LPARDesign Tool Update

Now that IBM has officially announced the new z16 mainframe, we expect that many of our readers will be receiving an upgrade proposal from IBM. Our 'IBM's Latest Mainframe - Meet the z16' on page 10 will help you get up to speed on the changes and enhancements in the new model.

One of the key activities in any CPC upgrade is selecting the 'Goldilocks upgrade' - that is, the target model that is just right for your needs; not too *much* capacity (because your hardware, maintenance, and some software costs are based on your installed capacity), and not too *little* capacity (because no one wants to have to go back to your Finance colleagues looking for more money because your new CPC doesn't provide the expected amount of capacity). You want the model that will deliver the performance and availability you need, and do so at the lowest overall cost of ownership.

IBM's zPCR tool is an invaluable tool for anyone planning a CPC upgrade. Another, possibly less well known, free tool that we love is LPARDesign, created by **Alain Maneville** and his customer and friend, **Thierry Deleris**. You might recall that we provided an article on this tool back in 2017. Alain and Thierry have made many enhancements over the last five years, and have just released the latest, z16-supporting version of the tool, so we thought this would an appropriate time to provide an update on the tool.

Before I go on, I want to thank Thierry, Alain, and **Seth Lederer** from IBM for all their help and patience during the writing of this article. I hope that you will take the time to download and use the latest version of this tool, and that you find it as valuable as we do.

Target Audience

This article should be of interest to anyone responsible for planning a CPC upgrade. But the LPARDesign tool continues to deliver value long after your migration to the new CPC is complete - we recommend that customers use this tool to model and optimize any configuration changes that might impact your HiperDispatch topology.

If you would like an introduction to the LPARDesign tool, see the LPARDesign Tool article in *Tuning Letter 2017 No. 4*.

Brief Introduction

Alain is a lifetime mainframer, with his most recent role being as an Executive IT Specialist in IBM France[13]. One of his many responsibilities in that role was to help his customers optimize their mainframe configurations. With the introduction of HiperDispatch on the z10 family of CPCs, optimizing a CPC's LPAR topology became an especially important ingredient in maximizing the capacity customers got out of their CPC. For a small or simple configuration, managing LPAR weight might not be not particularly difficult. However, finding the 'right' LPAR weights in configurations with many LPARs can be quite a challenge - it is very easy to overlook the impact that changing the weight of *one* LPAR could have on the *other* LPARs sharing the same CPC.

To address this challenge in a more scalable and user-friendly manner, Alain and his friend Thierry started developing a spreadsheet tool that they subsequently made available to customers as LPARDesign V3 in 2012. Rather than you having to take a 'try it and see' approach, or developing your own spreadsheet, you can load your LPAR configuration into the tool, and it will immediately show you your HiperDispatch topology - the number of Vertical High (VH), Vertical Medium (VM), and Vertical Low (VL) engines of each type in each LPAR (along with a host of other helpful information). It lets you easily model the impact on the HiperDispatch topology of planned configuration changes (such as performing a CPC upgrade!). It lets you easily define even the most complicated configurations by supporting import of zPCR files. And, perhaps most valuable of all, it will analyze your configuration against IBM's best practice guidelines and offer recommendations about changes you can make to optimize the configuration.

The original LPARDesign tool processes information for one CPC for one interval. Based on their experience with LPARDesign, Alain and Thierry found that there are times when it would be helpful to be able to easily compare multiple configurations. To address this requirement, they developed LPARDesign Extended - this add-on to the original LPARDesign tool was added since our last article, so we will cover it in a little more detail later in this article.

You might have seen a mention of LPARDesign in the recent enhancements to zPCR. zPCR 9.4e added a new HiperDispatch report, modeled after the LPARDesign tool. It has much of the information that is provided in the LPARDesign CONFIG sheets, however it is missing what I think is one of the most helpful features of LPARDesign - the ability to dynamically adjust an LPAR's weight and/or number of logical CPs and immediately see the impact that change would have on the overall HiperDispatch topology. It also doesn't have LPARDesign's 'Expert advice' function. On the other hand, LPARDesign is not intended to provide the detailed capacity modeling function that zPCR provides. I don't think it is a question of which tool is 'better' - they are both valuable, and each addresses a different need; they make a good team.

[13] Alain is retired now, but he and Thierry continue to support and enhance their LPARDesign tools.

Anyway, that's enough introduction. Now let's run through an example of how you would use LPARDesign and LPARDesign Extended as part of your planning and implementation of a new z16. You can get more detailed information on the use of LPARDesign in its *User's Guide*, or in the LPARDesign Tool article in *Tuning Letter 2017 No. 4*, so we won't go through every possible option here. Rather, the objective is to help you get real value from LPARDesign in as little time as possible.

