

AIX 5L V5.3 performance tools update (part 1)

Hsian-Fen Tsao

IBM eServer Solutions Enablement

April 2005

Table of Contents

Abstract.....	1
Overview of SMT and Micro-Partitioning.....	1
Impact on CPU utilization reporting.....	2
PURR and new metrics.....	3
AIX 5L V5.3 performance tools updates.....	4
Sample configuration for this paper.....	4
The smtctl command.....	4
The lparstat command.....	5
The mpstat command.....	8
Summary.....	10
Additional information.....	12
About the Author.....	13
Trademarks and Disclaimers.....	14

Abstract

IBM® POWER5™ technology introduces Simultaneous Multi-Threading (SMT) and Micro-Partitioning™ to significantly improve system performance and utilization. These technologies literally obsolesce the traditional, sample-based approach to measuring processor utilization. Therefore, a new hardware capability, a register called Process Utilization Resource Register (PURR), is implemented on POWER5 to aid performance management software in capturing relevant measurements. The PURR tracks the actual count of time a processor resource has consumed for a particular task, allowing highly granular measurement of utilization—both at the hardware-thread level and at the partition level. Consequently, performance tools in AIX 5L™ V5.3 have been enhanced to reflect new CPU utilization metrics. Additional performance tools are included and some tools have been updated.

In this paper (part 1), we first provide a brief overview of SMT and Micro-Partitioning technologies. We will also outline their impact on traditional utilization measurement, the usage of PURR, and new metrics. Then, we focus on the new text-based AIX 5L V5.3 performance tools, such as: `smtctl`, `lparstat`, and `mpstat`. These tools' usage and importance, from a tuning and analysis perspective, will be demonstrated by using real-world data collected under five micro-partitions within an SMT-enabled environment on a POWER5-based AIX 5L V5.3 server.

Overview of SMT and Micro-Partitioning

The processor-memory performance gap is one of the most challenging bottlenecks in microprocessor performance. System architects have attempted to bridge this gap by using various hardware solutions. IBM provided a hardware enhancement called Hardware Multithreading (HMT) for RS64 processor¹. HMT is a technique for tolerating memory latency by utilizing idle cycles in the CPU that normally would be stalled waiting on memory access. The RS64 implementation supports two hardware threads and switches between the two when the active thread experienced a long latency event such as a cache miss. Although HMT increases resource utilization capacity, it allows only one thread to be active at any given time. Using a completely different architecture, the POWER4™ chip is designed to have two processor cores on a single chip—each with its own on-chip L1 cache. Efficiency is increased by doubling the processors on a chip and having the two cores share a common L2 cache.

With POWER5, IBM not only brings in similar topology, as with POWER4, but also more advanced technologies, specifically, Simultaneous Multi-Threading and Micro-Partitioning.

- SMT is currently a preferred solution for deeply pipelined processors. It is a hardware enhancement and is quite different from the traditional context-switching multithreading design. SMT supports *simultaneous* execution of two hardware threads on each processor core, without switching between the two threads as is done in an HMT implementation. Thus, underutilized processing capacity can be utilized by *concurrently* executing two instruction streams. To the operating system, SMT makes a single processor appear as two logical processors, one per hardware thread. Each of these two threads is supported as a separate logical processor by AIX 5L V5.3.

¹See the document, "Performance Workloads in a Hardware Multithreaded Environment." A link to this document can be found in the "Additional information" section of this paper.

- Micro-Partitioning (a.k.a. shared processor partitions) is the mapping of virtual processors to physical processors. It provides the ability to share processors among more partitions in the system than there are physical processors. Stated another way, Micro-Partitioning allows fractional processor allocations. The result of this is that the number of partitions which can be created on a given platform is not restricted by the actual number of processors on the system. This enables enterprises to maximize the number of workloads that can be supported on a server simultaneously through the more flexible assignment of resources to various partitions.

Compared to the logical partitioning (LPAR) implementation delivered with POWER4, POWER5 virtualization technologies bring exciting new capabilities to the platform. These capabilities include Micro-Partitioning and virtual I/O (disk and communication adapters), both available in the Advanced POWER Virtualization option for the IBM eServer™ p5 family of servers. Physical processors, memory, and I/O devices are *virtualized*, which enables these resources to be *shared* by multiple partitions.

