Welcome to:

POWER Systems Micro-Partitioning
Part 2

Janel Barfield
jgbarfie@us.ibm.com

Objectives

- Use AIX commands to view partition configuration and processor performance information
- Describe performance considerations when configuring Micro-Partitioning options
- Configure a partition using the advanced processor options

Viewing Processor Information

- Processors are seen with the **`lsdev`** command
  - **`lsdev`** shows *physical* or *virtual* processors

```
# lsdev -Cc processor
proc0 Available 00-00 Processor
proc2 Available 00-02 Processor
```

- Processor attributes are seen with the **`lsattr`** command

```
# lsattr -El proc0
frequency   1656000000     Processor Speed       False
smt_enabled true           Processor SMT enabled False
smt_threads 2              Processor SMT threads False
state       enable         Processor state       False
type        PowerPC_POWER5 Processor type        False
```

- Logical processors are seen with the **`bindprocessor`** command

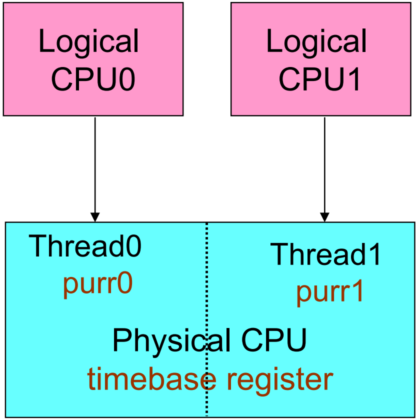
```
# bindprocessor -q
The available processors are:  0 1 2 3
```

Traditional CPU Utilization Statistics

- Data collection is sample based
- 100 samples per second sorted into the following categories:
  - user
  - sys
  - iowait
  - idle
- Each sample corresponds to a 10 ms clock tick
- Recorded in the kernel data structures: **sysinfo** and **cpuinfo**
- Performance tools convert:
  - Tick counts from the **sysinfo** structure into utilization percentages for the machine/partition (e.g., **`vmstat`**, **`iostat`**, **`sar`**)
  - Tick counts from the **cpuinfo** structure into utilization percentages for a processor/thread (e.g., **`sar -P ALL`**, **`topas`** hot CPU section)

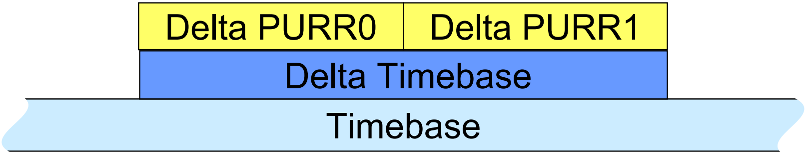
Processor Utilization of Resources Register (PURR)

- Traditional utilization metrics are misleading:
  - They think there are two physical processors
  - In AIX 5L V5.3, the number of dispatch cycles for each thread can be measured using a new processor utilization of resources register (PURR)
- Two PURR registers (one for each hardware thread)
- Units are the same as the timebase register
- Sum of the PURR values for both threads is equal to the timebase register



CPU Utilization

- In a simultaneous multi-threaded environment and/or a Micro-Partition, CPU utilization statistics:
  - Still collect 100 samples per second (for binary compatibility)
  - Collect additional state-based *PURR-based* metrics (in PURR increments)
- Utilization metrics:
  - Same categories are used: user, sys, iowait, and idle
  - Physical resource utilization metrics for a logical processor:
    - (delta PURR/delta TB) represents the fraction of the physical processor consumed by a logical processor
    - (delta PURR/delta TB)*100 over an interval represents the percentage of dispatch cycles given to a logical processor



