

# **IBM EnergyScale for POWER9 Processor-Based Systems**

*February 2021*

*Martha Broyles  
Christopher J. Cain  
Chris Francois  
Todd Rosedahl  
Brian Veale*

Executive Overview .....	4
EnergyScale Features .....	5
Common Features .....	5
Power and Thermal Trending .....	5
Performance Boost .....	5
Power Capping .....	5
“Soft” Power Capping .....	5
Memory Performance .....	6
Energy-Optimized Fan Control .....	6
Processor Sleep .....	7
Processor Core Nap .....	7
Processor Folding .....	7
Safe Mode .....	7
Reasons for Temporary Safe Mode .....	8
Permanent Safe Mode .....	8
PowerVM Specific Features .....	9
Static Power Saver Mode .....	9
Dynamic Performance Mode (DPM) .....	10
Maximum Performance Mode (MPM) .....	10
Tunable Parameters .....	12
Idle Power Saver .....	12
EnergyScale for I/O .....	12
Frequency and Performance Measurement .....	13
Non PowerVM Specific Features .....	15
Power, Thermal, and Performance Measurement .....	15
Power Management .....	15
Frequency and Performance Measurement .....	15
User Interfaces .....	16
Overview .....	16
ASMI .....	17
Setting System Power and Performance Mode .....	18
Tunable Parameters .....	19
Idle Power Saver .....	20
CIM Client .....	21
Script to Read Average Power .....	22

Average Power Output .....	23
Script to Read Average Frequency .....	24
Average Frequency Output .....	25
Script to Read Thermal Data.....	26
Thermal Data Output .....	27
HMC.....	28
Setting System Power Management Mode .....	28
DCMI .....	29
Redfish .....	30
BMC GUI.....	31
EnergyScale Operating System Support .....	32
Processor Folding in AIX.....	33
Processor Folding in IBM i.....	34
Processor Folding in Linux .....	35
Query System Power and Performance Mode in AIX .....	36
Query Idle Power Saver and Dynamic Power Saver Tunable Parameters in AIX.....	37
Appendix I: System Requirements .....	38
Release Level FW910 .....	39
Feature Support .....	39
Frequency.....	40
Maximum Frequency by Core Count.....	41
Release Level FW920 .....	42
Feature Support .....	42
Frequency.....	43
Maximum Frequency by Core Count.....	43
Release Level FW921 .....	45
Feature Support .....	45
Frequency.....	46
Maximum Frequency by Core Count.....	47
Release Level OP910 / OP920.....	48
Feature Support .....	48
Frequency.....	48
Maximum Frequency by Core Count.....	49
Appendix II: Processor Usage and Accounting .....	50
Appendix III: Resources .....	52

## **Executive Overview**

The energy required to power and cool computers can be a significant cost to a business – reducing profit margins and consuming resources. In addition, the cost of creating power and cooling infrastructure can be prohibitive to business growth. In response to these challenges, IBM developed EnergyScale™ Technology for IBM Power systems. EnergyScale provides functions that help the user to understand and control IBM server power and cooling usage. This enables better facility planning, provides energy and cost savings, enables peak energy usage control, and increases system availability. Administrators may leverage EnergyScale capabilities to control the power consumption and performance of POWER processor-based systems to meet their particular data center needs.

In this paper, the functions provided by EnergyScale are described along with usage examples, and hardware and software requirements. Support and awareness of EnergyScale extends throughout the system software stack, and is included in the AIX, IBM i, and Linux operating systems. This paper focuses on features and functions found in systems based on POWER9 processors. For previous-generation IBM Power systems, please refer to companion paper, “IBM EnergyScale for POWER8 Processor-based Systems”.

## EnergyScale Features

EnergyScale provides many features to measure, monitor, and control both the power consumption and energy efficiency of POWER9 Systems. Not all features may be supported, see [Appendix I: System Requirements](#) for features supported for a specific release and system.

### Common Features

The following features are common across PowerVM and non PowerVM systems.

#### Power and Thermal Trending

EnergyScale provides continuous collection of real-time server power consumption. Administrators may use the power information to predict data center power consumption at various times of the day, week, or month. In addition, data may be aggregated to identify anomalies, manage electrical loads, and enforce system-level power budgets.

A measured ambient temperature can help identify data center “hot-spots” that need attention. Please note that the ambient temperature reported may vary from system model to system model due to variations in the placement of the ambient temperature sensor in the system.

See [User Interfaces](#) for supported interfaces to collect power and thermal data.

#### Performance Boost

EnergyScale provides additional performance in both PowerVM and non PowerVM modes. The system is designed to provide enough power and cooling for the processors to run all workloads – even the heaviest and most power hungry -- at a specified base frequency. However, most workloads do not consume all available power at the base frequency. In these cases, the frequency can be increased above that base frequency to take advantage of the additional power and thermal headroom in the system.

#### Power Capping

Power Capping enforces a user-specified limit on power consumption. See [User Interfaces](#) for supported interfaces to set a power cap. In most data centers and other installations, when a server is installed, a certain amount of power is allocated to it. Generally, the amount is what is considered to be a “safe” value, and it typically has a large margin of reserved, extra power that is never used. This is called the *margin power*. The main purpose of the power cap is not to save power but rather to give a data center operator the capability to reallocate power from existing systems to new systems by reducing the margin assigned to the existing servers. That is, power capping gives an operator the capability to add extra servers to a data center which previously had all available power allocated to its existing systems. It does this by guaranteeing that a system will not use more power than assigned to it by the operator. This is also called a “hard power cap”.

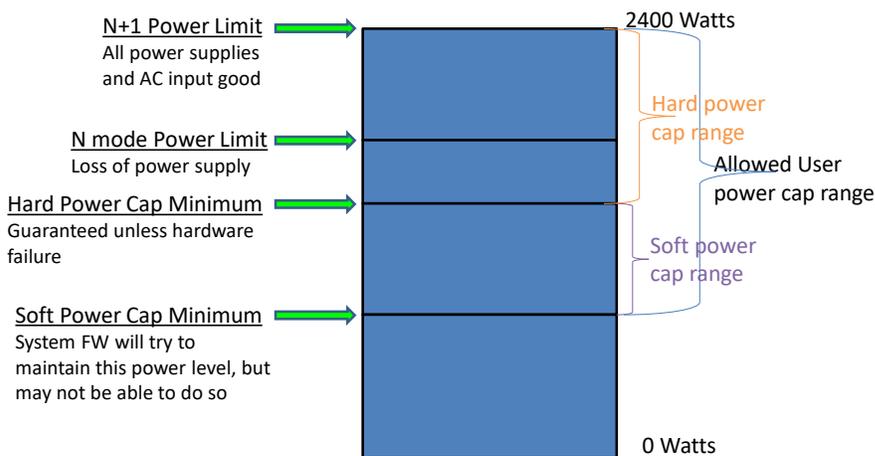
Previously, the data center administrator had to plan for the power consumption of the data center based on the Underwriters' Laboratories (UL) rating on the back of the servers being installed. The UL rating (commonly referred to as “label power”) on today's servers indicates the most power that a system could ever draw and is based on the capacity of the power supplies. It has to take into account a fully-configured system with the highest power-usage parts installed at the highest possible utilization.

#### “Soft” Power Capping

There are two power ranges into which the power cap may be set. When a power cap is set in the guaranteed range (“hard power cap” described above), the system is guaranteed to use less power than the cap setting. In order to meet this guarantee, extreme system configuration and environmental conditions must be accounted for. Setting a power cap in this region allows for the recovery of the

marginated power, but in many cases cannot be used to save power. Soft power capping extends the allowed power capping range further, beyond a region that can be guaranteed in all configurations and conditions. By setting a power cap in this soft region, the system can be set to save power by running at a lower power/performance point. If the power management goal is to meet a particular power consumption limit, then soft power capping is the mechanism to use. The performance impacts of a particular power cap setting can be determined by reading the power and CPU speed information from a CIM client. Note that the failure to enforce a “soft” power cap below the minimum guaranteed range is not an error, and will not result in any error. The system will continue to operate normally with all EnergyScale features at the minimum-supported frequency until the soft power cap is disabled or raised.

## Example Power Capping Ranges



20

## Memory Performance

As part of power capping the memory may be throttled to reduce additional power. This will only occur after the processors have been reduced to minimum frequency and the system power is still above the power cap. By default, system machine type models (MTM's) 9080-M9S, 9222-80H, and 9080-M98 will have memory throttled when there are 8 high power CDIMMs installed under any processor socket due to system power constraints which will impact applications that require a very high memory bandwidth workload. The high power CDIMMs previously mentioned are the 512GB DDR4 CDIMM (limited to about 160 GB/s/socket when socket has 8 CDIMMs) and the 128GB DDR3 CDIMM (limited to about 150 GB/s/socket when socket has 8 CDIMMs). Note that non-high power CDIMMs will be able to achieve up to 230 GB/s/socket when socket has 8 CDIMMs.

## Energy-Optimized Fan Control

Cooling fans contribute significantly to the overall power consumption of a given computer. In order to minimize energy expended on cooling and to minimize the energy wasted “over-cooling” a system,

firmware on all POWER9 systems will adjust fan speed in response to real-time temperatures of the system components. Note that in comparison to previous-generation systems, exhaust temperatures may increase, however all components will still be within allowed RAS temperature envelopes. This is a natural product of the optimization of fan speed, component temperature, and fan power consumption.

## **Processor Sleep**

New with Power9, the system can convert the power reduction from sleeping cores into increased performance on the active cores. On PowerVM systems, this will be done when the user de-configures a processor core. On non PowerVM systems, the user can also de-configure cores, but additionally, the operating system will put cores to sleep automatically when long periods of idleness are encountered.

## **Processor Core Nap**

IBM POWER processors use a low-power mode called Nap that stops processor execution when there is no work to do on a particular processor core. The latency of exiting Nap falls within a partition dispatch (context switch) such that the hypervisor firmware can use it as a general purpose idle state. When the Operating System detects that a processor thread is idle, it yields control of a hardware thread to the hypervisor. The hypervisor immediately puts the thread into Nap. If the processor core is in a shared processor pool (the set of cores being used for micro-partition dispatching) and there is no micro-partition to dispatch, the hypervisor puts the thread into Nap mode. When all hardware threads running on a given processor core enter Nap mode, the whole core enters Nap mode, which allows the hardware to clock off most of the circuits inside the processor core. Reducing active power consumption by turning off the clocks allows the temperature to fall, which further reduces leakage (static) power of the circuits resulting in a cumulative effect. On some systems, unlicensed cores are kept in nap mode until they are licensed and return to nap mode when they are unlicensed again.

