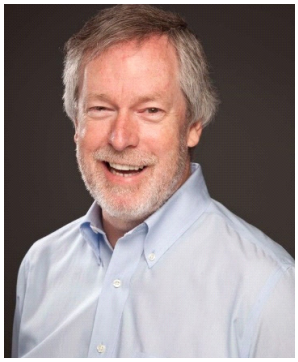
A PRACTICAL JOURNAL OF z/OS TUNING AND MEASUREMENT ADVICE

Cheryl Watson's Tuning Letter

REPRINT



This document is a reprint of a *Cheryl Watson's Tuning Letter 2021 No. 1* article by **Jim Horne,** titled **'A Different Way to Define LPAR Weights**'.

The ability to logically divide a CPC into multiple Logical Partitions (LPARs) has been with us since 1988. The share of the total capacity that each LPAR is guaranteed is determined by that LPARs weight compared to the total of the weights of all active LPARs. Traditionally, most sites tried to aim for all weights summing to 1000 - this conceptually made it easier to see the guaranteed capacity of each LPAR - if it had a weight of 350, that indicated that it should be guaranteed 35% of the total capacity.

But in practice, maintaining that total of 1000 didn't always work out. Someone might increase the weight of one LPAR, but forget to reduce the weight of other LPARs by a corresponding amount. Prior to HiperDispatch, small variances in LPAR weights didn't make much of a difference in practice. However, the introduction of HiperDispatch means that increasing an LPAR's weight by a value as small as 1 might mean the difference between having a Vertical Medium or a Vertical High CP.

With that in mind, Jim Horne, from Lowe's Companies, invented a new paradigm for managing LPAR weights. His methodology is not only intuitive, it is also far easier to manage than the old way of managing LPAR weights. We encourage all sites to consider Jim's model and see if it wouldn't be a better fit for your environment.

A Different Way to Define LPAR Weights

The classic way to define LPAR weights is to make the sum of all LPAR weights equal to 1,000. It provides an easy way to see the percent of the total CPC an LPAR's weight guarantees it, and is the most commonly-used methodology for managing LPAR weights. However, in today's world of HiperDispatch and its reliance on engine polarity, this information is not as useful to most of us as it used to be.

This concept was the basis behind **Jim Horne's** '*A Different "Weigh" to Define LPAR Weights*' session at SHARE in Dallas 2022. Jim's session had a large attendance and positive feedbacks from the attendees, and we really liked his idea, so he kindly agreed to write this article for us. If you are facing a CPC upgrade in the foreseeable future, this is an ideal time to review Jim's methodology and see how it would work in your environment.

For those of you that don't know Jim, he has been a mainframe systems programmer since his first day on the job in 1984, and (according to Jim) he has no idea what he wants to be when he grows up ☺. In the meantime, he is a specialist in performance tuning and capacity planning for the mainframe environment at Lowe's Companies, where he has been for the last 22 years.

Target Audience

This article will be of interest to anyone that is responsible for defining, managing, or understanding LPAR weights and their impacts. It offers a different approach, one this is more suited to a HiperDispatch world, for defining weights, so we recommend that both seasoned and less-experienced technicians should read it.

In this article, we will look at another approach to defining and managing LPAR weights, one that is more relevant to HiperDispatch.

But before we do that, we need to have a common understanding of the terms 'LPAR weight' and 'guaranteed share'.

- The weight of an LPAR provides a way to determine the share of the CPC an LPAR is guaranteed if it needs it. The share, or guaranteed percent available, is determined by dividing the LPAR's weight by the sum of the weights of all active LPARs.
- A guarantee means that the defined share will be available to an LPAR *if it needs it*. It does not mean that the LPAR *will* always use it, just that it always *can*. If an LPAR is not

using all its guaranteed share, that capacity is available to other LPARs until this LPAR needs it. This guaranteed share is independent of any LPAR capping, and is often indicative of the relative capacity of an LPAR over an extended time (days or weeks). This is true for each type of engine: GCPs, zIIPs, ICFs, or IFLs.

We want to be able to define LPAR weights that are meaningful indicators of how much of the CPC each LPAR needs if every LPAR is running at max capacity. You can determine how many cores an LPAR is guaranteed by multiplying its share of the CPC by the number of engines in the pool you are working with.

Table 1 shows how easy it is to determine each LPAR's guaranteed fair share when the LPAR weights total to a round number. In this example, the weights total to 1000. So you can very quickly see that LPARs SYS1, SYS2, and SYS3 each have a fair share of 30% of the CPC, and LPAR SYS4 has a fair share of 10% of the CPC.

Table 1 - Traditional Weights

LPAR	Defined CPs	Weight	Percent of Pool	Guaranteed Capacity
SYS1	3	300	30%	2.4
SYS2	3	300	30%	2.4
SYS3	3	300	30%	2.4
SYS4	1	100	10%	0.8
Total		1000	100%	8.0
Total installed CPs	8			

An alternative way to define LPAR weights, more suited to the world of HiperDispatch and the concept of vertical engine polarity, is what I call the "Guaranteed Engines" method. It differs from the traditional method in that the total weight of all LPARs in a processor pool for a given engine type is based on the total number of shared engines in that pool. This means we can instantly see how many engines of a particular type each LPAR is guaranteed.

