



# locust-性能测试工具

---

# 目录



介绍与安装



脚本编写



运行模式



扩展

# 性能自动化测试工具-Locust

## Locust介绍

概念, 基本特征, 背景的介绍

## 编写一个Locustfile

Locust类, TaskSet类, 发送HTTP请求

## 结果分析

测试结果分析, 参数的意义

## Locust安装

Windows安装, Linux安装, Mac安装, 支持Python的版本, 启动示例

## 测试及分布式

具体测试步骤, 单个, 分布式启动

## Locust扩展

支持的语言, 添加web路由

	loadrunner	jmeter	locust	wrk
分布式压力	支持	支持	支持	不支持
单机并发能力	低	低	高	低
并发机制	进程/线程	线程	协程	线程
开发语言	C/Java	Java	Python	C
报告与分析	完善	简单图标	简单图表	简单结果
授权方式	商业收费	开源免费	开源免费	开源免费
测试脚本形式	C/Java	GUI	Python	C
资源监控	支持	不支持	不支持	不支持

# Locust介绍

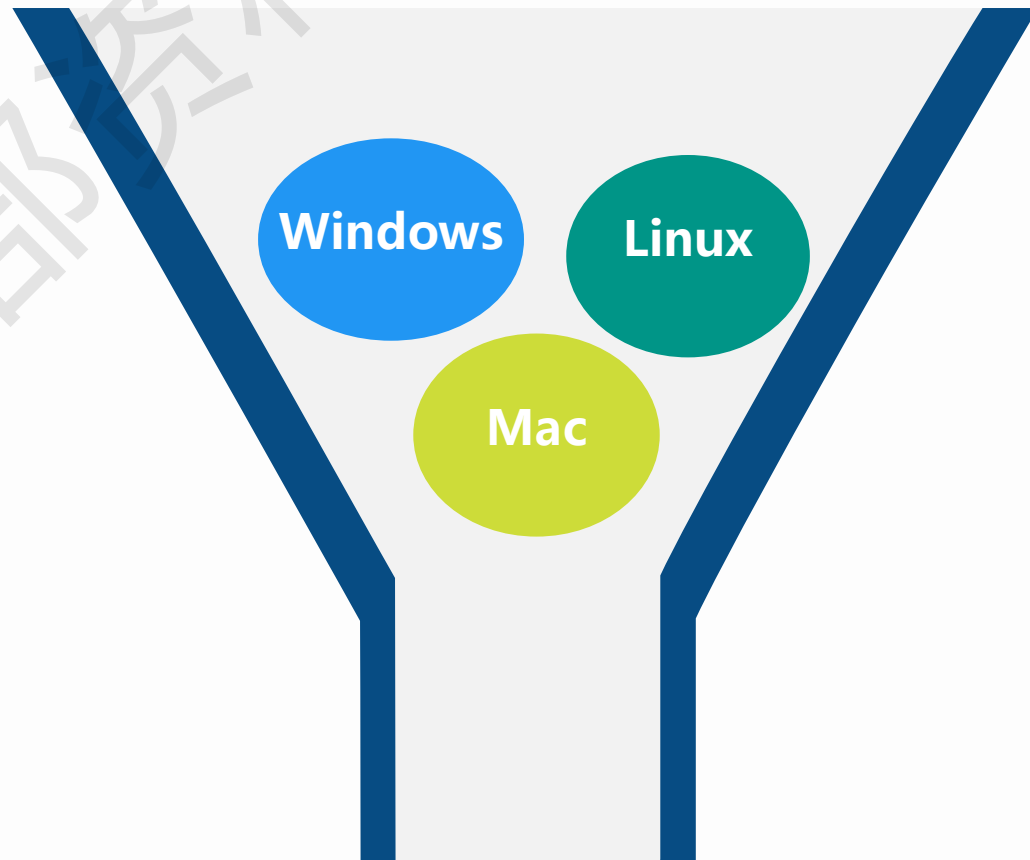
**简介：** Locust是一款易于使用的分布式用户负载测试工具。它用于对网站（或其他系统）进行负载测试，并确定系统可以处理多少并发用户。

