Siege

Install on MAC

https://jasonmccreary.me/articles/installing-siege-mac-os-x-lion/

******* Install Steps *******
curl -C - -O http://download.joedog.org/siege/siege-latest.tar.gz
tar -xvf siege-latest.tar.gz
cd siege-2.70/
./configure
make
make install

$ siege

sends a 10 requests across 10 concurrent connections for benchmarking
$ siege -c 10 -r 10 -b /

Siege a web server with 10 concurrent connections for 10 seconds:
$ siege -c 10 -b -t 10S http://example.com/

multi-mechanize Pythonpackage

1) Install Multi-Mechanize:

```
pip install multi-mechanize
```

2) Bootstrapping a new multi mechanize project is easy:

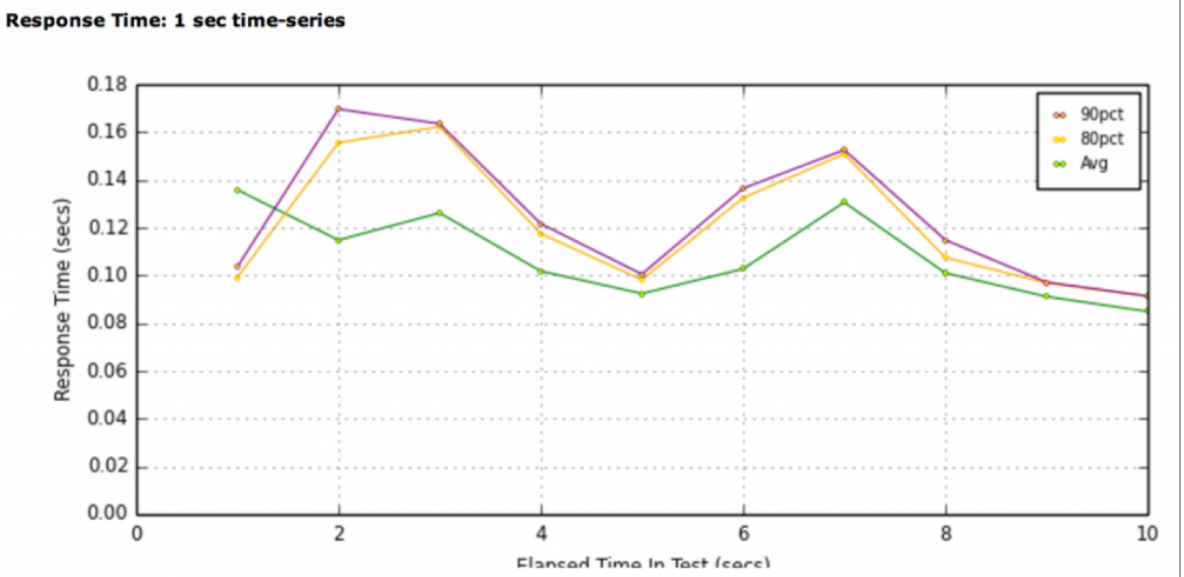
```
multimech-newproject demo
```

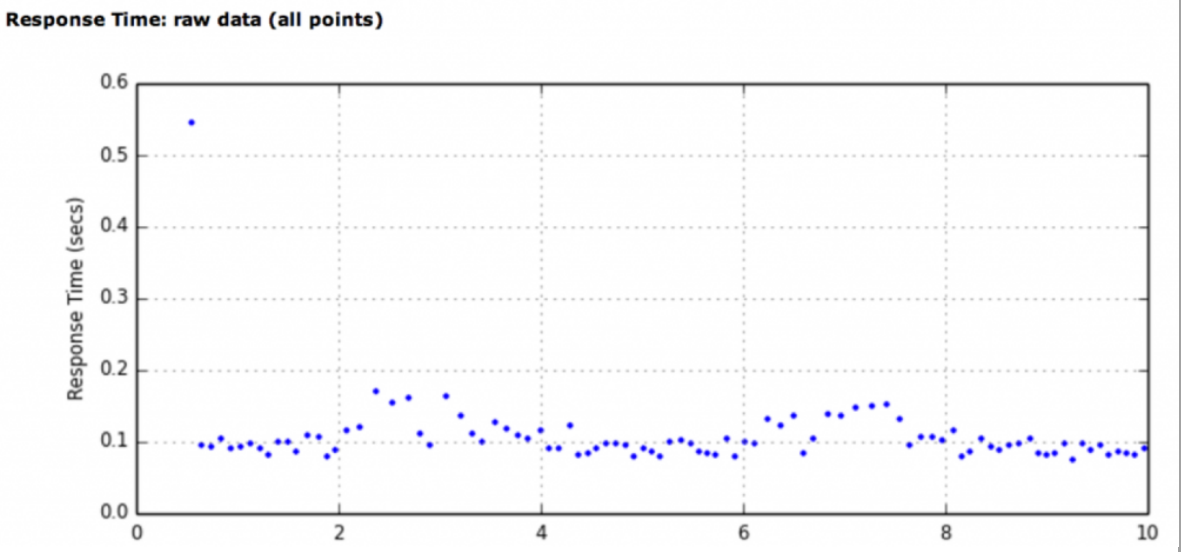
```
import mechanize
import time
class Transaction(object):
    def run(self):
        br = mechanize.Browser()
        br.set_handle_robots(False)

start_timer = time.time()
        resp = br.open('http://www.example.com/')
        resp.read()
        latency = time.time() - start_timer
```

```
self.custom_timers['homepage'] = latency

assert (resp.code == 200)
        assert ('Example' in resp.get_data())
```

3) Run the multi-mechanize project and review the outputted reports

```
multimech-run demo
```





Bees With Machineguns

1) Install Bees with Machine Guns:

```
pip install beeswithmachineguns
```

2) Configure Amazon Web Services credentials in ~/.boto:

```
[Credentials]
aws_access_key_id=xxx
aws_secret_access_key=xxx
[Boto]
ec2_region_name = us-west-2
ec2_region_endpoint = ec2.us-west-2.amazonaws.com