For instance, physical processors are abstracted into *virtual* processors² that can be assigned to partitions. Unlike LPAR on POWER4 that required at least one entire microprocessor per partition, POWER5 Micro-Partitioning has finer granularity, which allows allocating fractions of processors to a partition. The result of this, as already inferred, is the ability to run more partitions on a server than there are physical microprocessors (e.g., five partitions can exist on a 4-way POWER5 server). Each POWER5 physical processor can be “sliced” into as many as 10 partitions; and a maximum of 254 partitions may be active at the same time on an eServer p5 server.

Partition attributes are defined via a user interface for partitions—called the Hardware Management Console (HMC)³. One of the important attributes of Micro-Partitioning is the partition type: capped or uncapped. A capped partition is not allowed to exceed its processor entitlement until that entitlement is changed manually. An uncapped partition can exceed its processor entitlement if the partition needs more power and CPU resources are available in the shared processor pool. The “partition type” attribute is important since it affects both performance and license charges.

Impact on CPU utilization reporting

These new technologies (SMT, Micro-Partitioning, and the virtualization methodologies that exploit them) demonstrate their value in many areas. However, the added complexity affects the accurate reporting of processor utilization when performance measurements are based on the conventional, sample-based approach. The traditional sampling of utilization information is performed in the system clock interrupt handler, which occurs 100 times a second. Each sample corresponds to a 10-millisecond (1/100th of a second) local timer tick. This sampling rate is fixed. At each interrupt, one of four utilization categories is charged with one millisecond of processor time. These utilization categories are:

- user
- system
- iowait
- idle

² It is an attribute of a partition. It defines the number of processors exposed to the partition. With simultaneous multi-threading, the number of processors observed by an AIX 5L V5.3 partition is twice the number of physical CPUs.

³ More information can be found regarding the HMC by viewing the Redbooks Technote link provided in the “Additional information” section of this paper.

The utilization category is chosen based on the state of the interrupted thread. Using a performance monitor tool (e.g., vmstat), these system-wide or per-processor statistics are monitored for an interval and are calculated as averages—the values are expressed as utilization percentages for a machine or a processor. This statistic is most commonly used to determine *idle* capacity—indicating the available amount of performance capacity still remaining in a server. Idle time is usually more complex to calculate than are the other three categories (user, system, and iowait). With respect to POWER5 technologies, the combination of SMT and Micro-Partitioning makes idle capacity's calculation even more complex.

The problem with the traditional fixed incremental sampling approach is that it assumes the dispatch cycle of each virtual CPU is the same. This is not necessarily true in shared processor partition environments, where the variable rate of dispatch cycle is a result of the POWER5 scheduling algorithm invoked by the POWER Hypervisor™⁴. The opportunistic redirection of cycles is one of the strengths of POWER5. If a virtual processor has nothing to do, the operating system 'cedes' the virtual processor to the Hypervisor. In this case, the remaining cycles may be scheduled for others to use. Over a period of time, this redirected use of virtual processors can add up.

As an example, let's imagine that a shared 4-way partition with an entitlement of one physical processor is running one job. Using the fixed incremental sampling algorithm, this would be reported as 25% busy and 75% idle. But in reality, there might not be any available capacity because that one job could be dispatched to the physical processor almost all the time. This issue is even more obvious with SMT enabled. Each logical processor could be viewed as a separate processor. If one logical processor is 100% busy and the other logical processor is idle, utilization would be reported as only 50%. This implies that there is a remaining capacity of 50%; but the physical processor is actually 100% busy. The sampling methodology has over-estimated idle time and understated CPU utilization.

PURR and new metrics

As briefly mentioned earlier in this paper, to deal with this complex measurement and reporting issue, POWER5 introduces a Processor Utilization Resource Register (PURR) for each hardware thread. PURR provides proper process/system accounting and processor utilization measurements. PURR is a per-hardware thread register, incrementing a 64-bit counter (similar to the Time Base (TB) register), but the PURR is not incremented all the time (as is the TB). The PURR is incremented when one or more instructions are dispatched in a cycle. The increment is done automatically so the operating system can always get the up-to-date value. With SMT enabled, each logical processor has a PURR. Thus, it is possible to measure performance of each hardware thread separately. In the previous example where one logical processor is 100% busy and the other is idle, the reported utilization would no longer be 50%, but the correct 100%. This is because the busy logical processor would receive almost all the PURR increments, while the other would practically none—meaning 100% of the PURR increments would be reported as busy.