CPU Utilization Example

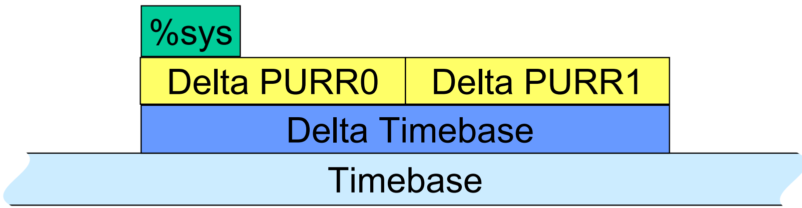
- Assume:
  - Two threads running on a single CPU system with simultaneous multi-threading enabled
  - One thread is 100% busy, the second one is idle:
    - Traditional, sample-based collection method would (incorrectly) show the system as 50% busy
    - New, state-based (PURR) collection method would (correctly) show the system as 100% busy

CPU Utilization Metrics

- Physical CPU utilization metrics are calculated using PURR statistics:
  - For example:

    %sys = (delta PURR in system mode/delta PURR in all modes)*100

    - For example, if a logical processor runs for 4 ms, and it was in system mode for 1 ms, %sys would report 25

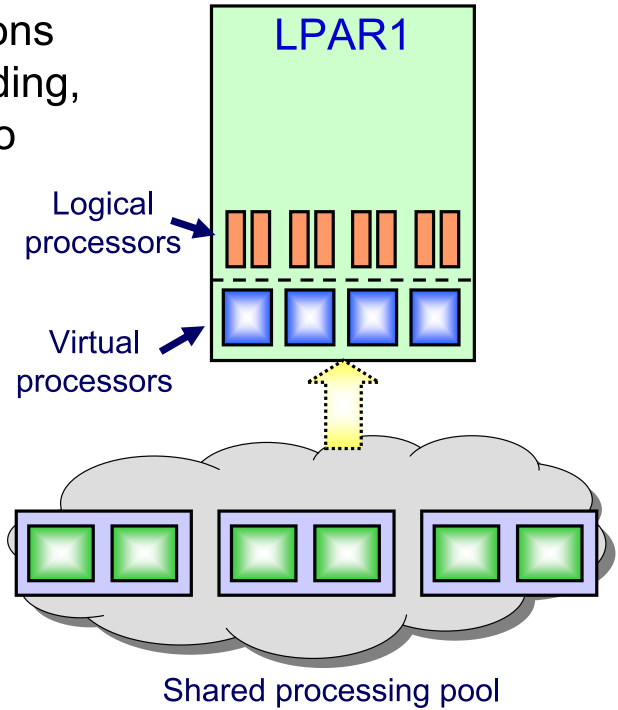


Additional CPU Utilization Metrics

- Physical Processor Consumed (PPC) = sum(delta PURR/delta TB)
  - This is how much physical processor time was consumed for each logical processor
  - Dedicated partitions will always show all processors consumed (because even if idle, it's being "consumed" by the partition)
  - Micro-Partitions show actual portions of physical processors consumed because a virtual processor gives up its excess cycles
- Micro-Partitions only:
  - Percentage of entitlement consumed = (PPC/ENT)*100
  - Available physical processors = (delta PIC/delta TB)
    - PIC = Pool Idle Count (Delta PURR when no VPs are dispatched):
      - All partition entitlements satisfied
      - No partition to dispatch
  - Logical processor utilization (%lbusy):
    - Sum of %sys and %user

