

Load Average: An Addendum

Neil J. Gunther

[Performance Dynamics](#)
Castro Valley, California, USA

Several readers have queried my statement, in **Section 4: Kernel Magic** of my original online article, that the sampling rate of the `calc_load()` function in the Linux kernel is **once every 5 seconds** rather than *once every 5-th of second*. This addendum tries to address that confusion.

1 Hz vs. HZ

Hz (note the lower case 'z') is the metric SI unit of *frequency* named after the nineteenth century German physicist Heinrich Hertz (not the car rental company). It is a relevant measurement unit for any periodic or cyclical phenomenon such as your pulse (e.g., 1.2 Hz) or a musical pitch (concert A is 440 Hz). The assumed timebase is *seconds* and therefore, 'Hz' can also be read as the number of *cycles per second*.

HZ, on the other hand, is the *name* of a programming variable (or constant in this case), not a unit of measurement. It is the name of a number. The actual value of that number is [machine-dependent](#). The following platforms use `HZ == 100`:

- [Intel Pentium](#)
- [Power PC](#)
- [Ultra-SPARC](#)

whereas the [HP Alpha](#) uses `HZ == 1024` or `HZ == 1200`, depending on the CPU model. We'll assume `HZ` represents 100 in the subsequent explanation. The next question is, 100 what?

Every computer platform has a clock implemented in hardware that has a constant ticking rate by which everything else in the system is synchronized. To make this ticking rate known to the system, the hardware sends an interrupt to the Linux kernel on every clock *tick*. A `HZ` value of 100 means that **one second of wall-clock time** corresponds to 100 CPU ticks. In other words, a clock interrupt occurs with a frequency of one interrupt every 100-th of a second or 100 interrupts per second (from which it follows that the `HZ` constant, representing 1 second, corresponds to 100 Hz). Conversely, 1 CPU tick = 1 second / 100 interrupts = 10 milliseconds. So, the number represented by the constant `HZ` is a *scale factor* between the system-clock and wall-clock time.

2 Sample Rate

The workhorse routine is the `CALC_LOAD` macro defined in [sched.h](#) We reproduce the relevant block of code here for easier reference:

```

61 #define FSHIFT          11          /* nr of bits of precision */
62 #define FIXED_1         (1<<FSHIFT) /* 1.0 as fixed-point */
63 #define LOAD_FREQ       (5*HZ)      /* 5 sec intervals */
64 #define EXP_1           1884        /* 1/exp(5sec/1min) as fixed-point */
65 #define EXP_5           2014        /* 1/exp(5sec/5min) */
66 #define EXP_15          2037        /* 1/exp(5sec/15min) */
67
68 #define CALC_LOAD(load,exp,n) \
69     load *= exp; \
70     load += n*(FIXED_1-exp); \
71     load >>= FSHIFT;

```

Looking at [line 63](#), it says that another constant `LOAD_FREQ` is equal to $(5 * \text{HZ})$ i.e., 5 multiplied by the value of the `HZ` constant. Also, note the comment (from Linus), which says that the value of `LOAD_FREQ` corresponds to *intervals of 5 seconds*; not intervals of 1/5-th of a second.

Thus, $5 * \text{HZ}$ means five times the value of the constant called `HZ`. Moreover, since `HZ` represents 100 ticks and 5×100 ticks = 500 ticks, it follows that 500 ticks is the same as 500×10 milliseconds or an interval of 5 seconds. All this can be summarized as follows:

$$\begin{aligned}
 1 * \text{HZ} &= 100 \text{ ticks} \\
 5 * \text{HZ} &= 500 \text{ ticks} \\
 1 \text{ tick} &= 10 \text{ milliseconds} \\
 500 \text{ ticks} &= 5000 \text{ milliseconds} == 5 \text{ seconds}
 \end{aligned}$$

So, $5 * \text{HZ}$ means that `CALC_LOAD` is called every 500 CPU ticks or **5 seconds** and not 5 times per second as some people mistakenly think. Also, be careful not to confuse this sampling period of 5 seconds with the *reporting* periods of 1, 5, and 15-minutes.

File translated from T_EX by [T_TH](#), version 3.38.

On 7 Apr 2011, 09:33.