A new "physical" CPU utilization metric based on PURR values is needed (Table 1). This new metric uses the per-thread PURR to measure the thread's processor time and uses the Time Base register to measure the elapsed time. As an example, the value expressed in *physc* of command "sar -P ALL" with SMT enabled is calculated using the new metric (delta PURR/delta TB). This metric represents the fraction of a physical processor consumed by a logical processor, which

⁴ The POWER Hypervisor is the firmware layer that runs underneath AIX, Linux and i5/OS on eServer pSeries machines. It provides the capabilities like hardware resource management, partition, Capacity on Demand.

indicates the relative SMT resource split between hardware threads. In a shared processor partition environment, PURR is used to measure the time a virtual processor actually runs on a physical processor. The partition time becomes “virtual,” so the Hypervisor maintains the virtual Time Base in the partition’s PURR. With SMT integrated with shared processor partitions, each virtual processor supports two logical processors. The threads’ PURRs are now virtualized by the POWER Hypervisor. In this case, PURR is used to measure both the relative SMT split between threads and the relative fraction of time a partition ran on a physical processor.

New Metrics	Information provided
(delta PURR/delta TB)	The fraction of a physical processor consumed by a logical processor
$\%sys = (\text{delta PURR in system mode} / \text{entitled PURR}) * 100$ where entitled PURR = (ENT * delta TB), and ENT is entitlement in # of processors (entitlement/100)	Physical processor utilization in the traditional category: user, system, iowait, and idle
Sum (delta PURR/delta TB) for each logical processor in a partition	Physical processor consumed (PPC) over an interval
$(PPC / ENT) * 100$	Partition percentage of entitlement consumed
(delta PIC/delta TB) where PIC is the pool idle count which represents clock ticks when PHYP was idle	Available pool of processors
(sum of old 10-millisecond tick-based %sys and %user)	Logical processor utilization

Table 1: New metrics using PURR-based method

AIX 5L V5.3 performance tools updates

The performance tools of AIX 5L V5.3 have been adapted for measuring the new POWER5 technologies and its processor utilization metrics.

Sample configuration for this paper

We will discuss these new performance tools by using sample data. The examples throughout this paper were taken from a 4-way POWER5 server with the following specifications:

- Virtualized into five Micro-Partitions
- Each partition configured as 8/10ths of a POWER5 CPU
- Two virtual CPUs with SMT enabled
- Running with capacity ‘capped’

The test workload consists of a Java™ 2 Enterprise Edition (J2EE™) application running under:

- AIX 5L V5.3
- DB2 Universal Database™ (UDB)
- WebSphere® environment

The smtctl command

The text-based **smtctl** command is new with AIX 5L V5.3. It controls the SMT mode of the partition. It turns SMT on or off system-wide, either immediately or at the next boot. The SMT mode persists across system boots and, by default, is enabled. Its usage and flags are explained below. There is also an example listing of SMT settings on the eServer p5 server just described.

Command	Usage	What it does
smtctl	This SMT command allows users to control the SMT mode of partition	
	-m off on	Mode flag that enables (default) or disables SMT
	-w boot now	Mode change takes effect immediately or at next reboot
	No flag	Report SMT setting on your system (<i>Example 1</i>)

```
# smtctl

This system is SMT capable.
SMT is currently enabled.
SMT boot mode is set to enable.
Processor 0 has 2 SMT threads
SMT thread 0 is bound with processor 0
SMT thread 1 is bound with processor 0
Processor 2 has 2 SMT threads
SMT thread 2 is bound with processor 2
SMT thread 3 is bound with processor 2
```

Example 1: SMT setting on the system

The lparstat command

The text-based **lparstat** command is also new with AIX 5L V5.3. It shows statistics for shared processor partitions, Hypervisor information and utilization statistics. The information is given and best viewed over intervals. The flags for the lparstat command are explained below. There are also three example lparstat listings for the eServer p5 server described earlier.

