time during which the LPAR was soft-capped during the interval. For example, if the SMF/RMF interval is 15 minutes and one LPAR has been capped 15 minutes, the WLM Capping% corresponding to that interval for this LPAR will be 100%.

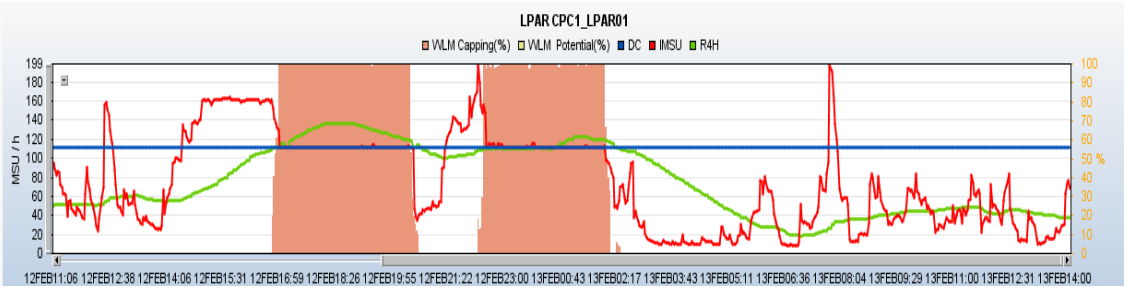

Figure 0

The R4HA (SMF70LAC) and the WLM Capping% (SMF70NSW) are written but not computed by RMF or CMF. WLM is the tool which computes the R4HA & the WLM Capping%; so it is WLM (and not RMF or CMF) which asks PR/SM to cap or uncap a LPAR.

Note: When a system IPLs, the 4 hour rolling average is calculated over too short of a time period. This can cause the system to be capped during IPL. This is described by IBM via an APAR:

http://www-01.ibm.com/support/docview.wss?uid=isg1OW55509

Jacky Hofbauer – zCost Management – jhof@zcostmanagement.com

How is MSU usage for the Software bill calculated?

The SCRT (Sub Capacity Reporting Tool by IBM) computes the customer's bill on a monthly basis per CPC (from the second day of the month at 00:00 to the first day of the following month 24:00). The process builds a MSU usage chart sorted by hour (E.g. there are 720 hours in one 30 day month) and by LPAR, by inserting in each cell the average of the R4HA (SMF70LAC) for one hour or the average of the Defined Capacity (SMF70MSU) for one hour; the retained value will be the lowest MSU usage metric. This metric will be retained for each product running on the LPAR (the SMF record 89 provide the product repartition by LPAR).

Figure 1 is an example involving a CPC with 4 LPARs, where all z/OS software products are running on the 4 LPARs and where the SMF interval is 15 minutes. The SCRT will retain hour 2 (left) as consumption peak without Soft-Capping (Pictured Left) and hour 1 with Soft-Capping (Pictured Right):

Hour/ LPAR	Hour 1	Hour 2	...	H 720
LP1	51/53/52/49 Av = 51,25 ➔ 51.25 Value retained to compute SCRT	48/53/52/55 Av = 52 ➔ 52	...	50/48/46/45 Av = 47,25 ➔ 47.25
LP2	70/68/69/74 ➔ 70.25	72/68/69/74 ➔ 70.75	...	75/72/67/71 ➔ 71.25
LP3	81/83/82/79 ➔ 81.25	80/84/84/82 ➔ 82.5	...	82/79/80/81 ➔ 80.5
LP4	11/13/12/15 ➔ 12.75	11/13/12/11 ➔ 11.75	...	13/11/13/12 ➔ 12.25
Total	**215.5**	**217**	...	**211.25**

Hour/ LPAR	Hour 1	Hour 2	...	H 720
LP1 DC=50	51/53/52/49 Av = 51,25 > DC ➔ 50 Value retained to compute SCRT	48/53/52/55 Av = 52 > DC ➔ 50	...	50/48/46/45 Av = 47,25 < DC ➔ 47.25
LP2 DC=69	70/68/69/74 ➔ 69	72/68/69/74 ➔ 69	...	75/72/67/71 ➔ 69
LP3 DC=81	81/83/82/79 ➔ 81	80/84/84/82 ➔81	...	82/79/80/81 ➔ 80.5
LP4 DC=12	11/13/12/15 ➔ 12	11/13/12/11 ➔ 11.75	...	13/11/13/12 ➔ 12
Total	**212**	**211.75**	...	**208.75**

Figure 1

Alternatively, SCRT will generate a similar chart for LP2 and LP3 only, should a software product only be active in LP2 and LP3.

