

---

# **charm-benchmark Documentation**

***Release 1.1.4***

**Adam Israel**

**Mar 27, 2017**



---

## Contents

---

<b>1</b>	<b>Getting Started</b>	<b>3</b>
1.1	Installing Charm Benchmark . . . . .	3
1.2	Using charm-benchmark . . . . .	3
<b>2</b>	<b>Contributing</b>	<b>5</b>
2.1	Source . . . . .	5
2.2	Submitting a Pull Request . . . . .	5
2.3	Open Bugs . . . . .	5
2.4	Documentation . . . . .	5
2.5	Getting Help . . . . .	6
<b>3</b>	<b>API Documentation</b>	<b>7</b>
<b>4</b>	<b>Indices and tables</b>	<b>9</b>



Contents:



Benchmarks are [Juju Actions](#) that follow a specific pattern. While they can be accomplished via `action_set`, we've created `charm-benchmark` to make that task easier.

## Installing Charm Benchmark

```
pip install charm-benchmark
```

## Using charm-benchmark

There are two ways of using `charm-benchmark`: via direct calls to the Python library...:

```
#!/usr/bin/env python
import subprocess

try:
    from charmhelpers.contrib.benchmark import Benchmark
except ImportError:
    subprocess.check_call(['apt-get', 'install', '-y', 'python-pip'])
    subprocess.check_call(['pip', 'install', 'charm-benchmark'])
    from charmbenchmark import Benchmark

def main():

    Benchmark.start()

    # Get your parameters via action_get and run your benchmark

    # Parse the benchmark results
    transactions = 1096
```

```
transaction_rate = 10.51
transferred = 346.08
response_time = 0.92

Benchmark.set_data({'results.transactions.value': transactions})
Benchmark.set_data({'results.transactions.units': 'hits'})

Benchmark.set_data({'results.transaction_rate.value': transactions})
Benchmark.set_data({'results.transaction_rate.units': 'hits/sec'})

Benchmark.set_data({'results.transferred.value': transferred})
Benchmark.set_data({'results.transferred.units': 'MB'})

Benchmark.set_data({'results.response_time.value': response_time})
Benchmark.set_data({'results.response_time.units': 'secs'})

# Set the composite, which is the single most important score
Benchmark.set_composite_score(
    transaction_rate,
    'hits/sec',
    'desc'
)

Benchmark.finish()

if __name__ == "__main__":
    main()
```

...or via cli commands, which can be called from bash or any other language:

```
#!/bin/bash
set -eux

# Make sure charm-benchmark is installed
if ! hash benchmark-start 2>/dev/null; then
    apt-get install -y python-pip
    pip install -U charm-benchmark
fi
benchmark-start

# Run your benchmark

# Grep/awk/parse the results

benchmark-data 'transactions' 1096 'hits'
benchmark-data 'transaction_rate' 10.51 'hits/sec'
benchmark-data 'transferred' 346.08 'MB'
benchmark-data 'response_time' 0.92 'ms'

# Set the composite, which is the single most important score
benchmark-composite 'transaction_rate' 10.51 'hits/sec' 'desc'

benchmark-finish || true
```



## CHAPTER 2

---

### Contributing

---

All contributions, both code and documentation, are welcome!

### Source

The source code is located at <https://github.com/juju-solutions/charm-benchmark>. To submit contributions you'll need to create a Github account if you do not already have one.

### Submitting a Pull Request

[Fork](#) the charm-benchmark repo to your personal space and make your changes. When you're ready, and `make test` both covers the new features added and passes on them cleanly, you can send a [pull request](#).

### Open Bugs

If you're looking for something to work on, the open bug/feature list can be found at <https://github.com/juju-solutions/charm-benchmark/issues>.

### Documentation

If you'd like to contribute to the documentation, please refer to the `HACKING` document in the root of the source tree for instructions on building the documentation.

## Getting Help

If you need help you can find it in #juju on the Freenode IRC network. Come talk to us - we're a friendly bunch!

## CHAPTER 3

---

### API Documentation

---



## CHAPTER 4

---

### Indices and tables

---

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