Using LPARDesign

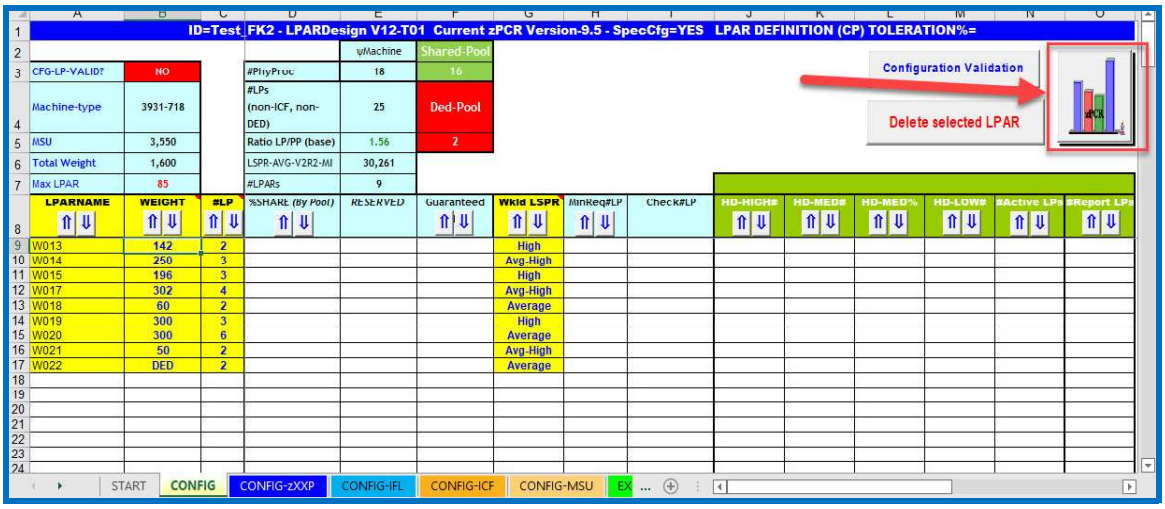
Before going any further, we suggest downloading both tools - LPARDesign and LPARDesign Extended from their github page: https://github.com/AlainManeville/z-OS-LPARDesign. The latest releases at this time are LPARDesign V12 T01, and LPARDesign Extended V2 T02. 'Installation' consists of nothing more than downloading one xls file for each tool.

The LPARDesign xls files come pre-loaded with a sample configuration. However, given that this article is aimed primarily at readers preparing for a CPC upgrade, we recommend that you use a .zpcr file to load *your* current configuration into LPARDesign. The IBM CP3KEXTR and zPCR documentation tells you how to create the zPCR input files using your SMF data (a minimum of type 70 and type 113 records, and preferably for every LPAR on the CPC that is to be upgraded). CP3KEXTR creates EDF files that you then load into zPCR, and zPCR in turn creates a .zpcr file. We would expect you to be doing this anyway as part of your upgrade evaluation process, so using LPARDesign should generate any additional work.

To load your configuration into LPARDesign, open the xls file you downloaded, specify a name for your study (this can be anything you want - hopefully something meaningful, but a maximum of 15 bytes), and then click on the 'CONFIG' tab. This will show you the tab containing the HiperDispatch information about the general purpose CPs in the sample configuration. But we want to load *your* information, so click on the zPCR logo on the top right of the sheet, as highlighted in Figure 19 on page 71.

Note: One of my laptops still has MS Office 2010, and that level of Excel had occasional problems with the Visual Basic in LPARDesign - on other PCs, Excel 2016 and Excel 365 did not have any problems. Therefore, based on my experience, I recommend using a more recent level than Excel 2010.

Figure 19 - Navigating to Import zPCR File



After you click the zPCR icon, a 'zPCR Actions' window will open. Select the 'Import' option and navigate your way to your zPCR file. Select the file you want to load and press OK. In the Import zPCR Configuration window that pops up, click on 'Load zPCR Configuration'. That will load your configuration into the tool, and you should see all your LPARs listed in the LPARNAME column.

