

This paper introduces IBM's Sub-Capacity Pricing, hard and soft capping, and the rolling 4-hour average (R4HA), as well as an overview of the Automated Capacity Management feature of ThruPut Manager, which can automatically defer or constrain low priority batch to lower the monthly peak R4HA and result in significant software savings.

Overview

Monthly software licensing costs have become a significant ongoing expense for z/OS installations. IBM's Sub-Capacity Pricing and IBM provided cost control mechanisms such as LPAR Group Limits help reduce monthly fees, but in absolute terms, the costs remain high. Additionally, mechanisms such as "soft capping" may have undesirable effects on system performance and responsiveness.

ThruPut Manager's Automated Capacity Management feature allows for additional savings by deferring the execution of less important batch workloads so they do not contribute to the monthly software bill while allowing greater use of soft capping by reducing or eliminating the side effects.

This paper introduces the issues related to control of mainframe software costs and describes how Automated Capacity Management provides significant savings on top of those offered by the standard IBM mechanisms.

IBM's Sub-Capacity Pricing

In general, each individual IBM software product is licensed to a specific CPC. Historically, the license fees were based on the total CPC capacity. However, since a physical machine can now be split into one or more logical partitions (LPARs), an installation may not use the entire capacity of the CPC and a product may not be used on all LPARs. The "size" of the machine in which a product runs becomes smaller.

IBM's Sub-Capacity Pricing handles this situation by tying the monthly fee for each product to a value that reflects the peak usage of the LPARs on which the product is run. That value is calculated as the peak rolling 4-hour average (R4HA) CPU consumption for the month in MSUs (Millions of Service Units) per hour across all of the LPARs on which a product executed. Note that the entire load on the LPARs involved including z/OS overhead is used, not just the CPU used to

run the product. Of course, one of the products is z/OS itself, which of necessity runs on all LPARs.

Although Sub-Capacity Pricing can reduce software costs somewhat, most installations need a way to guarantee a limit to the monthly charges. IBM provides several ways, each of which allows the installation to control the CPU consumption of one or more LPARs on a CPC by establishing a limit and "capping" the CPU consumption as required.

Hard Capping

So called "hard capping" is implemented by specifying the "initial capping" option in the LPAR definition using the Hardware Management Console (HMC). This means that the LPAR can never utilize CPU unused by other LPARs in excess of its share by LPAR weight and the LPAR is capped all of the time.

If the LPAR has a quick spike in CPU requirements that would exceed the cap, it will simply not get those cycles and will be delayed so that its share of the total cycles available does not exceed its guaranteed share. This behavior can significantly limit the responsiveness of the LPAR.

At this point it should be noted that the R4HA is indeed an average, and therefore an LPAR can use more than the desired limit as long as at other times it uses less, which is a very typical usage pattern. So unless an LPAR operates at peak value for 4 hours at a time or more, while hard capping will control the R4HA, it is not suitable for workloads that vary in short term CPU requirements.

Hard capping is entirely implemented by PR/SM, part of the CPC firmware.

Soft Capping

While hard capping limits the LPAR's share of processor cycles, soft capping controls the value of the R4HA and is

intended specifically to control software costs. The installation sets the limit on the value of the R4HA. WLM monitors the R4HA and when the limit is reached directs PR/SM to limit the LPAR's access to CPU to stop the growth of the R4HA value.

The limit can be set either by LPAR or by LPAR Group. Using the Hardware Management Console (HMC), the installation can provide a "Defined Capacity" for individual LPARs or define an LPAR Group and provide an overall "LPAR Group Limit". In both cases, the limit is defined in MSU/hr and represents the maximum value of the R4HA that the installation wishes for the individual LPAR or group. An LPAR can be configured as part of an LPAR Group with an associated limit and also be assigned a Defined Capacity.

Generally, use of LPAR Groups provides better resource usage than does Defined Capacity at no additional cost. In an LPAR Group, the heavy usage by one LPAR can be offset by the light usage of others so that the Group Limit is not exceeded.

If the Group Limit on the R4HA of the total usage by all LPARs in the group is exceeded, WLM determines an individual limit expressed in MSU/hr for each LPAR in the group based on the LPAR's weight relative to the total weight of all members in the group and the current CPU demands by each member of the group. This individual limit represents the LPAR's share of the Group Limit.

It is a common misperception that the LPAR Group Limit is divided up solely by the relative LPAR weights. This is incorrect. WLM uses a "receiver/donor" approach to assign a greater share of the limit to those LPARs that require additional cycles. Only if all LPARs in the group have a high demand for CPU will the limit be divided up strictly by relative LPAR weights.