What is the best Defined Capacity value to choose?

The goal is to retain a value allowing z/OS software cost control without significantly degrading performance. The Capacity Planner needs to analyze the behavior of each LPAR in detail by using SMF records 70 & 72 and then fix a value for Defined Capacity (DC), for example:

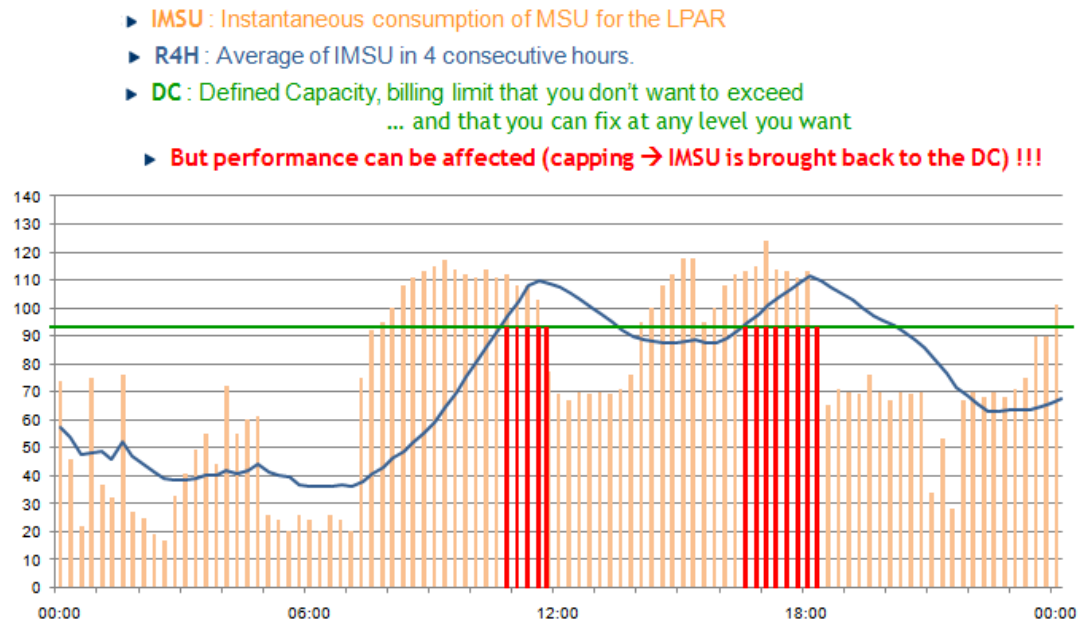

Figure 2

Therefore an optimal balance needs to be found, by defining a Defined Capacity that achieves the best compromise between performance and cost control.

Soft-Capping Business Case:

We can further explore the behavior of the Defined Capacity by using an example. CPC1 has got 4 LPARs: PROD, PREPROD, DEV & SYST. CPC1 is a z9 model 708 with 532 MSU. Without Soft-Capping, the SCRT is based on the R4HA only. The retained value is 505 MSU:

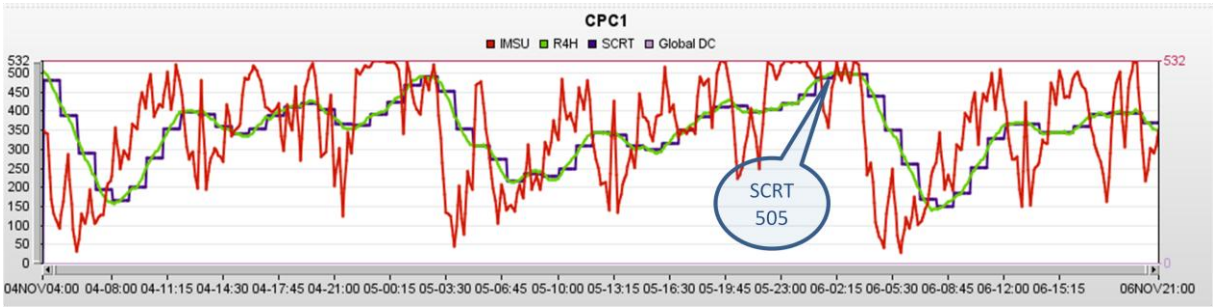

Figure 3: CPC1 activity

We shall now determine the R4HA peak for each LPAR, for example:

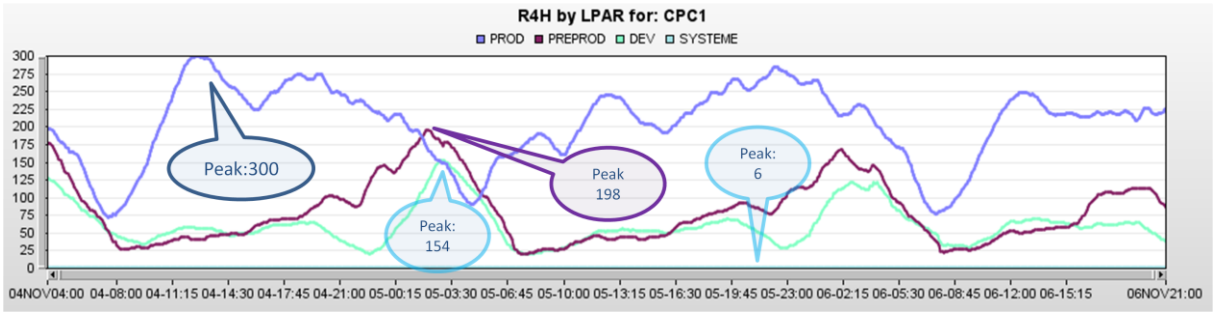

Figure 4

After a detailed analysis of SMF records 70 & 72, we are going to retain Defined Capacity values of 295 MSU for PROD, 185 MSU for PREPROD, 145 MSU for DEV and 5 MSU for SYST. The objective is to implement a modicum of MSU resource usage and thus financial control for extra-consumption periods, avoiding performance problems by setting each Defined Capacity Limit just below the peak of the associated R4HA:

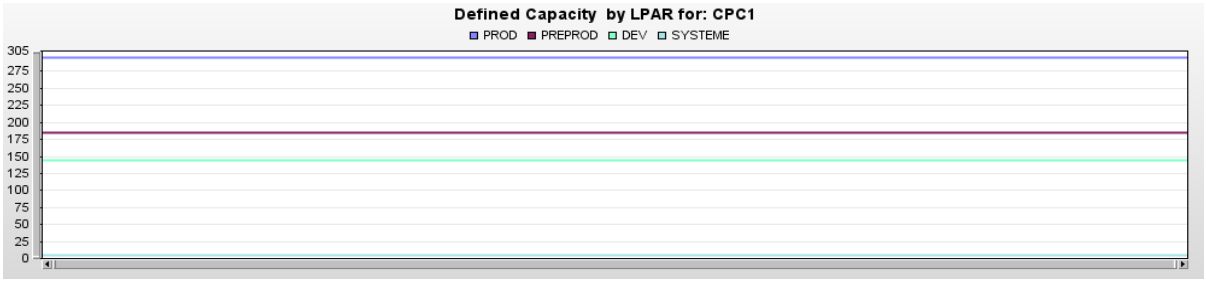

Figure 5

We will achieve such control with a Defined Capacity of 630 MSU, whereas the capacity of CPC1 is 532 MSU. Thus implementing a pragmatic Sub-Capacity policy, hence VWLC…

