MyWebUniversity
python
Rocky Linux
Manual Pages
FPDF Library
PDF generator
FPDF
Full credit is given to the above companies including the OS that this PDF file was generated!

Rocky Enterprise Linux 9.2 Manual Pages on command 'turbostat.8'

$ man turbostat.8

TURBOSTAT(8)   System Manager's Manual   TURBOSTAT(8)

NAME

turbostat - Report processor frequency and idle statistics

SYNOPSIS

turbostat [Options] command

turbostat [Options] [--interval seconds]

DESCRIPTION

turbostat reports processor topology, frequency, idle power-state sta?
tistics, temperature and power on X86 processors. There are two ways
to invoke turbostat. The first method is to supply a command, which is
forked and statistics are printed in one-shot upon its completion. The
second method is to omit the command, and turbostat displays statistics
every 5 seconds interval. The 5-second interval can be changed using
the --interval option.

Some information is not available on older processors.

Options

Options can be specified with a single or double '-', and only as much
of the option name as necessary to disambiguate it from others is nec?

essary. Note that options are case-sensitive.

--add attributes add column with counter having specified 'attributes'.

The 'location' attribute is required, all others are optional.

location: {msrDDD | msr0xXXX | /sys/path...}

msrDDD is a decimal offset, eg. msr16

msr0xXXX is a hex offset, eg. msr0x10

/sys/path... is an absolute path to a sysfs attribute

scope: {cpu | core | package}

sample and print the counter for every cpu, core, or package.

default: cpu

size: {u32 | u64 }

MSRs are read as 64-bits, u32 truncates the displayed value to 32-bits.

default: u64

format: {raw | delta | percent}

'raw' shows the MSR contents in hex.

'delta' shows the difference in values during the measurement interval.

'percent' shows the delta as a percentage of the cycles elapsed.

default: delta

name: "name_string"

Any string that does not match a key-word above is used

as the column header.

--cpu cpu-set limit output to system summary plus the specified cpu-set. If cpu-set is the string "core", then the system summary plus the

first CPU in each core are printed -- eg. subsequent HT siblings are

not printed. Or if cpu-set is the string "package", then the system

summary plus the first CPU in each package is printed. Otherwise, the

system summary plus the specified set of CPUs are printed. The cpu-set

is ordered from low to high, comma delimited with ".." and "-" permit?

ted to denote a range. eg. 1,2,8,14..17,21-44

--hide column do not show the specified built-in columns. May be in?

voked multiple times, or with a comma-separated list of column names.

--enable column show the specified built-in columns, which are other?

wise disabled, by default. Currently the only built-in counters dis?

abled by default are "usec", "Time_Of_Day_Seconds", "APIC" and "X2APIC". The column name "all" can be used to enable all disabled-by-default built-in counters.

--show column show only the specified built-in columns. May be invoked multiple times, or with a comma-separated list of column names.

--show CATEGORY --hide CATEGORY Show and hide also accept a single CATEGORY of columns: "all", "topology", "idle", "frequency", "power", "sysfs", "other".

--Dump displays the raw counter values.

--quiet Do not decode and print the system configuration header infor? mation.

--interval seconds overrides the default 5.0 second measurement inter? val.

--num_iterations num number of the measurement iterations.

--out output_file turbostat output is written to the specified out? put_file. The file is truncated if it already exists, and it is cre? ated if it does not exist.

--help displays usage for the most common parameters.

--Joules displays energy in Joules, rather than dividing Joules by time to print power in Watts.

--list display column header names available for use by --show and --hide, then exit.

--Summary limits output to a 1-line System Summary for each interval.

--TCC temperature sets the Thermal Control Circuit temperature for sys? tems which do not export that value. This is used for making sense of the Digital Thermal Sensor outputs, as they return degrees Celsius be? low the TCC activation temperature.

--version displays the version.

The command parameter forks command, and upon its exit, displays the statistics gathered since it was forked.

ROW DESCRIPTIONS

The system configuration dump (if --quiet is not used) is followed by statistics. The first row of the statistics labels the content of each column (below). The second row of statistics is the system summary line. The system summary line has a '-' in the columns for the Pack? age, Core, and CPU. The contents of the system summary line depends on the type of column. Columns that count items (eg. IRQ) show the sum across all CPUs in the system. Columns that show a percentage show the average across all CPUs in the system. Columns that dump raw MSR val? ues simply show 0 in the summary. After the system summary row, each row describes a specific Package/Core/CPU. Note that if the --cpu pa? rameter is used to limit which specific CPUs are displayed, turbostat will still collect statistics for all CPUs in the system and will still show the system summary for all CPUs in the system.

