

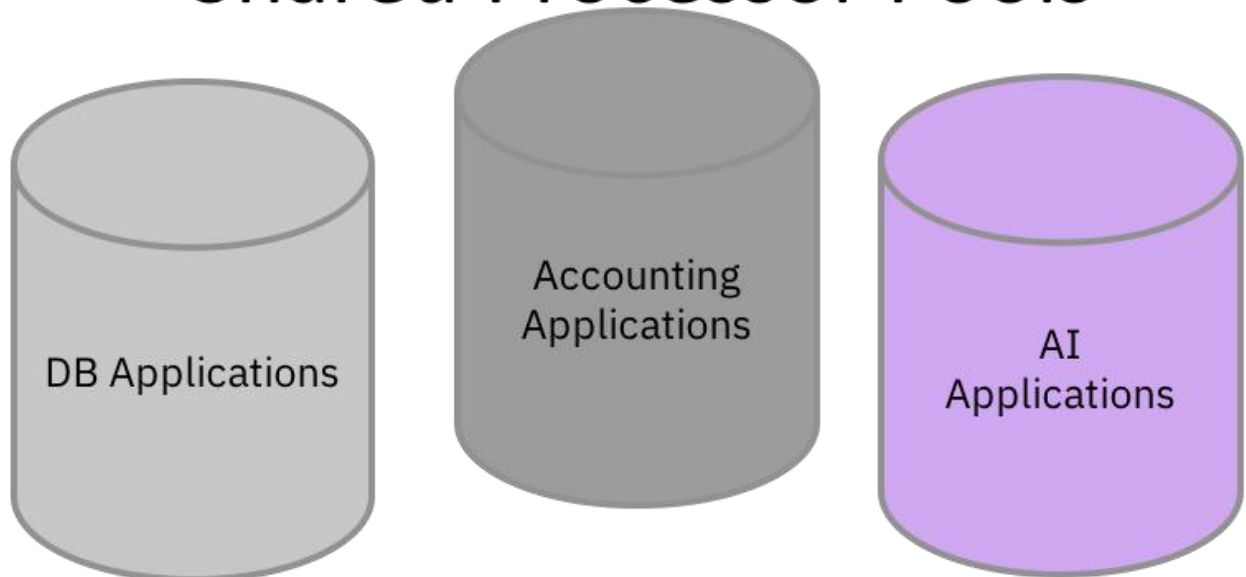
Using Shared Processor Pools to manage CPU resources

Note: This document was originally copied from

<https://www.ibm.com/developerworks/community/wikis/home?lang=en-us#!/wiki/Power%20Systems/page/Using%20Shared%20Processor%20Pools%20to%20manage%20CPU%20resources>

Author: Peter Heyrman

PowerVM Shared Processor Pools



Do you have situations where you need to limit the CPU consumption to ensure compliance with a software license? Do you want to ensure that non-production work does not interfere with production work on the same server? Shared processor pools are a technology available in PowerVM that can be leveraged to handle these challenging situations.

Basics of Shared Processors Partitions

When creating a partition you can create the partitions with dedicated CPUs or as a shared processor partition. With a dedicated processor partition, there is a defined maximum amount of capacity that is always available to that partition which is based on the whole number of CPUs allocated to the partition. If you create a shared processor partition, you can specify in fractional CPU quantities the amount of CPU guaranteed to be available to the partition (entitled CPU capacity). If you designated the partition as capped, the most CPU that can be consumed by the partition is defined by the entitled CPU capacity. If these were the only two types of partitions supported, limiting CPU consumption would be easy as the fixed amount assigned to the partitions has an upper limit.

There is a third type of processor configuration which is a shared uncapped partition. Shared uncapped is the most popular and flexible CPU configuration because the partition can be guaranteed a specific amount of CPU capacity coupled with the flexibility to consume CPU above the guarantee. With an uncapped partition the overall CPU that a given partition can consume is limited by the virtual CPUs defined for the partition. For example, a shared uncapped partition is created with a desired entitlement of 1.5 processors, maximum entitlement of 2.0, desired virtual processors of 3 and maximum virtual processors of 5. Since the partition is configured with 3 desired virtual processors, the CPU consumption is limited to 3.0 processor units. If, though the use of dynamic LPAR (DLPAR), you change the desired CPUs from 3 to 4, the limit on consumption of the partition will grow to 4.0 processor units.