## **Processor Folding**

While Processor Core Nap provides substantial energy savings when processors become idle, additional savings can be realized if processors remain idle by intent rather than by happenstance. Processor Folding is a consolidation technique that dynamically adjusts, over the short-term, the number of processors available for dispatch to match the number of processors demanded by the workload. As the workload increases, the number of processors made available increases; as the workload decreases, the number of processors made available decreases. Processor Folding increases energy savings during periods of low to moderate workload because unavailable processors remain in low-power idle states longer than they otherwise would. Since the idle condition is intentional, the hypervisor is also advised to exploit special purpose idle states available on some POWER9 systems that can reduce power consumption even further than with Nap mode alone, but without the stringent latency requirement. Processor Folding achieves power savings similar to those that could be achieved by intelligent, utilization-based logical partition (LPAR) configuration changes, but it does so with much greater efficiency and fidelity, and without impacting the configuration or processor utilization of the LPAR.

## **Safe Mode**

All systems and firmware releases support safe mode. “Safe Mode” is a system mode where the firmware will automatically drop to a fixed lower frequency to keep the system thermal and power safe

and any hard power cap set by the client is still held. While in safe mode all power and thermal trending data will no longer be reported.

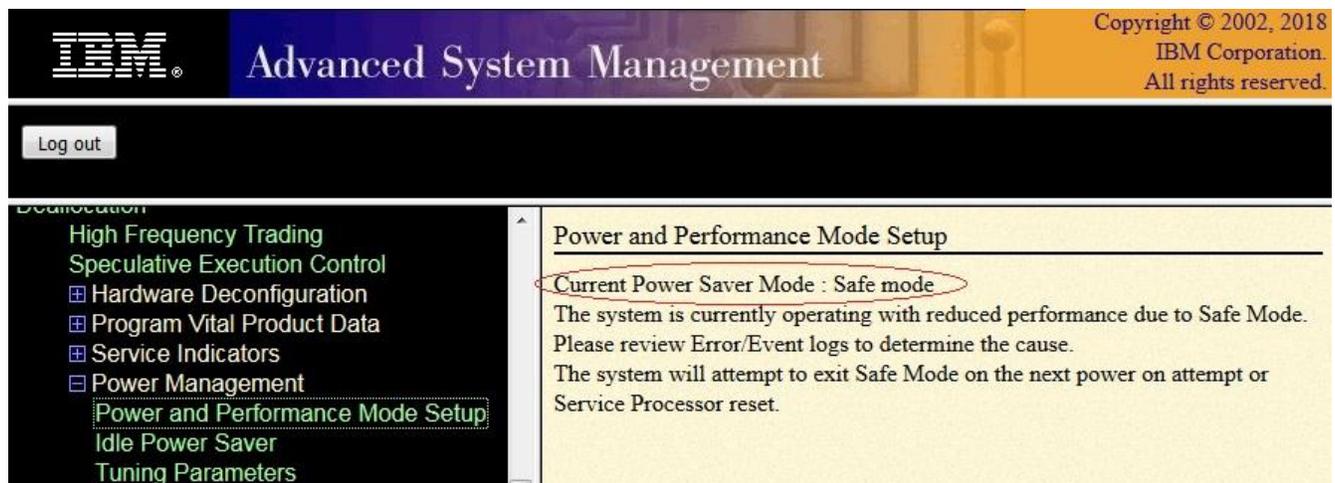
### **Reasons for Temporary Safe Mode**

The following *may* cause the system to temporarily enter safe mode, no user action is required to exit safe mode from these conditions and no user notification is provided when it occurs.

- Concurrent code update
- Mode changes to enter Dynamic Performance or Maximum Performance modes
- Error recovery

### **Permanent Safe Mode**

Certain hardware and firmware failures that cannot be recovered from can cause the system firmware to stay in safe mode. There will always be an error log generated when safe mode is entered due to a hard failure. In addition to a posted error log safe mode will also be indicated in the ASMI Power and Performance Mode Setup menu:



To re-enable full EnergyScale functionality, a reboot, firmware update, FRU replacement, or complete A/C power cycle of the system must be completed. If the problem which originally caused the system to enter safe mode is still present, the system will re-enter safe mode and generate an additional error log entry.

## ***PowerVM Specific Features***

The following features are specific to PowerVM systems.

### **Static Power Saver Mode**

Static Power Saver lowers the processor frequency and voltage a fixed amount, reducing the power consumption of the system while still delivering predictable performance. This percentage is predetermined to be within a safe operating limit and is not user-configurable. In addition, some EnergyScale-ready operating systems automatically enable processor folding in dedicated processor partitions when Static Power Saver mode is enabled. See [Appendix I](#) for details on the actual static power saver frequency by system and firmware release.

ASMI is the recommended user interface to enable/disable Power Saver mode. Power Saver could be enabled based on regular variations in workloads, such as predictable dips in utilization overnight, or over weekends. Power Saver can be used to reduce peak energy consumption, which can lower the cost of all power used. Please note that when Power Saver is enabled for certain workloads with low CPU utilization, workload performance will not be impacted, though CPU utilization may increase due to the reduced processor frequency.

The only time that a system does not support operating at the Power Saver voltage and frequency is during a system boot or re-boot. Power Saver may be enabled at any time; however, if Power Saver was enabled prior to a system boot the voltage and frequency will remain at the default boot values until the platform firmware reaches a standby or running state. Immediately before the platform firmware starts executing on the system's processors, the voltage and frequency will drop to the power saver mode values. If a system re-boot occurs while in Power Saver mode, the voltage and frequency will return back to boot values and following a successful system re-boot the voltage and frequency will be dropped back to power saver mode. The power saver mode setting persists across system boots, service processor resets and loss of ac power (unless the power outage is long enough to drain the Service Processor NVRAM battery).

## Dynamic Performance Mode (DPM)

DPM varies processor frequency and voltage based on the utilization of the system's POWER9 processors. When in DPM, the system will generally run above the nominal frequency<sup>1</sup> and may even get to the maximum frequency if the workloads are light enough, or many cores are not being used. The determining factor for what frequency the CPU runs at in Dynamic Performance mode, is power. The system limits the socket power draw to a base wattage (this varies by chip and system) which results in lower fan speeds due to the lower heat production from the socket. When in Dynamic Performance mode, the frequency will vary above nominal depending on available power headroom, i.e. the power draw at the socket. With a heavy workload and all the cores being used, the system will run at least at the nominal frequency. If some cores are not being used, the system can run at much higher frequencies before the power limit is reached. If there are enough cores not being used, the maximum frequency may be reached.

DPM mode provides deterministic behavior at all ambient conditions. This means that the same workload run in the same configuration on an identically configure system, will produce the same performance.

The frequency is managed at the socket level, so different sockets may run at different frequencies. However, the mode setting is system wide.

Dynamic Performance mode will lower the processor frequency if the entire processor socket is idle for 100s of milliseconds, thus, providing both a performance boost and a power savings when possible. Dynamic Performance mode is great for customers that want deterministic performance across the full range of environmental conditions or those that have acoustic or power consumption concerns.

## Maximum Performance Mode (MPM)

Maximum Performance mode takes advantage of lower active core counts and normal utilization workloads (just like Dynamic Performance mode). However, Maximum Performance mode will allow the system to reach the maximum frequency under more conditions by increasing the allowable maximum socket power and taking advantage of thermal headroom. This results in higher performance, but also higher power consumption and higher fan speeds. In MPM mode, the socket will be at maximum voltage and frequency even if all cores are completely idle. The exception to this is if Idle Power Saver Mode is enabled.

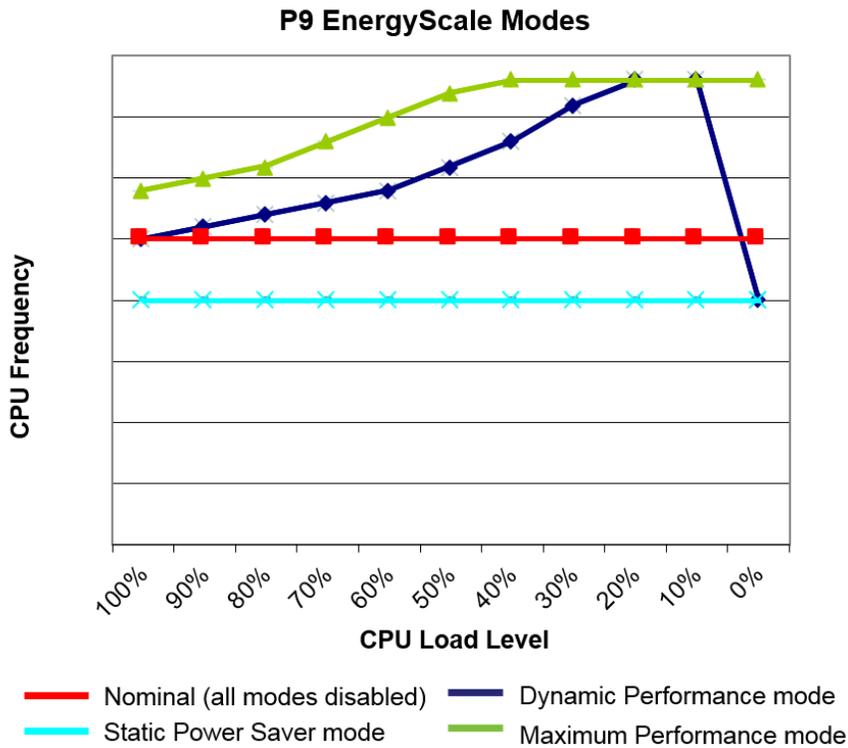
In order to provide adequate cooling, Maximum Performance mode will increase fan speeds, which can increase the associated acoustics by up to 15 decibels and increase the power that fans are using. If the datacenter ambient environment is less than a guaranteed ambient, the frequency in Maximum Performance mode will consistently be in the upper range of the maximum frequency (roughly 10% to 20% better than nominal). At this lower ambient temperature, MPM mode also provides deterministic performance. As the ambient temperature increases above the guaranteed ambient, determinism can no longer be guaranteed. See [Appendix I: System Requirements](#) for the guaranteed ambient temperature by release and system.

Note that the increased noise and wattage varies by system model, configuration, core count, and other factors.

---

<sup>1</sup> Nominal is the system base frequency when all power management modes are disabled

NOTE: In all modes if Idle Power Saver is enabled the frequency will drop to the system minimum when the Idle Power Saver entry time and utilization threshold are met. See [Idle Power Saver](#) section for more details.