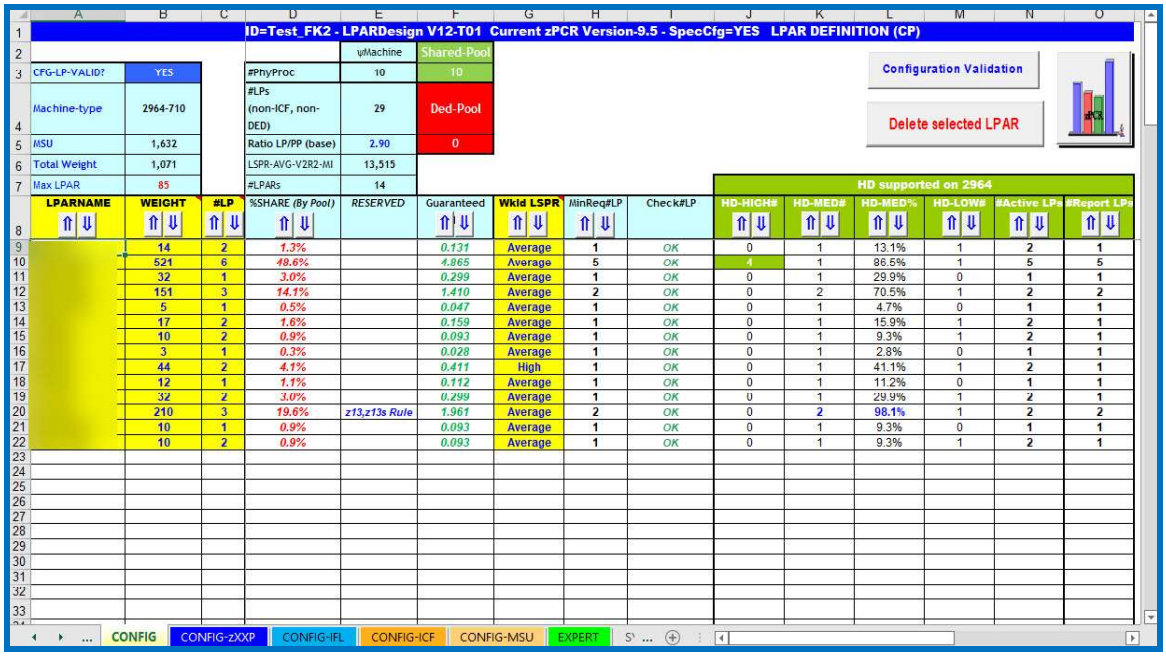
Figure 20 - Navigation Bar

Before you do anything else, locate the LPARDesign Navigation and Action Bar, shown in Figure 20. This is not part of the spreadsheet, so you might need to hunt around your desktop a little to find it. You will need to use the Navigation and Action bar to perform various functions while you are using LPARDesign.

Next, go to the START tab. Near the center of the sheet, there is a section called “Change Configuration”. In that section, confirm that the Machine Type, the number of zIIPs, number of ICFs, and number of IFLs for your imported CPC match your expectations. This is also the place where you would change the number of any of the different types of processors if you want to model a different configuration - we’ll come back a little later to discuss how to change those.

Next, go to the CONFIG tab - this is the one that shows the HiperDispatch topology of your general purpose CPs. To populate the empty columns, click on the Configuration Validation button. You should now have a screen that looks something like that shown in Figure 21.

Figure 21 - Validated CONFIG Tab



On a single screen you can now see the weight of each LPAR, the number of logical CPs defined for the LPAR, the LPAR’s share of the total shared general purpose CP pool as a percentage, the LPAR’s share of the total shared general purpose CP pool specified in terms of number of CPs, the LSPR workload categorization for each LPAR for the selected interval[14], the minimum required number of CPs to deliver the capacity as guaranteed by the LPAR’s weight, the number of Vertical High, Medium, and Low CPs, and the percent of a CP that each Vertical Medium CP is entitled to.

In this example shown in Figure 21, the current CPC is a z13 710 with 14 LPARs. Only one of the 14 LPARs has *any* Vertical High general purpose CPs (the second one down). However, you will notice that the 3rd LPAR from the bottom has a guaranteed capacity of 1.961 CPs, which results in it having two Vertical Medium CPs. In this case, you might want to try increasing the weight a little - increasing the LPAR’s weight from 210 to 216 results in

[14] The workload categorization is shown for information purposes, and LPARDesign carries it forward to any .zpcr files you might create for subsequent use by zPCR. It is not used by any of LPARDesign’s internal calculations.

that LPAR going from zero Vertical High CPs to one Vertical High. To determine the impact of your planned changes, you simply overwrite the weight or number of Logical Processors for whatever LPARs you want to change, and then click on Configuration Validation.

You should go through this process for each engine type on the CPC (there is a separate CONFIG tab for each one), always remembering to click the Configuration Validation button after you make a change.

The other thing we recommend is checking the Expert tab to see if the tool has any warnings or recommendations about ways to further optimize the configuration. You will see that there are four buttons across the top of the Expert sheet, one for each engine type (GCP, zIIP, ICF, and IFL). Click on each button and scroll through the list of LPARs to see if there are any recommendations.

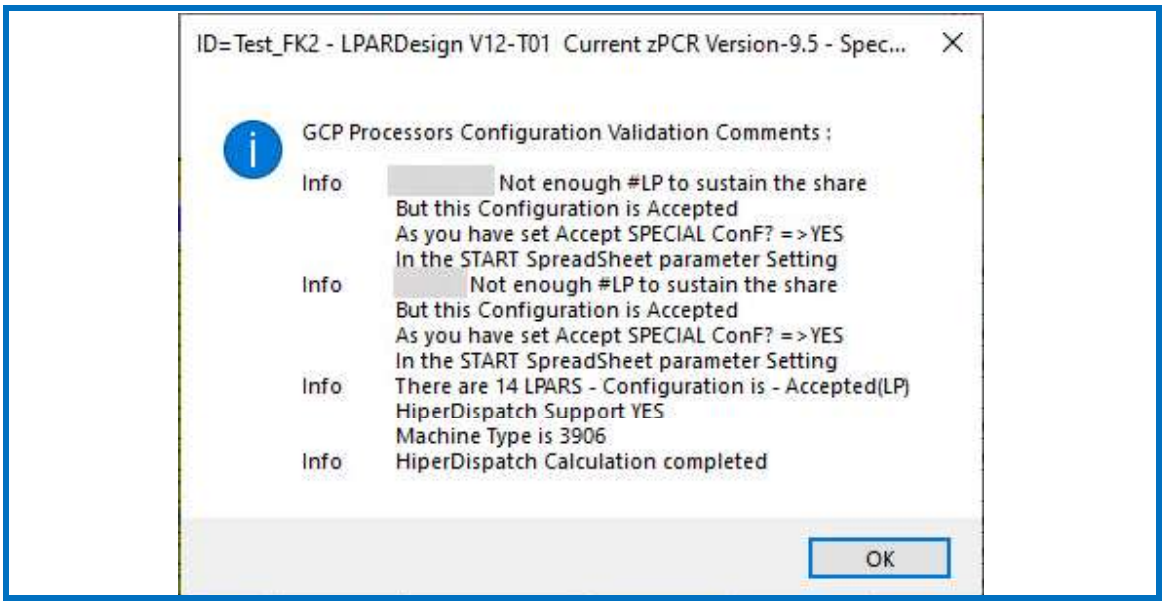
If you are preparing for an upgrade and have not checked your LPAR weights recently, performing an exercise like this can help you optimize your current configuration, *prior* to doing your analysis to identify the ideal target CPC. When you have reached your 'ideal' configuration, you can export the updated configuration into a new .zpcr file that you can then load into zPCR. It would be an interesting exercise to see zPCR's projections of the expected number of MIPS for your optimized configuration, and compare that to the 'before' configuration.