COLUMN DESCRIPTIONS

usec For each CPU, the number of microseconds elapsed during counter collection, including thread migration -- if any. This counter is disabled by default, and is enabled with "--enable usec", or --debug. On the summary row, usec refers to the total elapsed time to collect the counters on all cpus.

Time_Of_Day_Seconds For each CPU, the gettimeofday(2) value (seconds.subsec since Epoch) when the counters ending the measurement interval were collected. This column is disabled by default, and can be enabled with "--enable Time_Of_Day_Seconds" or "--debug". On the summary row, Time_Of_Day_Seconds refers to the timestamp following collection of counters on the last CPU.

Core processor core number. Note that multiple CPUs per core indicate support for Intel(R) Hyper-Threading Technology (HT).

CPU Linux CPU (logical processor) number. Yes, it is okay that on many systems the CPUs are not listed in numerical order -- for efficiency reasons, turbostat runs in topology order, so HT siblings appear together.

Package processor package number -- not present on systems with a single processor package.

Avg_MHz number of cycles executed divided by time elapsed. Note that this includes idle-time when 0 instructions are executed.

Busy% percent of the measurement interval that the CPU executes instructions, aka. % of time in "C0" state.

Bzy_MHz average clock rate while the CPU was not idle (ie. in "c0" state).

TSC_MHz average MHz that the TSC ran during the entire interval.

IRQ The number of interrupts serviced by that CPU during the measurement interval. The system total line is the sum of interrupts serviced across all CPUs. turbostat parses /proc/interrupts to generate this summary.

SMI The number of System Management Interrupts serviced CPU during the measurement interval. While this counter is actually per-CPU, SMI are triggered on all processors, so the number should be the same for all CPUs.

C1, C2, C3... The number times Linux requested the C1, C2, C3 idle state during the measurement interval. The system summary line shows the sum for all CPUs. These are C-state names as exported in /sys/devices/system/cpu/cpu*/cpuidle/state*/name. While their names are generic, their attributes are processor specific. They the system description section of output shows what MWAIT sub-states they are mapped to on each system.

C1%, C2%, C3% The residency percentage that Linux requested C1, C2, C3.... The system summary is the average of all CPUs in the system. Note that these are software, reflecting what was requested. The hardware counters reflect what was actually achieved.

CPU%c1, CPU%c3, CPU%c6, CPU%c7 show the percentage residency in hardware core idle states. These numbers are from hardware residency counters.

CoreTmp Degrees Celsius reported by the per-core Digital Thermal Sensor.

PkgTmp Degrees Celsius reported by the per-package Package Thermal Monitor.

GFX%rc6 The percentage of time the GPU is in the "render C6" state, rc6, during the measurement interval. From /sys/class/drm/card0/power/rc6_residency_ms.

GFXMHz Instantaneous snapshot of what sysfs presents at the end of the measurement interval. From /sys/class/graphics/fb0/device/drm/card0/gt_cur_freq_mhz.

Pkg%pc2, Pkg%pc3, Pkg%pc6, Pkg%pc7 percentage residency in hardware package idle states. These numbers are from hardware residency counters.

PkgWatt Watts consumed by the whole package.

CorWatt Watts consumed by the core part of the package.

GFXWatt Watts consumed by the Graphics part of the package -- available only on client processors.

RAMWatt Watts consumed by the DRAM DIMMS -- available only on server processors.

PKG_% percent of the interval that RAPL throttling was active on the Package. Note that the system summary is the sum of the package throttling time, and thus may be higher than 100% on a multi-package system. Note that the meaning of this field is model specific. For example, some hardware increments this counter when RAPL responds to thermal limits, but does not increment this counter when RAPL responds to power limits. Comparing PkgWatt and PkgTmp to system limits is necessary.

RAM_% percent of the interval that RAPL throttling was active on DRAM.

TOO MUCH INFORMATION EXAMPLE

By default, turbostat dumps all possible information -- a system con?
figuration header, followed by columns for all counters. This is ideal
for remote debugging, use the "--out" option to save everything to a
text file, and get that file to the expert helping you debug.

When you are not interested in all that information, and there are sev?

eral ways to see only what you want. First the "--quiet" option will skip the configuration information, and turbostat will show only the counter columns. Second, you can reduce the columns with the "--hide" and "--show" options. If you use the "--show" option, then turbostat will show only the columns you list. If you use the "--hide" option, turbostat will show all columns, except the ones you list.

