Load Testing with Locust.io

Robert Jerovšek

robert.jerovsek@gmail.com

Locust.io

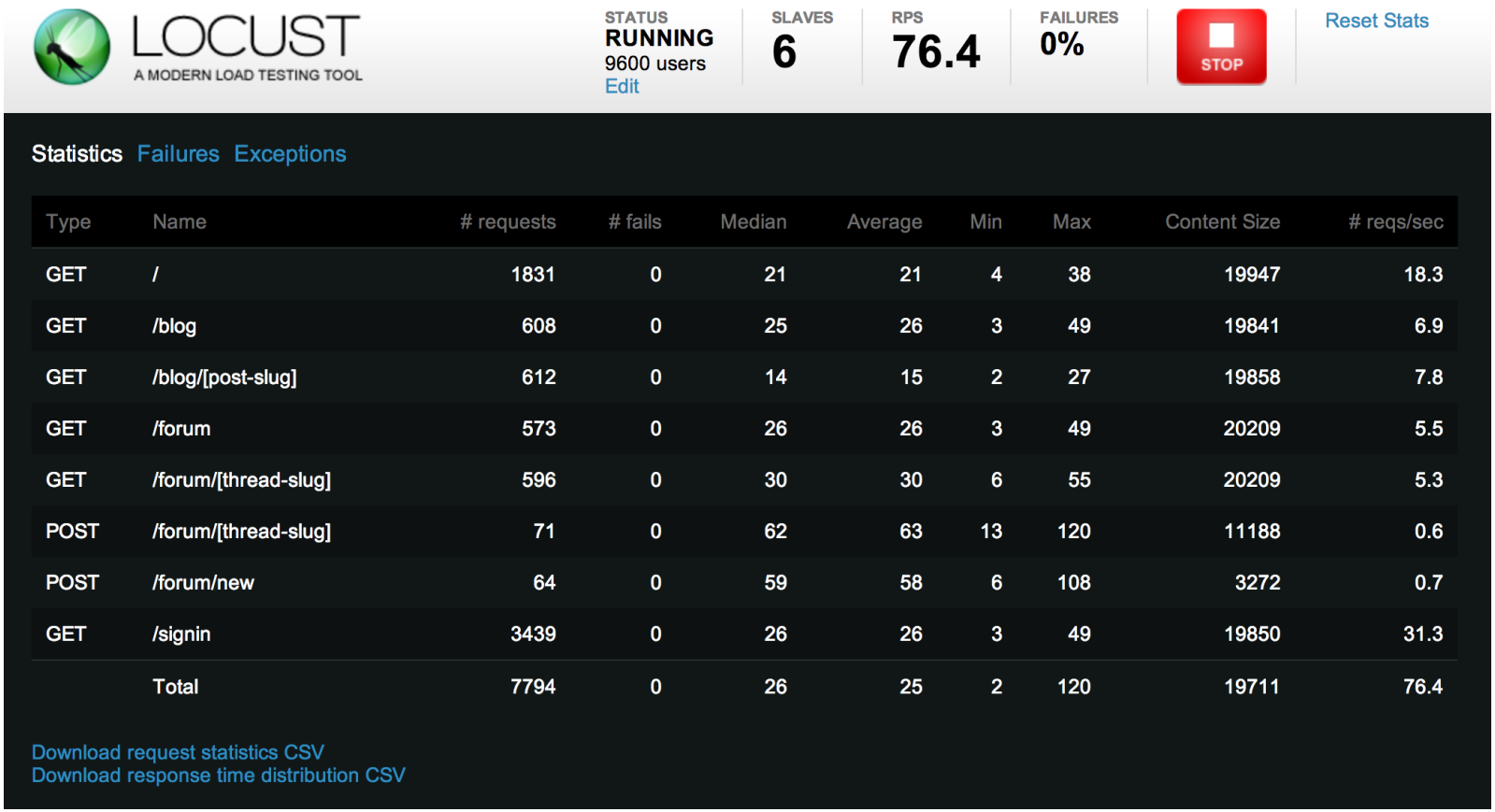
- Open Source Python tool for Load Testing
- Simple
- Scalable

```
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
```