Command	Flags	What it does
lparstat	-i	Lists partition configuration details (<i>Example 2</i>)
	-h	Displays the default monitoring information, including summary statistics on POWER Hypervisor (<i>Example 3</i>)
	-H	Displays POWER Hypervisor information (<i>Example 4</i>)

```
# lparstat -i

Node Name                : lcb21
Partition Name           : lcb21
Partition Number         : 7
Type                     : Shared-SMT
Mode                     : Uncapped
Entitled Capacity        : 0.80      <- Min. 1/10th of a processor. Would be an
                                   Integer number in dedicated partition

Partition Group-ID       : 32775
Shared Pool ID           : 0        <- Shared partition information only
Online Virtual CPUs      : 2
Maximum Virtual CPUs     : 8
Minimum Virtual CPUs     : 2
Online Memory            : 3072 MB
Maximum Memory           : 3072 MB
Minimum Memory           : 2816 MB
Variable Capacity Weight5 : 128
Minimum Capacity         : 0.80    <- Min. capacity partition will grant to the OS
Maximum Capacity         : 0.80    <- Max. capacity partition will grant to the OS
Capacity Increment       : 0.01    <- In increments of 1/100th of a processor
Maximum Physical CPUs in system : 4
Active Physical CPUs in system : 4
Active CPUs in Pool      : -
Unallocated Capacity     : 0.00
Physical CPU Percentage   : 40.00%  <- Would be 100% in dedicated partition
Unallocated Weight       : 0
```

Example 2: partition configuration details. Note: Comments/tips after "<-" are shown here in red for ease of reading.

⁵ If your partition is in uncapped mode, you must specify uncapped weight of that partition. Number between 0-255 that represents the relative share of additional capacity the uncapped partition is eligible to receive. An uncapped partition with uncapped weight at 0 is functionally identical to a capped partition.

Recommendation: Increasing the number of virtual processors may degrade performance for various reasons (i.e., smaller time slice, bigger dispatching latency, higher cache miss rate, and more lock contention). Therefore, the recommendation is that as few virtual processors as possible are configured for each partition. It is better to have a lesser number of virtual processors with higher capacity than a larger quantity, each with a small amount of processing power.

```
# lparstat -h 5 3

System configuration: type=Shared mode=Capped smt=On lcpu=4 mem=3072 psize=4 ent=0.80

%user  %sys  %wait  %idle  physc  %entc  lbusy  app    vcsw  phint  %hypv  hcalls
-----
 67.6  16.1   1.4   14.9   0.73  91.1   79.4   0.36   23709  7757   30.4   172921
 66.5  16.7   1.6   15.1   0.73  91.1   77.9   0.32   25587  8235   27.8   185169
 67.5  15.6   1.1   15.8   0.73  91.7   75.6   0.42   27311  7919   34.1   175459
```

Example 3: lparstat default monitoring

Each column of Information (shown above) with command “lparstat –h” is explained below:

Metrics	What it measures	Comments/Tips
%usr %sys %iowait %idle	Physical CPU utilization in user, system, iowait, & idle	<ul style="list-style-type: none"> This calculation is relative to partition entitlement. The number of logical or virtual processors is not relevant.
Shared Partition only Metrics:		
physc	Physical processor consumed	<ul style="list-style-type: none"> This indicates how much of the physical processor a partition is getting. Physical processor consumed is equal to percentage of %entc times entitlement.
%entc	Percentage of entitled capacity consumed	<ul style="list-style-type: none"> This shows relative entitlement consumption for each logical processor and system average utilization calculation from logical processors utilization It uses the new metrics (PPC/ENT)*100 where PPC=0.73 and ENT=0.8 (in this example). Physc and %entc are important as problem indicators (i.e., unbalanced workload). They can be used to determine if a partition has more than its entitlement in uncapped partition or if there is spare capacity in capped partition.
lbusy	Percentage of logical processor utilization	<ul style="list-style-type: none"> This describes how busy the online logical CPUs are. It can be used to see if more virtual processors should be added to a partition
app	Available physical processor in shared pool	<ul style="list-style-type: none"> This is used to determine how much capacity is available in the shared pool. This provides a measurement of spare cycles in the platform that could be obtained for uncapped partitions.
vcsw	Number of virtual processor's context switch	<ul style="list-style-type: none"> Virtual processor hardware preemption—one measure of POWER Hypervisor overhead. It may be best to minimize the number of virtual processors in each partition, if many partitions are activated
phint	Number of phantom interrupts received	<ul style="list-style-type: none"> A phantom interrupt is targeted for another partition that shares physical processors with the partition that inadvertently received the interrupt. Occurrences of 'phint' are extremely rare for dedicated partitions. Shared partitions observe them more often. Overall, the performance degradation of handling 'phint' is small because the CPU cost to process 'phint' is small.
Optional metrics only with –h flag specified:		