To find out what columns are available for --show and --hide, the "--list" option is available. For convenience, the special strings "sysfs" can be used to refer to all of the sysfs C-state counters at once:

```
sudo ./turbostat --show sysfs --quiet sleep 10
10.003837 sec
    C1   C1E  C3   C6   C7s  C1%  C1E% C3%  C6%  C7s%
    4    21   2    2    459  0.14 0.82 0.00 0.00 98.93
    1    17   2    2    130  0.00 0.02 0.00 0.00 99.80
    0    0    0    0    31   0.00 0.00 0.00 0.00 99.95
    2    1    0    0    52   1.14 6.49 0.00 0.00 92.21
    1    2    0    0    52   0.00 0.08 0.00 0.00 99.86
    0    0    0    0    71   0.00 0.00 0.00 0.00 99.89
    0    0    0    0    25   0.00 0.00 0.00 0.00 99.96
    0    0    0    0    74   0.00 0.00 0.00 0.00 99.94
    0    1    0    0    24   0.00 0.00 0.00 0.00 99.84
```

ONE SHOT COMMAND EXAMPLE

If turbostat is invoked with a command, it will fork that command and output the statistics gathered after the command exits. In this case, turbostat output goes to stderr, by default. Output can instead be saved to a file using the --out option. In this example, the "sleep 10" command is forked, and turbostat waits for it to complete before saving all statistics into "ts.out". Note that "sleep 10" is not part of turbostat, but is simply an example of a command that turbostat can fork. The "ts.out" file is what you want to edit in a very wide win? dow, paste into a spreadsheet, or attach to a bugzilla entry.

```
[root@hsw]# ./turbostat -o ts.out sleep 10
```

```
[root@hsw]#
```

PERIODIC INTERVAL EXAMPLE

Without a command to fork, turbostat displays statistics ever 5 sec? onds. Periodic output goes to stdout, by default, unless --out is used to specify an output file. The 5-second interval can be changed with the "-i sec" option.

```
sudo ./turbostat --quiet --hide sysfs,IRQ,SMI,CoreTmp,PkgTmp,GFX%rc6,GFXMHz,PkgWatt,CorWatt,GFXWatt
    Core CPU  Avg_MHz   Busy%     Bzy_MHz   TSC_MHz   CPU%c1    CPU%c3    CPU%c6    CPU%c7
    -    -    488  12.52     3900 3498 12.50     0.00 0.00 74.98
    0    0    5    0.13 3900 3498 99.87     0.00 0.00 0.00
    0    4    3897 99.99     3900 3498 0.01
    1    1    0    0.00 3856 3498 0.01 0.00 0.00 99.98
    1    5    0    0.00 3861 3498 0.01
    2    2    1    0.02 3889 3498 0.03 0.00 0.00 99.95
    2    6    0    0.00 3863 3498 0.05
    3    3    0    0.01 3869 3498 0.02 0.00 0.00 99.97
    3    7    0    0.00 3878 3498 0.03
    Core CPU  Avg_MHz   Busy%     Bzy_MHz   TSC_MHz   CPU%c1    CPU%c3    CPU%c6    CPU%c7
    -    -    491  12.59     3900 3498 12.42     0.00 0.00 74.99
    0    0    27   0.69 3900 3498 99.31     0.00 0.00 0.00
    0    4    3898 99.99     3900 3498 0.01
    1    1    0    0.00 3883 3498 0.01 0.00 0.00 99.99
    1    5    0    0.00 3898 3498 0.01
    2    2    0    0.01 3889 3498 0.02 0.00 0.00 99.98
    2    6    0    0.00 3889 3498 0.02
    3    3    0    0.00 3856 3498 0.01 0.00 0.00 99.99
    3    7    0    0.00 3897 3498 0.01
```

This example also shows the use of the --hide option to skip columns that are not wanted. Note that cpu4 in this example is 99.99% busy, while the other CPUs are all under 1% busy. Notice that cpu4's HT sib? ling is cpu0, which is under 1% busy, but can get into CPU%c1 only, be? cause its cpu4's activity on shared hardware keeps it from entering a deeper C-state.

