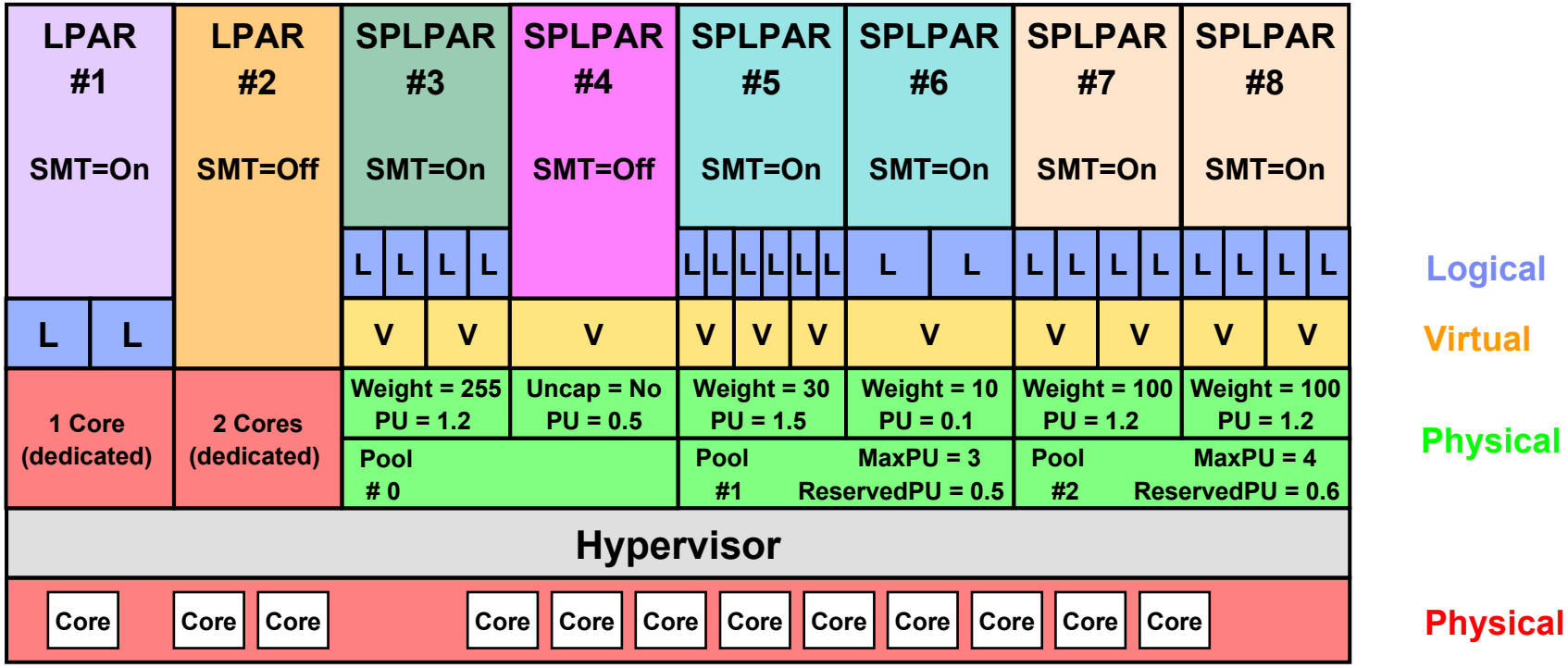
June 2012 @Dubai
IBM Power Academy

IBM PowerVM processor virtualization

Luca Comparini
STG Lab Services Europe
IBM FR

June,13th 2012
@IBM Dubai

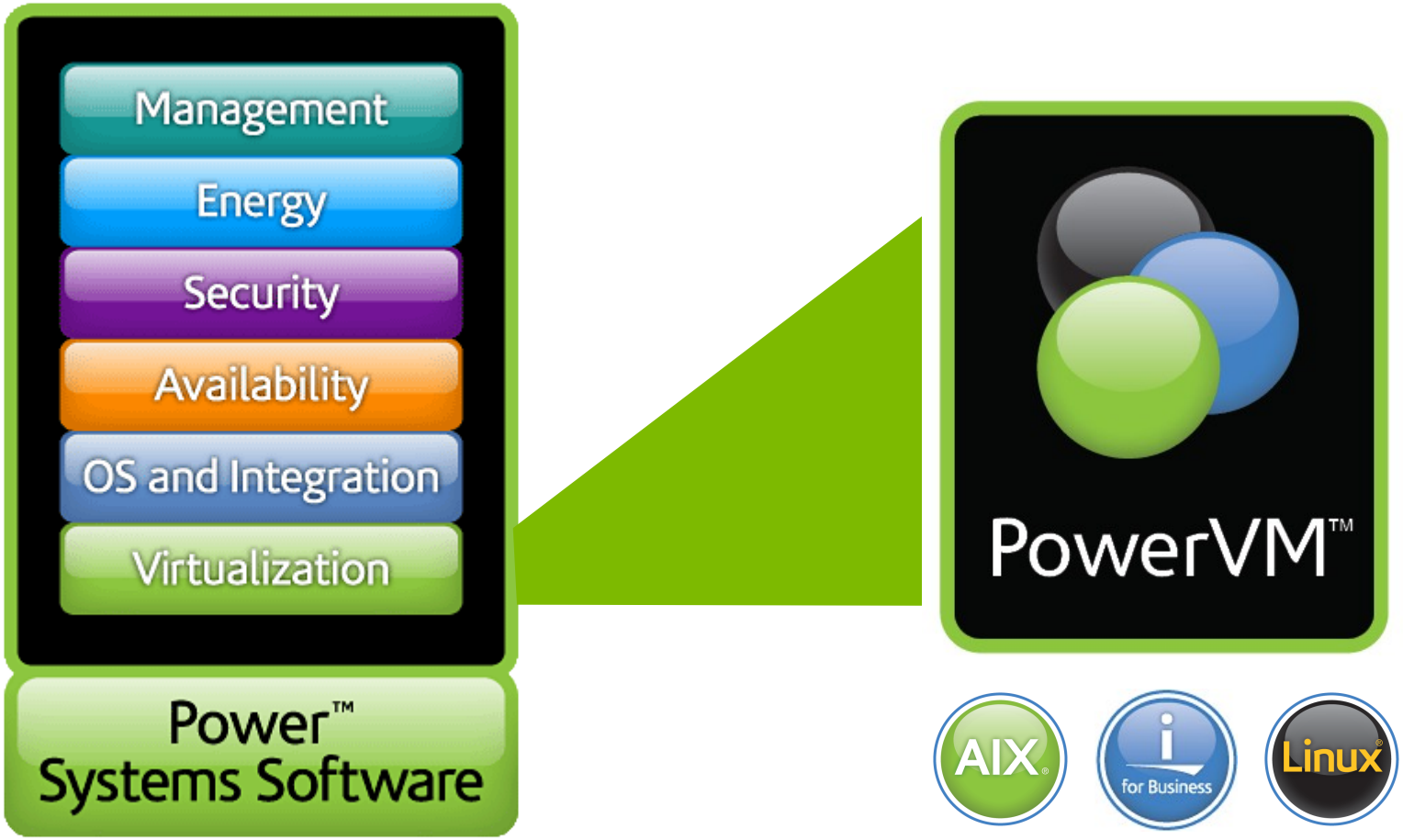
Objective of the session: understand this chart



Agenda

- Power Systems processor partitioning capabilities by platform
- The shared processor pool
- The difference between physical, virtual, and logical processors
- SPLPAR processor minimum, desired, and maximum settings
- Recommendations for SPLPAR processor settings
- Multiple Shared Processor Pools
- Suggestions for shared processor pool settings

