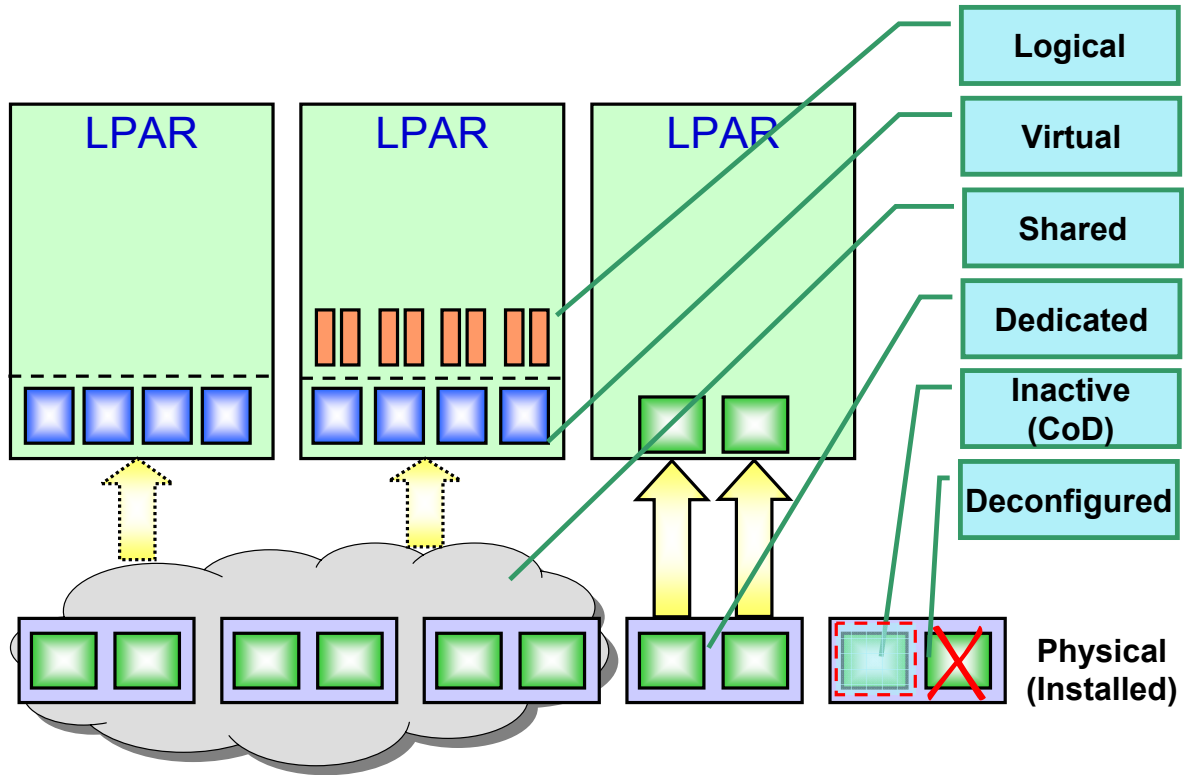**Welcome to:**

POWER Systems Micro-Partitioning
Part 1

Janel Barfield
jgbarfie@us.ibm.com

Objectives

- Describe the following terms:
  - Dedicated and shared processors, Micro-Partition, virtual processor, logical processor, and processing units
- Describe how the Hypervisor allocates processing power from the shared processor pool
- Describe the effect of using capped and uncapped processors
- Describe the effect of the uncapped processor weight feature
- Describe the minimum and maximum virtual processor configuration options

Processor Concepts - Overview

LPAR

LPAR

LPAR

Logical

Virtual

Shared

Dedicated

Inactive (CoD)
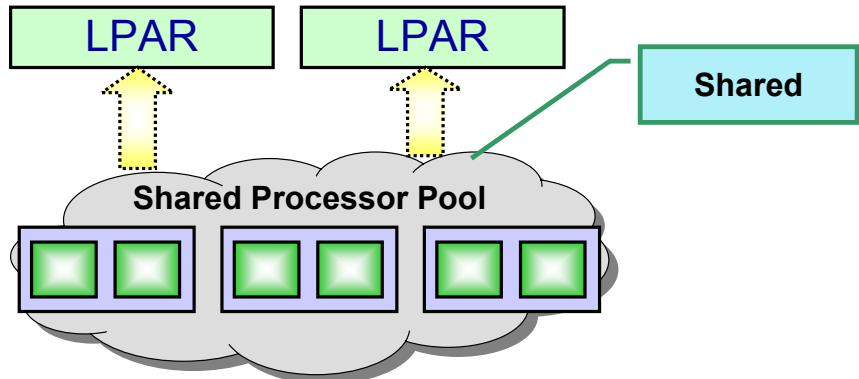
Deconfigured

Physical (Installed)

Dedicated Processors

- Allocated as whole processors to a specific partition

- Same physical processors are used for that partition while it is running
  - When partition is stopped, dedicated processors may or may not go to shared pool