SYSTEM CONFIGURATION INFORMATION EXAMPLE

By default, turbostat always dumps system configuration information be? fore taking measurements. In the example above, "--quiet" is used to suppress that output. Here is an example of the configuration informa? tion:

```
turbostat version 2017.02.15 - Len Brown <lenb@kernel.org>
CPUID(0): GenuineIntel 13 CPUID levels; family:model:stepping 0x6:3c:3 (6:60:3)
CPUID(1): SSE3 MONITOR - EIST TM2 TSC MSR ACPI-TM TM
CPUID(6): APERF, TURBO, DTS, PTM, No-HWP, No-HWPnotify, No-HWPwindow, No-HWPepp, No-HWPpkg, EPB
cpu4: MSR_IA32_MISC_ENABLE: 0x00850089 (TCC EIST No-MWAIT PREFETCH TURBO)
CPUID(7): No-SGX
cpu4: MSR_MISC_PWR_MGMT: 0x00400000 (ENable-EIST_Coordination DISable-EPB DISable-OOB)
RAPL: 3121 sec. Joule Counter Range, at 84 Watts
cpu4: MSR_PLATFORM_INFO: 0x80838f3012300
8 * 100.0 = 800.0 MHz max efficiency frequency
35 * 100.0 = 3500.0 MHz base frequency
cpu4: MSR_IA32_POWER_CTL: 0x0004005d (C1E auto-promotion: DISabled)
cpu4: MSR_TURBO_RATIO_LIMIT: 0x25262727
37 * 100.0 = 3700.0 MHz max turbo 4 active cores
38 * 100.0 = 3800.0 MHz max turbo 3 active cores
39 * 100.0 = 3900.0 MHz max turbo 2 active cores
39 * 100.0 = 3900.0 MHz max turbo 1 active cores
cpu4: MSR_CONFIG_TDP_NOMINAL: 0x00000023 (base_ratio=35)
cpu4: MSR_CONFIG_TDP_LEVEL_1: 0x00000000 ()
cpu4: MSR_CONFIG_TDP_LEVEL_2: 0x00000000 ()
cpu4: MSR_CONFIG_TDP_CONTROL: 0x80000000 ( lock=1)
cpu4: MSR_TURBO_ACTIVATION_RATIO: 0x00000000 (MAX_NON_TURBO_RATIO=0 lock=0)
cpu4: MSR_PKG_CST_CONFIG_CONTROL: 0x1e000400 (UNdemote-C3, UNdemote-C1, demote-C3, demote-C1, UNlocked: pkg-cstate-limit=0: pc0)
cpu4: POLL: CPUIDLE CORE POLL IDLE
cpu4: C1: MWAIT 0x00
cpu4: C1E: MWAIT 0x01
cpu4: C3: MWAIT 0x10
```

```
cpu4: C6: MWAIT 0x20
cpu4: C7s: MWAIT 0x32
cpu4: MSR_MISC_FEATURE_CONTROL: 0x00000000 (L2-Prefetch L2-Prefetch-pair L1-Prefetch L1-IP-Prefetch)
cpu0: MSR_IA32_ENERGY_PERF_BIAS: 0x00000006 (balanced)
cpu0: MSR_CORE_PERF_LIMIT_REASONS, 0x31200000 (Active: ) (Logged: Transitions, MultiCoreTurbo, Amps, Auto-HWP, )
cpu0: MSR_GFX_PERF_LIMIT_REASONS, 0x00000000 (Active: ) (Logged: )
cpu0: MSR_RING_PERF_LIMIT_REASONS, 0x0d000000 (Active: ) (Logged: Amps, PkgPwrL1, PkgPwrL2, )
cpu0: MSR_RAPL_POWER_UNIT: 0x000a0e03 (0.125000 Watts, 0.000061 Joules, 0.000977 sec.)
cpu0: MSR_PKG_POWER_INFO: 0x000002a0 (84 W TDP, RAPL 0 - 0 W, 0.000000 sec.)
cpu0: MSR_PKG_POWER_LIMIT: 0x428348001a82a0 (UNlocked)
cpu0: PKG Limit #1: ENabled (84.000000 Watts, 8.000000 sec, clamp DISabled)
cpu0: PKG Limit #2: ENabled (105.000000 Watts, 0.002441* sec, clamp DISabled)
cpu0: MSR_PP0_POLICY: 0
cpu0: MSR_PP0_POWER_LIMIT: 0x00000000 (UNlocked)
cpu0: Cores Limit: DISabled (0.000000 Watts, 0.000977 sec, clamp DISabled)
cpu0: MSR_PP1_POLICY: 0
cpu0: MSR_PP1_POWER_LIMIT: 0x00000000 (UNlocked)
cpu0: GFX Limit: DISabled (0.000000 Watts, 0.000977 sec, clamp DISabled)
cpu0: MSR_IA32_TEMPERATURE_TARGET: 0x00641400 (100 C)
cpu0: MSR_IA32_PACKAGE_THERM_STATUS: 0x884c0800 (24 C)
cpu0: MSR_IA32_THERM_STATUS: 0x884c0000 (24 C +/- 1)
cpu1: MSR_IA32_THERM_STATUS: 0x88510000 (19 C +/- 1)
cpu2: MSR_IA32_THERM_STATUS: 0x884e0000 (22 C +/- 1)
cpu3: MSR_IA32_THERM_STATUS: 0x88510000 (19 C +/- 1)
cpu4: MSR_PKGC3_IRTL: 0x00008842 (valid, 67584 ns)
cpu4: MSR_PKGC6_IRTL: 0x00008873 (valid, 117760 ns)
cpu4: MSR_PKGC7_IRTL: 0x00008891 (valid, 148480 ns)
```