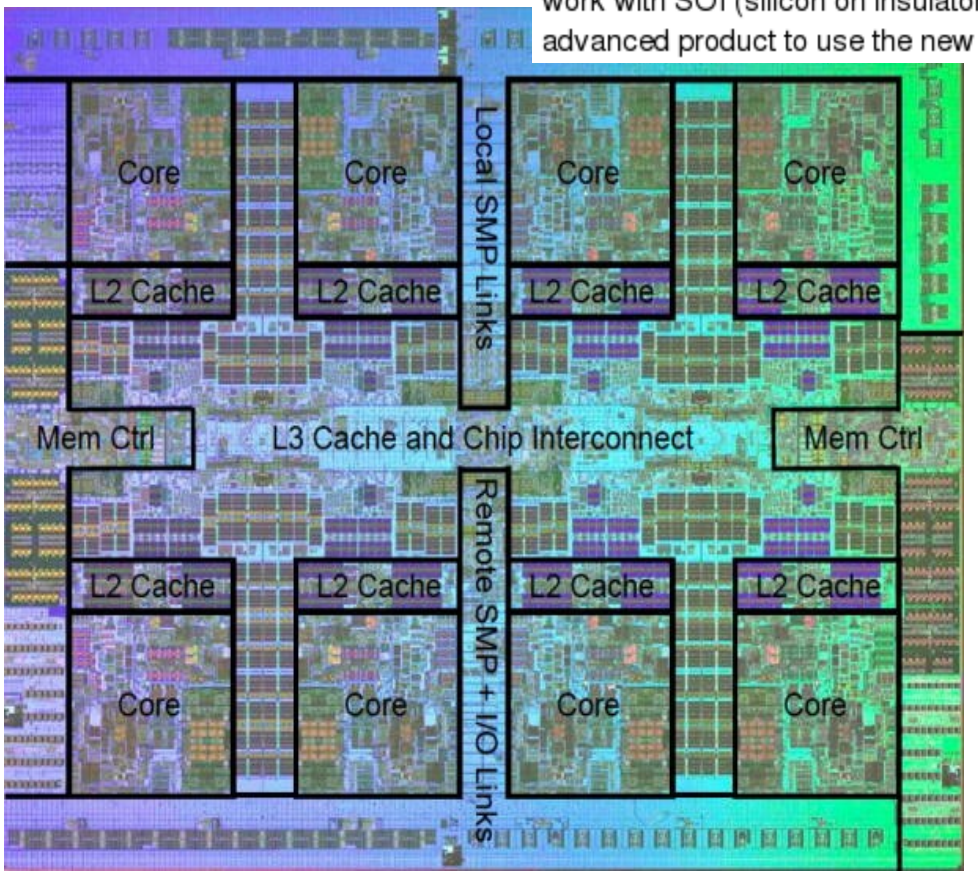
Power Systems virtualization = PowerVM



Why (processor) virtualization

news.cnet.com/8301-13512_3-10321740-23.html

What makes the Power7 so powerful? Each chip has eight cores, and each core supports four-way multithreading. There's 32MB of level-3 cache on the chip, made using embedded DRAM (eDRAM) cells. Most CPUs use SRAM for cache because it's generally easier to combine with high-performance logic, but DRAMs--with only one transistor per bit--offer compelling density advantages. IBM spent years developing a new kind of eDRAM that would work with SOI (silicon on insulator) manufacturing processes, and the Power7 is the most advanced product to use the new technology.



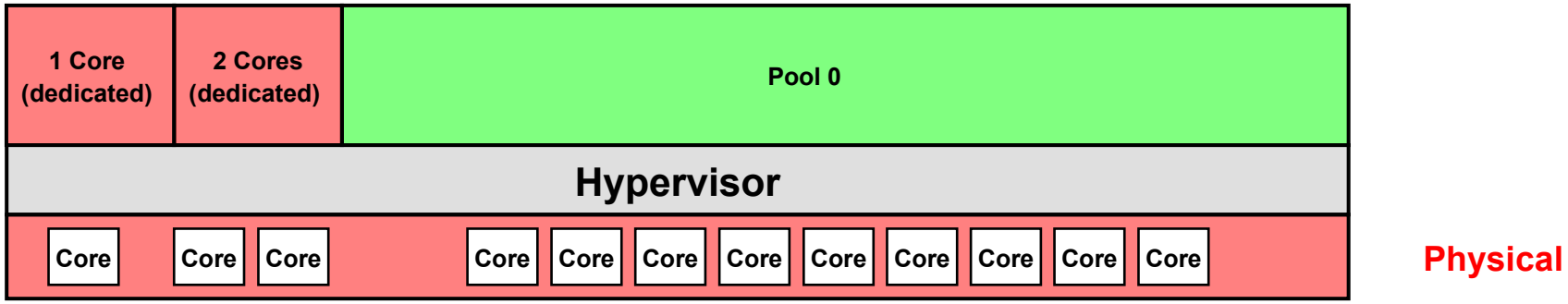
Do you really want to use one of this just for 1 application?

Without virtualization,
1 LPAR = 1 system

The basis of processor virtualization: Dedicated vs Shared

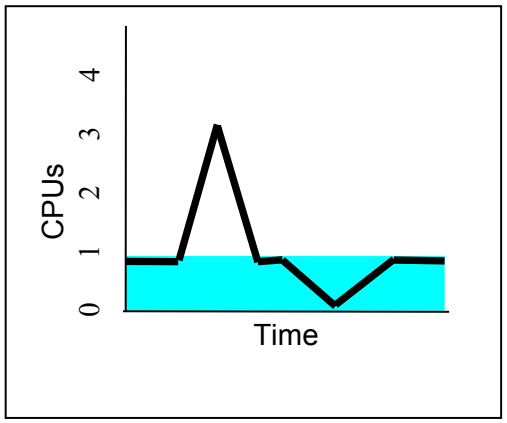
Dedicated: Need to have excess capacity to handle peak usage. LPARs have whole CPU assigned.

Shared: Excess capacity is shared and can be drawn upon when needed. Thanks to Micro Partion technology an LPAR can have fractions of CPUs assigned.

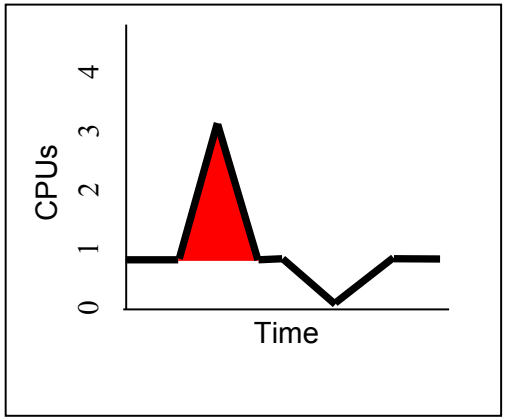


LPAR behaviour in shared mode

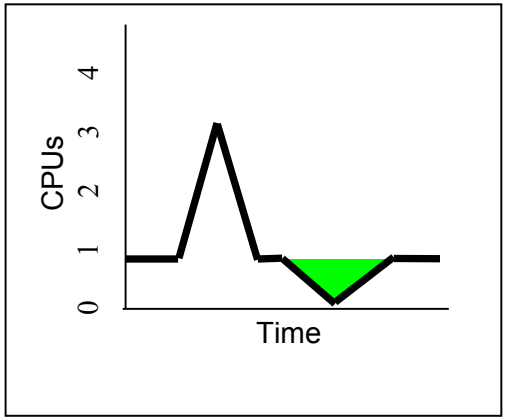
Each LPAR is operating in one of these modes



Guarantee



Using extra CPU cycles from the shared processor pool



Donating extra CPU cycles to the shared processor pool

We guarantee enough CPU to handle most of the workload to be processed.

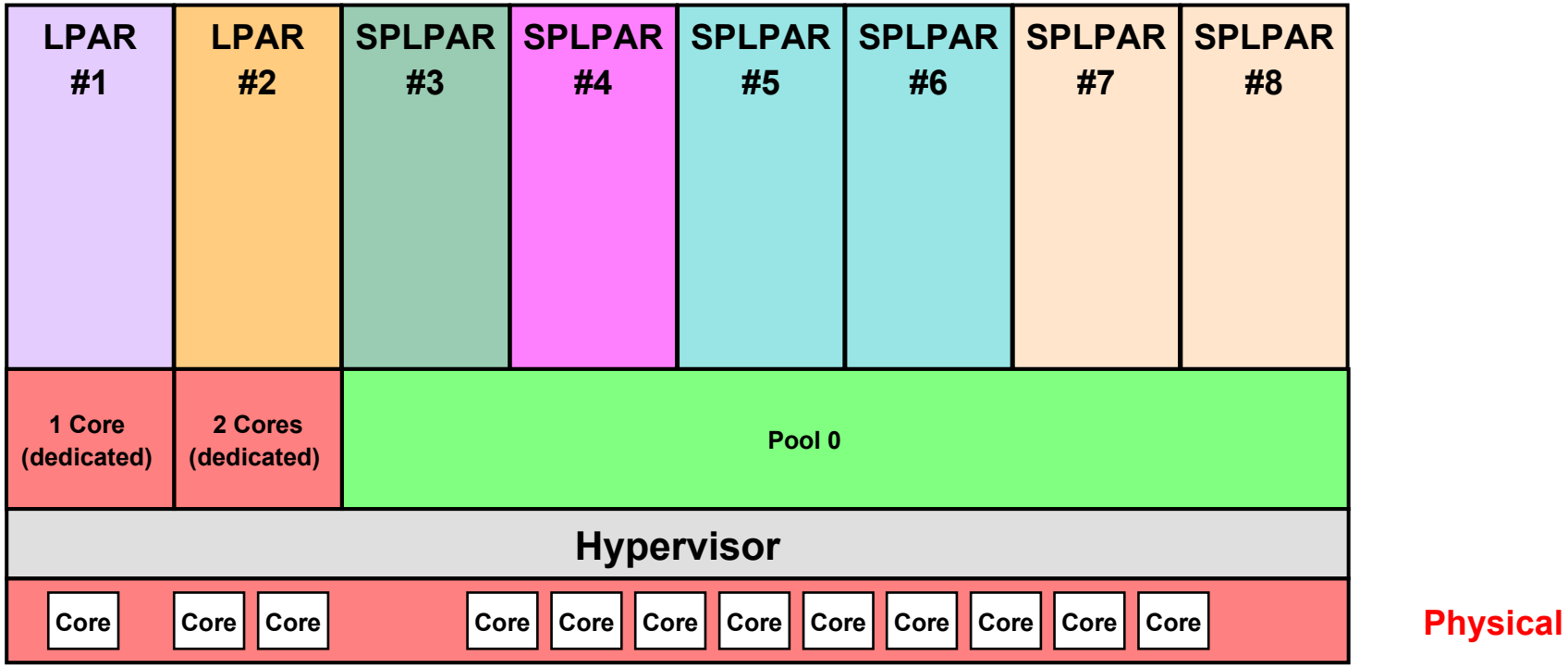
The red shows that at some times during the day we will need to “borrow” or “use” extra cycles from the pool.