When you are ready to plan for your future CPC, you would use a tool such as zPCR or *Cheryl Watson's CPU Chart* to identify a few CPCs that would deliver roughly the amount of capacity you expect to need on the new CPC. In this example, let's say that one of your candidates is a z15 618. So you would like to use LPARDesign to determine the number and mix of High, Medium, and Low CPs on that target CPC.

To do this, go to the START tab, click on the drop down beside the Machine-Type field, and select an 8561-618. Now, very importantly, at this point, do NOT go directly back to the CONFIG tab. Instead, click on the GCP button in the Navigation and Action Bar. This will bring you to the CONFIG tab *and* apply your changes to the Machine-Type.

To see what the HiperDispatch topology would look like on your target CPC, click on the Configuration Validation button. In addition to adjusting for the new processor type and number of physical CPs, the Configuration Validation also checks to see if the number of logical processors in each LPAR is sufficient to deliver the guaranteed capacity based on the LPAR weights. In this case, the general purpose speed of the target CPC is lower than the speed of the current CPC, so we get a warning message that two of the LPARs are not defined with enough logical processors, as shown in Figure 22 on page 74.

Figure 22 - Processor Change Warning Messages



Under-Provisioned LPARs

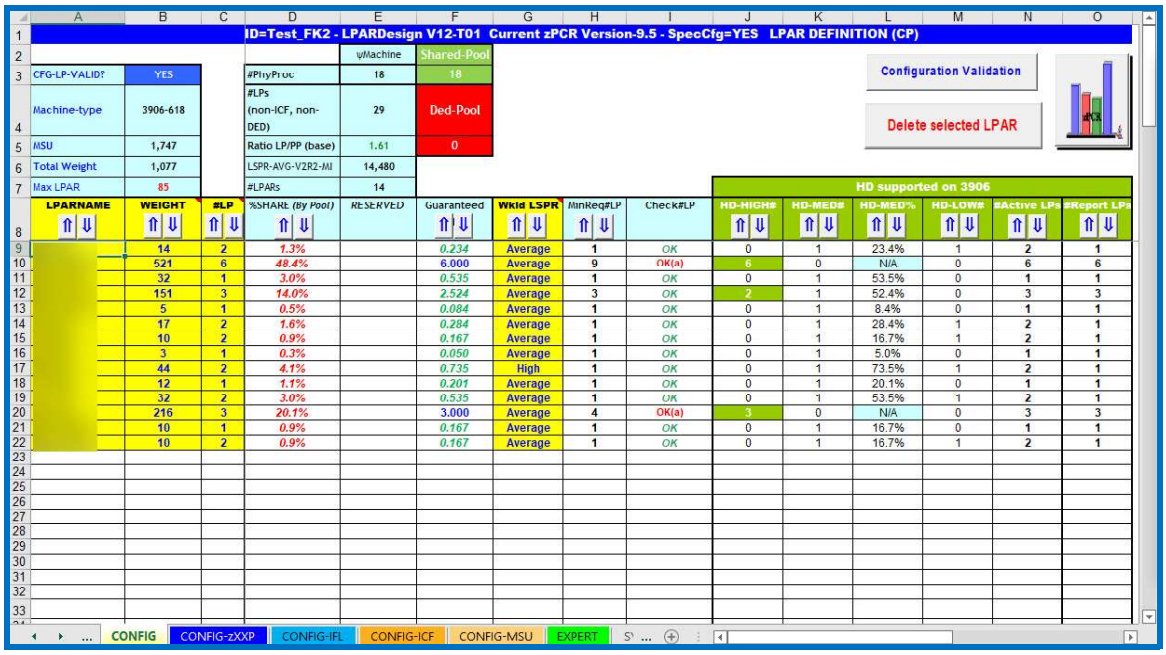
Finding LPARs with too few logical CPs to deliver the capacity guaranteed by their weight is a surprisingly common 'error'. You might think that it doesn't really matter that much (judging by how frequently we encounter this), however, the *LPAR weight* associated with the inaccessible capacity is *not* redistributed across the other LPARs when PR/SM is doing its HiperDispatch calculations.

For example, let's say that an LPAR's weight is 500 and that entitles it to 10 CPs worth of capacity, but the LPAR is only defined with 5 logical CPs. This effectively means that 250 weight units are going to waste from a HiperDispatch perspective. The *capacity* associated with those 5 logical CPs *is* available to other LPARs, but it would be made available to them in a less efficient way (via Vertical Low CPs for example), as opposed to redistributing the weights and possibly gaining VHs in other partitions.