## New Performance Modes – Summary of Differences

	Dynamic Performance	Maximum Performance	Comment
<b>Performance</b>	Higher performance than Static Nominal Mode	Highest performance (when operating in nominal environment)	
<b>Determinism</b>	Full operating ambient temp range	Below guaranteed ambient defined by MTM	
<b>Acoustics</b>	base	Typically 0.5 to 1.5 Bel higher	Many factors will affect acoustics difference – Specific MTM, Specific CPU feature, CPU PN and workload
<b>Energy consumption</b>	base	Typically 100W higher	Many factors will affect energy difference – Specific MTM, Specific CPU feature, CPU PN and workload

## Tunable Parameters

The tunable parameters can be used to modify the system behavior while Dynamic Performance Mode is enabled. This may be useful to properly balance the performance required with the energy savings desired. These parameters should not be changed unless the user is working directly with an IBM representative or has the proper level of expertise in the effects of these parameter changes. While the parameters will be shown in this paper, no attempt will be made to explain in detail the system energy saving or performance that will result from the parameter modification.

### Idle Power Saver

This mode, which is enabled/disabled independently from all other modes and functions, reduces the energy usage to a low level *when the entire system is determined to be idle*. When idle, the voltage/frequency of the processor is reduced to the minimum and static power saver mode is reported to the OS to enable processor folding. This can cause OS folding policy values to be updated at runtime. For example, in AIX, `vpm_fold_policy` may be updated to show 0x3 Dynamic. When not idle the system is managed in accordance to the configured power mode (i.e. Dynamic Performance, Maximum Performance). Idle Power Saver can be enabled/disabled via ASMI. Additionally, the utilization levels that determine idleness and the time delays for entry/exit can also be modified from ASMI. See the [ASMI section](#) for more details. See [Appendix I](#) for system support details such as which systems support this mode and which systems have it enabled by default.

### EnergyScale for I/O

IBM Power Systems automatically power off pluggable PCI adapter slots that are not being used to save approximately 14 watts per slot. A PCI adapter slot is considered not being used when the slot is empty, when the slot is not assigned to a partition, or when the partition to which the slot is assigned is not powered on. A PCI slot is powered off immediately by system firmware when it is dynamically removed from the partition to which it was assigned, and when the partition to which it is assigned is powered off. Furthermore, system firmware automatically scans all pluggable PCI slots at regular intervals looking for those that meet the criteria for being not in use and powers them off. This ensures among other things that slots left on after platform power-on are subsequently powered off if they are not in use. This is supported on all POWER9 processor-based systems, and the expansion units that they support. Note that it applies to hot-pluggable PCI slots only. Power controls for other types of I/O features and built-in, or embedded, PCI adapters are not available and so they cannot be powered off independently from their enclosure power.

## Frequency and Performance Measurement

Methods to measure the frequency and performance on PowerVM systems vary by operating system.

### AIX

lparstat -E and mpstat -E are used to view the current processor frequency

- lparstat -E reports the frequency averaged across all of the Virtual Processors assigned to the LPAR
- mpstat -E reports the frequency per Virtual Processor

Usage: lparstat/mpstat -E [ *Interval* [ *Count* ] ]

```
# lparstat -E 1 1
System configuration: type=Dedicated mode=Capped smt=4 lcpu=4 mem=4096MB Power=Disabled
Physical Processor Utilization:
-----Actual-----          -----Normalised-----
user  sys  wait  idle  freq          user  sys  wait  idle
-----  -----
0.001 0.002 0.000 0.997  3.0GHz [100%] 0.001 0.002 0.000 0.997

# mpstat -E 1 1
System configuration: lcpu=4 mode=Capped
vcpu    pbusy          physc          freq          scaled physc
-----  -----
0       0.0022 [0%]    1.0005 [100%] 3.0GHz [100%] 1.0048 [100%]
```

The *Interval* parameter must be supplied to lparstat/mpstat -E to read the current frequency.

Without the interval parameter, the commands generate a single report containing statistics since the last boot of the LPAR by default.

**NOTE: pmcycles command in AIX should NOT be used for reading the current processor frequency.** The recommended approach is to use lparstat -E and mpstat -E as discussed above.

References:

[https://www.ibm.com/support/knowledgecenter/en/ssw\\_aix\\_72/com.ibm.aix.cmds3/lparstat.htm](https://www.ibm.com/support/knowledgecenter/en/ssw_aix_72/com.ibm.aix.cmds3/lparstat.htm)  
[https://www.ibm.com/support/knowledgecenter/en/ssw\\_aix\\_72/com.ibm.aix.cmds3/mpstat.htm](https://www.ibm.com/support/knowledgecenter/en/ssw_aix_72/com.ibm.aix.cmds3/mpstat.htm)

## IBM i

### IBM iDoctor for IBM i

IBM iDoctor for IBM i displays the CPU rate for the IBM i partition over time on the Collection Overview graph. The CPU rate for the partition is the ratio of scaled to unscaled processor utilized time, expressed as a percentage. The processor utilized time is the accumulation of non-idle virtual processor SPURR and PURR<sup>2</sup> over each time interval. The ratio of SPURR and PURR accumulated over an interval represents the processor frequency versus nominal over the interval.

### WRKSYSACT

The Work with System Activity (WRKSYSACT) command displays the Average CPU rate since last refresh for the partition in output shown on the display station. The Average CPU rate for the partition is the ratio of scaled to unscaled processor utilized time, expressed as a percentage. The processor utilized time is accumulation of non-idle virtual processor SPURR and PURR for the interval since the last refresh.

---

<sup>2</sup> See [Appendix II](#) for more info.

### **IBM i Collection Services**

Database file QAPMJOBMI contains time series data by task, primary thread, and secondary thread. Scaled and unscaled CPU times, both charged and used, are available to calculate average CPU rate for processing activity of tasks and threads.

Database file QAPMSYSTEM contains time series system-wide (i.e. partition) accumulations of performance data. Scaled and unscaled CPU times are accumulated for various categories of processor usage. The ratio of scaled to unscaled time is the average CPU rate for the category of time accumulation. The processor utilized time is accumulation of non-idle virtual processor SPURR and PURR for the time interval.

Note: As of IBM i 7.3, the QAPMCONF database file key "NF" contains the processor nominal frequency in MHz. The processor nominal frequency can be used to convert average CPU rate to average processor frequency.

# ***Non PowerVM Specific Features***

## **Power, Thermal, and Performance Measurement**

Non Power VM systems follow standard linux power management features including the power and performance management governors.

## **Power Management**

There are new power, thermal, and performance sensors that can be read via the standard `lm_sensor` command.

## **Frequency and Performance Measurement**

#Nominal frequency range

```
# cat /sys/devices/system/cpu/cpu0/cpufreq/scaling_available_frequencies
```

```
3283000 ...
```

#Full Frequency range

```
# cat /sys/devices/system/cpu/cpu0/cpufreq/scaling_boost_frequencies
```

```
3800000 ..
```

#Current running frequency of any core:

```
# cat /sys/devices/system/cpu/cpu0/cpufreq/cpuinfo_cur_freq
```

```
2316000
```

#Test max frequency achieved in the system:

```
# ppc64_cpu --frequency
```

```
min:  3.776 GHz (cpu 143)
```

```
max:  3.777 GHz (cpu 73)
```

```
avg:  3.777 GHz
```

#Use `cpupower` tool to query and set frequency

Available frequency steps from `cpupower` will list only the nominal range, but user can select full frequency range to set and it will take effect.

## User Interfaces

### Overview

The table below summarizes the ASMI, CIM interface, HMC, DCMI, REST, Redfish and BMC GUI support. Refer first to [Appendix I: System Requirements](#) to know if the specific interface is supported for a particular release and system.

	<i>ASMI</i>	<i>CIM Client</i>	<i>HMC</i>	<i>DCMI</i>	<i>REST</i>	<i>Redfish</i>	<i>BMC GUI</i>
<b>Power Trending</b>	n/a	Y	n/a	Y	Y	n/a	Y
<b>Thermal Reporting</b>	n/a	Y	n/a	Y	Y	n/a	Y
<b>Maximum Performance Mode</b>	Y	n/a	Y	n/a	n/a	Y	n/a
<b>Static Power Saver</b>	Y	n/a	Y	n/a	n/a	Y	n/a
<b>Dynamic Performance Mode</b>	Y	n/a	Y	n/a	n/a	Y	n/a
<b>Tunable Parameters</b>	Y	n/a	n/a	n/a	n/a	Y	n/a
<b>Idle Power Saver</b>	Y	n/a	n/a	n/a	n/a	Y	n/a
<b>Power Capping</b>	n/a	n/a	n/a	Y	Y	n/a	n/a

# ASMI

Supported EnergyScale features can be found on the Advanced System Management Interface (ASMI) under the “Power Management” menu.

The screenshot displays the ASMI web interface. At the top, there is a header with the IBM logo on the left, the text "Advanced System Management" in the center, and "Copyright" on the right. Below the header is a black navigation bar containing a "Log out" button. The main content area is divided into two columns. The left column is a dark sidebar menu with a list of expandable items, each preceded by a small square icon with a plus sign. The "Power Management" item is highlighted with a red oval. The right column is a light yellow panel titled "System Name" with a horizontal line. Below the title is a text input field labeled "System Name:" followed by a question mark icon. A "Save settings" button is positioned below the input field.

IBM Advanced System Management Copyright

Log out

- Expand all menus
- Collapse all menus
- Power/Restart Control
- System Service Aids
- System Information
- System Configuration
  - System Name
  - Configure I/O Enclosures
  - Time Of Day
  - Firmware Update Policy
  - PCI Error Injection Policy
  - Monitoring
  - HSL Opticonnect Connections
  - I/O Adapter Enlarged Capacity
  - Hardware Management Consoles
  - Virtual I/O Connections
  - Firmware License Agreement
  - Floating Point Unit Computation Test
  - Virtual Trusted Platform Module
  - Hypervisor Dispatch Wheel Time
  - PCIe Hardware Topology
  - Hardware Page Table Size
  - Console Type
  - Predictive Dynamic Memory Deallocation
  - High Frequency Trading
  - Speculative Execution Control
  - Hardware Deconfiguration
  - Program Vital Product Data
  - Service Indicators
  - Power Management**
    - Power and Performance Mode Setup
    - Idle Power Saver
    - Tuning Parameters
  - Security
- Network Services

System Name

System Name:  ?

Save settings

# Setting System Power and Performance Mode



Log out

- Expand all menus
- Collapse all menus

- Power/Restart Control
- System Service Aids
- System Information
- System Configuration
  - System Name
  - Configure I/O Enclosures
  - Time Of Day
  - Firmware Update Policy
  - PCI Error Injection Policy
  - Monitoring
  - HSL Opticonnect Connections
  - I/O Adapter Enlarged Capacity
  - Hardware Management Consoles
  - Virtual I/O Connections
  - Firmware License Agreement
  - Floating Point Unit Computation Test
  - Virtual Trusted Platform Module
  - Hypervisor Dispatch Wheel Time
  - PCIe Hardware Topology
  - Hardware Page Table Size
  - Console Type
  - Predictive Dynamic Memory Deallocation
  - High Frequency Trading
  - Speculative Execution Control
- Hardware Deconfiguration
- Program Vital Product Data
- Service Indicators
- Power Management
  - Power and Performance Mode Setup**
  - Idle Power Saver
  - Tuning Parameters

### Power and Performance Mode Setup

Current Power Saver Mode : Enable Maximum Performance mode

- Disable all modes ?
- Enable Static Power Saver mode ?
- Enable Dynamic Performance mode ?
- Enable Maximum Performance mode ?