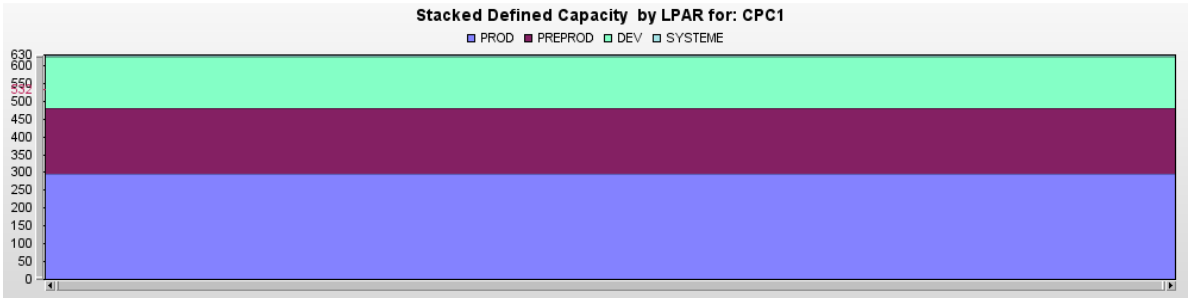

Figure 6

The result after the implementation of Soft-Capping shows a gain of 6 MSU (Reducing from 505 to 499 MSU) while avoiding performance problems.

Figure 7

<u>Group Capacity Limit (GCL) Considerations</u>

In 2008, IBM introduced one other feature named "Group Capacity Limit" (GCL) which makes the use of the Soft-Capping function easier.

GCL Terminology:

- ShareW%: This is the %SHARE calculated from the WEIGHT of the LPAR
- ShareW_GCL%: This is the %SHARE calculated from the WEIGHT in the GCL of the LPAR
- ShareD%: This is the %SHARE calculated from the DEFINED CAPACITY of the LPAR
- Target MSU from Weight: ShareW% translated in MSU
- Target MSU from GCL Weight: ShareW_GCL% translated in MSU

We should now consider the use of Group Capacity Limit for the same CPC configuration:

This GCL feature allows the definition of an aggregated group limit for all the LPARs defined within an associated group. As the sum of LPARs R4HA reaches the GCL, all LPARs IMSU will be capped on their target MSU from GCL Weight.

LPAR Name GCL	Weight	SHAREW% global	GCL Group	SHAREW% GCL	Target MSU based on Weight GCL	MSU limit from Weight	MSU Limit from #LP	Number of LP	Defined Capacity	SHARED%
PROD	600	60,00%	GCL1	60,00%	294	319	532	8	300	56,39%
PREPROD	200	20,00%	GCL1	20,00%	98	106	466	7	180	33,83%
DEV	180	18,00%	GCL1	18,00%	88	96	399	6	150	28,20%
SYST	20	2,00%	CGL1	2,00%	10	11	200	3	8	1,50%
Total	1000				490	532			638	
Capacity	532									
GCLimit	490									
Total Weight of GCL	1000									

Figure 8

In this example we plan to define a GCL of 490 MSU, a little below the initial VWLC.

By referring to the previously Defined Capacity settings (300, 180, 150 & 8) we are considering the definition of a GCL of 490 MSU. However, this does not provide the same performance guarantee during a capping period for the PROD LPAR with a Defined Capacity of 300 MSU with Soft-Capping, a MSU limit from Weight of 319 MSU and a target MSU based on weight GCL of only 294 MSU. The other LPARs within the GCL will be impacted by the MSU resource shortage. Therefore, to increase their "Target MSU from GCL", the GCL must be increased.

In this case, with a GCL limit at 490 MSU, the SCRT will take a base of 490 MSU for VWLC.