Locust.io

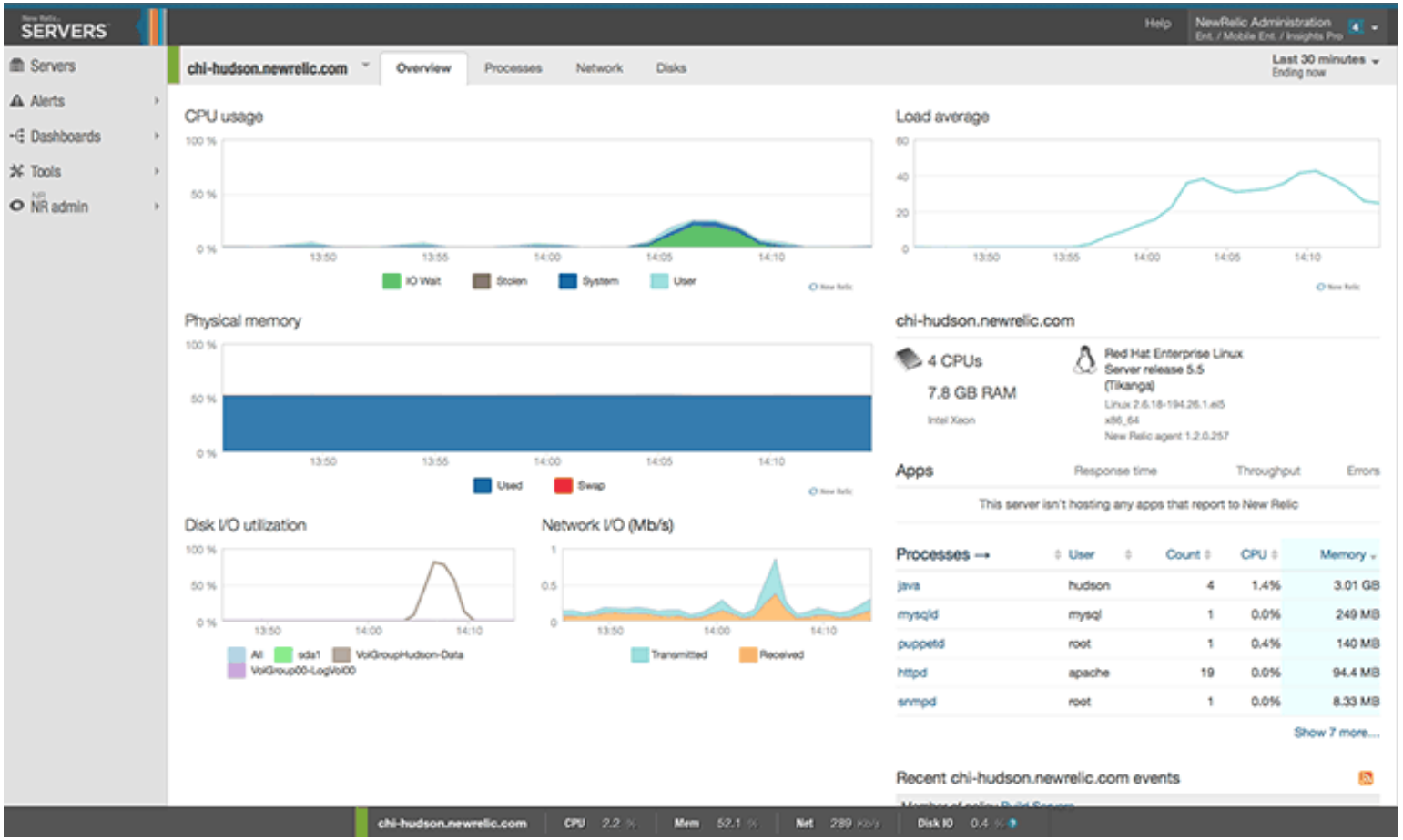
- Python 2.6+
- Python 3.x WIP (due to reliance on Gevent)
- No graphs out of the box (does only one thing well :))