Note: Enabling any of the Power Saver modes will cause changes in the processor frequencies, changes in processor utilization, changes in power consumption, and performance to vary. Other effects are possible as well. Please see the EnergyScale™ white paper for more information on power saving modes.

Continue ?

## Tunable Parameters

Note: These parameters are for advanced users only. Do not adjust them unless directed to do so by IBM. The “Reset Parameters” button on the bottom will restore all parameters to their default value.

IBM Advanced System Management Copyright © 2002, 2018 IBM Corpora All rights reserved

Log out

- Expand all menus
- Collapse all menus
- Power/Restart Control
- System Service Aids
- System Information
- System Configuration
  - System Name
  - Configure I/O Enclosures
  - Time Of Day
  - Firmware Update Policy
  - PCI Error Injection Policy
  - Monitoring
  - HSL Opticonnect Connections
  - I/O Adapter Enlarged Capacity
  - Hardware Management Consoles
  - Virtual I/O Connections
  - Firmware License Agreement
  - Floating Point Unit Computation Test
  - Virtual Trusted Platform Module
  - Hypervisor Dispatch Wheel Time
  - PCIe Hardware Topology
  - Hardware Page Table Size
  - Console Type
  - Predictive Dynamic Memory
- Deallocation
  - High Frequency Trading
  - Speculative Execution Control
- Hardware Deconfiguration
- Program Vital Product Data
- Service Indicators
- Power Management
  - Power and Performance Mode Setup
  - Idle Power Saver
  - Tuning Parameters**
- Security
- Network Services
- Performance Setup
- On Demand Utilities
- Login Profile

Number of samples for computing utilization statistics  
Current value: 16  
New value:  Range: MinVal-1 MaxVal-1024

Step size for going up in frequency  
Current value: 0.8%  
New value:  Range: MinVal- 0.1% MaxVal- 100.0%

Step size for going down in frequency  
Current value: 0.8%  
New value:  Range: MinVal- 0.1% MaxVal- 100.0%

Delta percentage for determining active cores  
Current value: 18%  
New value:  Range: MinVal-0% MaxVal-100%

Utilization threshold to determine active cores with slack  
Current value: 98.0%  
New value:  Range: MinVal- 0.0% MaxVal- 100.0%

Enable/disable frequency delta between cores  
Current value: Disabled  
New value:

Maximum frequency delta between cores  
Current value: 10%  
New value:  Range: MinVal-10% MaxVal-100%

Enable/disable workload optimized frequency  
Current value: Enabled  
New value:

Note: It is recommended that only advanced users modify these values since it can result in unexpected behavior and performance impacts. Please see the EnergyScale™ white paper for more information on tuning parameters.

Save settings ?

Reset Parameters ?

# Idle Power Saver



Log out

- Expand all menus
- Collapse all menus

- Power/Restart Control
- System Service Aids
- System Information
- System Configuration
  - System Name
  - Configure I/O Enclosures
  - Time Of Day
  - Firmware Update Policy
  - PCI Error Injection Policy
  - Monitoring
  - HSL Opticonnect Connections
  - I/O Adapter Enlarged Capacity
  - Hardware Management Consoles
  - Virtual I/O Connections
  - Firmware License Agreement
  - Floating Point Unit Computation Test
  - Virtual Trusted Platform Module
  - Hypervisor Dispatch Wheel Time
  - PCIe Hardware Topology
  - Hardware Page Table Size
  - Console Type
  - Predictive Dynamic Memory
- Deallocation
  - High Frequency Trading
  - Speculative Execution Control
- Hardware Deconfiguration
- Program Vital Product Data
- Service Indicators
- Power Management
  - Power and Performance Mode Setup
  - Idle Power Saver**
  - Tuning Parameters

### Idle Power Saver

Idle Power Saver Enable

Current value: Enabled

New value: Enabled ▾

Delay Time to Enter Idle Power

Current value: 240Seconds

New value: 240 Range: MinVal-10Seconds MaxVal-600Seconds

Utilization Threshold to Enter Idle Power

Current value: 8%

New value: 8 Range: MinVal-1% MaxVal-95%

Delay Time to Exit Idle Power

Current value: 10Seconds

New value: 10 Range: MinVal-10Seconds MaxVal-600Seconds

Utilization Threshold to Exit Idle Power

Current value: 12%

New value: 12 Range: MinVal-5% MaxVal-95%

Note: Selecting a utilization threshold to enter idle power that is higher than the utilization threshold to exit idle power will result in unexpected behavior. Please see the EnergyScale™ white paper for more information on Idle Power Saver.

Save settings ?

Reset Parameters ?

## ***CIM Client***

This section will provide a few example scripts for some features, this is not meant to show everything that can be done via CIM. For more information search for “EnergyScale” or “EnergyStar” on [www.ibm.com](http://www.ibm.com) to find the “Manual for Using WBEM CLI to Fetch Flexible Service Processor CIM Data”.

## Script to Read Average Power

```
#!/usr/bin/perl
my $system="";
my $uid="HMC";
my $pwd="xxxxxx";
my $verbose=0;

if ($#ARGV >= 0)
{
    $system="$ARGV[$i]";
}
else
{
    print "ERROR: Must supply system name or IP address\n";
    exit 1;
}
if ($verbose)
{
    print "==> wbemcli -nl -noverify ei
\"https://${uid}:${pwd}\@${system}:5989/root/ibmsd:fips_powermetricvalue\" | grep -e
\"-InstanceID\" -e \"-MeasuredElementName\" -e \"-TimeStamp\" -e \"-MetricValue\"";
}
@result=`wbemcli -nl -noverify ei
"https://${uid}:${pwd}\@${system}:5989/root/ibmsd:fips_powermetricvalue" | grep -e "-
InstanceID" -e "-MeasuredElementName" -e "-TimeStamp" -e "-MetricValue"`;
if ( $#result < 0 )
{
    print "ERROR: Failed to find Power Trending Data\n";
}
else
{
    print "Average Power Trending Data for $system: (Watts)\n";
    print "   Timestamp                Watts   Instance Identifier\n";
    print "   -----                -"
\n";
    foreach my $line (@result)
    {
        if ($line =~ /-MetricValue=(\d*)/)
        {
            $value=$1;
        }
        elsif ($line =~ /-TimeStamp=(\S*)/)
        {
            $timestamp=$1;
        }
        elsif ($line =~ /-MeasuredElementName=([\^"]*)/)
        {
            $element=$1;
        }
        elsif ($line =~ /-InstanceID=([\^"]*)/)
        {
            $instance=$1;
            if ($instance =~ /Avg/)
            {
                printf("   $timestamp   %5d   $element\n", $value);
            }
        }
    }
}
exit 0;
```

## Average Power Output

Timestamp	Watts	Instance Identifier
20140606140700.000000+000	343	CECDrawer 00001E00
20140606140730.000000+000	343	CECDrawer 00001E00
20140606140800.000000+000	343	CECDrawer 00001E00
20140606140830.000000+000	343	CECDrawer 00001E00
20140606140900.000000+000	343	CECDrawer 00001E00
:	:	:
:	:	:
20140606150400.000000+000	342	CECDrawer 00001E00
20140606150430.000000+000	342	CECDrawer 00001E00
20140606150500.000000+000	342	CECDrawer 00001E00
20140606150530.000000+000	342	CECDrawer 00001E00
20140606150600.000000+000	342	CECDrawer 00001E00
20140606140700.000000+000	226	Power Supply U78C9.001.D123456-E1 00001000
20140606140730.000000+000	226	Power Supply U78C9.001.D123456-E1 00001000
20140606140800.000000+000	226	Power Supply U78C9.001.D123456-E1 00001000
20140606140830.000000+000	226	Power Supply U78C9.001.D123456-E1 00001000
20140606140900.000000+000	226	Power Supply U78C9.001.D123456-E1 00001000
:	:	:
:	:	:
20140606150400.000000+000	226	Power Supply U78C9.001.D123456-E1 00001000
20140606150430.000000+000	226	Power Supply U78C9.001.D123456-E1 00001000
20140606150500.000000+000	226	Power Supply U78C9.001.D123456-E1 00001000
20140606150530.000000+000	226	Power Supply U78C9.001.D123456-E1 00001000
20140606150600.000000+000	226	Power Supply U78C9.001.D123456-E1 00001000
20140606140700.000000+000	194	Power Supply U78C9.001.D123456-E2 00001001
20140606140730.000000+000	194	Power Supply U78C9.001.D123456-E2 00001001
20140606140800.000000+000	194	Power Supply U78C9.001.D123456-E2 00001001
20140606140830.000000+000	194	Power Supply U78C9.001.D123456-E2 00001001
20140606140900.000000+000	194	Power Supply U78C9.001.D123456-E2 00001001
:	:	:
:	:	:
20140606150400.000000+000	194	Power Supply U78C9.001.D123456-E2 00001001
20140606150430.000000+000	194	Power Supply U78C9.001.D123456-E2 00001001
20140606150500.000000+000	194	Power Supply U78C9.001.D123456-E2 00001001
20140606150530.000000+000	194	Power Supply U78C9.001.D123456-E2 00001001
20140606150600.000000+000	194	Power Supply U78C9.001.D123456-E2 00001001

## Script to Read Average Frequency

```
#!/usr/bin/perl
my $system="";
my $uid="HMC";
my $pwd="xxxxxx";
my $verbose=0;

if ($#ARGV >= 0)
{
    $system="$ARGV[$i]";
}
else
{
    print "ERROR: Must supply system name or IP address\n";
    exit 1;
}

if ($verbose)
{
    print "==> wbemcli -nl -noverify ei
\"https://${uid}:${pwd}@${system}:5989/root/ibmsd:fips_cpuusagemetricvalue\" | grep
-e \"-InstanceID\" -e \"-MeasuredElementName\" -e \"-TimeStamp\" -e \"-
MetricValue\"\\n\";
}
@result=`wbemcli -nl -noverify ei
\"https://${uid}:${pwd}@${system}:5989/root/ibmsd:fips_cpuusagemetricvalue\" | grep -e
\"-InstanceID\" -e \"-MeasuredElementName\" -e \"-TimeStamp\" -e \"-MetricValue\"`;
if ( $#result < 0 )
{
    print "ERROR: Failed to find Processor Frequency Trending Data\n";
}
else
{
    print "Processor Frequency Trending Data for $system: (MHz)\n";
    print "   Timestamp                MHz      Instance Identifier\n";
    print "   -----                ----      -----
\n";
    foreach my $line (@result)
    {
        if ($line =~ /-MeasuredElementName=.(["]*)./)
        {
            $instance=$1;
        }
        elsif ($line =~ /-TimeStamp=(\S*)/)
        {
            $timestamp=$1;
        }
        elsif ($line =~ /-MetricValue=(\d*)./)
        {
            $value=$1;
            printf("   $timestamp   %4d   $instance\n", $value);
        }
    }
}

exit 0;
```