During the capping period and because the weight is static and linear, and above all because the weight was not originally designed for Soft-Capping but for Hard-Capping and to guarantee power when the CPC reaches its full capacity, the IMSU of all the LPARs will be capped. Hence, for example, if there is a loop in one LPAR, it can generate a Group Capping condition, so all the LPARs of the group will be capped. Clearly this is not a desirable scenario for any workload, but of course, a pragmatic and practicable compromise between the SCRT bill and associated workload performance must be found…

VWLC Evolution – AutoSoftCapping from zCost Management

During the previous two examples, we discovered that the Defined Capacity was a static limit; a certain amount of MSU is reserved for each LPAR. However, Soft-Capping does not allow the transfer of all or a subset of this reserved MSU resource from one LPAR to another; thus MSU resource sharing is not optimized. Also the R4HA metric is often higher than it could be, meaning that the z/OS software bill generated from the SCRT is higher than it could be. IBM Soft-Capping & GCL are good solutions to control the VWLC software bill every month at the same level; however, this control can cause performance concerns.

AutoSoftCapping (ASC) from zCost Management brings flexibility and dynamism in Soft-Capping by optimizing performance and controlling the associated VWLC.

How does ASC work?

1) Flexibility and Dynamism

ASC fulfills as efficiently as possible the MSU needs of each LPAR by dynamically adjusting each Defined Capacity value. Unlike IBM Soft-Capping and GCL which have a very basic algorithm associated with R4HA, ASC implements a more sophisticated and granular approach, which combined with relevant LPAR parameter settings (E.g. Customer MSU Policy), delivers the lowest SCRT bill possible per LPAR and thus LPAR Group, while maintaining the required performance levels. This is the primary objective of ASC architecture.

For an LPAR, the DC is defined higher but as close as possible to the major SCRT billing ethos, where If R4HA < IMSU, the goal is to avoid capping, as much as possible.

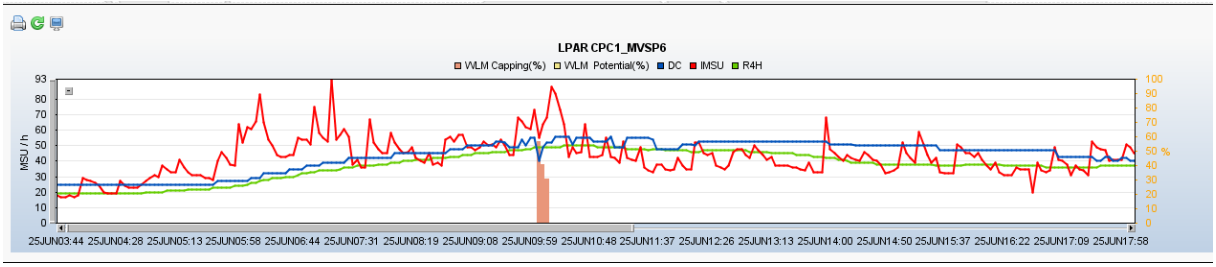

Figure 9

Thus it follows that If IMSU < R4HA; the goal is to decrease DC thus optimizing the associated SCRT bill

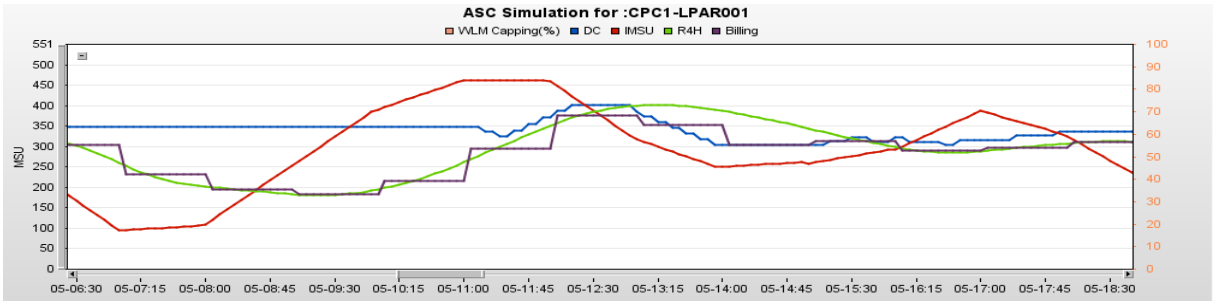

Figure 10

2) VWLC MSU Resource Control

ASC considers the MSU requirements for all LPARs, while the sum of the associated Defined Capacity (DC) remains constant.