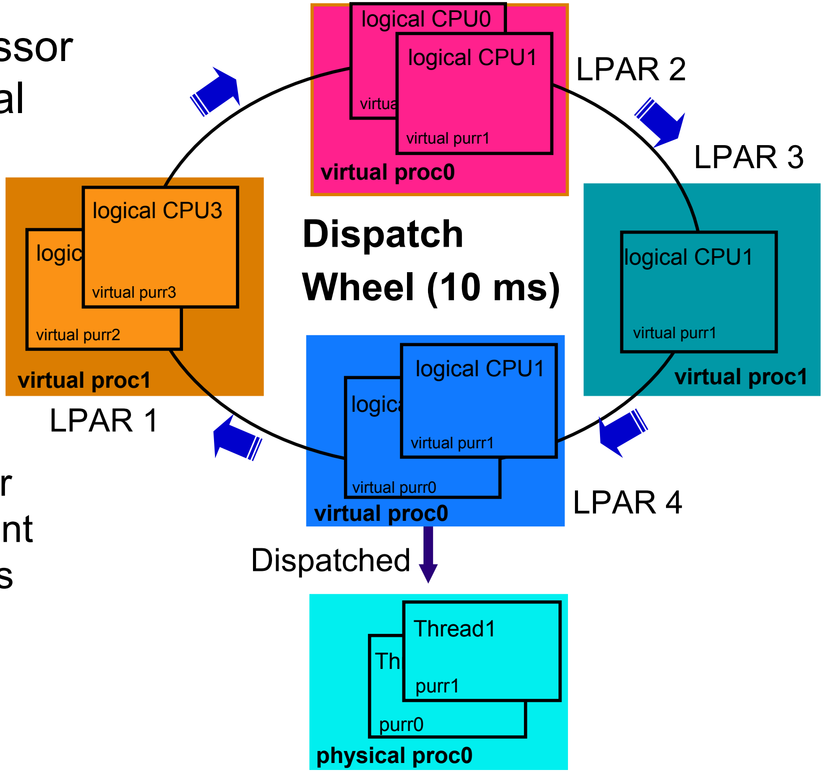
Simultaneous Multi-Threading and SPLPARs

- Simultaneous multi-threading can be used with Micro-Partitions
- With simultaneous multi-threading, each virtual processor runs two threads
  - Each thread is called a logical processor
- LPAR1 example:
  - 1.6 processing units
  - 4 virtual processors
  - Simultaneous multi-threading enabled
    - 8 logical processors



Metrics with Simultaneous Multi-Threading and SPLPAR

- Each virtual processor supports two logical processors
  - Dispatched at the same time
- PURR statistic
  - Still measures fraction of time partition runs on a physical processor (the relative amount of processing units consumed)



Viewing Partition Information (1 of 2)

- `lparstat -i` command

```
$ lparstat -i
Node Name                                  : lou162
Partition Name                             : LPAR2
Partition Number                           : 2
Type                                       : Shared-SMT
Mode                                       : Capped
Entitled Capacity                          : 0.50
Partition Group-ID                         : 32770
Shared Pool ID                             : 0
Online Virtual CPUs                        : 1
Maximum Virtual CPUs                       : 2
Minimum Virtual CPUs                       : 1
Online Memory                              : 1280 MB
Maximum Memory                             : 1536 MB
Minimum Memory                             : 1024 MB
Variable Capacity Weight                   : 0
Minimum Capacity                           : 0.10
Maximum Capacity                           : 1.00
Capacity Increment                         : 0.01
Maximum Physical CPUs in system            : 2
Active Physical CPUs in system             : 2
Active CPUs in Pool                        : 2
Unallocated Capacity                       : 0.00
Physical CPU Percentage                    : 50.00%
Unallocated Weight                         : 0
```

Viewing Partition Information (2 of 2)

- The HMC **`lshwres`** command

lshwres -r proc -m *mansys* --filter "lpar_names=*LPAR*" --level lpar

↑ Managed System Name (mansys)

↑ Partition Name (LPAR)

```
hscroot@hmc:~> lshwres -r proc -m sputnik --filter \
"lpar_names=LPAR2" --level lpar -F curr_proc_units,curr_procs \
--header

curr_proc_units,curr_procs
0.6,6
```

AIX SPLPAR Tool Impact (1 of 2)

- SPLPAR tool impact:
  - Many standard AIX analysis tools and the Perfstat library have been enhanced to use the new Processor Utilization of Resources Register (PURR) or Scalable Processor Utilization of Resources Register (SPURR) in POWER6 systems
  - Additional metrics have been added
  - New tools have been added such as **`lparstat`** and **`mpstat`**