## **Average Frequency Output**

Timestamp	MHz	Instance Identifier
20140606140700.000000+000	2060	CECDrawer 00001E00
20140606140730.000000+000	2060	CECDrawer 00001E00
20140606140800.000000+000	2060	CECDrawer 00001E00
20140606140830.000000+000	2060	CECDrawer 00001E00
:	:	:
:	:	:
20140606145530.000000+000	2060	CECDrawer 00001E00
20140606145600.000000+000	2060	CECDrawer 00001E00

## Script to Read Thermal Data

```
#!/usr/bin/perl
my $system="";
my $uid="HMC";
my $pwd="xxxxxx";
my $verbose=0;

if ($#ARGV >= 0)
{
    $system="$ARGV[$i]";
}
else
{
    print "ERROR: Must supply system name or IP address\n";
    exit 1;
}

if ($verbose)
{
    print "==> wbemcli -nl -noverify ei
\"https://${uid}:${pwd}@${system}:5989/root/ibmsd:fips_thermalmetricvalue\"\n";
}
@result=`wbemcli -nl -noverify ei
"https://${uid}:${pwd}@${system}:5989/root/ibmsd:fips_thermalmetricvalue`;
if ( $#result < 0 )
{
    print "ERROR: Failed to find Thermal Trending Data\n";
}
else
{
    print "Thermal Trending Data for $system: (1/100th degree C)\n";
    print "   Timestamp                Temp   Instance Identifier\n";
    print "   -----                ----   -----
\n";
#   print "   20140606094200.000000+000   3300   IBM:ExhaustAirTemp_U78C9.001.D123456
00001E00_10711";

    foreach my $line (@result)
    {
        if ($line =~ /-InstanceID=\.([\^"]*)\.\/)
        {
            $instance=$1;
        }
        elsif ($line =~ /-TimeStamp=(\S*)\/)
        {
            $timestamp=$1;
        }
        elsif ($line =~ /-MetricValue=\.(\d*)\.\/)
        {
            $value=$1;
            printf("   $timestamp   %4d   $instance\n", $value);
        }
    }
}

exit 0;
```

## Thermal Data Output

Thermal Trending Data for tul73fp: (1/100th degree C)

Timestamp	Temp	Instance Identifier
20140606135600.000000+000	3400	IBM:ExhaustAirTemp_U78C9.001.D123456 00001E00_11726
20140606135630.000000+000	3400	IBM:ExhaustAirTemp_U78C9.001.D123456 00001E00_11728
20140606135700.000000+000	3400	IBM:ExhaustAirTemp_U78C9.001.D123456 00001E00_11731
20140606135730.000000+000	3400	IBM:ExhaustAirTemp_U78C9.001.D123456 00001E00_11733
20140606135800.000000+000	3400	IBM:ExhaustAirTemp_U78C9.001.D123456 00001E00_11735
:	:	:
:	:	:
20140606145200.000000+000	3300	IBM:ExhaustAirTemp_U78C9.001.D123456 00001E00_11950
20140606145230.000000+000	3300	IBM:ExhaustAirTemp_U78C9.001.D123456 00001E00_11952
20140606145300.000000+000	3300	IBM:ExhaustAirTemp_U78C9.001.D123456 00001E00_11955
20140606145330.000000+000	3300	IBM:ExhaustAirTemp_U78C9.001.D123456 00001E00_11957
20140606145400.000000+000	3300	IBM:ExhaustAirTemp_U78C9.001.D123456 00001E00_11959
20140606135600.000000+000	2400	IBM:InletAirTemp_U78C9.001.D123456-D1 00006000_11727
20140606135630.000000+000	2400	IBM:InletAirTemp_U78C9.001.D123456-D1 00006000_11729
20140606135700.000000+000	2400	IBM:InletAirTemp_U78C9.001.D123456-D1 00006000_11730
20140606135730.000000+000	2400	IBM:InletAirTemp_U78C9.001.D123456-D1 00006000_11732
20140606135800.000000+000	2400	IBM:InletAirTemp_U78C9.001.D123456-D1 00006000_11734
:	:	:
:	:	:
20140606145200.000000+000	2300	IBM:InletAirTemp_U78C9.001.D123456-D1 00006000_11951
20140606145230.000000+000	2300	IBM:InletAirTemp_U78C9.001.D123456-D1 00006000_11953
20140606145300.000000+000	2300	IBM:InletAirTemp_U78C9.001.D123456-D1 00006000_11954
20140606145330.000000+000	2300	IBM:InletAirTemp_U78C9.001.D123456-D1 00006000_11956
20140606145400.000000+000	2300	IBM:InletAirTemp_U78C9.001.D123456-D1 00006000_11958

# HMC

The HMC (Hardware Management Console) is a management console that controls managed systems, logical partitions, managed frames, other features provided through the managed objects, and the HMC itself. The HMC provides both graphical user interface (GUI) and command line interfaces. Users can use the user interfaces to configure or manage various features offered by the managed objects.

## Setting System Power Management Mode

The user can list which power management modes are supported using the `lspwrmgmt` command on the command line as illustrated below. **NOTE in some HMC releases Maximum Performance Mode is referred to as `fixed_max_frequency` and Dynamic Performance Mode is referred to as `dynamic_favor_perf`:**

```
lspwrmgmt -m <managed system name> -r sys -F supported_power_saver_mode_types
```

Example output:

```
"static,dynamic_favor_perf,fixed_max_frequency"
```

A user can then enable one of the supported power management modes using the `chpwrmgmt` command on the command line as illustrated below enabling Fixed Maximum Frequency mode:

```
chpwrmgmt -m <managed system name> -r sys -o enable -t fixed_max_frequency
```

To query the current power management mode use `lspwrmgmt` command on the command line as illustrated below:

```
lspwrmgmt -m <managed system name> -r sys
```

Example output:

```
curr_power_saver_mode=Enabled,curr_power_saver_mode_type=fixed_max_frequency,desired_power_saver_mode=Enabled,desired_power_saver_mode_type=fixed_max_frequency,"supported_power_saver_mode_types=static,dynamic_favor_perf,fixed_max_frequency",idle_power_saver_mode=null
```

NOTE: The new mode may not take effect immediately. Normally, if the operation is performed before the system is powered on, the desired mode won't take effect until the system is up and running. If the mode is in transition, any changes will be blocked.

To disable power management mode:

```
chpwrmgmt -m <managed system name> -r sys -o disable
```

With the HMC GUI, users can reach this task by selecting the managed system -> Operations -> Power Management.

## ***DCMI***

On some systems and FW level the Data Center Manageability Interface (DCMI) may be used to access additional data such as processor temperatures, baseboard temperatures, power and the ability to set a power limit. Refer to the DCMI specification for full command list. See [Appendix I: System Requirements](#) for system and required FW that supports DCMI.

### NOTES:

- Power readings and power limits are input (AC) power.
- Set Power Limit exception actions cannot be changed on POWER9 systems. The Set Power Limit command will be rejected with 0x8A OEM completion code if the exception action is anything besides 0x11 (log event only)
- Set Power Limit correction time limit cannot be changed from 1000ms. The Set Power Limit command will be rejected with 0x85 completion code for correction time out of range if any other value is sent
- Set Power Limit sampling period cannot be changed from 1s. The Set Power Limit command will be rejected with 0x89 completion code for sampling period out of range if any other value is sent
- If a Set Power Limit command is sent while a power limit is already active the new power limit will take effect immediately without another activate power limit command being sent.

## Redfish

On some systems and FW level Redfish may be used to access the same power management interfaces available on ASM. See [Appendix I: System Requirements](#) for system and required FW that supports Redfish.

Example Redfish output:

```
{
"@odata.context": "/redfish/v1/$metadata#IBMEnterpriseEnergyScale",
"@odata.id": "/redfish/v1/Systems/Server-8375-42A-13C707W/EnergyScale",
"@odata.type": "#IBMEnterpriseEnergyScale.v1_0_0.IBMEnterpriseEnergyScale",
"CPUIIdlePowerSaver": {
"DelayTimeEnter": 15,
"DelayTimeExit": 15,
"State": "Disabled",
"UtilizationThresholdEnter": 1,
"UtilizationThresholdExit": 50
},
"DeltaThresholdPercentage": 0,
"DownFrequencyStepPercentage": 0.80000000000000004,
"DownThresholdPercentage": 99.900000000000006,
"FrequencySamplingSize": 1,
"Manufacturer": "IBM",
"MaxFrequencyDeltaPercentage": 10,
"MulticoreFrequencyDeltaEnabled": false,
"Name": "System EnergyScale",
"PowerSaveMode": "MaxPerformance",
"SlackThresholdPercentage": 10.0,
"UpFrequencyStepPercentage": 100.0,
"UpThresholdPercentage": 99.900000000000006,
"WorkloadOptimizedFrequency": "Enabled"
}
```

## ***BMC GUI***

Power and thermal information is available from the BMC GUI, see [Appendix I: System Requirements](#) for systems that supports the BMC GUI. For more information on OpenBMC see [https://www.ibm.com/support/knowledgecenter/POWER9/p9eih/p9eih\\_managing\\_with\\_openbmc.htm](https://www.ibm.com/support/knowledgecenter/POWER9/p9eih/p9eih_managing_with_openbmc.htm)

## EnergyScale Operating System Support

Operating system support for EnergyScale has been included since the introduction of the POWER6 processor. Certain capabilities, however, require more recent editions of each operating system. Please refer to the table below for a summary of available features by operating system level. As a best practice, the Fix Level Recommendation Tool ( <https://www-304.ibm.com/webapp/set2/flrt/> ) and/or System Software Maps ( <http://www-01.ibm.com/support/docview.wss?uid=ssm1maps> ) should be used to determine the latest recommended code level when planning system installs and upgrades.