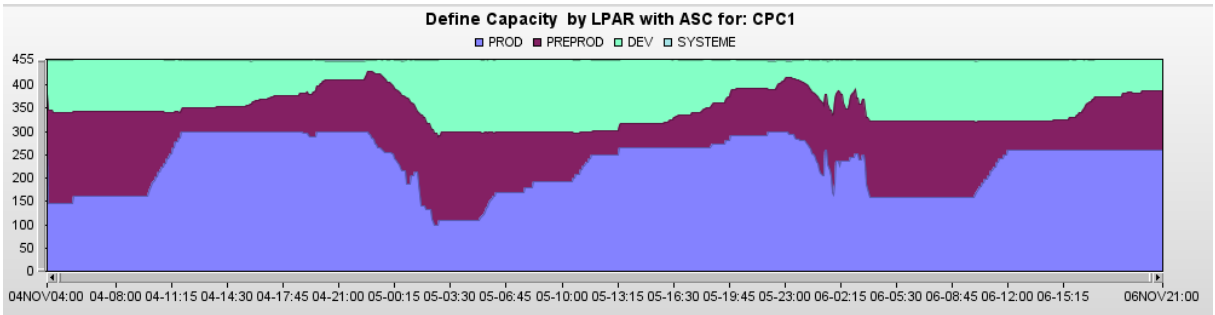

Figure 11

ASC Conclusion

Using the same configuration as in "Figure 2" with CPC1 and 4 LPARs (PROD, PREPROD, DEV & SYST). CPC1 is a z9 model 708 with 532 MSU.

Because SCRT without Soft-Capping is solely based on R4HA only (Figure 3), a VWLC of 505 MSU applies; the SCRT with Soft-Capping (Figure 7) shows a VWLC of 499 MSU; and the SCRT with GCL will be at 490 MSU but with a risk of performance constraint. Therefore a maximum 15 MSU (~3%) reduction (505 vs. 490 MSU), but with the possibility of performance constraints issues…

With the dynamic sampling associated with ASC and the transfer of unused MSU resource from one LPAR to another, facilities not possible with IBM Soft-Capping and GCL, a VWLC of 449 MSU without performance constraint is possible. Thus a 56 MSU (~11%) reduction (505 vs. 449 MSU)

For example, the new CPC1 ASC MSU activity profile:

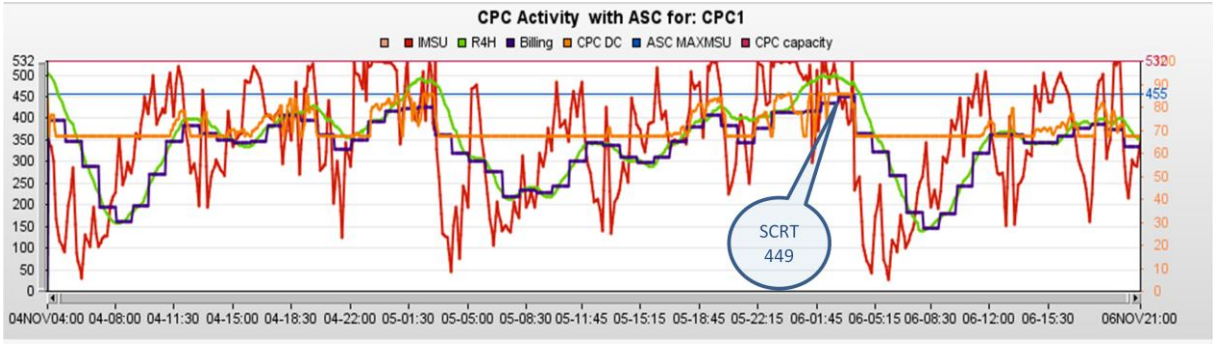

Figure 12

Furthermore, the sum of the DCs remains constant with an efficient load-balancing of MSU resource.