The green shows that during quiet times for our application, this LPAR will not be using all of its guarantee and therefore extra cycles (“Donor”) will be available for use by other LPARs.

In reality, the LPAR never really owns the CPU cycles, the LPAR is just guaranteed to have access to a certain number of cycles from the pool.

Shared Processor Pool

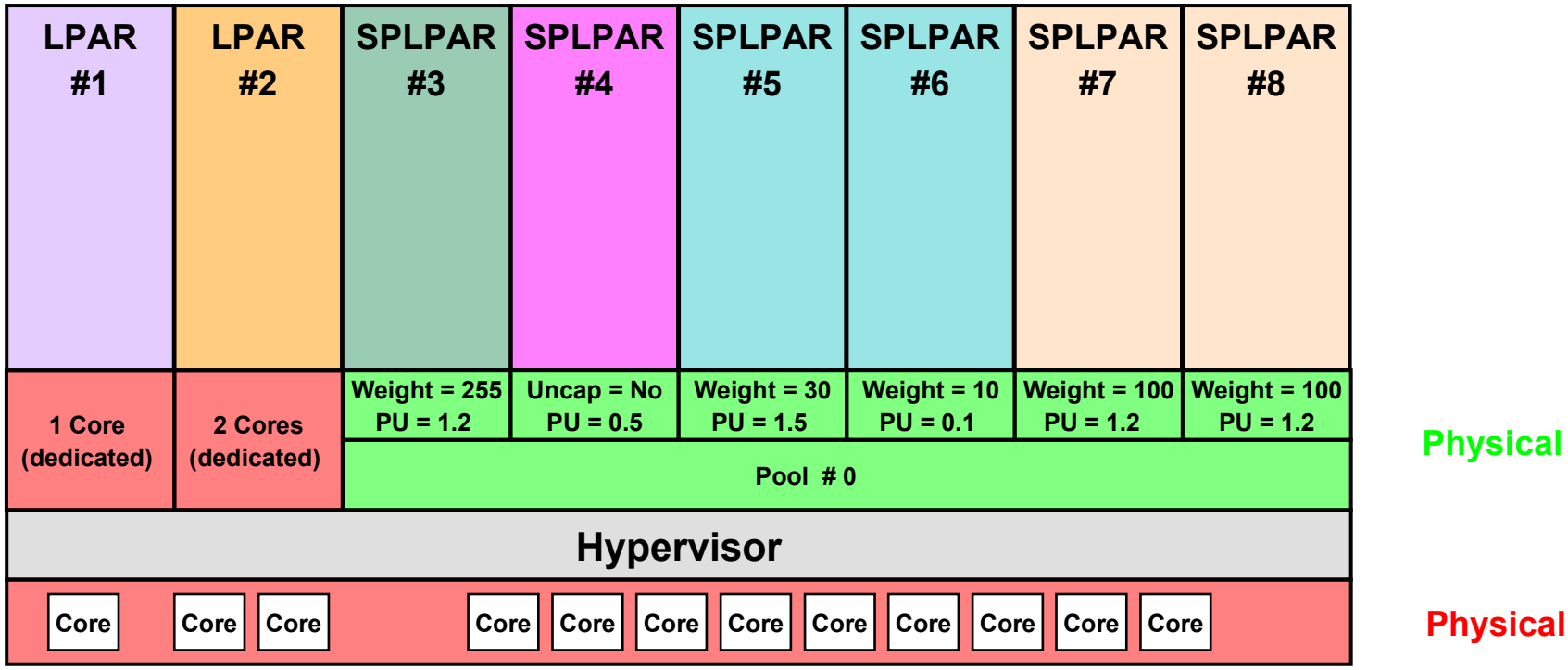
Shared Processor Pool LPAR are based on Micro Partitioning technology



Learning points: (1) All activated, non-dedicated cores are automatically used by the shared processor pool.
(2) The shared processor pool size can change as dedicated LPARs are started/stopped.

Shared Processor Pool – Processing Units

Physical processors are allocated to SP LPARs using processing units

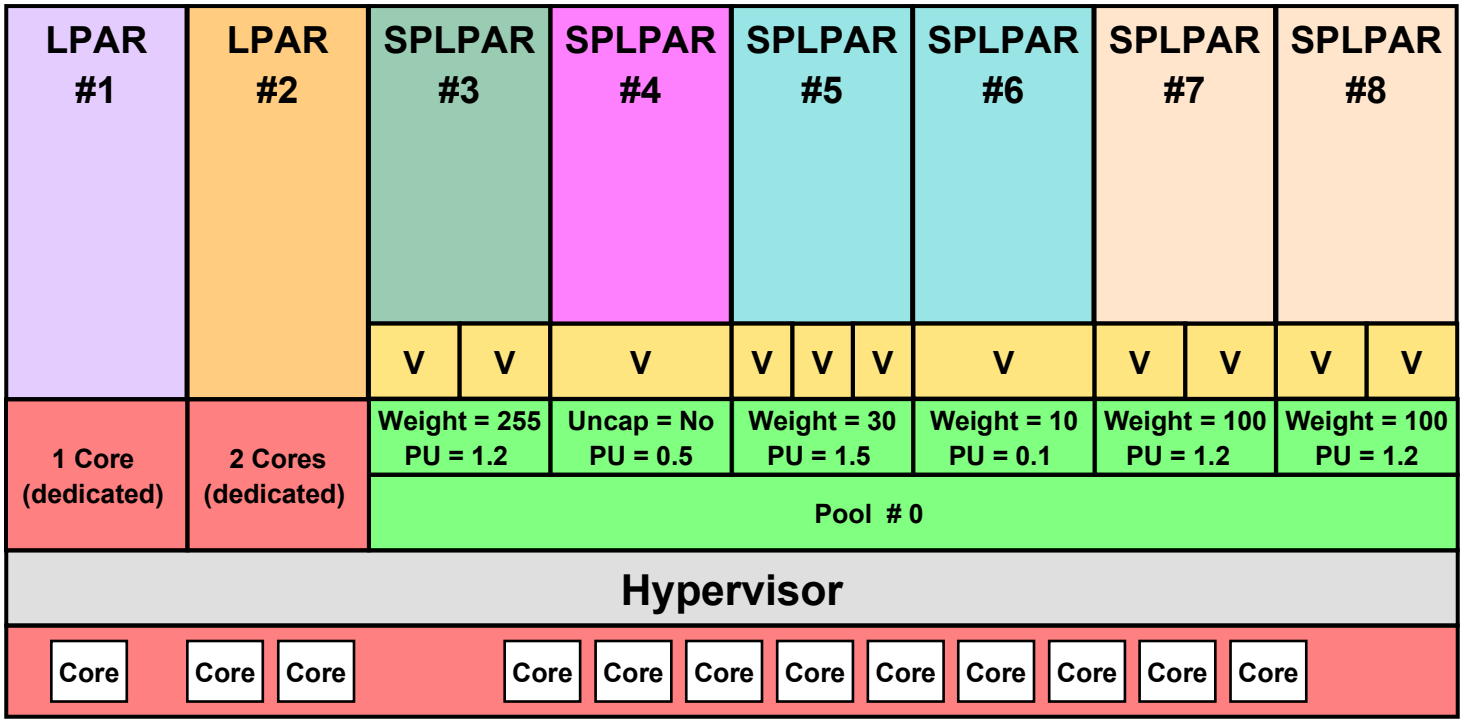


Think "P__ " **P=Physical**

Learning points: (1) One processing unit is equivalent to one core's worth of compute cycles
(2) The sum total of assigned processing units cannot exceed the size of the shared pool

Shared Processor Pool – Processing Units

Processing Units are presented to LPARs through Virtual Processors



Think "PV_ " **P=Physical V=Virtual**

Learning points: (1) Each virtual processor can represent 0.1 to 1 processing units.
(2) You will not be sharing pooled processors until the number of virtual processors exceeds the size of the shared pool.

Virtual Processors and Processing Units relationship

Virtual Processors Assigned to LPAR	Range Of Processing Units that the SPLPAR can utilize
1	0.1 - 1
2	0.2 - 2
3	0.3 - 3
4	0.4 - 4
x	0.1x - x

Example: An SPLPAR has two virtual processors. This means that the assigned processing units must be somewhere between 0.2 and 2. The maximum processing units that the SPLPAR can utilize is two.

If we want this SPLPAR to be able to use more than two processing units worth of cycles, we need to add more virtual processors, perhaps 2 more. Assigned processing units must now be at least 0.4 and the maximum utilization will be 4.

Learning point: The number of virtual processors establishes the maximum number of processing units that an SPLPAR can access.