- Processor affinity utilized for best performance

LPAR

Dedicated

Physical Processors
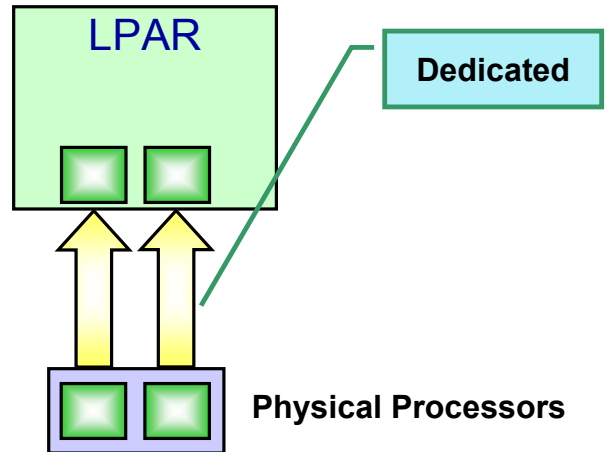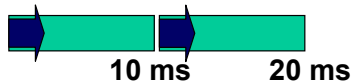
Shared Processors

- Processor capacity assigned in *processing units* from a *shared processing pool*
  - Minimum per partition is 0.1 processing units
  - Additional capacity allocated in 0.01 processing units
- Partition's guaranteed amount is its *Entitled Capacity*
- Advantages:
  - Configuration flexibility
  - Excess capacity can be used by other partitions
- *Micro-Partition*: A partition utilizing shared processors

Shared Processor Pool (1 of 2)

- Each partition is configured with a percentage of execution dispatch time for each 10 ms timeslice
- Examples:
  - A partition with 0.2 processing units is entitled to 20% capacity of one processor during each timeslice
  - A partition with 1.8 processing units is entitled to 18 ms of processing time for each 10 ms timeslice (using multiple processors)
- The Hypervisor dispatches excess idle time back to pool
- Processor affinity algorithm takes into account *hot* cache
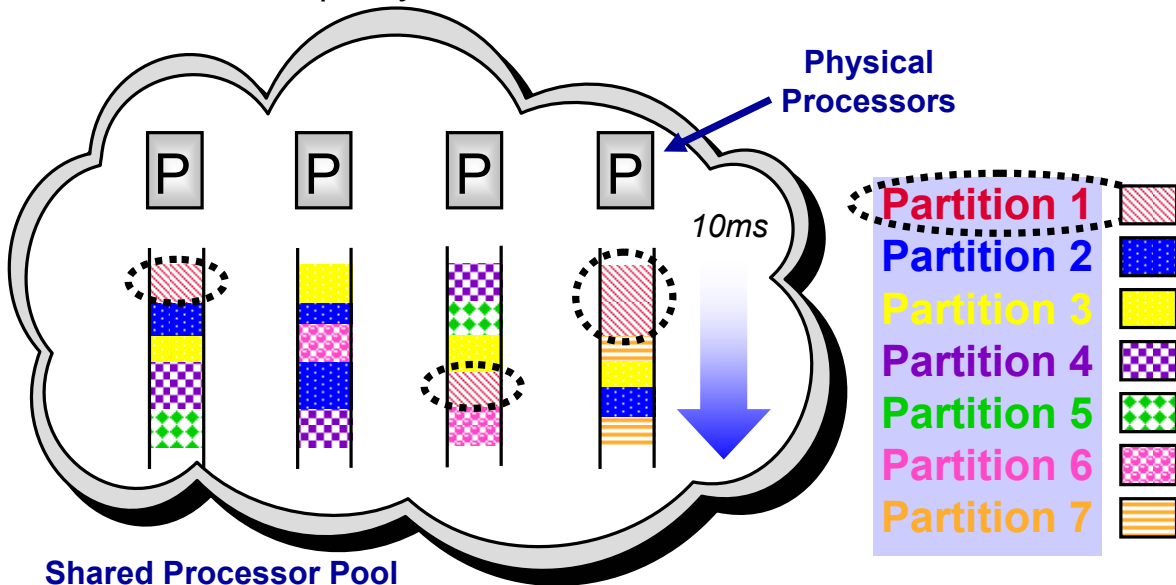
0.2 = 2 ms per timeslice

10 ms   **20 ms**

1.8 = 18 ms per 10 ms timeslice

10 ms   **20 ms**

10 ms   **20 ms**
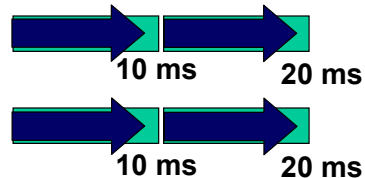
Shared Processor Pool (2 of 2)

- This example shows one 10 ms timeslice, 7 running partitions, and 4 processors
  - A partition may run on multiple processors depending on interrupts and its *entitled capacity*

Physical Processors

10ms

Shared Processor Pool

Partition 1
Partition 2
Partition 3
Partition 4
Partition 5
Partition 6
Partition 7

Capped Partitions

- Partitions with shared processors are either *capped* or *uncapped*
- *Capped*: Limited to the entitled capacity
- Example: 1.5 capped processing units means a partition can use *up to* 15 ms of execution time during each timeslice, but no more than that

Total processing units →
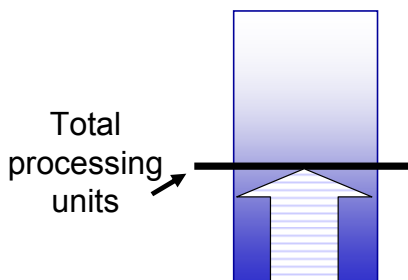
The partition *may* utilize CPU cycles up to its capacity – guaranteed!

