Multiple Shared Processor Pools
By Jaqui Lynch

Micro partitioning and shared processor pools have now been around since POWER5. Micro partitioning allows LPARs to share processor resources while still being guaranteed a certain amount of resource. The use of the shared processor pool (SPP) takes this one step further and allows you to manage processor resources for several LPARs that are grouped together into one processor pool. The shared processor pool capability requires PowerVM Standard as a minimum.

By default, all micro partitions run in pool 0 which is the default pool. All cores are part of the shared processor pool unless the LPAR using them is running dedicated or is powered off and the administrator has told the system to not make those cores available to anyone else. In POWER5 there was only one shared pool. For POWER6 and higher there is always at least 1 pool (pool 0) but there can be up to 64 pools. This capability is called multiple shared processor pools (MSPPs) and it isolates work-loads in a shared processor pool and prevents the work load from exceeding an upper limit. For applications that use sub-capacity licensing this can save you significant amounts of money.

The concept of micro partitioning is well understood as are the concepts of entitlement and virtual processors. However, things change a little when you start using MSPPs. Now you have the ability to pool LPARs into different groups depending on their needs – this allows you to isolate workloads for licensing reasons and it allows you to group LPARs together in a logical manner. As an example you may have a group of LPARs that each peak at 2 cores but they do not peak together. If you were designing this unshared then you would need a lot more cores than if you put them in a properly sized pool and let them share the resources as they need them. The key is to ensure that the pool is sized for the peak processor units needed.

The pools need to be defined before they can be allocated to an LPAR. This is done using the HMC by going into System management, Servers, selecting the system and then clicking on Task, Configuration, Virtual Resources and finally Shared Processor Pool Management. From here you can click on a pool and rename it and change its settings with the exception that the default pool 0 cannot be changed. The first step is to set the pool name to something meaningful – if this is your oracle pool then call it something like orapool. The name must be unique.

There are two additional fields you need to set – the first is reserved processing units and the second is maximum processing units. Maximum processing units defines the upper boundary of processor units that the LPARs in this pool can use. So if you are licensed for 4 cores total for your application then you would set this to 4. Reserved processing units is the number of processing units that

you want to reserve for uncapped LPARs in the SPP. You can leave reserved at 0. PowerVC has now been updated and supports MSPPs as of version 1.2.3 fixpack 2.

Once the pools are set up you can then click on the partitions tab and reassign existing micro partitions from their current pool to the new one you just created. If you decide you no longer want to use a pool you should first reassign the LPARs in the pool to other pools and then unconfigure the SPP by setting its maximum and reserve to 0 units.

One key item to understand is that all processor activations actually take place in the default pool 0. The other pools are basically a mask over the default pool and are used to cap utilization of the underlying cores by the LPARs in the assigned pool.

It is also possible to migrate an LPAR that is in an SPP but there are two conditions that must be met. There must be an SPP on the target machine that has the same name and the SPP on the target must have sufficient capacity for the LPAR being migrated.

Monitoring SPPs

When creating the LPAR it is important to go into properties after the LPAR is created and check the box for “Allow performance information collection”. Without this being checked you will not be able to monitor the whole box and the statistics shown by various commands, including nmon, for APP (available pool) are not valid. Every LPAR needs to have this value set for proper monitoring. You can now use topas -C (topas -cecdisp on VIO) or nmon to monitor usage of the pools. In topas press p to see the pools. There are three fields you are interested in:
maxc – maximum pool capacity for each pool
app – the number of available processors in the pool
physb – the sum of physical busy for processors in shared LPARs in the pool
You can also go down to the pool you are interested in and press the f key to get details of the LPARs within the pool.

Variable Capacity Weighting

This is also referred to as uncapped weight. VCM is specified in the LPAR profile and each micro partition is assigned a weight from 0 to 255 with 128 being the default. Typically, we run VIO servers at 255, production and something like 160 to 192 and everything else at the default or less. The idea of VCW is to determine who gets resources when LPARs want to exceed their entitlement. The higher the uncapped weight the more the system biases the allocation of additional processor resources to that LPAR. As an example if LPAR1 has a weight of 128 and LPAR2 has a weight of 128 and they are both requesting additional resources then one of two things will happen

1. If there are sufficient resources, then they both get what they request

2. If there are not sufficient resources, then each will get 50% of what they request as they have the same weight. The weights get summed and then the LPAR is given a bias which consists of its weight over the total – in this case 128/256=128.

The expectation was that the VCW would provide active weighting between LPARs in the pool at all times. However, this was not correct and has led to a lot of confusion. With firmware 8.4.0 this has changed.

Firmware 8.3.0 and earlier

At 8.3.0 and prior, uncapped weight is only used when there is contention for resources. If there is no contention, then available resources are distributed to the virtual processors immediately regardless of the uncapped weights for the LPARs. Effectively, they are equally biased. If there is contention, then the weights are evaluated and the hypervisor uses those weights to bias how it allocates the resources. Basically, the SPP has to be constrained before the VCW is used.

Firmware 8.4.0 (Nov 2015) and later

At 8.4.0 the hypervisor considers VCW at all times. If multiple LPARs are assigned to a SPP the system now uses the uncapped weight to determine how processor resources will be distributed to the LPARs in the pool. Basically, at 8.4.0 weighting in the SPP is enabled even when there is no contention within the pool. This may impact some environments because the hypervisor is now more proactive about managing uncapped capacity.

Licensing

MSPPs can be used to provide licensing savings when the application supports sub capacity licensing. This means that you could have a license for 4 cores but have 8 LPARs sharing them as long as the maximum that can be used at any one time is 4 cores. I highly recommend that you talk to any application vendor that you are working with and find out if they support sub capacity licensing as there can be considerable savings. Make sure they understand that you are talking about having a pool of cores shared by more than one LPAR running their application and that you can control it so that they cannot exceed the maximum cores you specify.

Summary

The use of Multiple shared processor pools is extremely useful when consolidating workloads as it allows you to group LPARs together such that they can share resources without you having to oversize the server. Most customers use MSPPs to control licensing costs allowing high and low priority work to share cores and using weighting to provide preference to the higher priority work. As of firmware 8.4.0 those weights are always reviewed by the hypervisor which matches the customer's expectations. MSPPs can make a significant difference to bot performance and costs so I highly recommend reviewing where they might be appropriate in your environment.

References

PowerVM Managing and Monitoring – SG24-7590-04
https://www.redbooks.ibm.com/redbooks/pdfs/sg247590.pdf

PowerVM Introduction and Configuration – SG24-7940-05
https://www.redbooks.ibm.com/redbooks/pdfs/sg247940.pdf