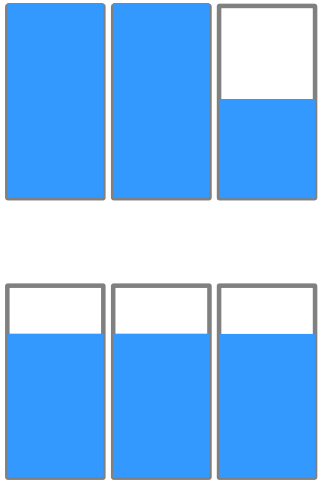
Virtual Processors

A major confusion to many

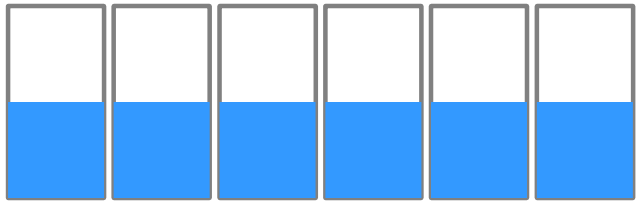
If it was called “spreading factor” it would help

→ the Entitlement CPU cycles are spread across the VP

EC = 2.5 spread across 3 VP



EC = 2.5 spread across 6 VP



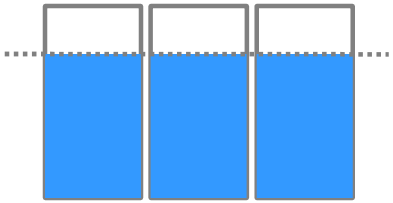
→ VP must be same or higher than EC
(EC = 2.5 cannot run on just 2 CPU ; so must be 3 VP or more)

To cap or not to cap

Capped

The LPAR cannot use more than Entitled; unused cycles go in the pool for the other LPARs or are wasted.

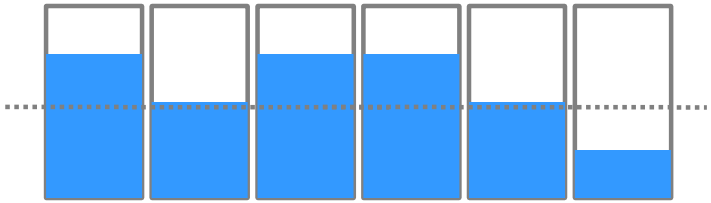
Can be used for SW license control



EC = 2.5 spread across 3 VP capped

Uncapped

The LPAR competes for spare pool CPU cycles and can get more work done.



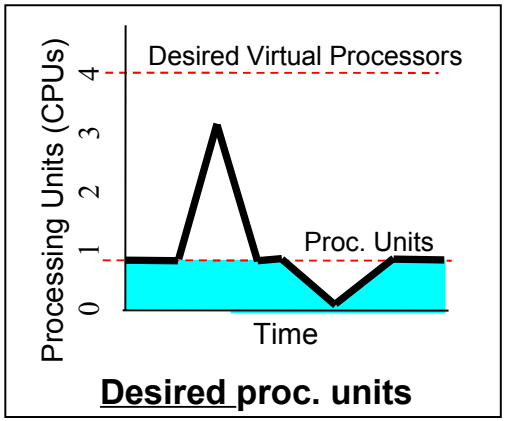
EC = 2.5 spread across 6 VP uncapped

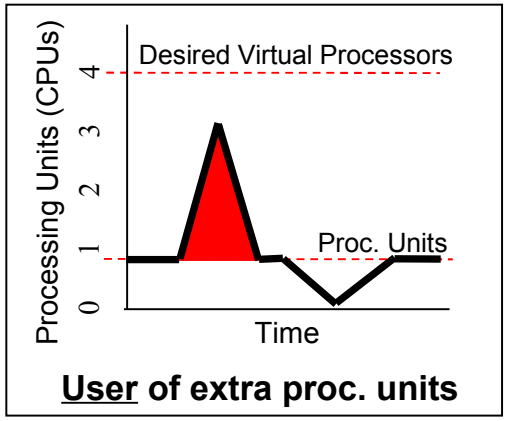
Capped vs Uncapped – single seat vs taking extra space for legs

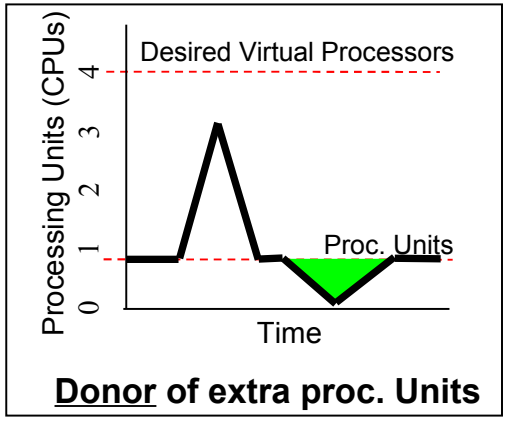
Capped

Uncapped

LPAR behaviour in shared mode - explained







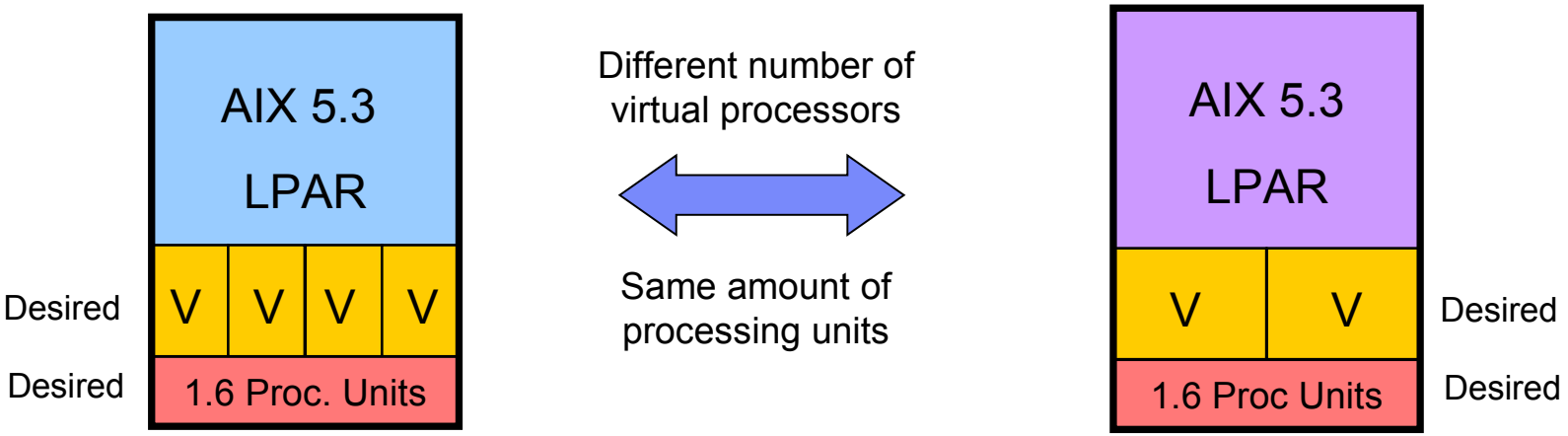
Desired Processing Units

establishes a guaranteed amount of processor cycles for each LPAR
uncapped → LPAR can utilize excess cycles
capped → LPAR is limited to the Desired Processing Units

Desired Virtual Processors

establishes an upper limit for possible processor consumption by an LPAR (when it is uncapped)

Virtual Processors and Processing Units relationship



If all four virtual processors have work to be done, each will receive 0.4 processing units.

The maximum processing units possible to handle peak workload is 4.

Individual processes/threads may run slower

Workloads with a lot of processes/threads may run faster

If both virtual processors have work to be done, each will receive 0.8 processing units.

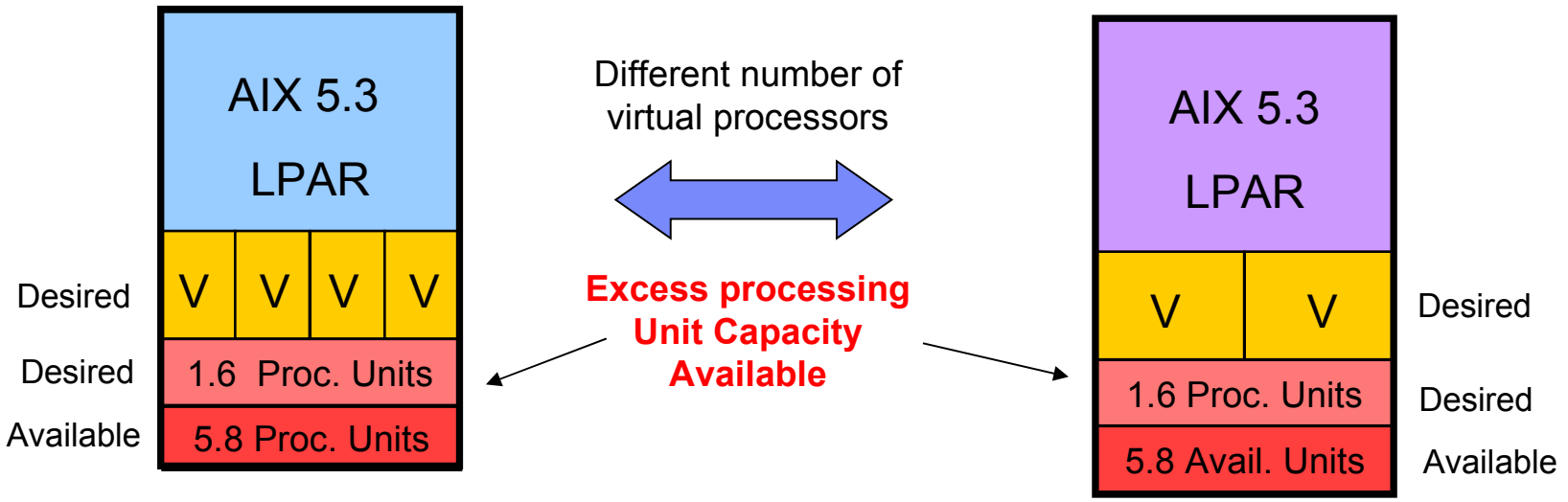
The maximum processing units possible to handle peak workload is 2.