For uncapped partition to actually receive CPU time above the entitled capacity there needs to be available CPU in the server. Available CPU can come from:

- Other shared processor partitions that have not fully consumed their entitled capacity. For example, if the desired entitlement is set to 1.5 processor units, but the partition is only using 0.80 processor units, there are 0.70 processor units of uncapped capacity available
- Dedicated partitions that share CPU time with other partitions
- Processors that are licensed but have not been assigned to either shared or dedicated partitions
- Shared processor pool reserved units not consumed by partitions assigned to a given shared processor pool
- Utility COD

When defining a partitions as uncapped, the partition is assigned an uncapped weight. The uncapped weight is a way to indicate to the hypervisor a priority in the assignment of available CPU time across the various uncapped processor partitions. Later in this blog there will be more details about how uncapped weight is used by the hypervisor.

Basics of Shared Processor Pools

A PowerVM based server always has at least one defined shared processor pool which is named the *DefaultPool*. The default pool is special in that there is no upper limit on the CPU consumption of partitions assigned to the default pool. Also, if you do not explicitly specify a shared processor pool for a partition, the partition is placed in the default pool.

In addition to the default pool, PowerVM supports up to 63 user defined shared processor pools. The default name of the pools are *SharedPool/01* through *SharedPool/63* but you can rename this pools to names that make sense to your business needs. For example, you could have a pool

for database work named *DB-Pool*. You could also have names like *Web-Pool*, *Prod*, *Non-Prod* and so on.

Along with the pool name there are two other attributes for the pool, maximum processing units and reserved processing units. Maximum processing units is a whole number that limits the overall consumption of all the partitions in the pool to the specified value. For example, if you had three uncapped partitions with 4 VPs each, in the default shared pool the total CPU consumption of these three partitions could be as high as 12 processors. If you placed these three partitions into a shared processor pool with a limit of 6, the hypervisor would ensure that the sum total of the CPU consumption of the three partitions was limited to 6 processors. This is valuable to many customers to maintain software license agreements or to limit consumption of some partitions for business reasons. One common misconception surrounding shared processor pools is that the maximum processing units actually reserves physical processors for this group of partitions or has some effect on the affinity of the resources assigned to the partition. Neither of these are true, in fact if you had a server with 16 cores, the sum total of the maximum processing units across all the pools can exceed the 16 physical cores. Net, there is no direct tie between the maximum values and physical resources.

In addition to maximum, there is a value for reserved processing units. Reserved units can be set aside to be used as additional capacity for the partition in a specific shared processor pool. From a hypervisor dispatching point of view, a partition would first consume its entitlement, the partitions would then consume the reserve capacity and finally the partitions in the pool would compete for server-wide available uncapped capacity. The drawback in the use of reserved processing units is that these processing units are actually assigned similar to entitled capacity so designating reserved processor units will reduce the overall processor units that can be assigned directly to partition.





Managing Partitions and Shared Processor Pools

The HMC has a simple to use interface to manage shared processor pool attributes and partitions. One of the views is the pool view

Shared Processor Pool: myServer

Pools | Partitions

Select a pool name from the table to modify pool attributes





 --- Select Action ---

Pool Name ^	Pool ID ^	Reserved Processing Units ^	Maximum Processing Units ^
DefaultPool	0	N/A	N/A
SharedPool01	1	0	0
DB-Pool	2	0	10
Web-Pool	3	0	5
SharedPool04	4	0	0
SharedPool05	5	0	0
SharedPool06	6	0	0
SharedPool07	7	0	0
SharedPool08	8	0	0
SharedPool09	9	0	0
SharedPool10	10	0	0
SharedPool11	11	0	0
SharedPool12	12	0	0
SharedPool13	13	0	0
SharedPool14	14	0	0
SharedPool15	15	0	0
SharedPool16	16	0	0
SharedPool17	17	0	0
SharedPool18	18	0	0
SharedPool19	19	0	0