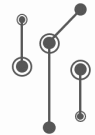
Locust完全基于事件，因此可以在一台机器上支持数千个并发用户。与许多其他基于事件的应用程序相比，它不使用回调。相反，它通过gevent使用轻量级过程。每个蝗虫蜂拥到你的网站实际上是在自己的进程内运行（或者是greenlet，这是正确的）。这允许您在Python中编写非常富有表现力的场景，而不会使代码与回调复杂化。

**特点：** 可扩展的，分布式的，性能测试的，开源的，用Python编写的性能测试工具

# Locust安装

- Windows安装： 安装python  
进入python安装目录  
`python3 -m pip install locustio`
- Linux安装： 安装python  
进入python安装目录  
`pip install locustio`
- Mac安装： 安装Homebrew。  
安装libev  
安装python  
进入python安装目录  
`pip install locustio`





# Locust各个参数的意义



Start new Locust swarm

Close

Number of total users to simulate

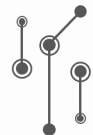
Hatch rate (users spawned/second)

Host

Number of users to simulate: 需要产生总的用户数

Hatch rate(users spawned/second): 每秒产生的用户数

start swarming: 执行

[Statistics](#) [Charts](#) [Failures](#) [Exceptions](#) [Download Data](#) [Slaves](#)

Type	Name	# Requests	# Fails	Median (ms)	90%ile (ms)	Average (ms)	Min (ms)	Max (ms)	Average size (bytes)	Current RPS	Current Failures/s
GET	aya-1588751546941092	346	0	3	4	4	3	14	0	4.6	0
PUT	a-1588766065724639	1	0	321	320	321	321	321	0	0	0
PUT	1588766095651373	1	0	26	26	26	26	26	0	0	0
PUT	158876610034240	1	0	24	24	24	24	24	0	0	0

### Statistics:

Type: 请求类型;

Name: 请求路径;

requests: 当前请求的数量;

fails: 当前请求失败的数量;

Median: 中间值, 单位毫秒, 一般服务器响应时间低于该值, 而另一半高于该值;

Average: 所有请求的平均响应时间, 毫秒;

Min: 请求的最小的服务器响应时间, 毫秒;

Max: 请求的最大服务器响应时间, 毫秒;

Content Size: 单个请求的大小, 单位字节;

reqs/sec: 每秒钟请求的个数。

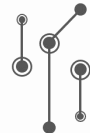
### Charts:

吞吐量/每秒响应事务数 (rps) 实时统计

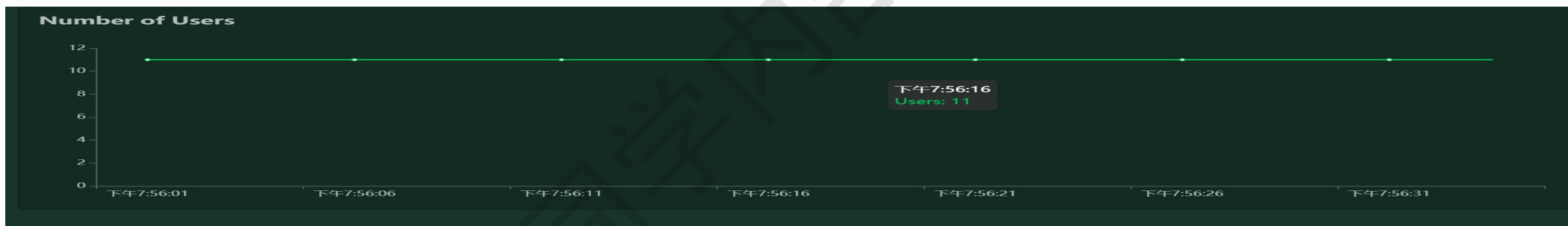
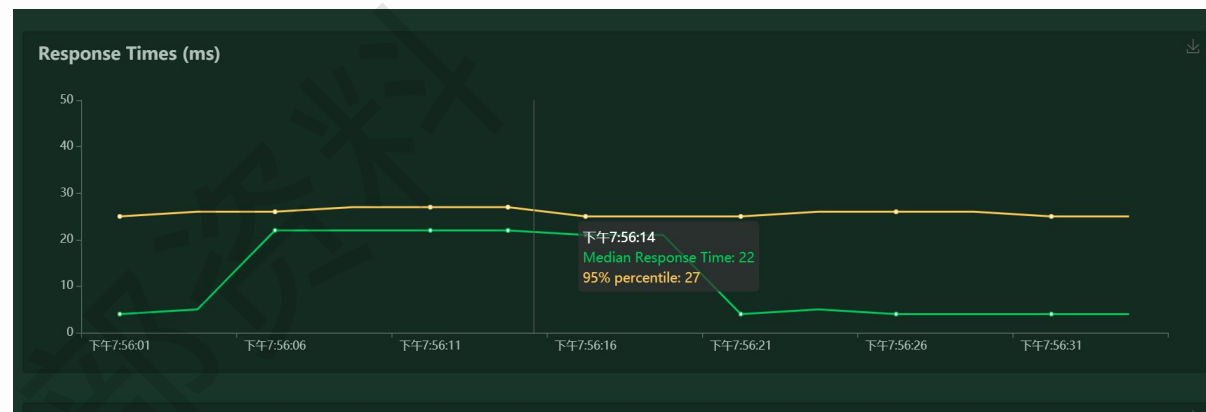
平均响应时间/平均事务数实时统计