In the example in Table 2 on page 4, we want SYS1, SYS2, and SYS3 to each have a guaranteed share of 30% of the total available capacity. The CPC has 8 shared CPs, so that results in a weight of 240 (8 CPs * 30%) for each of those LPARs. As you can see, the total of all the weights is now 800, rather than 1000. The number of engines guaranteed to each

LPAR is immediately obvious, and it is a simple calculation to determine each LPAR's guaranteed percent share of the total available capacity.

Table 2 - Guaranteed Engines Weights

LPAR	Defined CPs	Weight	Percent of Pool	Guaranteed Capacity
SYS1	3	240	30%	2.4
SYS2	3	240	30%	2.4
SYS3	3	240	30%	2.4
SYS4	1	80	10%	0.8
Total		800	100%	8.0
Total installed CPs	8			

One of the benefits of the new method is that calculating HiperDispatch polarities is much easier. The rules for determining polarity are:

- Less than 1.5 engines:
  - Two Vertical Mediums.
- Between 1.5 up to just less than 2 engines:
  - z13: 2 Vertical Mediums.
  - All other machines: 1 Vertical High, 1 Vertical Medium.
- 2 or more engines (N greater than or equal to 2):
  - If the fraction is *less* than 0.5:
    - N-1 Vertical Highs.
    - 2 Vertical Mediums.
  - If the fraction is *greater* than or equal to 0.5:
    - N Vertical Highs.
    - 1 Vertical Medium.

If we apply these rules to the configuration in Table 2 on page 4, we see that the HiperDispatch topology is not very efficient, as shown in Table 3.

Table 3 - Weights Based on Guaranteed Cores (With Polarities)

LPAR	Defined CPs	Weight	Percent of Pool	Guaranteed Capacity	Vertical Highs	Vertical Mediums
SYS1	3	240	30%	2.4	1	2
SYS2	3	240	30%	2.4	1	2
SYS3	3	240	30%	2.4	1	2
SYS4	1	80	10%	0.8	0	1
Total		800	100%	8.0	3	7
Total installed CPs	8					

Each of the three large LPARs has only one Vertical High (VH) and 2 Vertical Mediums (VM). The fourth LPAR is guaranteed one VM, which is what it would get regardless of its share, because only one engine (CP) has been defined to it. Table 4, shows how a small change in weight for the three primary LPARs results in a much better polarity: SYS1, SYS2, and SYS3 each have 2 VH and 1 VM, rather than 1 VH and 2 VMs.

Table 4 - Optimized Weights Based on Guaranteed Engines Weights (With Polarities)

LPAR	Defined CPs	Weight	Percent of Pool	Guaranteed Capacity	Vertical Highs	Vertical Mediums
SYS1	3	250	31%	2.5	2	1
SYS2	3	250	31%	2.5	2	1
SYS3	3	250	31%	2.5	2	1
SYS4	1	50	6%	0.5	0	1
Total		800	100%	8.0	6	4
Total installed CPs	8					

Adding Temporary Capacity

I find that the guaranteed engines method is especially helpful when you want to define temporary capacity, such as a Capacity Backup (CBU) or On Off Capacity on Demand (OOCoD) situation. Looking back at the original configuration in Table 1 on page 3, our objective is that the guaranteed capacity of LPARs SYS2, SYS3, and SYS4 would remain

unchanged, and the guaranteed capacity of the SYS1 LPAR would increase from 2.4 to 3.4. The following examples illustrate three ways you can add 1 CP of temporary capacity to a CPC, and show how close they get to our objective.

1. Add the temporary engines to the CPC and the LPAR(s) that you want to help, but do not make any weight changes. As seen in Table 5, this approach means that the additional capacity will be available to *all* LPARs, rather than just the one we are specifically trying to help (SYS1). This is clearly not what we wanted, but it highlights the need to adjust LPAR weights when making capacity changes.

Table 5 - Add 1 CP, But No Weight Changes

LPAR	Defined CPs	Weight	Percent of Pool	Guaranteed Capacity	Vertical Highs	Vertical Mediums
SYS1	4	300	30%	2.7	2	1
SYS2	3	300	30%	2.7	2	1
SYS3	3	300	30%	2.7	2	1
SYS4	1	100	10%	0.9	0	1
Total		1000	100%	9.0	6	4
Total installed CPs	9					

2. Add the temporary engines to the CPC and LPAR(s) and adjust the weight of the target LPAR(s) to give them a larger share of the enlarged capacity, but do *not* bother trying to keep all the weights adding up to 1000.