AIX 5L V5.3 performance tools update
Part 1

Metrics	What it measures	Comments/Tips
%hypv	Percentage of total time spend in POWER Hypervisor	<ul style="list-style-type: none"> This indicates the %hypv relative to entitlement is around 30% of system resources on this shared partition. It is the result of the H_CEDE call (<i>Example 4</i>) being made to put the virtual processors into a wait state since there is no useful work to do.
hcalls	Number of POWER Hypervisor calls executed	<ul style="list-style-type: none"> See <i>Example 4</i> for details.

```
# lparstat -H
System configuration: type=Shared mode=Capped smt=On lcpu=4 mem=3072 psize=4 ent=0.80
Detailed information on Hypervisor Calls
```

Hypervisor Call	Number of Calls	%Total Time Spent	%Hypervisor Time Spent	Avg Call Time(ns)	Max Call Time(ns)
remove	5062	0.0	0.1	541	10801
read	36394	0.1	0.2	280	9434
nclear_mod	0	0.0	0.0	1	0
page_init	4146	0.0	0.1	1185	18454
clear_ref	0	0.0	0.0	1	0
protect	0	0.0	0.0	1	2917
put_tce	48578	0.2	0.6	560	11198
xirr	24451	0.2	0.7	1424	11971
eoi	16694	0.1	0.3	800	9260
ipi	0	0.0	0.0	1	2690
cppr	0	0.0	0.0	1	0
asr	0	0.0	0.0	1	0
others	0	0.0	0.0	1	0
enter	7713	0.0	0.1	477	9463
cede	19177	29.5	97.3	246606	384518607
migrate_dma	0	0.0	0.0	1	0
put_rtce	0	0.0	0.0	1	0
confer	200	0.2	0.5	120819	2521932
prod	9712	0.1	0.2	826	10371
get_ppp	6	0.0	0.0	1689	4545
set_ppp	0	0.0	0.0	1	0
purr	0	0.0	0.0	1	0
pic	6	0.0	0.0	1030	4690
bulk_remove	785	0.0	0.0	1710	10333
send_crq	0	0.0	0.0	1	0
copy_rdma	0	0.0	0.0	1	0
get_tce	0	0.0	0.0	1	0
send_logical_lan	0	0.0	0.0	1	0
add_logical_lan_buf	0	0.0	0.0	1	0

Example 4: Detailed breakdown of Hypervisor time by call type

The POWER Hypervisor uses a small percentage of memory and CPU resources. This overhead is associated with Virtual Memory Management (VMM) and is used for the POWER Hypervisor dispatcher. It is also used for virtual processor data structures that may cause performance overhead (i.e., dispatching overhead of virtual processor in terms of saving and restoring state), as well as for other similar purposes. The overhead should be minor for most workloads, but its impact increases with extensive amounts of page-mapping activity.

