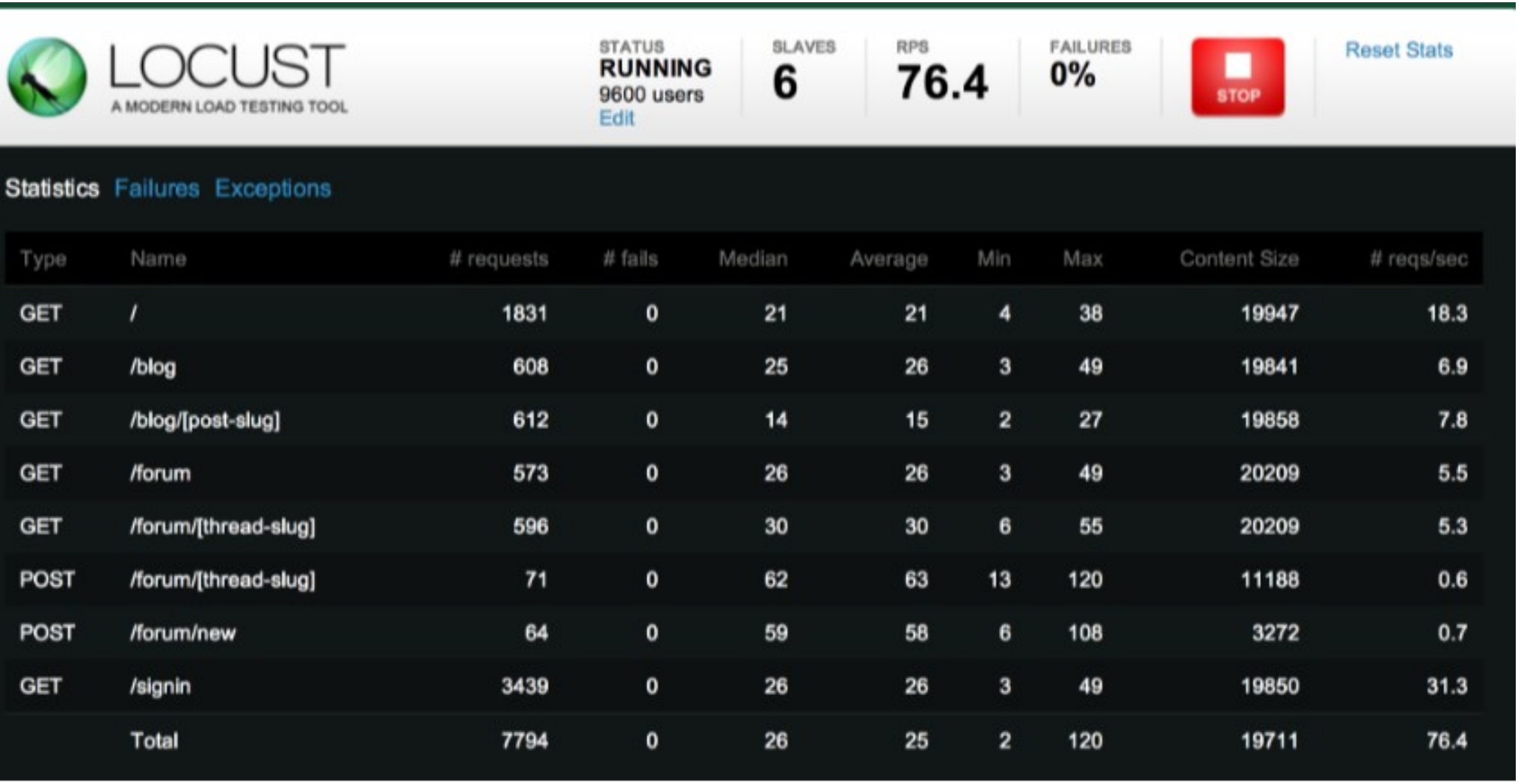
An open source load testing tool.

```
> pip install locustio
```

locustfile.py

```python
from locust import HttpLocust, TaskSet, task

class WebsiteTasks(TaskSet):
    def on_start(self):
        self.client.post("/login", {
            "username": "test_user",
            "password": ""
        })

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

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

class WebsiteUser(HttpLocust):
    task_set = WebsiteTasks
    min_wait = 5000
    max_wait = 15000
```

```
$ locust -f locustfile.py
```



Type	Name	# requests	# fails	Median	Average	Min	Max	Content Size	# reqs/sec
GET	/	1831	0	21	21	4	38	19947	18.3
GET	/blog	608	0	25	26	3	49	19841	6.9
GET	/blog/[post-slug]	612	0	14	15	2	27	19858	7.8
GET	/forum	573	0	26	26	3	49	20209	5.5
GET	/forum/[thread-slug]	596	0	30	30	6	55	20209	5.3
POST	/forum/[thread-slug]	71	0	62	63	13	120	11188	0.6
POST	/forum/new	64	0	59	58	6	108	3272	0.7
GET	/signin	3439	0	26	26	3	49	19850	31.3
	Total	7794	0	26	25	2	120	19711	76.4

```python
import random
from locust import HttpLocust, TaskSet, task
from pyquery import PyQuery


class BrowseDocumentation(TaskSet):
    def on_start(self):
        # assume all users arrive at the index page
        self.index_page()
        self.urls_on_current_page = self.toc_urls

    @task(10)
    def index_page(self):
        r = self.client.get("/")
        pq = PyQuery(r.content)
        link_elements = pq(".toctree-wrapper a.internal")
        self.toc_urls = [
            l.attrib["href"] for l in link_elements
        ]

    @task(50)
    def load_page(self, url=None):
        url = random.choice(self.toc_urls)
        r = self.client.get(url)
        pq = PyQuery(r.content)
        link_elements = pq("a.internal")
        self.urls_on_current_page = [
            l.attrib["href"] for l in link_elements
        ]

    @task(30)
    def load_sub_page(self):
        url = random.choice(self.urls_on_current_page)
        r = self.client.get(url)
```

```
class AwesomeUser(HttpLocust):
    task_set = BrowseDocumentation
    host = "http://docs.locust.io/en/latest/"

    # we assume someone who is browsing the Locust docs,
    # generally has a quite long waiting time (between
    # 20 and 600 seconds), since there's a bunch of text
    # on each page
    min_wait = 20  * 1000
    max_wait = 600 * 1000
```

Summary

- Define user behaviour using
  - taskset and task
  - Provide weight to task
  - max_wait and min_wait
- Use command line to automate the testing
- Tests can be run in distributed mode