

Load testing with locust

Installation

```
$ pip install locustio
```

locustfile.py

```
from locust import HttpLocust, TaskSet, task

class UserBehavior(TaskSet):

    @task
    def get_something(self):
        self.client.get("/something")

class WebsiteUser(HttpLocust):
    task_set = UserBehavior
```

locustfile.py

```
class UserBehavior(TaskSet):  
  
    @task  
    def get_something(self):  
        self.client.get("/something")  
  
    @task  
    def get_something_else(self):  
        self.client.get("/something-else")
```

locustfile.py

```
class UserBehavior(TaskSet):  
  
    @task(2)  
    def get_something(self):  
        self.client.get("/something")  
  
    @task(1)  
    def get_something_else(self):  
        self.client.get("/something-else")
```

locustfile.py

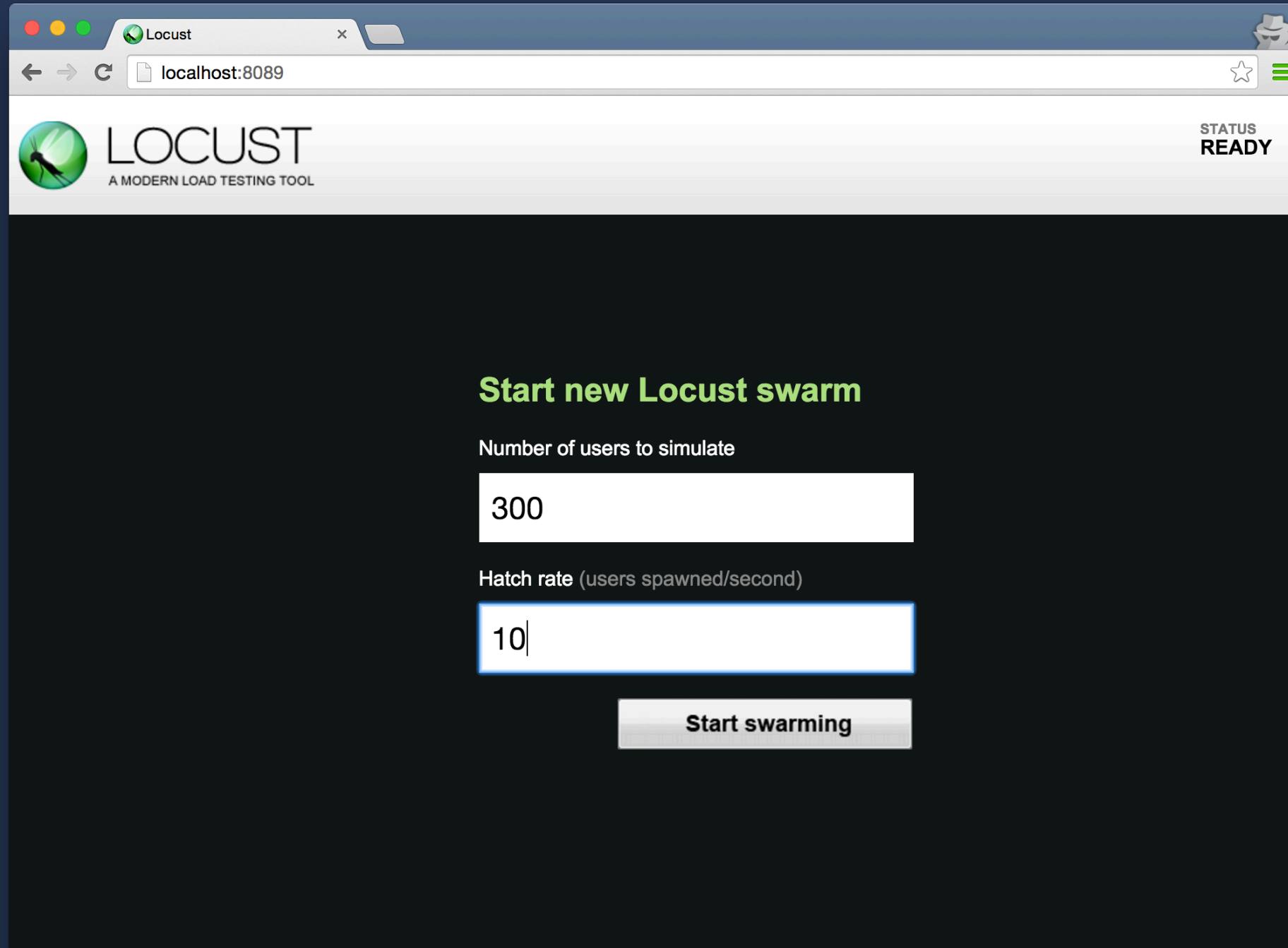
```
class UserBehavior(TaskSet):  
  
    @task  
    def get_something(self):  
        self.client.get("/something")  
  
    @task  
    def get_something_else(self):  
        self.client.get("/something-else")
```

locustfile.py

```
class UserBehavior(TaskSet):  
  
    def on_start(self):  
        self.client.post("/login", {  
            'username': 'foo', 'password': 'bar'  
        })  
  
    @task  
    def get_something(self):  
        self.client.get("/something")
```


Running

```
$ locust --host=http://localhost:5000
```

Locust

localhost:8089

 **LOCUST**
A MODERN LOAD TESTING TOOL

STATUS

HATCHING

189 users

Edit

RPS

27.2

FAILURES

11%

STOP

[Reset Stats](#)

Statistics

Failures

Exceptions

Download Data

Type	Name	# requests	# fails	Median	Average	Min	Max	Content Size	# reqs/sec
GET	/	189	60	4	5	3	12	12	10.7
POST	/login	189	0	4	5	3	23	6	9.7
GET	/profile	118	0	4	5	3	17	6	6.8
	Total	496	60	4	5	3	23	8	27.2

Running (distributed)

```
$ locust --host=http://localhost:5000 --master
```

```
$ locust --host=http://localhost:5000 --slave\  
--master-host=192.168.10.100
```

Downsides

- Stats are pretty bad;
- Hard to get details about error responses;
- Non-trivial to do non-HTTP or non-RESTful requests.

Questions?