```

3) Create 2 EC2 instances using the default security group in the us-west-2b availabily zone using the ami-bc05898c image and login using the ec2-user user name.

```
bees up -s 2 -g default -z us-west-2b -i ami-bc05898c -k aws-us-west-2 -l ec2-user

Connecting to the hive.
Attempting to call up 2 bees.
Waiting for bees to load their machine guns...
.
.
.
.
Bee i-3828400c is ready for the attack.
Bee i-3928400d is ready for the attack.
The swarm has assembled 2 bees.
```

4) Check if the ec2 instances are ready for battle

```
bees report

Read 2 bees from the roster.
Bee i-3828400c: running @ 54.212.22.176
Bee i-3928400d: running @ 50.112.6.191
```

5) Attack a url if the ec2 instances are ready for battle

```
bees attack -n 100000 -c 1000 -u http://example.com/

Read 2 bees from the roster.
Connecting to the hive.
Assembling bees.
```

```
Each of 2 bees will fire 50000 rounds, 125 at a time.
Stinging URL so it will be cached for the attack.
Organizing the swarm.
Bee 0 is joining the swarm.
Bee 1 is joining the swarm.
Bee 0 is firing his machine gun. Bang bang!
Bee 1 is firing his machine gun. Bang bang!
Bee 1 is out of ammo.
Bee 0 is out of ammo.
Offensive complete.
     Complete requests:   100000
     Requests per second: 1067.110000 [#/sec] (mean)
     Time per request:    278.348000 [ms] (mean)
     50% response time:   47.500000 [ms] (mean)
     90% response time:   114.000000 [ms] (mean)
Mission Assessment: Target crushed bee offensive.
The swarm is awaiting new orders.
```

6) Spin down all the EC2 instances

```
bees down
```