Excess cycles are ceded back to shared pool

Uncapped Partitions

- *Uncapped*: If a partition needs extra CPU cycles (more than its entitled capacity), it can utilize unused capacity in the shared pool

- Example: An uncapped partition with 1.5 processing units is guaranteed to be able to use 1.5 units, but may use more if necessary (and more is available)
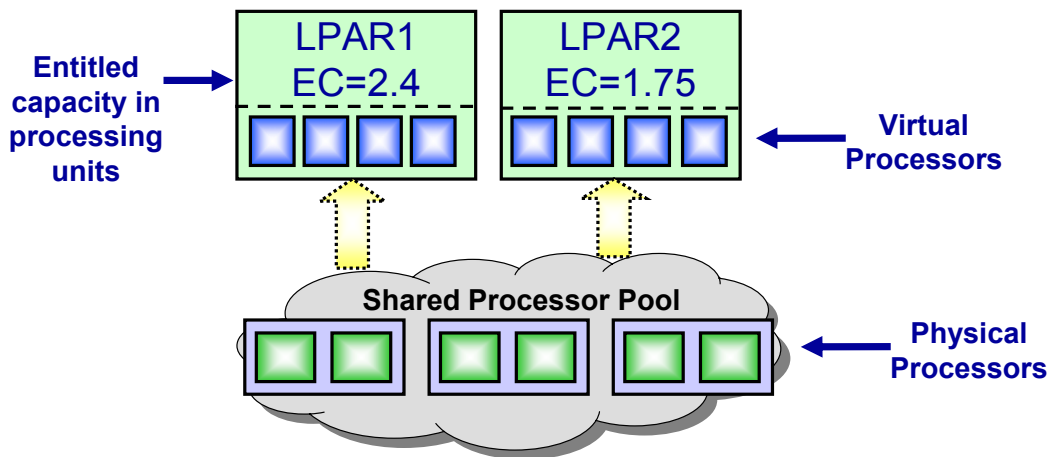
Total processing units
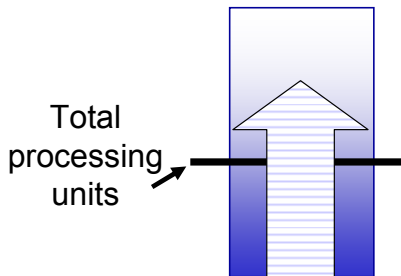
All uncapped partitions share the unused capacity based on user-defined weighting

Weight scale is 0 to 255

Virtual Processors (1 of 3)

- Virtual processors are used to tell the operating system how many physical processors it thinks it has
  - Operating system in **LPAR2** does not see 1.75 processing units; it will see the configured virtual processors
    - In this example, each partition sees 4 processors:

Virtual Processors (2 of 3)

- By default, for each 1.00 of a processor, or part thereof, a virtual processor will be allocated
  - Example: 3.6 processing units would have 4 virtual processors

- Up to 10 virtual processors *can* be assigned per processing unit
  - Example: 3.6 processing units can have up to 36 virtual processors

- Both entitled capacity and number of virtual processors can be changed dynamically for tuning
  - Maximum virtual processors per partition is 64

Example:
Partition with 4.2 entitled capacity:

Minimum virtual processors = _____

Maximum virtual processors = _____

Virtual Processors (3 of 3)

- The number of virtual processors does not change the *entitled capacity*

- Example: A partition has 1.5 *capped* processing units
  - For each 10 ms timeslice, the partition is only entitled to 15 ms of processing time
  - If 2 virtual processors
    - 7.5 ms runs on 2 physical processors
  - If 4 virtual processors
    - 3.75 ms runs on 4 physical processors

- For uncapped partitions, performance may be limited if the virtual processor setting is too low
  - Number of simultaneous physical processors is limited by the virtual processor setting
  - Each virtual processor is worth up to 10 ms of processing time

Virtual Processors – What to Do?

- How many virtual processors should be used?
  - For *uncapped* partitions, increase number of virtual processors to up to the number of processors in shared pool
    - Start with 25% more than the minimum, then monitor and tune
  - For *capped* partitions, start with minimum and monitor
  - Did performance improve?

- If virtual processor number is too low or too high, performance may be affected
  - *Too low:* Uncapped partitions will not be able to take advantage of excess cycles
  - *Too high*: May cause excessive processor context switching
    - As of AIX 5L ML3, *virtual processor folding* will fold virtual processors if there are too many for the current system load

  Monitor partitions and try increasing and decreasing VP quantity to determine the best configuration