<b>Feature</b>	<b>AIX</b>	<b>IBM i</b>	<b>Linux</b>
<b>Scaled Processor Time API(s)</b> Provides programmatic access to POWER9 SPURR <sup>3</sup> registers	5.3 TL9 6.1 TL2 7.1	7.2	Ubuntu 16.04 RHEL 7.5
<b>Performance Tool Support for Scaled Processor Time</b> PURR / SPURR support in OS Performance Tools	5.3 TL11 6.1 TL4 7.1	7.2	n/a
<b>Normalized Process and Job Accounting</b> Job accounting is SPURR-based	5.3 TL 9 6.1 TL2 7.1	7.2	n/a
<b>Processor Folding</b> Processor folding using processor idle states	5.3 TL 9 6.1 TL2 7.1	7.2	Ubuntu 16.04 RHEL7.5
<b>Processor Folding Control / Status</b> Processor folding may be enabled, disabled, or tuned from the OS	5.3 TL9 6.1 TL2 7.1	7.2	Ubuntu 16.04 RHEL7.5
<b>EnergyScale Configuration Status</b> EnergyScale Power Mgmt mode exposed through the OS	6.1 TL6 7.1	7.2	Ubuntu 16.04 RHEL7.5
<b>Query Idle Power Saver and Dynamic Power Saver Tunable Parameters</b> Query only, parameters are not settable from the OS.	7.1 TL4 SP1 7.2	n/a	n/a
<b>Im_sensors</b> Query EnergyScale sensors	n/a	n/a	Ubuntu 16.04 RHEL7.5

<sup>3</sup>Refer to Appendix II: Processor Usage and Accounting

## Processor Folding in AIX

In AIX the processor folding policy can be configured from the command line via the `schedo` command. The `vpm_fold_policy` tunable is a 4-bit value where each bit indicates the configuration of a different setting. The following table shows the various settings that are controlled.

<i>Bit</i>	<i>Setting</i>
<b>0</b>	=1 processor folding is enabled when the partition is using shared processors
<b>1</b>	=1 processor folding is enabled when the partition is using dedicated processors
<b>2</b>	=1 disables the automatic setting of processor folding when the partition is in static power saver mode
<b>3</b>	=1 processor affinity will be ignored when making folding decisions

Table 1: `vpm_fold_policy` is a 4-bit value, in which each bit controls an aspect of folding.

The following command displays the current setting of `vpm_fold_policy`:

```
# schedo -L vpm_fold_policy
NAME                                CUR  DEF  BOOT  MIN  MAX  UNIT  TYPE
DEPENDENCIES
-----
vpm_fold_policy                    1    1    1     0   15    D
-----
```

Note: If ASMI setting **Idle Power Saver** mode is set to **Enabled**, and processor utilization falls below the specified threshold values, processors can be set into low frequency/low voltage state, and the AIX `vpm_fold_policy` can be adjusted at runtime to 0x3.

To enable processor folding on a partition using dedicated partitions when the current value of `vpm_fold_policy` is set to 1, the following command would be issued to set the value to 3:

```
# schedo -o vpm_fold_policy=3
```

To disable processor folding, the value of `vpm_fold_policy` can be set to the value 4 using the following command:

```
# schedo -o vpm_fold_policy=4
```

By default, AIX will attempt to consider processor affinity (or topology) information when making processor folding decisions. This allows for the workload to remain spread across the processor nodes (e.g., chips depending on the system) and benefit from improved performance. Bit 3 of the `vpm_fold_policy` tunable allows this default behavior to be disabled. For example, if `vpm_fold_policy` is currently set to 6, indicating that processor folding is enabled when the partition is using dedicated partitions and that the partition will automatically enable processor folding when in static power saver mode, the following command would change the setting to indicate that the operating system should no longer consider processor affinity when making folding decisions:

```
# schedo -o vpm_fold_policy=14
```

For more information see the help information via the `-h` option of the `schedo` command:

```
# schedo -h vpm_fold_policy
```

## Processor Folding in IBM i

In an IBM i partition, processor folding is configured and controlled by the operating system by default. On POWER9 servers, the operating system enables processor folding by default in shared processor LPARs or when Static Power Saver mode is enabled. Operating system control of processor folding may be overridden via the **QWCCTLSW** limited availability API, which provides a key-based control language programming interface to a limited set of IBM i tunable parameters. Processor folding control is accessed via **QWCCTLSW** key 1060. The following sequence of calls cycles through the various options. Changes to key 1060 take effect immediately but do not persist across IPLs.

Get current status:

```
> CALL QWCCTLSW PARM('1060' '1')
KEY 1060 IS *SYSCTL.
KEY 1060 IS SUPPORTED ON THE CURRENT IPL.
KEY 1060 IS CURRENTLY ENABLED.
```

Explicitly disable processor folding:

```
> CALL QWCCTLSW PARM('1060' '3' )
KEY 1060 SET TO *OFF.
```

Explicitly enable processor folding:

```
> CALL QWCCTLSW PARM('1060' '2' 1)
KEY 1060 SET TO *ON.
```

Re-establish system control of processor folding:

```
> CALL QWCCTLSW PARM('1060' '2' 2)
KEY 1060 SET TO *SYSCTL.
```

## Processor Folding in Linux

It is essential to install a daemon package based on the host OS to enable utilization-based processor folding for Static Power Saver and Idle Power Saver modes:

```
pseries-energy-1.4.0-1.el7.ppc64.rpm  
pseries-energy-1.4.0-1.el6.ppc64.rpm  
pseries-energy-1.4.0-1.sles11.ppc64.rpm
```

Version 5.4 has the necessary user space tools required to enable CPU Folding.<sup>4</sup>

Once this package is installed, the `energyd` daemon will monitor the system power mode and activate processor folding when system power mode is set to "Static Power Saver" and deactivate processor folding in all other modes. The utilization-based CPU folding daemon will deactivate unused cores and transition them to low power idle states until the CPU utilization increases and those cores are activated to run a workload.

Utilization-based processor folding can be manually disabled using the following commands:

```
/etc/init.d/energyd stop #Stop daemon now, activate all cores  
chkconfig energyd off    #Do not restart daemon on startup
```

-or-

```
rpm -e pseries-energy    #un-install the package completely
```

Alternatively, CPU cores can be folded or set to low power idle state in any power mode manually using the following command line:

```
echo 0 > /sys/devices/system/cpu/cpuN/online #Where N is the  
logical CPU number
```

Please note that all active hardware threads of a core needs to be taken off-line using the above command in order to move the core to a low power idle state.

The cores can be activated again with the following command:

```
echo 1 > /sys/devices/system/cpu/cpuN/online #Where N is the  
logical CPU number
```

---

<sup>4</sup>Refer to <http://www14.software.ibm.com/webapp/set2/sas/f/lopdiags/installtools/home.html> for more details.

## Query System Power and Performance Mode in AIX

lparstat -i will output “Power Saving Mode”. Currently in POWER9 the OS cannot distinguish between Maximum and Dynamic Performance modes. A user should query for the mode from ASM or the HMC to determine if the system is in Dynamic or Maximum Performance mode. The following table shows what each interface will display for each mode.

	<b>lparstat -i “Power Saving Mode”</b>	<b>ASM “Current Power Saver Mode”</b>	<b>HMC lspwrmgmt</b>
<b>All modes disabled</b>	Disabled	Disable all modes	curr_power_saver_mode= Disabled
<b>Static Power Saver</b>	Static Power Savings	Enable Static Power Saver mode	curr_power_saver_mode= Enabled, curr_power_saver_mode_type= static
<b>Dynamic Performance</b>	Dynamic Power Savings (Favor Performance)	Enable Dynamic Performance mode	curr_power_saver_mode= Enabled, curr_power_saver_mode_type= dynamic_favor_perf or dynamic_performance
<b>Maximum Performance</b>	Dynamic Power Savings (Favor Performance)	Enable Maximum Performance mode	curr_power_saver_mode= Enabled, curr_power_saver_mode_type= fixed_max_frequency or maximum_performance

## Query Idle Power Saver and Dynamic Power Saver Tunable Parameters in AIX

Idle Power Saver and Dynamic Power Saver Tunable parameters may be queried in AIX. The parameters are not settable from the OS. Example query from AIX:

```
# lparstat -P
```

```
Dynamic Power Saver Tunables
```

```
-----  
Util threshold for increasing frequency : 98.0%  
Util threshold for decreasing frequency : 98.0%  
Number of samples for computing util stats : 16 samples  
Step size for going up in frequency : 0.8%  
Step size for going down in frequency : 0.8%  
Delta % for determining active cores : 18%  
Util threshold to determine active cores...: 98.0%  
Enable/Disable freq delta between cores : Disabled  
Maximum frequency delta between cores : 10%  
Idle Power Saver
```

```
-----  
Idle Power Saver Enable : Enabled  
Delay Time to Enter Idle Power Saver : 240 seconds  
Util Threshold to Enter Idle Power Saver : 8%  
Delay Time to Exit Idle Power Saver : 10 seconds  
AIX Power Saving Action : Folding Enabled (IPS)
```

## **Appendix I: System Requirements**

Due to differences in each release, this appendix details the systems and EnergyScale features supported by release including the actual frequency limits for various power management modes.

# Release Level FW910

## Feature Support

Refer to the EnergyScale Features chapter earlier in this document for a definition of each feature.

	Any system in OPAL Hypervisor Mode	Power S914* (9009-41A)	Power S/H922* (9009-22A)	Power S/H924* (9009-42A)	Power L922** (9008-22L)
Power Trending	CIM	CIM	CIM	CIM	CIM
Thermal Reporting	CIM	CIM	CIM	CIM	CIM
Static Power Saver	n/a	ASM, HMC, Redfish	ASM, HMC, Redfish	ASM, HMC, Redfish	ASM, HMC, Redfish
Maximum Performance Mode	n/a	ASM, HMC, Redfish	<b>Enabled by Default</b> ASM, HMC, Redfish	<b>Enabled by Default</b> ASM, HMC, Redfish	<b>Enabled by Default</b> ASM, HMC, Redfish
Dynamic Performance Mode	n/a	<b>Enabled by Default</b> ASM, HMC, Redfish	ASM, HMC, Redfish	ASM, HMC, Redfish	ASM, HMC, Redfish
Dynamic Power Saver Tunable Parameters	n/a	ASM, Redfish	ASM, Redfish	ASM, Redfish	ASM, Redfish
Idle Power Saver	n/a	<b>Enabled by Default</b> ASM, HMC, Redfish			
User Set Power Capping	n/a	n/a	n/a	n/a	n/a

### Support Notes

\*Capabilities shown are for the system default PowerVM Hypervisor Mode

\*\*Capabilities shown are for PowerVM. The system default is OPAL Hypervisor Mode. Refer to “Any system in OPAL Hypervisor Mode” column for default capabilities

## Frequency

Depending upon the Power savings setting selected, the maximum and minimum frequency may change. For a definition of each setting, please refer to the EnergyScale Features chapter earlier in this document.

### Support Notes

<sup>1</sup> Note that CPU frequencies greater than the system base are *not* guaranteed. The actual maximum frequency may vary based on environmental conditions, system configuration, firmware version, component tolerances, and workload.