Individual processes/threads may run faster

Workloads with a lot of processes/threads may run slower

Learning point: You need to consider peak processing requirements and the job stream (single or multi-threaded) when setting the desired number of virtual processors.

Virtual Processors and Processing Units relationship



Each virtual processor will receive 1.0 processing units from the 5.8 available.

Max processing units that can be consumed is 4 because we have 4 virtual processors.

Each virtual processor will receive 1.0 processing units from the 5.8 available.

Max processing units that can be consumed is 2 because we only have 2 virtual processors.

Learning point: In the presence of excess processing units, SPLPARs with a higher desired virtual processor count will be able to access more excess processing units.

Weight factor

0 to 255 (0 → capped) - you can tweak it dynamically

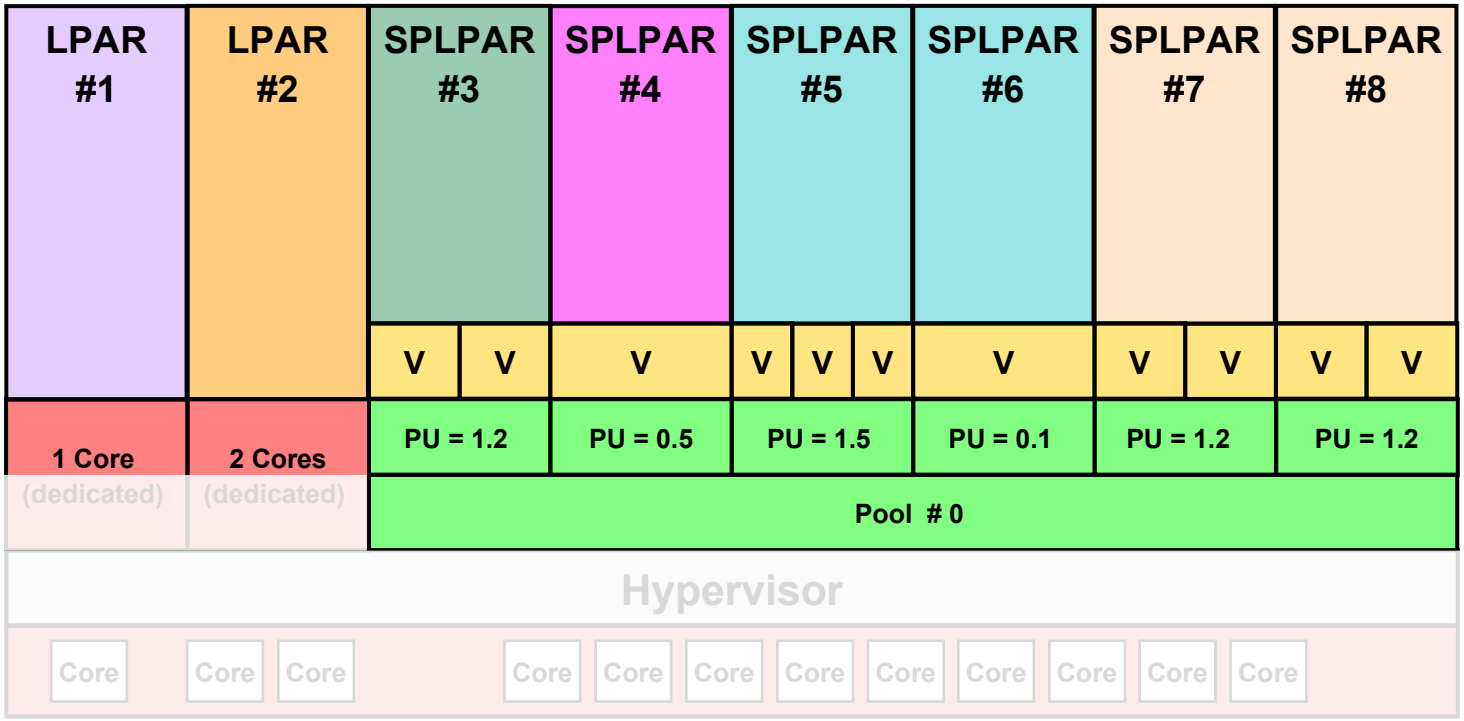
How much of the “spare” can each LPAR get?
The weights of all the LPARs who have had their entitlement are compared and unused resources are shared accordingly

$$AdditionalCapacityShare = \frac{UCk \times \frac{WPn}{rP}}{\sum We}$$

Where:

AdditionalCapacityShare	Share of unused processing capacity to be allocated to a particular partition (in *processor units* x 100)
UCk	Unused processor capacity available in their Shared-Processor Pool for the dispatch window (in processor units)
WPn	Uncapped weight of the particular uncapped micro-partition
rP	The number of runnable (eligible) micro-partitions for this dispatch window
∑We	Sum of the uncapped weights of all runnable uncapped micro-partitions

SP LPAR Utilization – Greater than 100% ???



PU Consumption	PU Utilization	VP Utilization
0.50	33%	16.7%
1.50	100%	50.0%
2.25	150%	66.0%
3.00	200%	100.0%

Property of an LPAR

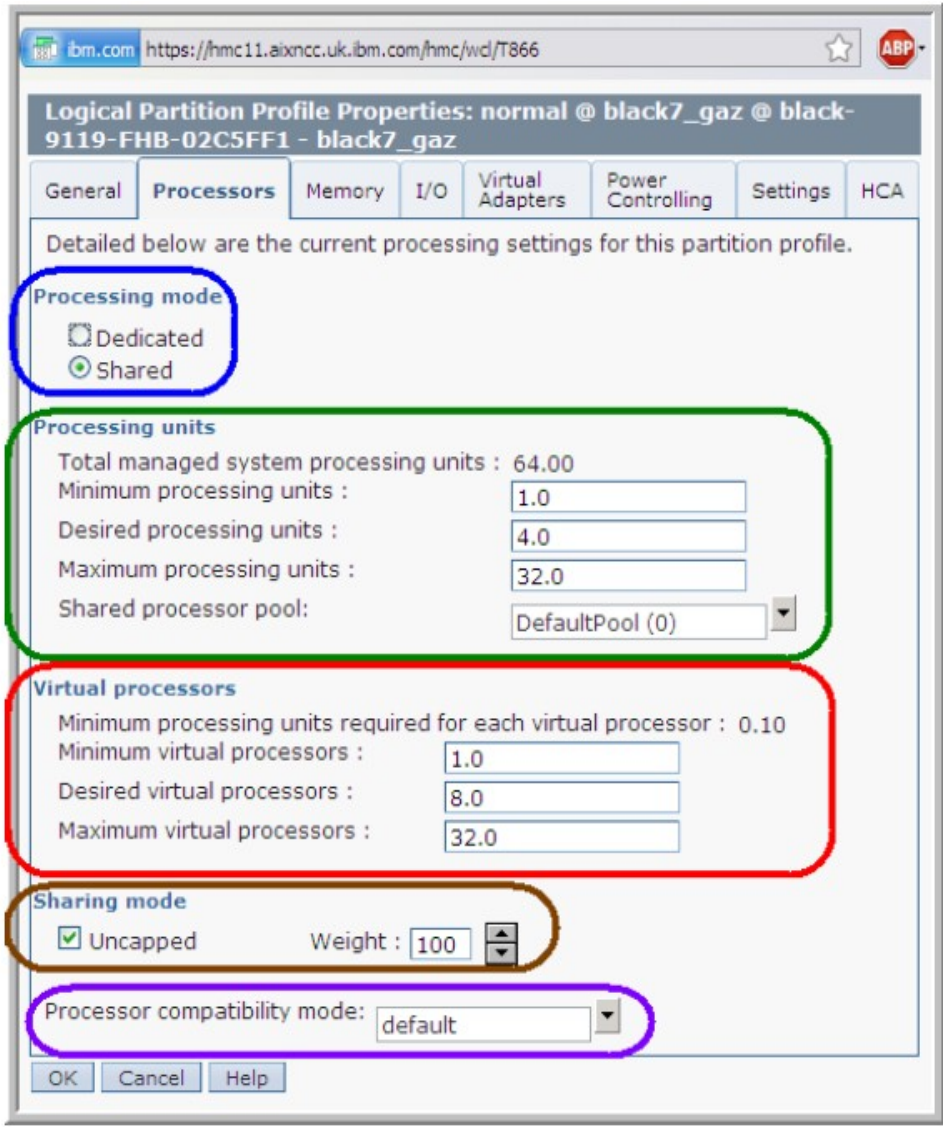
So, why **minimum** and **maximum**?