Table 6 - Add 1 CP, Adjust Weight of Subset of LPARs

LPAR	Defined CPs	Weight	Percent of Pool	Guaranteed Capacity	Vertical Highs	Vertical Mediums
SYS1	4	400	36%	3.27	2	2
SYS2	3	300	27%	2.45	1	2
SYS3	3	300	27%	2.45	1	2
SYS4	1	100	9%	0.82	0	1
Total		1100	100%	9.0	4	7
Total installed CPs	9					

As you can see in Table 6, we got close to what we wanted, but not *exactly*. In this example, the polarity for the LPAR we want to help is the same as it would be if it had gotten the entire engine, but that may not be the case in a real situation.

3. Add the temporary engines to the CPC and LPAR(s) and manually recalculate *all* the LPAR weights so they still add up to 1,000 as shown in Table 7.

Table 7 - Add 1 CP, Adjust Weight of ALL LPARs

LPAR	Defined CPs	Weight	Percent of Pool	Guaranteed Capacity	Vertical Highs	Vertical Mediums
SYS1	4	378	38%	3.40	2	2
SYS2	3	267	27%	2.40	1	2
SYS3	3	267	27%	2.40	1	2
SYS4	1	88	9%	0.79	0	1
Total		1000	100%	9.0	4	7
Total installed CPs	9					

Here we have the exact engine distribution we wanted, with the entire engine being guaranteed to SYS1. However, look at the weights of the LPARs - this took some calculating, probably not something you want to have to do in a high pressure situation.

4. Finally, let's look at adding the engine to SYS1 using the Guaranteed Engines method. The entire engine weight went where we wanted it to go, and only there, as you can see in Table 8. No complicated calculations are needed because the weight of the additional engine is reflected in the total weight of the CPC.

Table 8 - Add 1 CP, Adjust Weights Using Guaranteed Engines Method

LPAR	Defined CPs	Weight	Percent of Pool	Guaranteed Capacity	Vertical Highs	Vertical Mediums
SYS1	4	340	38%	3.40	2	2
SYS2	3	240	27%	2.40	1	2
SYS3	3	240	27%	2.40	1	2
SYS4	1	80	9%	0.80	0	1
Total		900	100%	9.0	4	7
Total installed CPs	9					

Handling Transient LPARs

The final situation I want to mention is dealing with LPARs that are not up all the time. There is no good, one size fits all answer to this, regardless of the method you use to compute total weights on the CPC. You can either define the weights for all LPARs such that the seldom

used LPARs will not affect the polarity of your most loved LPARs, or that if their polarity must change, you know about it ahead of time.

You will have to decide what is best for your shop. The key takeaway from this is to make sure you are aware of the implications of this situation.

Lessons Learned

We originally used the traditional method for managing LPAR weights. However, as the importance of a good HiperDispatch topology became more apparent to us, we realized that we needed a better, more effective, way to manage our LPAR weights. We also needed a less manually-intensive method that would make it easier to manage growing numbers of LPARs, and more use of temporary capacity.

To address these requirements, we invented a new way to calculate LPAR weights, one that is consistent with how HiperDispatch works so you can *easily* define polarity of the PUs in each engine pool on your CPC. We have a reasonably large, reasonably complex configuration, and our guaranteed-engine method has stood the test of time for us.

Some of you may find it useful, while others may decide it is better to stay with the traditional method, or whatever method you use today. As with any change, there are some things to be aware of if. One of the simplest is the multiplication factor you use for the guaranteed engines in each LPAR pool. In all the examples in this article, I use a factor of 100, but if you have, or may have, LPARs with more than 10 CPs, you may want to use a factor of 10. The reason for this is that PR/SM will not allow you to enter a weight value greater than 999.

There may be other issues that are unique to your installation - naturally you should discuss the benefits and drawbacks of the various approaches with the interested parties in your site. You can also use the LPARDesign tool to easily determine the new weights you might use, and how they would affect your HiperDispatch topology.

References

For more information on this topic, refer to:

- SHARE in Dallas 2022 Session '*A Different "Weigh" to Define LPAR Weights*', by **Jim Horne**.
- 'HiperDispatch Questions and Answers' article in *Tuning Letter 2015 No. 4*.
- SHARE in Fort Worth 2020 Session '*Understanding LPAR Controls for Better Performance*' by **Kathy Walsh**.

Summary

We want to thank Jim for taking the time to share his experiences with us. Jim presented this methodology at SHARE in Dallas this March, and it was very well received by the large audience.

As Jim points out, there is no perfect methodology that will fit every situation. Nevertheless, we frequently see customer configurations with sub-optimal LPAR weights, where a little tweaking and tuning can make a real difference.

We like Jim's methodology because it puts more emphasis on the HiperDispatch topology, and less on the LPAR's guaranteed share of capacity. In reality, we don't know many customers that run with *all* LPARs using *all* their available capacity for extended periods of time - and that is the only scenario where an LPAR's weight will limit the capacity available to the LPAR. For all other times, moving an LPARs weight up or down by a few units will make no discernible difference to the capacity that is given to the LPAR, but it can make a real difference to the efficiency of the overall configuration.

If you combine Jim's methodology with the LPARDesign tool we think you will be well positioned to achieve the optimal HiperDispatch topology without spending undue amounts of time on this activity.