The max efficiency frequency, a.k.a. Low Frequency Mode, is the fre?quency available at the minimum package voltage. The TSC frequency is the base frequency of the processor -- this should match the brand string in /proc/cpuinfo. This base frequency should be sustainable on all CPUs indefinitely, given nominal power and cooling. The remaining rows show what maximum turbo frequency is possible depending on the number of idle cores. Note that not all information is available on all processors.

ADD COUNTER EXAMPLE

Here we limit turbostat to showing just the CPU number for cpu0 - cpu3.

We add a counter showing the 32-bit raw value of MSR 0x199 (MSR_IA32_PERF_CTL), labeling it with the column header, "PRF_CTRL", and display it only once, afte the conclusion of a 0.1 second sleep.

```
sudo ./turbostat --quiet --cpu 0-3 --show CPU --add msr0x199,u32,raw,PRF_CTRL sleep .1
0.101604 sec
CPU    PRF_CTRL
-    0x00000000
0    0x00000c00
1    0x00000800
2    0x00000a00
3    0x00000800
```

INPUT

For interval-mode, turbostat will immediately end the current interval when it sees a newline on standard input. turbostat will then start the next interval. Control-C will be send a SIGINT to turbostat, which will immediately abort the program with no further processing.

SIGNALS

SIGINT will interrupt interval-mode. The end-of-interval data will be collected and displayed before turbostat exits.

SIGUSR1 will end current interval, end-of-interval data will be col?lected and displayed before turbostat starts a new interval.

NOTES

turbostat must be run as root. Alternatively, non-root users can be enabled to run turbostat this way:

```
# setcap cap_sys_admin,cap_sys_rawio,cap_sys_nice=+ep ./turbostat
# chmod +r /dev/cpu/*/msr
```

turbostat reads hardware counters, but doesn't write them. So it will not interfere with the OS or other programs, including multiple invoca?

tions of itself.

turbostat may work poorly on Linux-2.6.20 through 2.6.29, as acpi-cpufreq periodically cleared the APERF and MPERF MSRs in those kernels.

AVG_MHz = APERF_delta/measurement_interval. This is the actual number of elapsed cycles divided by the entire sample interval -- including idle time. Note that this calculation is resilient to systems lacking a non-stop TSC.

TSC_MHz = TSC_delta/measurement_interval. On a system with an invari?ant TSC, this value will be constant and will closely match the base frequency value shown in the brand string in /proc/cpuinfo. On a sys?tem where the TSC stops in idle, TSC_MHz will drop below the proces?sor's base frequency.

Busy% = MPERF_delta/TSC_delta

Bzy_MHz = TSC_delta/APERF_delta/MPERF_delta/measurement_interval

Note that these calculations depend on TSC_delta, so they are not reli?able during intervals when TSC_MHz is not running at the base fre?quency.

Turbostat data collection is not atomic. Extremely short measurement intervals (much less than 1 second), or system activity that prevents turbostat from being able to run on all CPUS to quickly collect data, will result in inconsistent results.

The APERF, MPERF MSRs are defined to count non-halted cycles. Although it is not guaranteed by the architecture, turbostat assumes that they count at TSC rate, which is true on all processors tested to date.

REFERENCES

Volume 3B: System Programming Guide" https://www.intel.com/prod?ucts/processor/manuals/

FILES

/dev/cpu/*/msr

SEE ALSO

msr(4), vmstat(8)

AUTHOR

Written by Len Brown <len.brown@intel.com>