In this example, if the LPAR in question *really* only needs 5 CPs' worth of capacity, you should reduce its weight, and allocate the freed-up weight units across other LPARs. Such an exercise might result in one or more other LPARs getting an additional Vertical High CP, which would help that LPAR and improve the overall efficiency of the CPC. On the other hand, if the LPAR really *does* need 10 CP's worth of capacity, then you should leave the weight as it is, and increase the number of logical CPs in the LPAR's Image Profile on the HMC.

Figure 23 on page 75 shows the LPAR Topology on the proposed configuration, assuming that you simply carry forward your LPAR definitions from the current CPC.

Figure 23 - CONFIG Worksheet for Target CPC



There are a number of items of interest:

- If you look at the HD-HIGH# column, you will see that the total number of Vertical High CPs, based on the current weights and number of logical processors in each LPAR is 11 (it was 4 on the current CPC).

- If you look at the 'Guaranteed' column, you will see that two of the cells are blue. That indicates that the number of logical CPs defined for that LPAR is not large enough to deliver the capacity guaranteed by the LPAR's weight.

  For example, if we look at the second LPAR in the list, the #LP is 6, the Guaranteed number is 6, but the minimum number of CPs required to deliver the guaranteed capacity is (MinReq#LP) is 9.

- Looking through the other LPARs, we don't see any that have a Guaranteed number that is just under a CP (for example, one with 3.**9** CPs). Such LPARs might benefit from a slightly higher weight that could give that LPAR an additional Vertical High. The Expert tab can be used to identify such LPARs - if it finds one, it provides a message like:

  lparname: (R3-GCP-base-Less Flexibility) - You have 1-VM with an entitlement of 95.1%

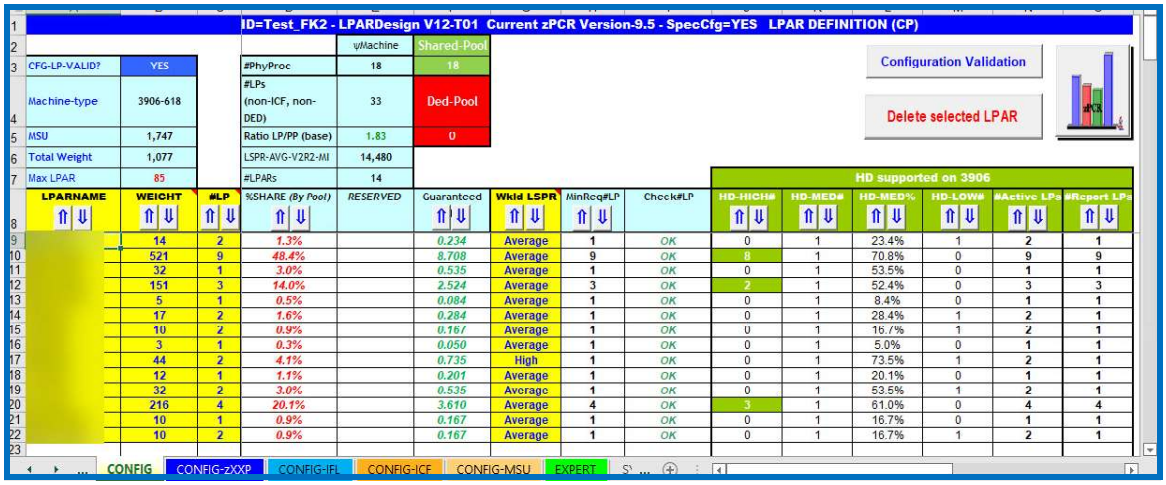
  Your current Guaranteed#PP is 8.951 - raising the Weight AND REMOVING 1-LP would give you 9-VH but with LESS FLEXIBILITY.

  The New Weight should be: 554 - The current Weight is: 550 - So you must decrease another LPAR Weight by: 4 to keep Total Weight(1106) constant.

In this case, we have two LPARs that need a larger number of logical processors if they are to receive the amount of capacity guaranteed by their weights. The second LPAR down has

a MinReq#LP of 9, and the third last one has a MinReq#LP of 4. Figure 24 shows the impact of increasing the number of logical processors to match those values.

Figure 24 - Optimized Target Configuration



CFG-LP-VALID?	YES		vMachine	Shared-Pool
Machine-type	3906-618	#PhyProc	18	18
MSU	1,747	#LPs (non-ICF, non-DED)	33	Ded-Pool
Total Weight	1,077	Ratio LP/PP (base)	1.83	0
Max LPAR	85	LSPR-AVG-V2R2-MI	14,480	
		#LPARs	14	