<sup>2</sup> When in Maximum Performance mode for frequency to be guaranteed to be in the MPM normal operating frequency range the ambient must be below this temperature

	<i>Guaranteed Ambient Temperature<sup>2</sup></i>	<i>Dynamic Performance Mode Normal Operating Frequency Range<sup>1</sup></i>	<i>Maximum Performance Mode Normal Operating Frequency Range<sup>1</sup></i>	<i>Static Power Saver / Minimum Frequency</i>
<b>S914 @2.3GHz</b>	25C	2.3 – 3.8 GHz	2.8 - 3.8 GHz	2.3 GHz
<b>S914 @2.8GHz</b>	25C	2.8 – 3.8 GHz	3.15 - 3.8 GHz	2.3 GHz
<b>S/H922 @2.3GHz</b>	25C	2.3 – 3.8 GHz	2.8 - 3.8 GHz	2.3 GHz
<b>S/H922 @2.5GHz</b>	25C	2.5 – 3.8 GHz	2.9 - 3.8 GHz	2.3 GHz
<b>S/H922 @3.0GHz</b>	25C	3.0 – 3.9 GHz	3.4 - 3.9 GHz	2.3 GHz
<b>S/H924 @2.75GHz</b>	25C	2.75 – 3.9 GHz	3.4 - 3.9 GHz	2.3 GHz
<b>S/H924 @2.9GHz</b>	25C	2.9 – 3.9 GHz	3.5 - 3.9 GHz	2.3 GHz
<b>S/H924 @3.3GHz</b>	25C	3.3 – 4.0 GHz	3.8 - 4.0 GHz	2.3 GHz
<b>L922 @2.3GHz</b>	25C	2.3 – 3.8 GHz	2.7 - 3.8 GHz	2.3 GHz
<b>L922 @2.5GHz</b>	25C	2.5 – 3.8 GHz	2.9 - 3.8 GHz	2.3 GHz
<b>L922 @3.0GHz</b>	25C	3.0 – 3.9 GHz	3.4 - 3.9 GHz	2.3 GHz

## Maximum Frequency by Core Count

Maximum frequency in MPM and DPM based on number of cores active. Values in MHz.

	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8
<b>S914 4 BC @2.3GHz</b>																	3800
<b>S914 6 BC @2.3GHz</b>													3800	3800	3800	3800	3800
<b>S914 8 BC @2.8GHz</b>									3800	3800	3800	3800	3800	3800	3800	3800	3800
<b>S/H922 12 BC @2.3GHz</b>	3449	3525	3603	3686	3771	3800	3800	3800	3800	3800	3800	3800	3800	3800	3800	3800	3800
<b>S/H922 10 BC @2.5GHz</b>					3778	3800	3800	3800	3800	3800	3800	3800	3800	3800	3800	3800	3800
<b>S/H922 8 BC @3.0GHz</b>									3900	3900	3900	3900	3900	3900	3900	3900	3900
<b>S/H924 12 BC @2.75GHz</b>	3666	3721	3775	3838	3900	3900	3900	3900	3900	3900	3900	3900	3900	3900	3900	3900	3900
<b>S/H924 10 BC @2.9GHz</b>					3850	3894	3900	3900	3900	3900	3900	3900	3900	3900	3900	3900	3900
<b>S/H924 8 BC @3.3GHz</b>									4000	4000	4000	4000	4000	4000	4000	4000	4000
<b>L922 @2.3GHz</b>																	
<b>L922 @2.5GHz</b>																	
<b>L922 @3.0GHz</b>																	

Additional in Eric's doc:

ZZ 2S4U – S924 11 BC @ 2.817

ZZ 2S2U – S922 11 BC @ 2.4

ZZ 2S2U – S922 6 BC @3.2

# Release Level FW920

## Feature Support

Refer to the EnergyScale Features chapter earlier in this document for a definition of each feature.

	Power E950
Power Trending	CIM
Thermal Reporting	CIM
Static Power Saver	ASM, HMC, Redfish
Maximum Performance Mode	<b>Enabled by Default</b> ASM, HMC, Redfish
Dynamic Performance Mode	ASM, HMC, Redfish
Dynamic Power Saver Tunable Parameters	ASM, Redfish
Idle Power Saver	<b>Enabled by Default</b> ASM, HMC, Redfish
User Set Power Capping	DCMI

## Frequency

Depending upon the Power savings setting selected, the maximum and minimum frequency may change. For a definition of each setting, please refer to the EnergyScale Features chapter earlier in this document.

### Support Notes

<sup>1</sup> Note that CPU frequencies greater than the system base are *not* guaranteed. The actual maximum frequency may vary based on environmental conditions, system configuration, firmware version, component tolerances, and workload.

<sup>2</sup> When in Maximum Performance mode for frequency to be guaranteed to be in the MPM normal operating frequency range the ambient must be below this temperature

	<i>Guaranteed Ambient Temperature<sup>2</sup></i>	<i>Dynamic Performance Mode Normal Operating Frequency Range<sup>1</sup></i>	<i>Maximum Performance Mode Normal Operating Frequency Range<sup>1</sup></i>	<i>Static Power Saver / Minimum Frequency</i>
<b>E950 @2.8GHz</b>	25C	2.8 – 3.8 GHz	3.15 - 3.8 GHz	2.7 GHz
<b>E950 @2.85GHz</b>	25C	2.85 – 3.8 GHz	3.2 - 3.8 GHz	2.7 GHz
<b>E950 @3.0GHz</b>	25C	3.0 – 3.8 GHz	3.4 - 3.8 GHz	2.7 GHz
<b>E950 @3.3GHz</b>	25C	3.3 – 3.8 GHz	3.6 - 3.8 GHz	2.7 GHz

### Maximum Frequency by Core Count

Maximum frequency in MPM and DPM based on number of cores active. Values in MHz.

	<b>24</b>	<b>23</b>	<b>22</b>	<b>21</b>	<b>20</b>	<b>19</b>	<b>18</b>	<b>17</b>	<b>16</b>	<b>15</b>	<b>14</b>
<b>E950 12 BC @2.8GHz</b>	3410	3470	3537	3603	3679	3760	3800	3800	3800	3800	3800
<b>E950 11 BC @2.85GHz</b>			3510	3585	3663	3744	3800	3800	3800	3800	3800
<b>E950 10 BC @3.0GHz</b>					3700	3775	3800	3800	3800	3800	3800
<b>E950 8 BC @3.3GHz</b>									3800	3800	3800



# Release Level FW921

## Feature Support

Refer to the EnergyScale Features chapter earlier in this document for a definition of each feature.

	Power E950	Power E980
<b>Power Trending</b>	CIM	n/a
<b>Thermal Reporting</b>	CIM	n/a
<b>Static Power Saver</b>	ASM, HMC, Redfish	ASM, HMC, Redfish
<b>Maximum Performance Mode</b>	<b>Enabled by Default</b> ASM, HMC, Redfish	<b>Enabled by Default</b> ASM, HMC, Redfish
<b>Dynamic Performance Mode</b>	ASM, HMC, Redfish	ASM, HMC, Redfish
<b>Dynamic Power Saver Tunable Parameters</b>	ASM, Redfish	ASM, Redfish
<b>Idle Power Saver</b>	<b>Enabled by Default</b> ASM, HMC, Redfish	n/a
<b>User Set Power Capping</b>	DCMI	n/a

## Frequency

Depending upon the Power savings setting selected, the maximum and minimum frequency may change. For a definition of each setting, please refer to the EnergyScale Features chapter earlier in this document.

### Support Notes

<sup>1</sup> Note that CPU frequencies greater than the system base are *not* guaranteed. The actual maximum frequency may vary based on environmental conditions, system configuration, firmware version, component tolerances, and workload.

<sup>2</sup> When in Maximum Performance mode for frequency to be guaranteed to be in the MPM normal operating frequency range the ambient must be below this temperature

	<i>Guaranteed Ambient Temperature<sup>2</sup></i>	<i>Dynamic Performance Mode Normal Operating Frequency Range<sup>1</sup></i>	<i>Maximum Performance Mode Normal Operating Frequency Range<sup>1</sup></i>	<i>Static Power Saver / Minimum Frequency</i>
<b>E950 @2.8GHz</b>	25C	2.8 – 3.8 GHz	3.15 - 3.8 GHz	2.7 GHz
<b>E950 @2.85GHz</b>	25C	2.85 – 3.8 GHz	3.2 - 3.8 GHz	2.7 GHz
<b>E950 @3.0GHz</b>	25C	3.0 – 3.8 GHz	3.4 - 3.8 GHz	2.7 GHz
<b>E950 @3.3GHz</b>	25C	3.3 – 3.8 GHz	3.6 - 3.8 GHz	2.7 GHz
<b>E980 @2.9GHz</b>	22C	2.9 – 3.9 GHz	3.55 - 3.9 GHz	2.7 GHz
<b>E980 @3.0GHz</b>	22C	3.0 – 3.9 GHz	3.584 - 3.9 GHz	2.7 GHz
<b>E980 @3.15GHz</b>	22C	3.15 – 3.9 GHz	3.7 - 3.9 GHz	2.7 GHz
<b>E980 @3.4GHz</b>	22C	3.4 – 4.0 GHz	3.8 - 4.0 GHz	2.7 GHz

### Maximum Frequency by Core Count

Maximum frequency in MPM and DPM based on number of cores active. Values in MHz.

	24	23	22	21	20	19	18	17	16	15	14	13	12
<b>E950 12 BC @2.8GHz</b>	3410	3470	3537	3603	3679	3760	3800	3800	3800	3800	3800	3800	3800
<b>E950 11 BC @2.85GHz</b>			3510	3585	3663	3744	3800	3800	3800	3800	3800	3800	3800
<b>E950 10 BC @3.0GHz</b>					3700	3775	3800	3800	3800	3800	3800	3800	3800
<b>E950 8 BC @3.3GHz</b>									3800	3800	3800	3800	3800
<b>E980 12 BC @2.9GHz</b>	3771	3826	3884	3900	3900	3900	3900	3900	3900	3900	3900	3900	3900
<b>E980 11 BC @3.0GHz</b>			3861	3900	3900	3900	3900	3900	3900	3900	3900	3900	3900
<b>E980 6 BC @3.0GHz</b>													3900
<b>E980 10 BC @3.15GHz</b>					3900	3900	3900	3900	3900	3900	3900	3900	3900
<b>E980 8 BC @3.4GHz</b>									4000	4000	4000	4000	4000

# Release Level OP910 / OP920

## Feature Support

Refer to the EnergyScale Features chapter earlier in this document for a definition of each feature.

	Power LC921	Power LC922	Power AC922
Power Trending	BMC GUI DCMI	BMC GUI DCMI	BMC GUI DCMI
Thermal Reporting	BMC GUI DCMI	BMC GUI DCMI	BMC GUI DCMI
User Set Power Capping	DCMI	DCMI	DCMI

## Frequency

### Support Notes

<sup>1</sup> Note that CPU frequencies greater than the system base are *not* guaranteed. The actual maximum frequency may vary based on environmental conditions, system configuration, firmware version, component tolerances, and workload.