虚拟用户数运行





# Locust各个参数的意义



## Charts:

吞吐量/每秒响应事务数 (rps) 实时统计  
平均响应时间/平均事务数实时统计  
虚拟用户数运行

# Locust扩展

01

添加web路由

Locust使用Flask来提供Web UI, 因此很容易将Web端点添加到Web UI。只需在locustfile中导入Flask应用程序并设置新路径

```
(  
from locust import web  
  
@web.app.route("/added_page")  
def my_added_page():  
    return "Another page"  
)
```

02

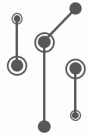
支持不同协议测试

HTTP  
HTTPS  
TCP  
UDP  
WebSocket  
...

03

支持多种语言

Golang  
Java  
python  
...



## 官网地址

- <https://docs.locust.io/en/stable/>



码同学内部资料



```
1 |
2 from locust import HttpLocust, TaskSet, task, between
3
4
5 class UserBehavior(TaskSet):
6
7     @task(1)
8     def index(self):
9         self.client.get("http://www.testfan.cn/")
10
11     @task(2)
12     def search(self):
13         self.client.get("http://www.testfan.cn/list/2/220.htm")
14
15
16 class WebsiteUser(HttpLocust):
17     task_set = UserBehavior
18     wait_time = between(1, 3)
19
```

被测域名

用户总数

每秒事物

运行时间

开始压测



建议smtp的服务

```
smtp = smtplib.SMTP()
smtp.connect('smtp.163.com')
username = 'mtx_testfan@163.com'
password = 'KQWHOUZDCUDHPXZY'
# 登录邮箱
smtp.login(username,password)
smtp.sendmail(sender,receiver,message.as_string())
print(f'email send {receiver} success')
```

发送邮件

```
title = '天天测试平台报警邮件'
message =
MIMEText(template,'html','utf-8')
message['From']=sender
message['To']=receiver
```

```
message['Subject']=Header(title,'utf-8')
```

```
smtp_service(sender,receiver,message)
```



```
template = ""  
    <h3>报警信息</h3>  
    <label style='background-color:red'>请求次数:  
{requests}</label></br>  
    <label style='background-color:red'>平均响应时间:  
{averageResponseTime}</label></br>  
    <label style='background-color:red'>最小响应时间:  
{minResponseTime}</label></br>  
    <label style='background-color:red'>最大响应时间:  
{maxResponseTime}</label></br>
```

```
""
```



**psutil**是一个开源且跨平台的库，其提供了便利的函数用来获取系统的信息，比如CPU，内存，磁盘，网络等。此外，psutil还可以用来进行进程管理，包括判断进程是否存在、获取进程列表、获取进程详细信息等。而且psutil还提供了许多命令行工具提供的功能，包括：ps, top, lsof, netstat, ifconfig, who, df, kill, free, nice, ionice, iostat, iotop, uptime, pidof, tty, taskset, pmap。

psutil是一个跨平台的库，在官方网站上查到其支持如下操作系统。

Linux

Windows

OSX

FreeBSD

OpenBSD

NetBSD

Sun Solaris

AIX

Works with Python versions from 2.6 to 3.X.

psutil包含了异常、类、功能函数和常量，其中功能函数用来获取系统的信息，如CPU、磁盘、内存、网络等。类用来实现进程的管理功能  
根据函数的功能，主要分为CPU、磁盘、内存、网络几类