LPARNAME	WEIGHT	#LP	%SHARE (By Pool)	RESERVED	Guaranteed	Wkld LSPR	MinReq#LP	Check#LP	HD supported on 3906 HD-HIGH#	HD-MED#	HD-MED%	HD-LOW#	#Active LPs	#Report LPs
	14	2	1.3%		0.234	Average	1	OK	0	1	23.4%	1	2	1
	521	9	48.4%		8.708	Average	9	OK	8	1	70.8%	0	9	9
	32	1	3.0%		0.535	Average	1	OK	0	1	53.5%	0	1	1
	151	3	14.0%		2.524	Average	3	OK	2	1	52.4%	0	3	3
	5	1	0.5%		0.084	Average	1	OK	0	1	8.4%	0	1	1
	17	2	1.6%		0.284	Average	1	OK	0	1	28.4%	1	2	1
	10	2	0.9%		0.167	Average	1	OK	0	1	16.7%	1	2	1
	3	1	0.3%		0.050	Average	1	OK	0	1	5.0%	0	1	1
	44	2	4.1%		0.735	High	1	OK	0	1	73.5%	1	2	1
	12	1	1.1%		0.201	Average	1	OK	0	1	20.1%	0	1	1
	32	2	3.0%		0.535	Average	1	OK	0	1	53.5%	1	2	1
	216	4	20.1%		3.610	Average	4	OK	3	1	61.0%	0	4	4
	10	1	0.9%		0.167	Average	1	OK	0	1	16.7%	0	1	1
	10	2	0.9%		0.167	Average	1	OK	0	1	16.7%	1	2	1

You will see that the total number of Vertical High CPs has increased from 11 to 13. The two LPARs that had blue Guaranteed cells are now green, and the Guaranteed values for all LPARs are now less than their #LP number.

It is important to remember that LPAR weights only place a hard cap on an LPAR's capacity if you are using the (very) old Initial Capping function, and no one should be using that anymore - there are a variety of better ways to achieve the same end result. As long as you are not using Initial Capping, small changes in an LPAR's weight are unlikely to have a discernible impact on the amount of capacity used by the LPAR. However, changes as small as '1' in an LPAR's weight can change the number of Vertical High CPs, so it really is worthwhile to invest a little time in ensuring that your LPAR weights still reflect your desired distribution of capacity and that they optimize your HiperDispatch topology. We have found that using LPARDesign can be a big time saver when performing this exercise.

Comparing Configurations with LPARDesign Extended

And that brings us to the LPARDesign *Extended* tool. When you are fine-tuning your weights, you might have a number of options available to you (different combinations of weights and numbers of logical CPs). If you are looking at upgrade options, you will have a minimum of two configurations - your current one, and the target CPC. You can certainly model just about any change you would like, however you can only *view* one configuration at a time. It would be nice, and certainly time-saving, to be able to do a side-by-side comparison of two configurations. To address that need, Thierry and Alain developed the LPARDesign Extended tool as an add-on to the original LPARDesign tool.

The LPARDesign Extended tool (and its User's Guide) is downloaded from the same github page as LPARDesign and has a similar look and feel as LPARDesign. It uses XML files created by LPARDesign as its input. You can use LPAR Design Extended to:

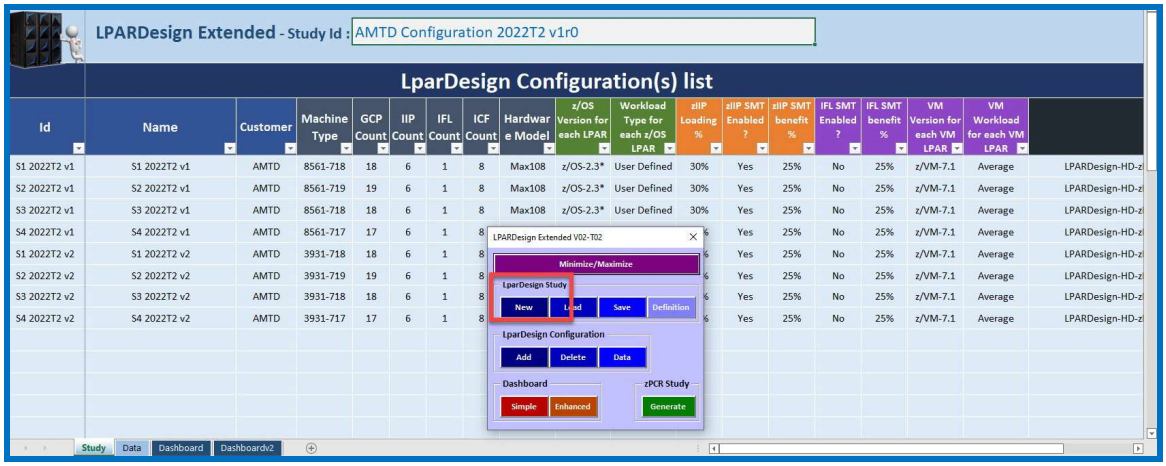
- Load multiple LPARDesign configurations and then do a side-by-side comparison.
- Visualize an LPARDesign configuration for a CPC by displaying both GCPs and zIIPs on a single screen.
- Combine multiple LPARDesign configurations into a single .zpcr file that can then be loaded into zPCR.

To use LPARDesign Extended, open the downloaded xlsm file. This will present you with a sheet similar to that shown in Figure 25. You will see the sample spreadsheet contains eight configurations. LPARDesign Extended provides two options for loading your *own* configurations:

- You can add configurations one at a time from an XML file that you created using LPARDesign.
- You can load a study file (also XML format) that you created previously using LPARDesign Extended.