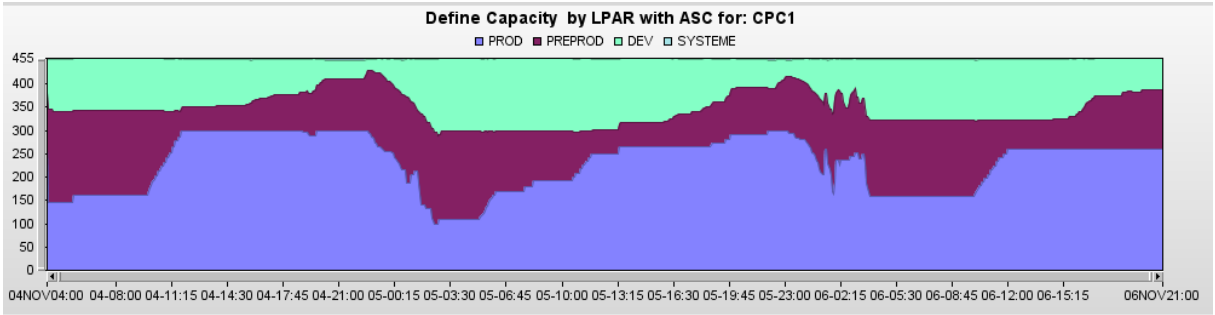

Figure 13

Additionally, the SCRT profile of ASC optimizing MSU resource usage at 449:

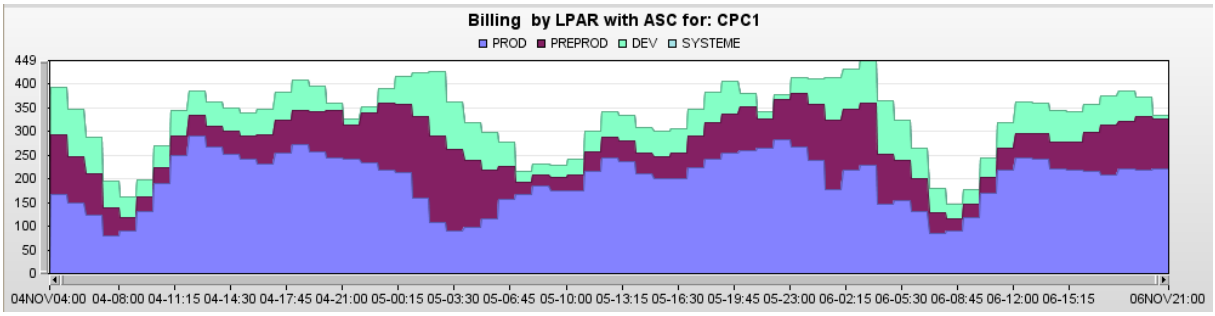

Figure 14

ASC Benefits

With **fully integrated technical (E.g. vis-à-vis SCRT) & management (E.g. SAM) reporting**, other typical benefits are:

- Delayed processor capacity acquisition or reduced technology (E.g. z9-z10) upgrade costs
- Perpetual & on-going reduced software costs
- Mechanism to control MSU usage for looping processes
- Optimized Production service for "controlled costs"
- Possible redesign of the PR/SM configuration daily, weekly or monthly.

IBM Soft-Capping and Group Capacity Limit (GCL) provide good mechanisms for controlling MSU usage, but by definition, "Capping" can cause "Performance" concerns, because MSU resource becomes finite as per Defined Capacity (DC), as opposed to pseudo infinite, as per the associated CPC capacity. However on the flip-side, the zSeries Mainframe customer must find the best balance between cost (E.g. VWLC bill) and performance. Therefore ASC provides this flexibility by constantly sampling MSU usage and refining DC allocations, using standard IBM API facilities (E.g. CMF/RMF and HMC), while uniquely dynamically allocating unused MSU from one LPAR to another. Furthermore the customer can create their own ASC policy to control this MSU transfer, safeguarding their own business profile.