Benchmark calibration

Table of Contents

[Measurement precision: .time() vs .perf\_counter() 1](#_Toc150091793)

[Measurement precision: # iterations 2](#_Toc150091794)

[Measurement precision: gtime overhead 5](#_Toc150091795)

[Measurement precision: warmup 5](#_Toc150091796)

## Measurement precision: .time() vs .perf\_counter()

Compare average execution time measured by two methods from Python [time](https://docs.python.org/3/library/time.html) module:

Iterations: 10

App Implementation: v1

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
|  | **.time()** | | **.perf\_counter()** | |  | |
| **maxPrime** | **avg exec time, s** | **std deviation** | **avg exec time, s** | **std deviation** | **time diff, %** | **std deviation diff, %** |
| 10 | 0,0104 | 0,000246 | 0,0104 | 0,000203 | 0,0% | 17,4% |
| 100 | 0,0130 | 0,001315 | 0,0126 | 0,000755 | 3,1% | 42,6% |
| 1000 | 0,0250 | 0,000900 | 0,0243 | 0,000275 | 2,8% | 69,5% |
| 10000 | 0,2110 | 0,027382 | 0,1978 | 0,006296 | 6,3% | 77,0% |
| 100000 | 5,7095 | 0,045748 | 5,6265 | 0,028201 | 1,5% | 38,4% |

*.****time()*** introduces up to 70% higher standard deviation than *.****perf\_counter()***

## Measurement precision: # iterations

**Hypothesis**: avg exec time should stabilize by increasing # of iteration (because any high or low outliers are averaged out), while std dev should decrease (according to the square root law, which is a part of the central limit theorem)  
  
App Implementation: v1  
  
Avg. time:

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **avg time, s** | **maxPrime** | | | | |
| **# iterations** | **10** | **100** | **1000** | **10000** | **100000** |
| 1 | 0,0136 | 0,0143 | 0,0313 | 0,2098 | 5,6241 |
| 2 | 0,0130 | 0,0140 | 0,0293 | 0,1958 | 5,7847 |
| 5 | 0,0106 | 0,0139 | 0,0257 | 0,1941 | 5,6699 |
| 10 | 0,0116 | 0,0123 | 0,0257 | 0,1882 | 5,7477 |
| 20 | 0,0117 | 0,0132 | 0,0244 | 0,1878 | 5,7622 |
| 50 | 0,0113 | 0,0119 | 0,0234 | 0,1895 | 5,7531 |

Std. deviation:

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **std deviation** | **maxPrime** | | | | |
| **# iterations** | **10** | **100** | **1000** | **10000** | **100000** |
| 1 | - | - | - | - | - |
| 2 | 0,002227 | 0,000471 | 0,001361 | 0,000218 | 0,125810 |
| 5 | 0,000275 | 0,000608 | 0,001611 | 0,004820 | 0,050500 |
| 10 | 0,001206 | 0,000238 | 0,001413 | 0,006318 | 0,095963 |
| 20 | 0,001240 | 0,001012 | 0,000689 | 0,006584 | 0,073244 |
| 50 | 0,002043 | 0,000669 | 0,000237 | 0,002520 | 0,149871 |

A graph of a function

Description automatically generated with medium confidenceSurprisingly, there’s no visible pattern of std.dev being decreased while increasing # of iterations.   
Avg. time seems to stabilize after **10 iterations**, which will be chosen as **default**:

## Measurement precision: gtime overhead

Run 10 iterations of & measure mean exec time difference for different maxPrime:

gtime -v ./bin/{binary} {max\_prime} > /dev/null

vs

./bin/{binary} {max\_prime} > /dev/null

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **avg time, s** | **maxPrime** | | | | |
| **mode** | **10** | **100** | **1000** | **10000** | **100000** |
| no gtime | 0,00773 | 0,01014 | 0,02184 | 0,19408 | 5,88091 |
| gtime | 0,01059 | 0,01284 | 0,02483 | 0,19412 | 5,86720 |
| error | 27,007% | 21,028% | 12,042% | 0,021% | -0,234% |

The overhead introduced by ***gtime*** diminishes as the number of iterations within the binary increases, starting at approximately 30% for a low iteration count and approaching negligible levels at 1000 iterations. Therefore, the impact of ***gtime*** on performance metrics is minimal for sufficiently long-running tasks.

## Measurement precision: warmup

Warmup – dry execution of a benchmark that doesn’t count into a total statistic. Run N warmup iteration, then run 10 benchmark iteration that count.  
  
Avg. time & std deviation over [0, 1, 2, 5] warmup iterations:

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **avg time, s** | **maxPrime** | | | | |
| **# iterations** | **10** | **100** | **1000** | **10000** | **100000** |
| 0 | 0,0113 | 0,0152 | 0,0307 | 0,1993 | 5,9789 |
| 1 | 0,0110 | 0,0138 | 0,0254 | 0,1888 | 5,8425 |
| 2 | 0,0111 | 0,0146 | 0,0239 | 0,1935 | 5,8943 |
| 5 | 0,0111 | 0,0142 | 0,0250 | 0,1918 | 5,8926 |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **std deviation** | **maxPrime** | | | | |
| **# iterations** | **10** | **100** | **1000** | **10000** | **100000** |
| 0 | 0,000763 | 0,000985 | 0,003865 | 0,003432 | 0,127359 |
| 1 | 0,000716 | 0,001105 | 0,001119 | 0,002835 | 0,203318 |
| 2 | 0,000961 | 0,000796 | 0,000188 | 0,003115 | 0,199401 |
| 5 | 0,000957 | 0,000335 | 0,000743 | 0,005228 | 0,162940 |

Neither avg. time nor std. deviation don’t seem to stabilize visibly after 1 warmup iteration. **1 is chosen as default**.