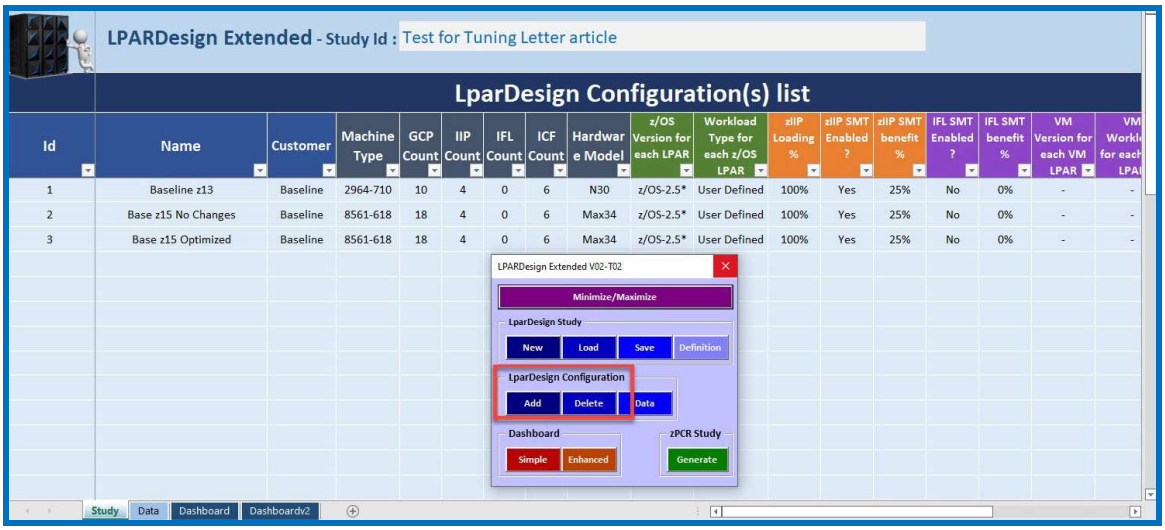
Because this is our first time to use LPARDesign Extended, we'll start with the first option (which uses the output from LPARDesign), and come back to Loading an existing study file a little later. Click on 'New' in the LPARDesign Study section of the Navigation Bar, as shown in Figure 25.

Figure 25 - LPARDesign Extended Initial Screen



To load the configurations you created using the LPARDesign tool, click on Add in the LPARDesign Configuration section, as shown in Figure 26 on page 78.

Figure 26 - Adding a Configuration to LPARDesign Extended

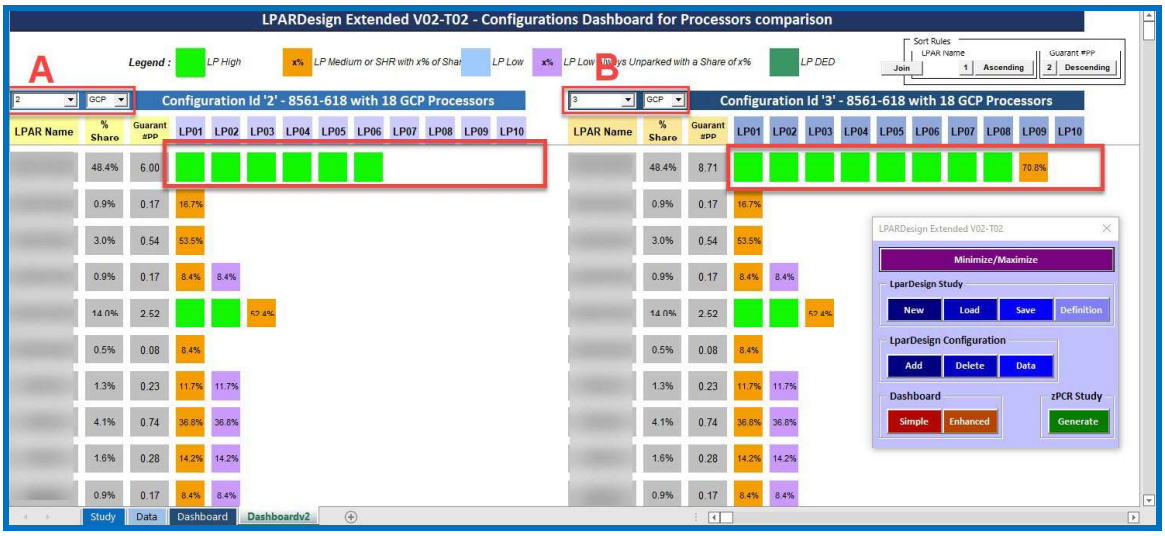


The tool will ask you for an ‘ID’ and ‘Name’ for each configuration (the value in the Customer column is obtained from the XML file, which in turn was specified by you when you created the configuration in LPARDesign). You can use any values you want, but we recommend using a different ID for each configuration - you will use the ID when you are selecting the configurations you want to compare. We used the CPU type and a sequential number (for example, 3931_100) and that worked well for us, but you can use the ID and Name fields in whatever would be the most meaningful way to you. Then you click on “Select a LPARDesign Configuration File” and navigate to and select the XML file that contains the first configuration file you want to load. Then click on “Go to zPCR elements definition” and complete the requested information (any zPCR information in the original LPARDesign configuration is *not* carried over in the XML file).

You go through that process once for each configuration you want to load. In this example, you can see that we’ve loaded three configurations - the current z13 one, the z15 one using the original weights and number of logical CPs, and the z15 one with the adjusted number of logical CPs.

The Dashboardv2 tab lets you visually compare the two configurations, side by side. You select the configurations you want to use by clicking the ID and engine type dropdowns as shown in Figure 27 on page 79 - the two places where you enter this information are marked ‘A’ and ‘B in the figure.