<sup>2</sup> For frequency to be guaranteed to be in the normal operating frequency range the ambient must be below this temperature.

	<i>Guaranteed Ambient Temperature<sup>2</sup></i>	<i>Minimum Frequency</i>	<i>Normal Operating Frequency Range<sup>1</sup></i>
<b>LC921 @2.13GHz 40 core</b>	25C	2.1 GHz	2.3 - 3.8 GHz
<b>LC921 @2.2GHz 32 core</b>	25C	2.1 GHz	2.5 - 3.8 GHz
<b>LC922 @2.6GHz 44 core</b>	35C	2.1 GHz	2.6 - 3.8 GHz
<b>LC922 @2.7GHz 40 core</b>	35C	2.1 GHz	2.7 - 3.8 GHz
<b>LC922 @2.91GHz 32 core</b>	35C	2.1 GHz	2.91 - 3.8 GHz
<b>AC922 @3.0GHz 20 core</b>	35C	2.3 GHz	3.0 - 3.8 GHz
<b>AC922 @3.3GHz 16 core</b>	35C	2.3 GHz	3.3 - 3.8 GHz
<b>AC922 @3.1GHz 22 core</b>	35C	2.3 GHz	3.1 - 3.8 GHz
<b>AC922 @3.45GHz 16 core</b>	35C	2.3 GHz	3.45 - 3.8 GHz

## **Maximum Frequency by Core Count**

Maximum frequency based on number of cores active. Values in MHz.

	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10
<b>LC921 20 SC @2.13GHz</b>					2934	3027	3121	3215	3332	3449	3566	3707	3800	3800	3800
<b>LC921 16 SC @2.2GHz</b>									3334	3455	3578	3719	3800	3800	3800
<b>LC922 22 SC @2.6GHz</b>															
<b>LC922 20 SC @2.7GHz</b>															
<b>LC922 16 SC @2.91GHz</b>															
<b>AC922 20 SC @3.0GHz</b>															
<b>AC922 16 SC @3.3GHz</b>															
<b>AC922 22 SC @3.1GHz</b>															
<b>AC922 16 SC @3.45GHz</b>															

LC921 → Boston 1U LC

LC922 → Boston 2U LC

AC922 → Witherspoon S922HPC

Additional in Eric's doc:

Boston 2U LC 22USC @2.4

Boston 2U LC 20SC @2.2

Boston 2U LC 16SC @2.25

Witherspoon 22USC @2.8

Witherspoon 20SC @2.4

Witherspoon 18SC @3.15

Witherspoon 16SC @2.7

## Appendix II: Processor Usage and Accounting

For historical reasons, process accounting charges and processor utilization are usually formulated in terms of time. The processors used in early time-sharing computer systems were fixed-frequency and single-threaded; processing capacity per unit of time was relatively constant. Consequently, it was convenient to express process accounting charges in terms of processor time and processor utilization as the percentage of time that the processor was not idle over an interval of interest.

With the introduction of multi-threaded processors (i.e. processors capable of executing multiple programs simultaneously), it was no longer desirable to report processor utilization based on the percentage of time that the processor was not idle. Consider a processor that supports 2-way Simultaneous Multi-Threading (SMT). When both threads are idle, utilization should be 0%. When both threads are not idle, utilization should be 100%. When one thread is idle and the other is not idle, there are several options:

1. Treat the processor as idle (0% utilization). This is obviously wrong, as the process is consuming more than 50% of the processing capacity.
2. Continue to treat the processor as not idle and charge the process with 100% of the time (100% utilization). This is a little better than the first option, but it causes utilization to be over-reported since it does not recognize the capacity available in the idle thread. The process is also significantly over-charged compared to when it shares the processor with another process.
3. Treat processor threads as individual processors, charging the process with 100% of the time and reporting processor-thread based utilization (50% utilization). This causes utilization to be under-reported because SMT efficiencies are much less than 100%, that is, the not idle thread typically represents only 15%-30% additional capacity. The process is also significantly over-charged compared to when it shares the processor with another process.

To address this problem, the Processor Utilization Resource Register (PURR) was introduced on POWER5™ for the purpose of apportioning processor time among the processor's threads. The PURR is defined such that  $\sum (\Delta \text{PURR}) = \Delta \text{TIME}$  over any interval. For each SMT context, PURR ticks are apportioned among the idle and non-idle processor threads, so that non-idle PURR as a portion of available time for the processor reflects the relationship between throughput and processor utilization observed for a representative commercial processing workload. By expressing processor utilization as the ratio of not idle PURR ticks to available time and by expressing process accounting in terms of PURR ticks, the historical definition and relationship between processor utilization and process accounting was maintained.

IBM POWER9™ processor-based systems with EnergyScale employ variable processor speed technology to dynamically optimize the speed and energy usage of the processor to the demand of the workload. Because the PURR ticks at a constant rate independent of processor speed, PURR-based processor utilization remains a useful and accurate metric, but PURR-based process accounting charges can vary depending on processor speed. To address this problem, the IBM POWER6 processor included a new per-thread processor timekeeping facility to normalize the relationship between processor time and processor speed. The new facility was named the Scaled Processor Utilization Resource Register (SPURR), and it represented processor time at nominal (i.e. 100%) speed.<sup>5</sup> The SPURR was primarily intended to improve process accounting consistency, but it can also be used in conjunction with the PURR in processor speed and capacity calculations. For example, a SPURR to

---

<sup>5</sup>When the POWER9 processor is operating at full speed, the PURR and SPURR tick in lockstep; when the POWER9 processor is operating at reduced speed, the SPURR ticks slower than the PURR; when the POWER9 processor is operating in excess of full speed, the SPURR ticks faster than the PURR.

PURR percentage of 85% indicates that the processor operated at 85% nominal speed over a sample interval.

While the PURR and SPURR provide the information necessary to measure accurate processor utilization and to perform consistent process accounting in the variable processor speed environment, they do not address the ambiguity of CPU time in the historical context. Simply stated, whether a CPU time value in a legacy software interface should represent PURR-based CPU time or SPURR-based CPU time is open to some interpretation. There is no single best choice to handle all cases, and in fact, the issue has not been uniformly addressed by all operating systems or even among versions of the same system. There is general agreement that SPURR-based process accounting is preferable to PURR-based accounting, since the results are more consistent across EnergyScale modes and within modes that vary the processor speed dynamically. This is particularly true for POWER9, which can operate across a wider range of processor speeds than POWER6.

## **Appendix III: Resources**

### **AIX:**

<http://publib.boulder.ibm.com/infocenter/pseries/v6r1/index.jsp>  
(Search for "spurr" to discover references regarding enablement in APIs and/or tools)

### **DCMI:**

<http://www.intel.com/content/dam/www/public/us/en/documents/technical-specifications/dcmi-v1-5-rev-spec.pdf>

### **IBM i:**

[http://www-01.ibm.com/support/knowledgecenter/ssw\\_ibm\\_i/welcome](http://www-01.ibm.com/support/knowledgecenter/ssw_ibm_i/welcome)  
(Search for "Energy management", "Processor folding", "Scaled processor time attribute", "Processor time", "Scaled processor time", "Processor utilized time", "Processor scaled utilized time", "Processor interrupt time", "Processor scaled interrupt time", "Processor stolen time", "Processor scaled stolen time", "Processor donated time", "Processor scaled donated time", "Processor idle time", "Processor scaled idle time")

### **IBM EnergyScale for POWER8 Processor-Based Systems:**

[http://www-01.ibm.com/common/ssi/cgi-bin/ssialias?subtype=WH&infotype=SA&appname=STGE\\_PO\\_PO\\_USEN&htmlfid=POW03125USEN&attachment=POW03125USEN.PDF#loaded](http://www-01.ibm.com/common/ssi/cgi-bin/ssialias?subtype=WH&infotype=SA&appname=STGE_PO_PO_USEN&htmlfid=POW03125USEN&attachment=POW03125USEN.PDF#loaded)

### **Power Systems:**

<https://www.ibm.com/it-infrastructure/power>

### **OpenBMC:**

[https://www.ibm.com/support/knowledgecenter/POWER9/p9eih/p9eih\\_managing\\_with\\_openbmc.htm](https://www.ibm.com/support/knowledgecenter/POWER9/p9eih/p9eih_managing_with_openbmc.htm)

The Power Architecture and Power.org wordmarks and the Power and Power.org logos and related marks are trademarks and service marks licensed by Power.org.

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

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

Java and all Java-based trademarks and logos are trademarks of Sun Microsystems, Inc. In the United States and/or other countries.

All performance information was determined in a controlled environment. Actual results may vary. Performance information is provided "AS IS" and no warranties or guarantees are expressed or implied by IBM. Buyers should consult other sources of information, including system benchmarks, to evaluate the performance of a system they are considering buying.

Photographs show engineering and design models. Changes may be incorporated in production models.

Copying or downloading the images contained in this document is expressly prohibited without the written consent of IBM.



© IBM Corporation 2018  
IBM Corporation  
Systems and Technology Group  
Route 100  
Somers, New York 10589

Produced in the United States of America  
September 2018  
All Rights Reserved

This document was developed for products and/or services offered in the United States. IBM may not offer the products, features, or services discussed in this document in other countries.

The information may be subject to change without notice. Consult your local IBM business contact for information on the products, features and services available in your area.

All statements regarding IBM future directions and intent are subject to change or withdrawal without notice and represent goals and objectives only.

IBM, the IBM logo, ibm.com, AIX, BladeCenter, EnergyScale, i5/OS, Power, POWER, POWER6, POWER7, POWER8, POWER9, System i and System p are trademarks or registered trademarks of International Business Machines Corporation in the United States or other countries or both. If these and other IBM trademarked terms are marked on their first occurrence in this information with a trademark symbol (® or ™), these symbols indicate U.S. registered or common law trademarks owned by IBM at the time this information was published. Such trademarks may also be registered or common law trademarks in other countries. A current list of IBM trademarks is available on the Web at "Copyright and trademark information" at [www.ibm.com/legal/copytrade.shtml](http://www.ibm.com/legal/copytrade.shtml)

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

IBM hardware products are manufactured from new parts, or new and used parts. . In some cases, the hardware product may not be new and may have been previously installed. Regardless, our warranty terms apply.

This equipment is subject to FCC rules. It will comply with the appropriate FCC rules before final delivery to the buyer.

Information concerning non-IBM products was obtained from the suppliers of these products or other public sources. Questions on the capabilities of the non-IBM products should be addressed with those suppliers.

When referring to storage capacity, 1 TB equals total GB divided by 1000; accessible capacity may be less.

The IBM home page on the Internet can be found at: <http://www.ibm.com>.

The IBM Power Systems home page on the Internet can be found at: <http://www.ibm.com/systems/power/>

49019149USEN-02