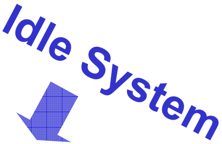
- **`lparstat`** is an easy way to view partition's configuration and processor usage:

```
# lparstat

System configuration: type=Shared mode=Capped smt=On lcpu=4
mem=1024 psize=2 ent=0.80

%user  %sys  %wait  %idle physc %entc  lbusy   app  vcsw phint
----- ----- ------ ------ ----- ----- ------   ---  ---- -----
  0.1   0.1    0.0   99.8  0.00   0.3    0.0  1.99   580     0
```

AIX SPLPAR Tool Impact (2 of 2)

- **`vmstat, iostat, sar`**:
  - Automatically use new PURR-based metrics for %user, %sys, %wait, %idle with simultaneous multi-threading or shared processor partitions
  - New metrics:
    - Physical Processor Consumed (**pc** or **physc**) by the partition
    - Percentage of Entitlement Consumed (**pec** or **%entc**) by the partition which can go as high as 1000% for uncapped partitions
- **`mpstat`** can be used to monitor logical and virtual processor activity
- **`topas –L`** adds LPAR view (shows new metrics)
- Additional tools such as **`trace`**-based tools can utilize additional metrics

Using `sar` with SPLPAR (1 of 2)




```
# sar -P ALL 1 2

AIX train172 3 5 00C88F7D4C00    03/01/05

System configuration: lcpu=2 ent=1.00

20:20:57 cpu    %usr    %sys    %wio   %idle   physc   %entc
20:20:58  0       23      58       0      19    0.01     0.7
          1        0       2       0      98    0.00     0.2
          U        -       -       0      99    0.99    99.1
          -        0       0       0      99    0.01     0.9
20:20:59  0       15      60       0      25    0.01     0.5
          1        0       2       0      98    0.00     0.2
          U        -       -       0      99    0.99    99.3
          -        0       0       0     100    0.01     0.7

Average   0       20      59       0      21    0.01     0.6
          1        0       2       0      98    0.00     0.2
          U        -       -       0      99    0.99    99.2
          -        0       0       0     100    0.01     0.8
```

Using `sar` with SPLPAR (2 of 2)

Busy System

```
# sar -P ALL 1 1
AIX bud152 3 5 00CDEF8E4C00    03/21/06

System configuration: lcpu=4 ent=0.80

16:19:23 cpu    %usr    %sys    %wio   %idle   physc   %entc
16:19:24  0        0       7       0      93    0.03     3.3
          1      100       0       0       0    0.37    46.8
          2      100       0       0       0    0.38    46.9
          3        0       1       0      99    0.02     3.1
          -       94       0       0       6    0.80   100.0
```

```
# mpstat -s 1 1
System configuration: lcpu=4 ent=0.8

     Proc0                Proc2
     39.99%               39.76%
 cpu0       cpu1       cpu2       cpu3
  2.55%    37.45%     37.57%      2.19%
```