Figure 27 - Side-by-Side Configuration Comparison Tab



In this example, you can see that the first LPAR has six Vertical High CPs in the configuration shown on the left, and 8 Vertical Highs and one Vertical Medium with a 70.8% share in the config on the right. To change the configurations that you want to compare, update the ID and/or engine type dropdowns.

Use the Navigation Bar Save button to save multiple configurations in an LPARDesign Study XML file.

Note that LPARDesign Extended is intended to let you *view* configurations that you create using LPARDesign. It doesn't provide any functions to *update* the configuration. However, if you would like to come back and review these configurations again in the future, you can save the current set of configurations as an LPARDesign Study - you can then load all configurations at one time, rather than having to add them one at a time as we did here. To create the LPARDesign Study XML file, click on Save in the Navigation Bar (beside the New button we used earlier) - that will save all the configurations in a single XML file. You subsequently load all those configurations into LPARDesign Extended using the Load button in the Navigation Bar.

Having compared a variety of configurations, if you decide on a subset of configurations that you would like to load back into zPCR, click on the green Generate button in the Navigation Bar and then select the subset of LPARs that you want to pass to zPCR.

Other Related Information

Limitations

While LPARDesign makes it much easier to see the effect of proposed changes to your LPAR weights or the number of logical processors, one thing that you are still responsible for is ensuring that the combined weights of all LPARs add up to the same number after you make your changes *if that is important to you*. Many installations aim for the weight of all LPARs to sum to 1000. The advantage of that methodology is that it lets you quickly see the relative share that each LPAR is guaranteed. For example, if all weights sum to 1000, you can look at an LPAR with a weight of 49 and immediately know that it is 'guaranteed' 4.9% of the CPC capacity.

However, there are a few alternatives:

- The weights don't *have* to sum to a round number. For example, you can use the CONFIG sheets in LPARDesign to see the guaranteed share of each LPAR (in the %SHARE column). However, I have to admit that this is not as simple as just being able to look at the LPAR's weight and knowing its share from that one number.
- In his excellent SHARE 2022 in Dallas '*A Different "Weigh" to Define LPAR Weights*' presentation, **Jim Horne** described an alternative methodology that he uses, based on the number CPs (or zIIPs or IFLs or ICFs) on the CPC. Jim kindly provided an article for our readers, illustrating his concept for those that were not able to attend his SHARE session - see 'A Different Way to Define LPAR Weights' on page 82.
- If enough LPARDesign users ask Thierry and Alain, they *might* be willing to investigate enhancing the tool to let you specify a desired total weight, and the tool would then adjust the weights so they total up to your desired value.

One other thing to bear in mind is that while LPARDesign helps you optimize the HiperDispatch topology, it is not able to predict how those changes will impact the LSPR workload characterization of the affected LPARs. As a generalization, you would expect an increase in the number of Vertical Highs to result in an improvement in the LPAR's workload categorization, however it is not possible to accurately predict the change. Even IBM's zPCR tool is unable to perform such modeling.

Also, note that the LPAR-level information on the CONFIG-MSU sheet reflects the published MSU value for that CPC, and that LPAR's %SHARE of that CPC. While the tool lets you alter an LPAR's workload category, it does not take that change into account when calculating the MSUs available to the LPAR.

References

For more information about optimizing your HiperDispatch topology, refer to the following articles and presentations:

- *The LPARDesign and LPARDesign Extended Tools Part 1* and *Part 2* presentations on the LPARDesign github page.
- The *LPARDesign* and *LPARDesign Extended Users Guides* on the LPARDesign github page.
- 'LPAR Design Tool' article in *Tuning Letter 2017 No. 4*.
- 'HiperDispatch Questions and Answers' article in *Tuning Letter 2015 No. 4*.
- 'CPU MF Part 3 - Optimizing CPC Cache' article in *Tuning Letter 2017 No. 4*.
- SHARE 2019 in Pittsburgh, Session 25634, *Customer Experiences Saving MSUs Through CPC Optimization*, by Todd Havekost and Frank Kyne.
- SHARE 2020 in Fort Worth Session 27021, *Understanding LPAR Controls for Better Performance*, by **Kathy Walsh**.
- IBM Techdoc 6354843 *Number of Logical CPs Defined for an LPAR* by Kathy Walsh.

Summary

I find more and more people saying to me "nothing can ever be simple any more". And this doesn't only apply to IT - the most recent person to point this out was a waiter in a tiny restaurant, wondering why he needs to enter customer orders on a tablet rather than his little paper notepad that he had used for the last 30 years. It is certainly a reasonable observation that life in general is getting more complex. Thankfully, we have people like Thierry and Alain creating tools like LPARDesign and LPARDesign Extended to make one part of our life (managing LPAR weights) a little easier.

If you have never used these tools, we highly recommend that you take them for a test drive. Like all tools, they take a little while to get familiar with. But before you know it, you will find yourself using the tools to understand the impact of small changes in LPAR weights and configurations, and then eagerly moving the result over to zPCR to see what difference it predicts your changes will make. Between these tools and **Jim Horne's** LPAR weight management methodology described in 'A Different Way to Define LPAR Weights' on page 82, we are sure you will achieve a more efficient HiperDispatch configuration *and* free up some valuable time - that you can then use to figure out how to use your latest 'productivity' tool☺.