pgtesttiming

- Description
- Options
- Usage
  - Interpreting Results
- See Also

pg_test_timing

pg_test_timing

1

Application

pg_test_timing

measure timing overhead

pg_test_timing

option

Description

pg_test_timing is a tool to measure the timing overhead on your system and confirm that the system time never moves backwards. It reads supported clock sources over and over again as fast as it can for a specified length of time, and then prints statistics about the observed differences in successive clock readings, as well as which clock source will be used.

Smaller (but not zero) differences are better, since they imply both more-precise clock hardware and less overhead to collect a clock reading. Systems that are slow to collect timing data can give less accurate `EXPLAIN ANALYZE` results.

This tool is also helpful to determine if the `track_io_timing` configuration parameter is likely to produce useful results, and whether the TSC clock source (see ???) is available and if it will be used by default.

Options

pg_test_timing accepts the following command-line options:

`-c cutoff`; `--cutoff=cutoff`
Specifies the cutoff percentage for the list of exact observed timing durations (that is, the changes in the system clock value from one reading to the next). The list will end once the running percentage total reaches or exceeds this value, except that the largest observed duration will always be printed. The default cutoff is 99.99.

`-d duration`; `--duration=duration`
Specifies the test duration, in seconds. Longer durations give slightly better accuracy, and are more likely to discover problems with the system clock moving backwards. The default test duration is 3 seconds.

`-V`; `--version`
Print the pg_test_timing version and exit.

`-?`; `--help`
Show help about pg_test_timing command line arguments, and exit.

Usage

Interpreting Results

The first block of output has four columns, with rows showing a shifted-by-one log2(ns) histogram of timing durations (that is, the differences between successive clock readings). This is not the classic log2(n+1) histogram as it counts zeros separately and then switches to log2(ns) starting from value 1.

The columns are:

- nanosecond value that is >= the durations in this bucket
- percentage of durations in this bucket
- running-sum percentage of durations in this and previous buckets

- count of durations in this bucket

The second block of output goes into more detail, showing the exact timing differences observed. For brevity this list is cut off when the running-sum percentage exceeds the user-selectable cutoff value. However, the largest observed difference is always shown.

On platforms that support the TSC clock source, additional output sections are shown for the `RDTSCP` instruction (used for general timing needs, such as `track_io_timing`) and the `RDTSC` instruction (used for `EXPLAIN ANALYZE`). At the end of the output, the TSC frequency, which may either be sourced from CPU information directly, or the alternate calibration mechanism are shown, as well as whether the TSC clock source will be used by default.

The example results below show system clock timing where 99.99% of loops took between 16 and 63 nanoseconds. In the second block, we can see that the typical loop time is 40 nanoseconds, and the readings appear to have full nanosecond precision. Following the system clock results, the TSC clock source results are shown, in the same fashion. The `RDTSCP` instruction shows most loops completing in 20–30 nanoseconds, while the `RDTSC` instruction is the fastest with an average loop time of 20 nanoseconds. In this example the TSC clock source will be used by default, but can be disabled by setting `timing_clock_source` to `system`.

```
System clock source: clock_gettime (CLOCK_MONOTONIC)
Testing timing overhead for 3 seconds.
Average loop time including overhead: 44.67 ns
Histogram of timing durations:
   <= ns   % of total  running %      count
       0       0.0000     0.0000          0
       1       0.0000     0.0000          0
       3       0.0000     0.0000          0
       7       0.0000     0.0000          0
      15       0.0000     0.0000          0
      31      24.0606    24.0606    5385707
      63      75.8342    99.8948   16974658
     127       0.0900    99.9848      20143
     255       0.0069    99.9917       1542
     511       0.0014    99.9932        322
    1023       0.0003    99.9935         68
    2047       0.0001    99.9936         23
    4095       0.0036    99.9972        813
    8191       0.0018    99.9990        402
   16383       0.0005    99.9995        120
   32767       0.0001    99.9997         32
   65535       0.0001    99.9998         24
```

```
Observed timing durations up to 99.9900%:
      ns   % of total  running %      count
      29       3.6921     3.6921     826442
      30      16.6755    20.3676    3732628
      31       3.6930    24.0606     826637
      40      75.7761    99.8368   16961658
      41       0.0019    99.8387        431
...
     190       0.0003    99.9901         65
...
29657159       0.0000   100.0000          1

Clock source: RDTSCP
Average loop time including overhead: 37.32 ns
Histogram of timing durations:
   <= ns   % of total  running %      count
       0       0.0000     0.0000          0
       1       0.0000     0.0000          0
       3       0.0000     0.0000          0
       7       0.0000     0.0000          0
      15       0.0000     0.0000          0
      31      99.9499    99.9499   26782299
      63       0.0381    99.9880      10220
     127       0.0008    99.9889        224
     255       0.0052    99.9941       1403
     511       0.0013    99.9954        340
    1023       0.0001    99.9954         17
    2047       0.0000    99.9955          7
    4095       0.0021    99.9976        569
    8191       0.0013    99.9989        357
   16383       0.0005    99.9994        128
   32767       0.0003    99.9996         70
   65535       0.0001    99.9997         19

Observed timing durations up to 99.9900%:
      ns   % of total  running %      count
      20      16.9064    16.9064    4530201
      29      41.5214    58.4279   11125972
      30      41.5220    99.9499   11126126
```

```
      40       0.0089    99.9587       2374
...
     130       0.0007    99.9902        181
...
18501572       0.0000   100.0000          1

Fast clock source: RDTSC
Average loop time including overhead: 27.12 ns
Histogram of timing durations:
   <= ns   % of total  running %      count
       0       0.0000     0.0000          0
       1       0.0000     0.0000          0
       3       0.0000     0.0000          0
       7       0.0000     0.0000          0
      15       1.2247     1.2247     456231
      31      98.7566    99.9813   36789785
      63       0.0109    99.9921       4049
     127       0.0029    99.9951       1087
     255       0.0008    99.9959        305
     511       0.0007    99.9966        279
    1023       0.0000    99.9966          7
    2047       0.0001    99.9967         22
    4095       0.0018    99.9985        673
    8191       0.0010    99.9995        383
   16383       0.0003    99.9998         94
   32767       0.0001    99.9999         38
   65535       0.0000    99.9999          9

Observed timing durations up to 99.9900%:
      ns   % of total  running %      count
       9       0.6316     0.6316     235290
      10       0.5931     1.2247     220941
      20      91.4328    92.6574   34061442
      29       3.6427    96.3001    1357007
      30       3.6811    99.9813    1371336
      40       0.0089    99.9902       3325
...
61594291       0.0000   100.0000          1

TSC frequency source: x86, cpuid 0x15
```

```
TSC frequency in use: 2449228 kHz
TSC frequency from calibration: 2448603 kHz

TSC clock source will be used by default, unless timing_clock_source is set to 'system'.
```

See Also

Wiki discussion about timing

□□□ #2
□ □□□ □□□ 2026-10-31 14:13:47 UTC
□ □□□ □□□ 2026-10-31 14:13:48 UTC