topas - Example Main Screen

```
Topas Monitor for host:    bud151                 EVENTS/QUEUES    FILE/TTY
Wed Nov 23 20:29:20 2005   Interval:  2           Cswitch      260  Readch     1078
                                                  Syscall      261  Writech     161
Kernel     0.2   |#                            |  Reads          1  Rawin         0
User      99.6   |############################|   Writes         1  Ttyout      162
Wait       0.0   |                             |  Forks          0  Igets         0
Idle       0.2   |#                            |  Execs          0  Namei         0
Physc =   0.80                    %Entc= 100.0    Runqueue     8.0  Dirblk        0
                                                  Waitqueue    0.0
Network  KBPS   I-Pack  O-Pack   KB-In  KB-Out
en0       0.3      2.5     0.5     0.1     0.2    PAGING            MEMORY
lo0       0.1      1.0     1.0     0.1     0.1    Faults         0  Real,MB    1024
                                                  Steals         0  % Comp     28.0
Disk     Busy%     KBPS     TPS KB-Read KB-Writ   PgspIn         0  % Noncomp   3.9
hdisk0    0.0      0.0     0.0     0.0     0.0    PgspOut        0  % Client    4.7
                                                  PageIn         0
Name            PID  CPU%  PgSp Owner             PageOut        0  PAGING SPACE
spload       221394  39.8   1.3 root              Sios           0  Size,MB     512
topas        188422   0.0   1.1 root                                % Used      1.0
gil           65568   0.0   0.1 root              NFS (calls/sec)   % Free     98.9
aixmibd      237726   0.0   0.6 root              ServerV2        0
rpc.lock     147580   0.0   0.2 root              ClientV2        0   Press:
rmcd         176292   0.0   2.5 root              ServerV3        0   "h" for help
snmpdv3n     258276   0.0   1.0 root              ClientV3        0   "q" to quit
```

Partition Data with `topas -L`

```
# topas -L

Interval:    2    Logical Partition: LPAR1           Wed Nov 23 20:32:04 2005
Psize:   2                    Shared SMT  ON         Online Memory:   1024.0
Ent: 0.80                      Mode: Capped          Online Logical CPUs:  4
Partition CPU Utilization                            Online Virtual CPUs:  2
%usr %sys %wait %idle physc %entc %lbusy   app   vcsw phint %hypv   hcalls
 100    0     0     0   0.8  99.98 50.00  1.18    240     0   0.0        0
==============================================================================
LCPU  minpf majpf  intr   csw  icsw  runq lpa scalls usr sys _wt idl   pc   lcsw
Cpu0      0     0   263   134   119     2 100     99 100   0   0   0 0.41   100
Cpu1      0     0   106   112   103     2 100    123 100   0   0   0 0.39   100
Cpu2      0     0    10     0     0     0   0      0   0  23   0  77 0.00    20
Cpu3      0     0    10     0     0     0   0      0   0  27   0  73 0.00    20
```

Cross Partition Data with `topas -C`

```
Topas CEC Monitor                Interval:  10          Mon Nov 28 17:10:46
2005
Partitions     Memory (GB)                Processors
Shr:  3        Mon: 2.5  InUse: 1.2       Shr:  1  PSz:  0   Shr_PhysB:  0.01
Ded:  1        Avl:   -                   Ded:  1  APP:  0.0 Ded_PhysB:  0.00

Host         OS  M Mem InU Lp  Us Sy Wa Id  PhysB  Ent  %EntC Vcsw PhI
-------------------------------------shared--------------------------------------
--
bud151       A53 S 1.0 0.3  4   0  0  0 99   0.00  0.80   0.5  325   0
bud155       A53 S 0.5 0.3  2   0  1  0 98   0.00  0.10   3.7  279   0
bud152       A53 S 0.5 0.3  2   0  1  0 98   0.00  0.10   3.5  262   0
```

Shared processor partitions

```
------------------------------------dedicated------------------------------------
bud153       A53 C 0.5 0.3  2   0  0  0 99   0.00
```

Dedicated processor partition

Micro-Partitioning and Applications

- Applications do not need to be aware of Micro-Partitioning
- Applications that may not benefit from Micro-Partitioning:
  - Applications with a strong response time requirement for transactions may find Micro-Partitioning detrimental:
    - Because virtual processors can be dispatched at various times during a timeslice
    - May result in longer response time with too many virtual processors:
      - Each virtual processor with a small entitled capacity is in effect a slower CPU
    - Compensate with more entitled capacity (2-5% PUs over plan)
  - Applications with polling behavior
  - CPU intensive application examples: DSS, HPC
- Applications that are good candidates for Micro-Partitioning:
  - Ones with low average CPU utilization, with high peaks:
    - Examples: OLTP, web applications, mail server, directory servers