The new POWER Hypervisor calls support the scheduling heuristics of minimizing idle time and improving locking in shared processor partitions. The operating system can *cede* the virtual processor to the POWER Hypervisor, enabling it to schedule the rest of the dispatch cycle for

other purposes. When one process cannot make forward progress, the operating system can make the virtual processor *confer* the remainder of a dispatch cycle to another virtual processor in the partition. When a virtual processor cedes, it is put into a sleep state by the POWER Hypervisor and can be awakened by a *prod* from another virtual processor or by an external interrupt. These primitives are used for performance reasons, but may also be associated with dispatching, timer (virtualized hardware Decrementer), and interrupt latencies, which may cause their own overhead. The latencies statistics can be monitored and reported using 'mpstat' (Example 5).⁶

The mpstat command

The text-based mpstat command is new with AIX 5L V5.3. It collects and displays performance statistics for all logical processors operating in the logical partition. Its flags are explained below. There are also two example listings of mpstat commands issued on the eServer p5 server described earlier.

Command	Flags	What it does
mpstat	-a	Displays all the statistics; 29 metrics. (Example 5)
	-s	Displays SMT utilization; available only in SMT enabled mode. (Example 6)
	-i	Displays detailed <i>interrupts</i> statistics; subset of '-a'.
	-d	Displays detailed <i>affinity</i> and <i>migration</i> statistics for AIX threads and <i>dispatching</i> statistics for logical processors; subset of '-a'.

When the `-a` flag is specified, the mpstat command reports 29 sets of metrics, as you can see in the two-tiered report listing shown in Example 5.