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

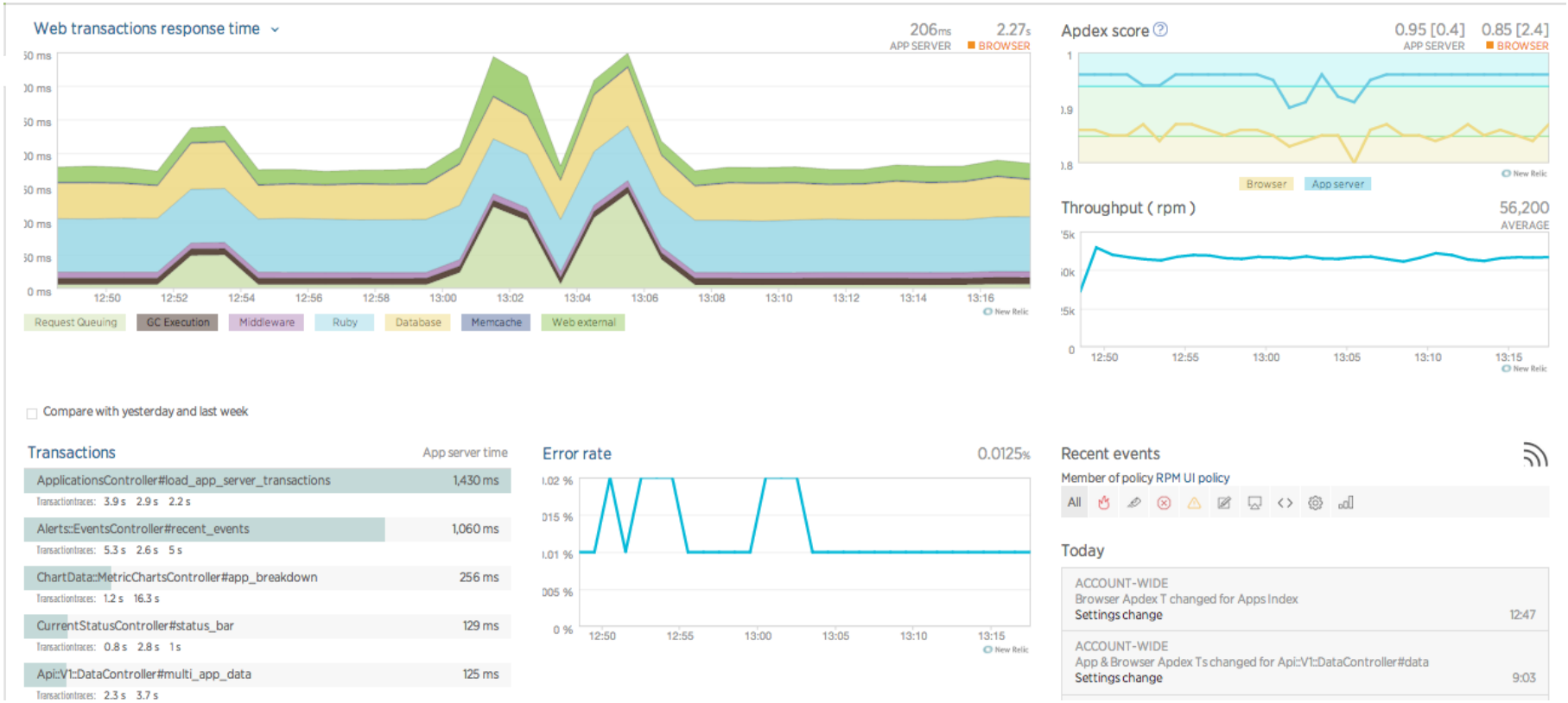
New Relic

- Monitor performance and load
- See bottlenecks
- Detect regressions



New Relic

- Everything in “the cloud”
- No local version
- Delay in data



Code!

https://github.com/robertjerovsek/pycon-kr-2015

Reasons - Capacity planning

- Can we handle 2/4/10x the traffic/users/orders?
- What do we need to handle that?
- When no “autoscaling” available

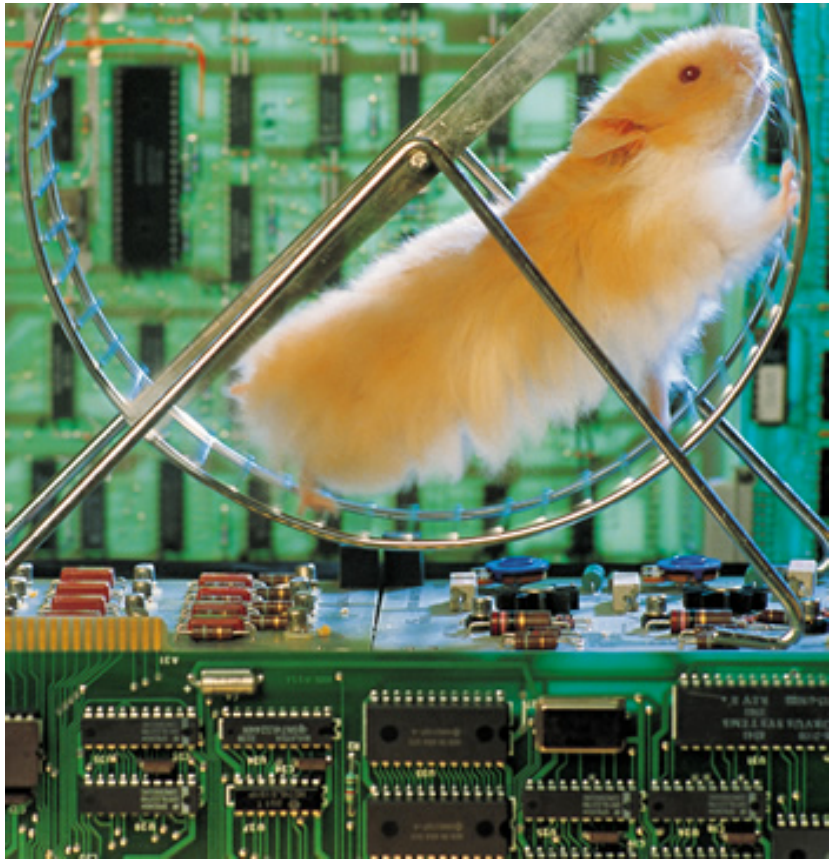
Reasons - Bottlenecks

- Connection limits
- Locking issues
- CPU, IO, Memory
- Network congestion

Reasons

- Reducing operational costs (wasted resources)
- Better performance
- Peaceful mind (no alarms)

Reasons - Happy hamsters :)



Solutions - More servers

- Frontend servers
- Database slaves
- Workers

Solutions

- Caching
  - old data visible to the user = bad UX
- Reducing connections to the database, external services
- Pre-calculating results
- Background tasks
  - slow change propagation = bad UX
- Distribute (slaves, sharding, …)

When to load test?

- Always :)
- Before and after releases
- Simple with Locust

Thank you