Minimum: if less than this available = don't start

Desired: what you'd really like

Lot of things can be changed dynamically:

- EC → +/- 0.01 beween MIN and MAX
- VP → integers between MIN and MAX
- Uncap weight → integers between 0 and 255



Dedicated vs Shared: lparstat -i

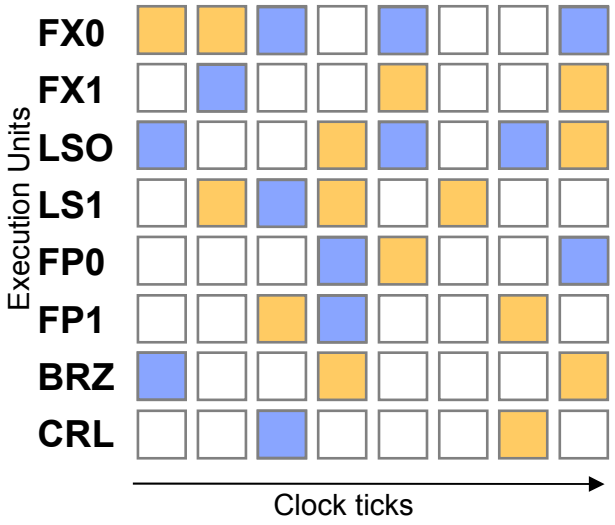
```
# lparstat -i
Node Name                                  : mantova
Partition Name                             : purple3-hpc
Partition Number                           : 13
Type                                       : Dedicated-SMT-4
Mode                                       : Capped
Entitled Capacity                          : 17.00
Partition Group-ID                         : 32781
Shared Pool ID                             : -
Online Virtual CPUs                        : 17
Maximum Virtual CPUs                       : 32
Minimum Virtual CPUs                       : 1
```

```
# lparstat -i
Node Name                                  : mantova
Partition Name                             : purple3-hpc
Partition Number                           : 13
Type                                       : Shared-SMT-4
Mode                                       : Uncapped
Entitled Capacity                          : 6.00
Partition Group-ID                         : 32781
Shared Pool ID                             : 0
Online Virtual CPUs                        : 8
Maximum Virtual CPUs                       : 30
Minimum Virtual CPUs                       : 2
```

So, what does it mean **SMT**?

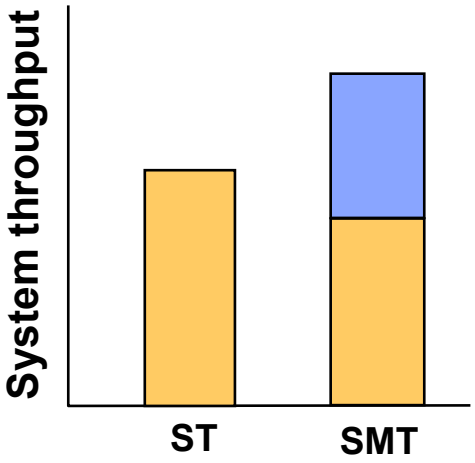
Simultaneous Multi-threading (SMT)

POWER5/6 (simultaneous multithreading)



Thread0 active
No thread active
Thread1 active

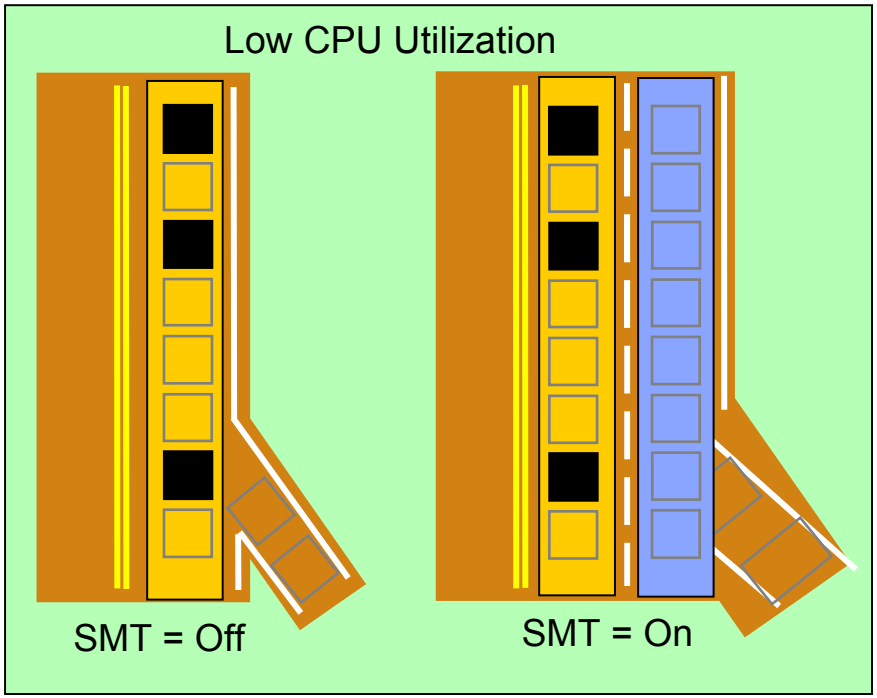
One processor (dedicated or virtual) appears as two logical processors to the operating system (AIX 5L V5.3 and Linux)

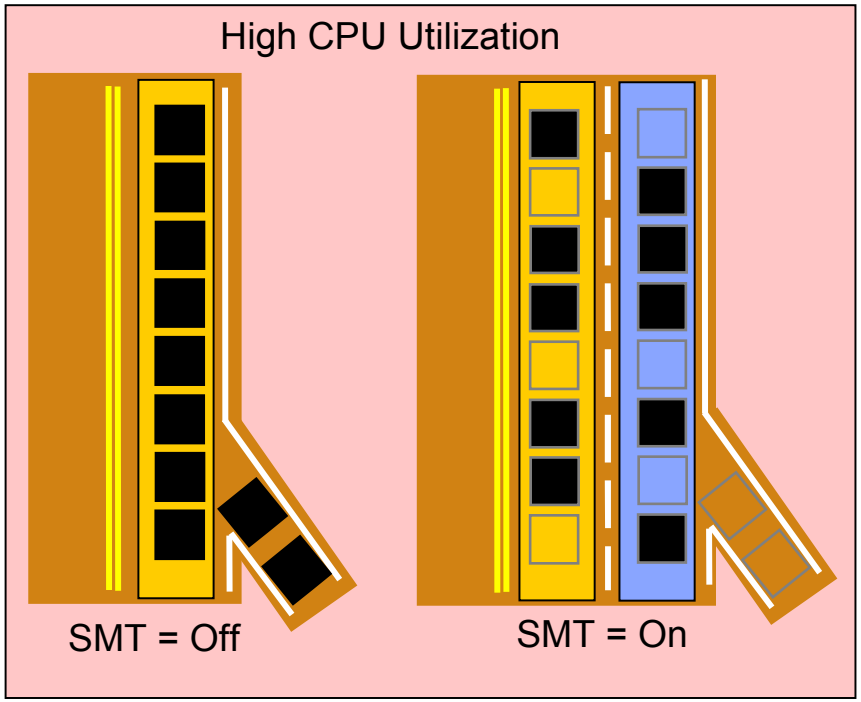


- Utilizes unused execution unit cycles
- Dispatch two threads per processor: *"It's like doubling the number of processors."*

Learning point: SMT = On → Logical processors present
SMT = Off → No logical processors

Simultaneous Multi-threading (SMT)



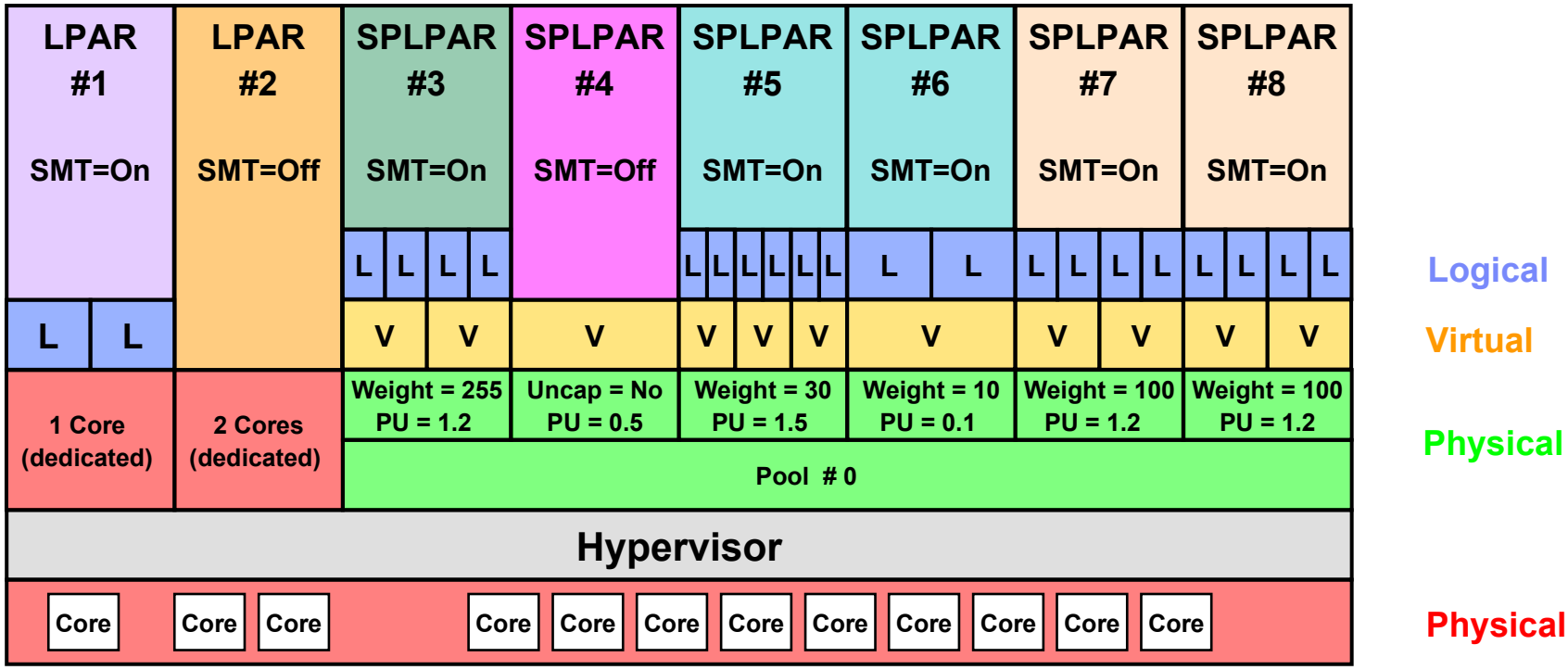


- SMT does not improve system throughput on a lightly loaded system
- SMT does not make a single thread run faster