Total: 64 Filtered: 64





OK Cancel Help

Select the shared pool you want to modify and options are presented to set the name, reserved processing units and maximum processing units. One note is that for a pool to actual be considered in use, you must set a non-zero value for maximum processing units. The other shared processor view is the partition view

Shared Processor Pool: myServer

Pools **Partitions**

Select a partition name from the table to move the partition to a different pool





 --- Select Action ---

Partition Name (ID) ^	Pool Name (ID) ^	Assigned Processing Units ^	State ^	Workload Group ^
hype13 SLES12 SP2 LE(13)	DefaultPool(0)	0.00	Not Activated	None
hype21(21)	DefaultPool(0)	1.00	Running	None
hype17 aix(17)	DefaultPool(0)	0.00	Not Activated	None
hype11 aix(11)	DefaultPool(0)	0.00	Not Activated	None
hype14 aix(14)	DefaultPool(0)	0.00	Not Activated	None
hype20(20)	DB-Pool(2)	0.00	Not Activated	None
hype16(16)	Web-Pool (3)	0.00	Not Activated	None
Total: 7		Filtered: 7		

OK Cancel Help

By selecting individual partitions, you can dynamically move the partitions between the various shared processor pools. There is also a partition profile attribute that can be utilized to define the shared processor pool for a given partition.

Technical details of hypervisor dispatching of uncapped partitions

The PowerVM hypervisor is responsible for deciding when and which virtual processors for given partitions are dispatched to run on the physical processor cores. The hypervisor maintains the state of every virtual processor on the server and a virtual processor is only dispatched on a physical processor when the virtual processor is ready to run. It could be the virtual processor just received an external interrupt from an I/O device, maybe a timer popped or some other event that indicates there is work for the virtual processor. The hypervisor maintains a list of ready to run virtual processors and in situations where there are more virtual processors ready to run than available physical cores, the hypervisor runs a lottery to determine which virtual processor should be dispatched. Let say there are only 2 partitions ready to run, partitionA has an uncapped weight of 1 and partitionB has an uncapped weight of 99. The hypervisor conceptually puts 1 ball into a drum for partitionA and 99 balls in the drum for partitionB and then reaches into the drum to select a lottery winner. So, as you can see, partitionA has 1 chance in 100 of being dispatched and partitionB has 99 chances in 100. When the hypervisor does an uncapped dispatch, the partition is allowed to run for a short period of time (less than a millisecond), the partition is interrupted and the lottery is re-run if required (only have to run the lottery if more virtual processors are needed than available physical processors).

For partitions assigned to one of the non-default shared processor pools, if the maximum processing units defined for the shared processor pool is reached, the partitions in the pool must wait until the next hypervisor dispatch window (i.e. the partition are NOT considered ready to run because they have exhausted the CPU limit of the pool). Currently the PowerVM hypervisor has a 10ms dispatch window so every 10ms every non-default shared processor pool is refreshed with a new allotment of CPU that can be consumed in the next 10ms window.

For the non-default shared processor pool, the hypervisor performs the uncapped dispatch lottery across all the partition in the entire server. Net, when configuring the uncapped weight, the priority applies across all the uncapped partition and is not scoped to an individual shared processor pool.

Also, in FW840 a change was made in the hypervisor to improve the way dispatching worked within a shared processor pool. Let say there was production partitions in the pool with an uncapped weight of 99 and test partitions in the same pool with an uncapped weight of 1. In the past, as long as there was available uncapped CPU available in the server, both the production and test partition were dispatched to run. In many situations this led to test partitions being given the same amount of CPU time as the production partitions. A change went into FW840 and later to correct this deficiency such that the hypervisor keeps track of a little history about the CPU to weigh the lottery more in favor of production partitions over the development partitions.

Contacting the PowerVM Team

Have questions for the PowerVM team or want to learn more? Follow our discussion group on LinkedIn [IBM PowerVM](#)