When the R4HA for an LPAR exceeds the LPAR's share of a Group Limit or its Defined Capacity, the LPAR is eligible for capping and WLM calculates how much the LPAR's access to CPU should be capped. These calculations determine the instructions that WLM provides to PR/SM to control the dispatching of the LPAR. This capping may be mild or severe depending on the size of the current CPU demand by the LPAR. If the load on the LPAR has dropped significantly it may not be necessary to cap the LPAR at all.

In an LPAR Group, some LPARs may be very busy and others only experiencing very light load. In this case the busy LPARs will be given more than their share by weight of the Group Limit and thus not be capped as heavily as they otherwise might have been.

If an LPAR has a Defined Capacity and is also a member of an LPAR Group the soft capping limit is the lower of the Defined Capacity and the LPAR's share of the LPAR Group Limit. When the R4HA for an LPAR Group is below the Group Limit, every LPAR's share of the Group Limit is set to be the value of the Group Limit.

Soft Capping – the Good and the Bad

Soft Capping is preferable to hard capping as it allows the LPAR to use resources above the limit for short periods, resulting in better use of available resources and more responsive performance. In general, hard capping should only be used if required for contractual purposes.

Soft capping is highly effective in controlling software costs. Once a Defined Capacity or a Group Limit is in place, IBM's Sub Capacity Reporting Tool (SCRT) used to generate the monthly usage report to be sent to IBM will not attribute a R4HA value for the LPAR or Group that is higher than the appropriate limit, regardless of the actual value of the R4HA.

If an LPAR's current CPU demand is very high when it reaches the limit, it will be severely capped by WLM and PR/SM. It's like hitting a brick wall at high speed. This level of capping causes the amount of CPU resource available to the LPAR to drop significantly, causing slow and uneven response for online and generally unacceptable response and throughput. If a significant period of high CPU usage preceded the start of soft capping, not only will the capping be severe, but it will last for a long period as it will take some time to get the R4HA below the limit.

Concerns over the consequences of soft capping have prevented many installations from implementing capping, resulting in greatly increased software licensing costs.

Everything Contributes to the R4HA

It seems obvious, but it is surprising how often this point is overlooked: every piece of load, even if it is running as discretionary, contributes to the R4HA. A second of CPU at low priority contributes exactly the same amount to the value of the R4HA as a second at high priority.

Clearly, some load is more important than others, yet all workloads can contribute to the size of the monthly software bill. Less important workloads can also significantly raise the CPU demand and result in longer and more severe periods of capping affecting the critical online applications.

Running low importance workloads in a discretionary service

class does not reduce the R4HA. If the CPU cycles are available, that workload will use them and the R4HA will increase as a result. A CPU cycle is a CPU cycle.

How to Save on Software License Fees

It sounds simple. Save on software license fees by implementing soft capping (most installations would use LPAR Groups) and reducing the amount of low priority work that is run when the R4HA is at peak. Low importance workloads are almost always batch jobs.

Obviously, it's not as simple as it sounds or more installations would be doing it.

The primary concern is that online response and turnaround for high importance batch must remain acceptable. To do this, the effects of soft capping must be minimized by reducing the amount of low priority batch work leading up to reaching the limit and while at or over the limit. Controlling a batch workload not only means restricting the number of executing jobs and the available initiators, but also controlling how much of the total CPU resource is consumed.

Ideally the less important batch workload should be gradually reduced as the R4HA approaches the limit, be under the tightest control while soft capping is active, and then the restrictions should be gradually removed as the R4HA decreases. Such an approach permits execution of as much non-critical workload as possible while keeping the R4HA low and minimizing the effects of soft capping.

Since the R4HA is an average, the peak value of the R4HA will often not correspond with peak demand for CPU and will lag by some period of time that varies based on usage patterns. If you consult your SCRT report, you may be surprised at when your peak R4HA value is occurring. This further complicates the task of deciding when to restrict some workloads.

Doing this manually would be a daunting task; yet, without control, low importance batch can drive the peak R4HA higher and soft capping can have more serious side effects.

Automated Capacity Management is a built-in feature of ThruPut Manager that automatically controls low importance batch workload in order to reduce the effects of soft capping or avoid soft capping altogether and enables the installation to lower the R4HA limit providing significant savings in software license fees.

A Brief Overview of ThruPut Manager with automation

ThruPut Manager (TM) allows the installation to define subsets of their batch workload. For each subset they can specify: Importance, Service goals, and Optional restrictions on how many can execute at once in the JESplex and on each member.