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

cpu  min  maj  mpcs  mpcr  dev  soft  dec  ph  cs  ics  bound  rq  push  S3pull  S3grd  S0rd
S1rd  S2rd  S3rd  S4rd  S5rd  sysc  us  sy  wa  id  pc  %ec  ilcs  vlcs
0  0  0  0  0  4294 19074 2059 2133 31577 10431 0 0 0 0 0 87.6
8.3 0.0 4.1 0.0 0.0 62543 72.2 17.9 0.8 9.0 0.18 22.4 1141 5724
1  0  0  0  0  4309 19135 2049 1872 31992 10661 0 0 0 0 0 88.1
8.3 0.0 3.6 0.0 0.0 62169 72.4 17.5 0.6 9.5 0.18 22.1 1148 5742
2  0  0  0  0  4254 20869 2050 2001 32426 9502 1 1 0 0 0 88.0
8.3 0.0 3.7 0.0 0.0 65730 72.8 18.0 0.6 8.6 0.18 23.1 1232 5737
3  0  0  0  0  4320 22094 3436 1895 33579 10196 0 0 0 0 0 88.6
7.9 0.0 3.5 0.0 0.0 61976 71.4 18.3 0.8 9.5 0.18 22.2 1180 6058
ALL 0 0 0 0 17177 81172 9594 7901 129574 40790 1 1 0 0 0 88.1
8.2 0.0 3.7 0.0 0.0 252418 64.8 16.1 1.3 17.7 0.72 89.8 2350 11630
```

Example 5: mpstat detailed monitoring

The information provided with command "mpstat -a" is summarized below:

⁶ See a detailed definition of POWER Hypervisor calls in the "Advanced POWER Virtualization on IBM eServer p5 Servers" reference. A link to this document is found in the "Additional information" section of this paper.

AIX 5L V5.3 performance tools update
Part 1

Metrics	What it measure	Comments/Tips
us, sy, wa, id	Physical CPU utilization in user, system, iowait, and idle for each logical processor	
min, maj	Minor and major page faults	
sysc	Number of system calls	System calls are invoked by the program directly and indirectly.
pc	Fraction of physical processor consumed	This indicates the percentage of SMT thread that is busy; SMT or shared mode only.
%ec	Percentage of entitlement consumed	Shared mode only.
Detailed interrupts statistics:		
mpcs,mpcr	Number of mpc send and receive interrupts	Interrupt used to communicate between processors.
dev	Number of device interrupts	This is a hardware I/O interrupt.
soft	Number of software interrupts	This is a machine instruction similar to a hardware interrupt that saves some state and branches to a service routine. System calls are implemented with software interrupt instructions that branch to the system call handler routine.
dec	Number of Decrementer interrupts	This is the hardware facility used to generate time-based interrupts. AIX loads a value in the register and the processor decrements it. When it reaches 0, an interrupt is sent.
ph	Number of phantom interrupts	This is the number of device interrupts received by the partition, but targeted to another partition in the pool. The operating system simply returns them to the POWER Hypervisor.
Detailed affinity (Statistics of percentage of thread re-dispatches with scheduling affinity domain 0-5) and migration statistics for AIX threads and dispatching statistics for logical processor:		
S0rd	The process re-dispatch occurs within the same logical processor.	This happens in the case of SMT enabled systems.
S1rd	The process re-dispatch occurs within the same physical processor, among different logical processors.	This involves sharing of the L1, L2, and L3 cache.
S2rd	The process re-dispatch occurs within the same processor chip, but among different physical processors.	This involves sharing of the L2 and L3 cache.
S3rd	The process re-dispatch occurs within the same MCM module, but among different processor chips.	
S4rd	The process re-dispatch occurs in the same Central Processing Complex (CPC) plane, but among different Multichip Modules (MCMs).	This involves access to the main memory or L3-to-L3 transfer.
S5rd	The process re-dispatch occurs outside of the CPC plane.	
cs, ics	Number of context switches and involuntary context switches	A context switch occurs at the operating system level. An involuntary context switch is caused typically by the thread's time slice expiring.
ilcs, vlcs	Number of involuntary and voluntary logical CPU context switches	Switches initiated by H_CEDE, H_CONFER, and H_PROD are voluntary context switches, while time slice-related context switches are involuntary. They are a measure of Hypervisor activity; meaning the hardware preemptions; shared mode only.
cpu	Logical CPU number	In this case, the logical CPU number is 0 – 3.
bound	Total number of threads bound to a particular processor	
rq	Number of threads on the run queue; run queue size for each logical processor	This metric, similar to 'pc,' is useful to see if the workload is balanced.
push	Number of thread migrations to another processor due to starvation load balancing	
mig	Total number of thread migrations to another logical processor	
S3pull	Number of migrations outside S3rd scheduling affinity domain	This measures migration of threads across MCM boundaries due to idle stealing.
S3grd	Number of dispatches from global run queue, outside S3rd scheduling affinity domain	This indicates the dispatches of threads from the global run queue across MCM boundaries.

General memory affinity considerations: Memory and processors that are directly connected are said to fall within a single affinity domain. A processor can access memory attached to its local memory domain faster, due to lower latency, than it can access memory attached to other memory domains. AIX 5L V5.3 attempts to satisfy page faults from the memory closest to the processor that generated the page fault. Memory affinity is meaningful for dedicated partitions, but not for shared partitions, because virtual processors may be dispatched on different physical processors during the time a partition is running. Thus, applications that require memory affinity should not be implemented in a shared processor partition environment.

Moving on to Example 6, when the **-s** flag is specified, the **mpstat** command reports SMT utilization if SMT is enabled. The report displays the virtual processor utilization, with the logical processor (thread) data associated with each virtual processor. For a micro-partition environment, the number of logical processors is also displayed with the entitled processor capacity for the partition.

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

Proc0          Proc2          <- physical CPUs
39.61%         39.57%         <- Would be 100% in dedicated partition
cpu0   cpu1   cpu2   cpu3   <- 4 logical CPUs with SMT enabled
20.50% 19.11% 20.92% 18.66%
```

Example 6: mpstat SMT utilization

In Example 6, physical processor Proc0 is busy at 39.61%, which is dispatched on logical processor cpu0 (20.50%) and on logical processor cpu1 (19.11%).

Summary

The technologies introduced with the delivery of the IBM POWER5 processor are nothing short of enormous, when compared to its predecessor POWER4—and even when compared to the industry, in general. The implementation of both Simultaneous Multi-Threading (SMT) and Micro-Partitioning supports the flexibility for IT shops to maximize the efficiency of their large number of applications through as many as 254 logical partitions. Put more plainly, partitions can take advantage of much more granular assignments of processing power. This processing power can be modified dynamically or during reboot. And, there are a host of new metrics and tools to precisely observe and tweak each partition's use of its processor, memory, and 'I/O wait' resources. These metrics are captured at both the hardware-thread level and at the partition level.

This paper provided an overview of SMT and Micro-Partitioning. The reader also now has a rather thorough grasp of three performance commands (smtctl, lparstat, and mpstat) that are critical to managing logical partitions on the eServer p5 platform.

