
Benchmark Harness Documentation

Release 0.0.1

Chris Adams

Sep 27, 2017

Contents

1 Quick Start	3
2 Contents	5
2.1 API	5
3 Indices and tables	7
Python Module Index	9

benchmark-harness is designed to make it easy to create simple suites of standalone benchmarks while avoiding some common pitfalls in benchmarking. In particular, benchmarks are always run for a specified duration to avoid reporting anomalies due to background system activity, startup costs, garbage collection or JIT activity, etc.

CHAPTER 1

Quick Start

A simple benchmark looks like this:

```
1 from benchmark_harness import run_benchmark
2
3
4 def fib(n):
5     if n == 0:
6         return 0
7     elif n == 1:
8         return 1
9     else:
10        return fib(n - 1) + fib(n - 2)
11
12
13 def benchmark():
14     """fib!"""
15     fib(20)
16
17
18 run_benchmark(benchmark, meta={"title": "Everyone loves fib()})
```

This script can be run directly:

```
$ python benchmarks/fib/benchmark.py
fib: completed 67 trials
Min: 0.007
Max: 0.010
```

Output can be redirected to get a full JSON record:

```
$ python tests/fib/benchmark.py | python -m json.tool
{
  "meta": {
    "title": "Everyone loves fib()"
  },

```

```
"times": [  
    0.00791311264038086,  
    ...  
]  
}
```

benchmark-harness installs the command-line benchmark-harness utility which makes it easy to run many benchmarks if you organize them into a directory containing one directory per benchmark with a benchmark.py file. If the above file were saved to benchmarks/fib/benchmark.py, a sample run would look like this:

```
$ benchmark-harness --benchmark-dir=benchmarks/  
fib: completed 59 trials  
    Min: 0.008  
    Max: 0.010
```

API

`benchmark_harness.runners.run_benchmark` (*args, **kwargs)

Run a benchmark a few times and report the results.

Arguments:

benchmark The benchmark callable. `run_benchmark` will time the execution of this function and report those times back to the harness. However, if `benchmark` returns a value, that result will be reported instead of the raw timing.

setup A function to be called before running the benchmark function(s).

max_time The number of seconds to run the benchmark function. If not given and if `handle_argv` is `True` this'll be automatically determined from the `--max_time` flag.

handle_argv `True` if the script should handle `sys.argv` and configure itself from command-line arguments

meta Key/value pairs to be returned as part of the benchmark results.

`benchmark_harness.runners.run_comparison_benchmark` (*args, **kwargs)

Benchmark the difference between two functions.

Arguments are as for `run_benchmark`, except that this takes 2 benchmark functions, an A and a B, and reports the difference between them.

For example, you could use this to test the overhead of an ORM query versus a raw SQL query – pass the ORM query as `benchmark_a` and the raw query as `benchmark_b` and this function will report the difference in time between them.

For best results, the A function should be the more expensive one (otherwise `djangobench` will report results like “-1.2x slower”, which is just confusing).

`benchmark_harness.suite.discover_benchmarks` (base_dir)

`benchmark_harness.suite.run_benchmark` (*benchmark*, *env=None*, *max_time=None*,
python_executable=None, *stderr=None*)

`benchmark_harness.suite.run_benchmarks` (*benchmarks*, *max_time=None*, *output_dir=None*,
includes=None, *excludes=None*, *con-*
tinue_on_error=False, *python_executable=None*,
env=None)

`benchmark_harness.utils.format_output` (*f*)

Allow functions to return normal Python data structure

If stdout is a tty, basic stats and a human-meaningful result will be displayed. If not, JSON will be returned for a script to process

CHAPTER 3

Indices and tables

- `genindex`
- `modindex`
- `search`

b

benchmark_harness, 5
benchmark_harness.runners, 5
benchmark_harness.suite, 5
benchmark_harness.utils, 6

B

benchmark_harness (module), 5
benchmark_harness.runners (module), 5
benchmark_harness.suite (module), 5
benchmark_harness.utils (module), 6

D

discover_benchmarks() (in module benchmark_harness.suite), 5

F

format_output() (in module benchmark_harness.utils), 6

R

run_benchmark() (in module benchmark_harness.runners), 5
run_benchmark() (in module benchmark_harness.suite), 5
run_benchmarks() (in module benchmark_harness.suite), 6
run_comparison_benchmark() (in module benchmark_harness.runners), 5