- SMT does improve system throughput on a heavily loaded system
- SMT does not make a single thread run faster (unless it is waiting in the queue)

Logical processors

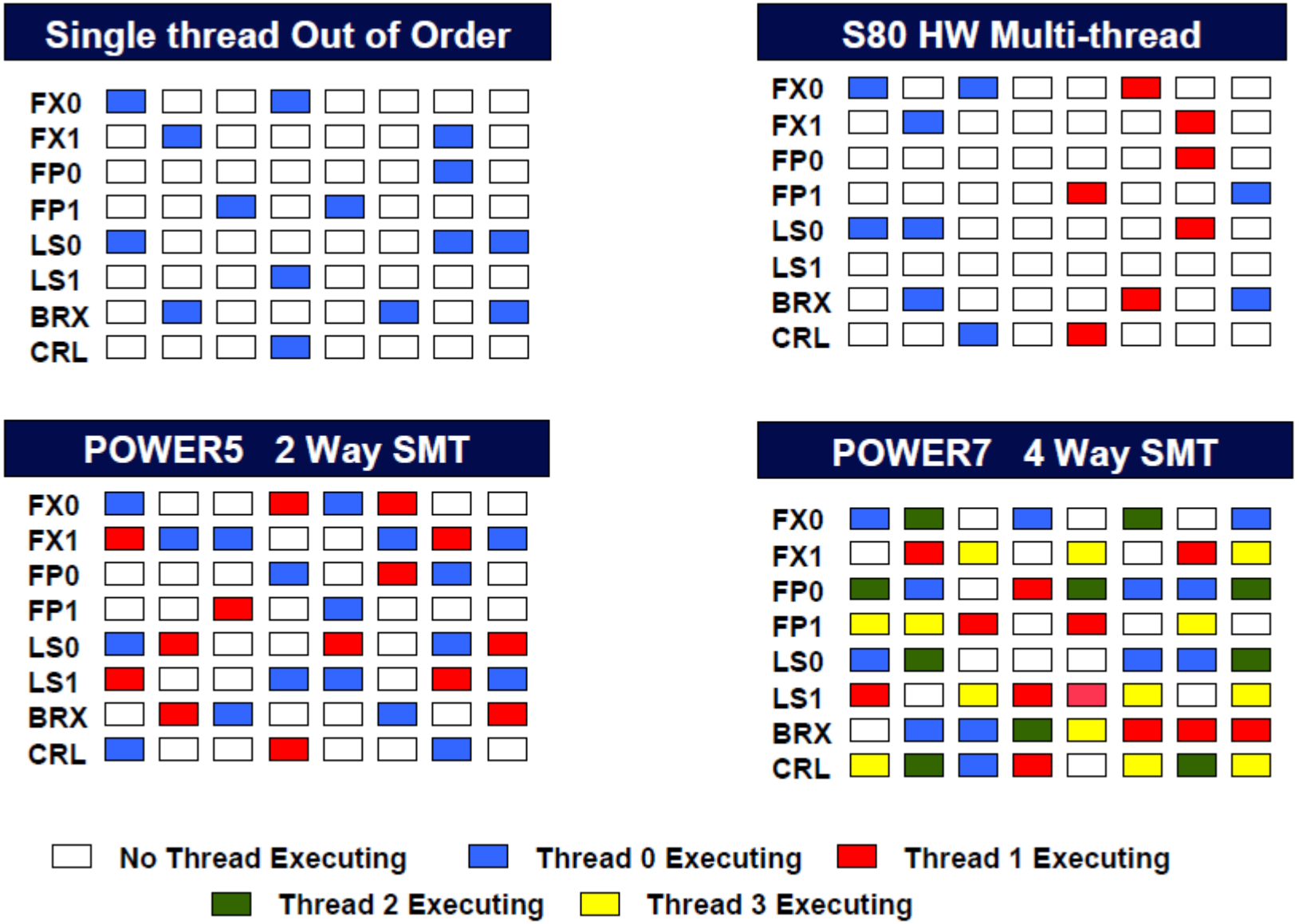
Simultanous Multi-threading (SMT) threads are represented by logical proc



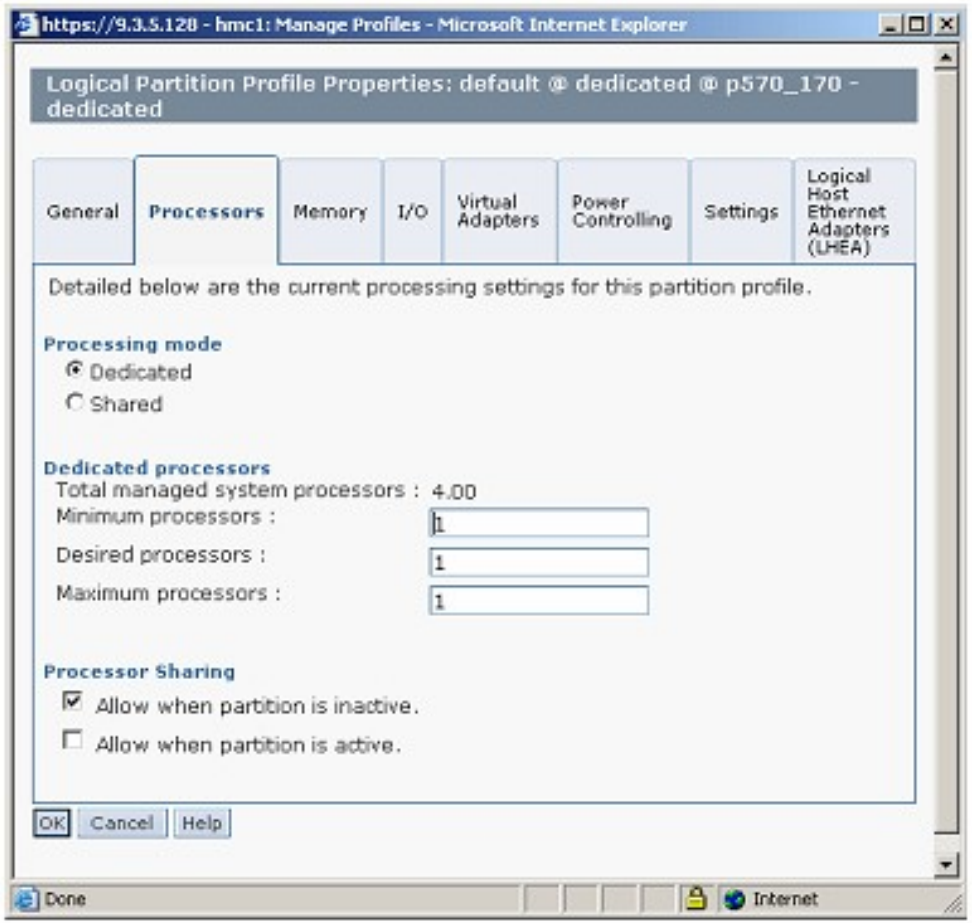
Think "PVL " **P=Physical V=Virtual L=Logical (SMT)**

Learning point: SMT requires a minimum of POWER5 hardware and AIX 5.3 (or supported Linux ver.)
SMT can be dynamically enable/disable via an AIX command.

Multi-threading evolution



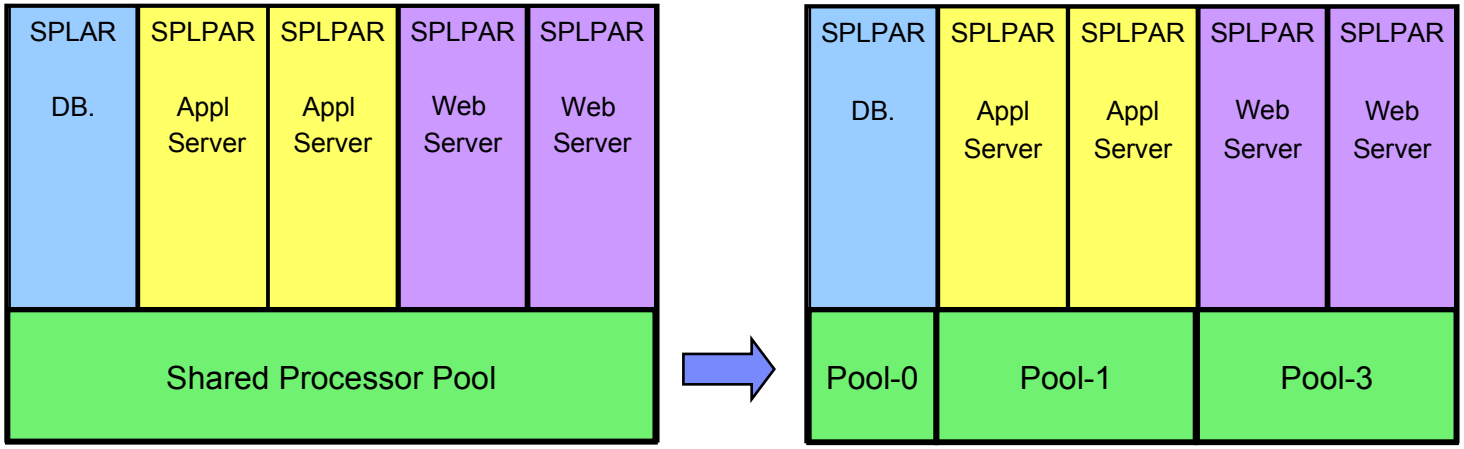
Shared Dedicated Capacity



Even if the partition is in Dedicated mode, spare cycles can be given back to the pool:

- if the partition is inactive
- if the partition is active

Multiple Shared Processor Pools



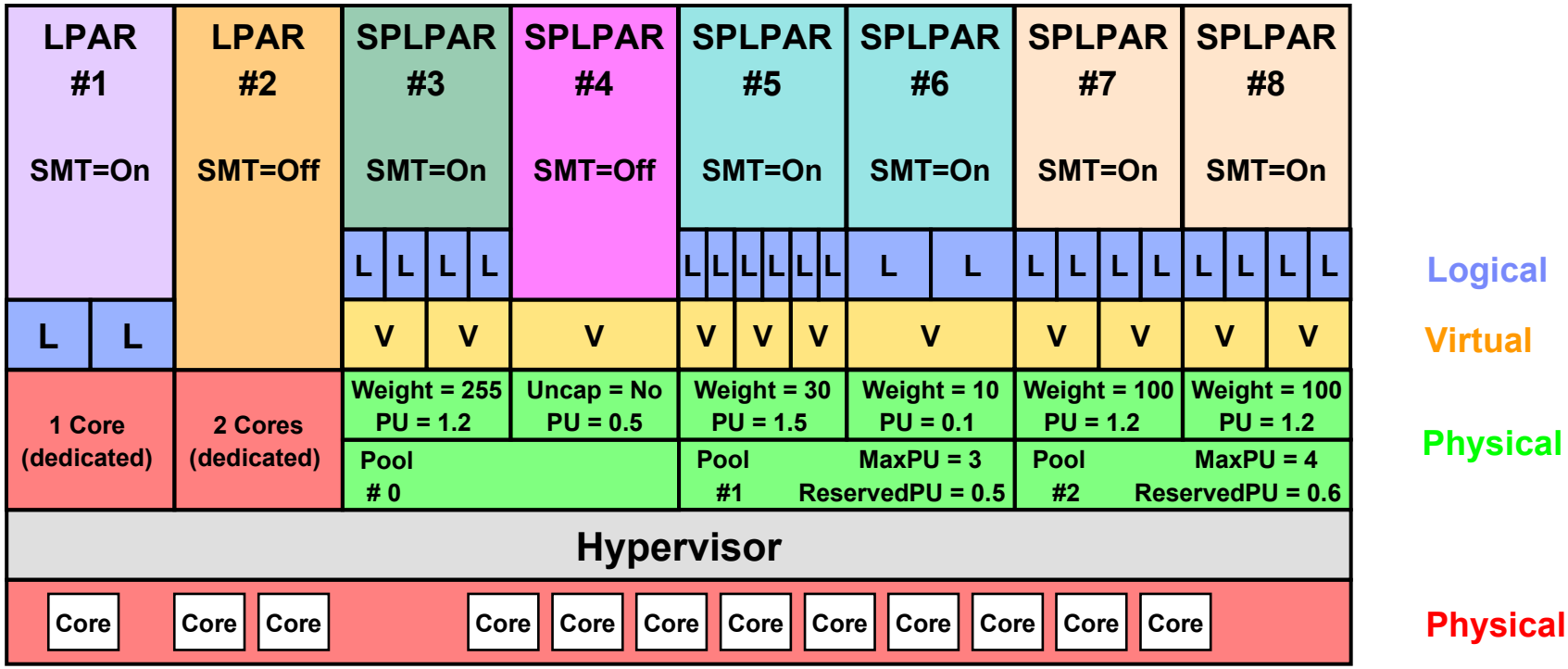
Limit processor resources to a group of LPARs

Up to 64 Processor Pools per server

Can help with software licensing

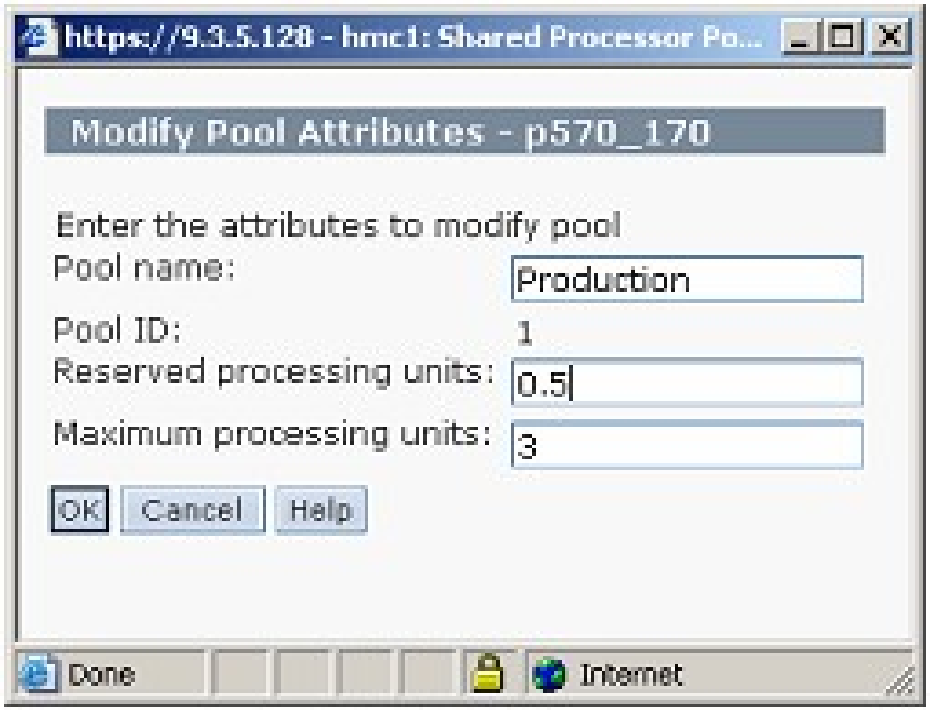
Help balance Prod / Dev on the same server

Multiple Shared Processor Pools



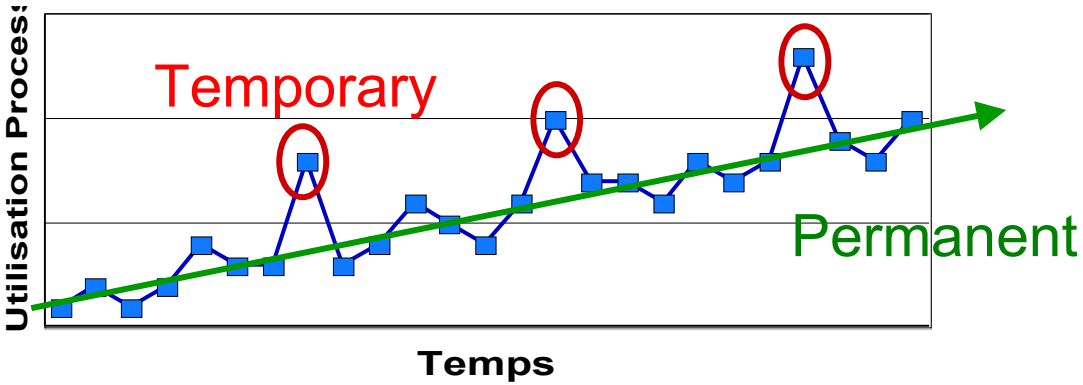
- MaxPU … A Whole number, specifies maximum processing units that can be consumed by all of the SPLPARs running in this pool,
- ReservedPU = Additional, guaranteed Processing Units for each pool (could be 0)
- Default Pool ID = 0 (cannot specify MaxPU or ReservedPU for the Default Pool)

Multiple Shared Processor Pools - configuration



- **Pool IDs are fixed and numbered 0...63**
- **SPLPARs can dynamically be moved to a different pool**
- **Disable a pool by setting its Maximum processing units to zero**
- **Default Pool ID = 0**
- **You cannot set reserved processing units or Maximum processing units for the Default Pool**

Capacity on demand



POWER7	proc	mem
720 8202-E4B		
740 8205-E6B		
750 8233-E8B		
755 8236-E8C		
770 9117-MMB	✓	✓
780 9179-MHB	✓	✓
795 9119-FHB	✓	✓

- **How it works:**
  - Activate dormient processor and memory within a system without any application disruption
  - Temporary: pay per day or per minute
  - Permanent: planned growth
- **Client benefits:**
  - **Permanent: Sustain rapid business growth without changing a system**
  - **Temporary: Sustain business peak (i.e. web site during sales / Christmas)**

Capacity on demand

Active

Dormient

Questions?

CREDITS to

Gareth Coates
IBM Power ATS EMEA
IBM UK

Charlie Cler
Executive IT Specialist
IBM US

Luca Comparini
STG Lab Services Europe
IBM FR

THANKS