While this paper focused on the minutiae related to SMT and Micro-Partitioning on POWER5, we should take a final moment to talk, at a higher level, about why these technologies are more critical than ever in today's competitive, business environment. POWER5 takes virtualization to a much higher level—out of the realm of theory and into the world where we use a host of disparate applications (running under many operating systems) to conduct real business tasks and processes. Virtualization capabilities increase overall system resource utilization and position POWER5-driven servers as excellent candidates for server consolidation. As most IT professionals know, server consolidation is becoming a critical issue for many enterprises of all

AIX 5L V5.3 performance tools update
Part 1

sizes. It simplifies and optimizes the IT environment—both its hardware resources and the skill sets required to support it. This leads to dramatic cost savings in delivering IT services, while at the same time, lifting reliability and availability of those same IT services.

[NOTE: Additional AIX 5L V5.3 performance tools and examples will be discussed in Part 2 of this series of technical papers.]

Additional information

- Performance Workloads in a Hardware Multithreaded Environment
www.hpcaconf.org/hpca8/caecw-02/s3p2.pdf
- IBM eServer p5, pSeries, and AIX Information Center
<http://publib.boulder.ibm.com/infocenter/pseries/index.jsp>
- IBM Publications Center
www.elink.ibm.link.ibm.com/public/applications/publications/cgibin/pbi.cgi?CTY=US
- Redbooks
ibm.com/redbooks
 - Redbook: Advanced POWER Virtualization on IBM eServer p5 Servers: Introduction and Basic Configuration (SG24-7940-00)
 - Redbook: Partitioning Implementations for IBM eServer p5 Servers (SG24-7039-02)
 - Redbooks Technote: What is a Hardware Management Console?
<http://publib-b.boulder.ibm.com/Redbooks.nsf/5193609f3941e9cf85256bc300724cfc/c9cbd629131dc2d485256d790059ac16?OpenDocument>
- White paper: IBM eServer p5 AIX 5L Support for Micro-Partitioning and Simultaneous Multi-threading
ibm.com/servers/aix/whitepapers/aix_support.pdf
- AIX 5L V5.3 Performance Management Guide and AIX 5L V5.3 Performance Tools Guide and Reference
<http://publib.boulder.ibm.com/infocenter/pseries/index.jsp>

About the Author

Hsian-Fen Tsao
IBM eServer Solutions Enablement

Hsian-Fen Tsao is an IBM technical consultant for eServer p5/AIX software vendors. Before joining the Solution Enablement Group in Austin, Texas; she worked in the pSeries/AIX Performance group for 10 years. Her experiences include database (On-Line Transaction Processing, Decision Support Systems), Web server, TCP/IP, Java/WebSphere, and POWER5 virtualization engine. You can contact her at htsao@us.ibm.com

Trademarks and Disclaimers

© IBM Corporation 1994-2005. All rights reserved.

References in this document to IBM products or services do not imply that IBM intends to make them available in every country.

The following terms are trademarks of International Business Machines Corporation in the United States, other countries, or both: IBM, eServer, pSeries, AIX, AIX 5L, POWER, POWER5, Micro-Partitioning, Hypervisor, DB2, and WebSphere.

Java and all Java-based trademarks are trademarks of Sun Microsystems, Inc. in the United States, other countries, or both.

Microsoft, Windows, Windows NT, and the Windows logo are trademarks of Microsoft Corporation in the United States, other countries, or both.

UNIX is a registered trademark of The Open Group in the United States and other countries.

Linux is a trademark of Linus Torvalds in the United States, other countries, or both.

Other company, product or service names may be trademarks or service marks of others.

Information is provided "AS IS" without warranty of any kind.

All customer examples described are presented as illustrations of how those customers have used IBM products and the results they may have achieved. Actual environmental costs and performance characteristics may vary by customer.

Performance is based on measurements and projections using standard IBM benchmarks in a controlled environment. The actual throughput or performance that any user will experience will vary depending upon considerations such as the amount of multiprogramming in the user's job stream, the I/O configuration, the storage configuration, and the workload processed. Therefore, no assurance can be given that an individual user will achieve throughput or performance improvements equivalent to the ratios stated here.