A single batch queue is used for all TM managed jobs. It orders that queue by importance and how the job is doing relative to its service goals. The queue is reordered frequently which means that high importance jobs that are missing their service goals will be first on the queue.

ThruPut Manager controls the number of "ThruPut Manager initiators" used to run TM-managed batch work based on current system and individual batch service class performance and the urgency of the work in the queue. TM also controls whether or not JES2 selects a job for execution in a ThruPut Manager initiator. It honors the installation specified restrictions and considers how well the LPAR and the WLM service class for the job is performing. When the job is selected, TM assigns the service class based primarily on the job's importance.

There is a lot more to ThruPut Manager automation, but it is this control over initiators, job selection and service class assignment that is extended by Automated Capacity Management to provide the installation with the necessary automation to reduce the contribution of less important batch workload to the peak R4HA, moderate the effects of soft capping and lower the peak value of the R4HA.

Introducing Automated Capacity Management

The underlying philosophy of Automated Capacity Management (ACM) is to defer less important batch workload to reduce or eliminate its contribution to the peak value of the R4HA. ACM adds more criteria to the job selection decisions being made by ThruPut Manager. With ACM configured and active, TM is in a unique position to control batch workload by taking into account the value of the R4HA relative to the limit, providing real savings on monthly software license fees.

On each LPAR on which it is active, ACM regularly determines the current R4HA and the recent rate of CPU consumption for the LPAR and LPAR Group (if applicable). It uses the R4HA of

the LPAR and the LPAR Group, as well as the recent consumption figures, compares them to the Defined Capacity and/or LPAR Group Limit, depending on which is present, and calculates a capacity percentage.

Automated Capacity Management introduces the concept of a “Capacity Level” for an LPAR which represents how close the current usage is to the limit. Capacity Level values range from 1 to 5, where a value of 1 represents full usage. The values 2 to 5 represent percentages of the limit. By default these are 95%, 90%, 85% and 80% but may be altered by the installation.

The installation specifies which subsets of the batch workload they wish to have constrained by ACM and specify what constraints should apply at each Capacity Level. Typically the installation will specify increasing constraints as the Capacity Level indicates higher usage.

For each subset of the batch workload, the installation can specify the maximum number of jobs in the subset that can be executing on the LPAR at once at each of the 5 possible values of Capacity Level. If the maximum for the Capacity Level has been reached, jobs in the subset will not be selected.

What about the jobs that are running? How do we stop a heavy CPU job from consuming all of the available CPU and driving up the R4HA?

Automated Capacity Management solves this by allowing the installation to define “Capacity Management” WLM service classes to be used exclusively by ACM to restrict the CPU consumption of the less important batch work. There can be one service class defined for each of the 5 possible values of Capacity Level. These service classes are intended to be defined as part of WLM resource groups with varying maximum CPU consumption limits on a system basis. By decreasing the limit for the associated resource groups as the Capacity Level moves from 5 to 1, the installation can gradually decrease the CPU consumption of designated subsets of the batch load as the R4HA increases and gradually relax these constraints as the R4HA decreases.

For each subset of the batch load for which the CPU usage is to be restricted, the installation specifies the Capacity Level at which the jobs are to be assigned to a Capacity

Management service class. If this is specified for a particular subset and a job in that subset is selected for execution, the job will be started in the appropriate service class.

When the Capacity Level changes on an LPAR, Automated Capacity Management reevaluates the service class for all ThruPut Manager managed jobs currently executing on the LPAR and resets the service class for each job as required. If a job belongs to a subset that is not affected by ACM, then its service class will not be changed.

Once these settings are configured, Automated Capacity Management does everything automatically. ACM monitors the R4HA and the current usage on each LPAR and makes the appropriate decisions and adjustments without any operator intervention. Batch workloads identified by the installation as less important will be gradually constrained as the R4HA rises. These constraints are relaxed as the R4HA and current demand decreases.

How Automated Capacity Management Delivers Savings

By reducing the contribution of low importance batch load to the R4HA, Automated Capacity Management reduces the negative effects of soft capping and makes it possible to implement the Defined Capacity and/or LPAR Group Limits or reduce current limit values, resulting in lower monthly software costs while maintaining online response times and the required throughput for high importance batch workloads.

At the same time, because ACM is constantly monitoring the R4HA and current demand, it will still provide the best service possible to low importance load under the circumstances by relaxing constraints as soon as the peak value of the R4HA will no longer be affected.

Installations that were hesitant to implement soft capping due to possible negative effects can now go ahead and reap not only the savings that soft capping itself provides but also the increased savings made possible by ACM.

These are not one-time savings. Automated Capacity Management will keep reducing monthly software license fees for years to come.