

# 10大主流压力测试工具

市面上流行的压力/负载/性能测试工具多是来自国外，近年来国内的性能测试工具也如雨后春笋崛起。同时由于开发的目的和侧重点不同，其功能也有很大差异，下面就为您介绍简单介绍10款目前最常见的测试产品。

## 1、kylinTOP测试与监控平台（商用版）

kylinTOP测试与监控平台是一款B/S架构的跨平台的集性能测试、自动化测试、业务监控于一体的测试平台，它是深圳是奇林软件有限公司旗下的一款产品，该工具开放10个免费虚拟用户可供学习和使用。在易用性上较好，录制脚本支持最新版本的浏览器，对谷歌和火狐都支持非常好。对一些https的网站证书问题，都为用户自动处理好了，可以轻松录制。录制过程高效便捷这是其它性能工具无法比拟的。仿真能力上是目前业界做的最好的性能工具，可以做到完全仿真浏览器行为，也就是单用户的HTTP请求瀑布图可以和浏览器器完全一样。总之它是目前国内一款非常难得好用的性能测试工具，可以完全替代国外的同类产品。目前在军工领域、测评检测机构、国有企业、银行体系、大型企业有着广泛的应用。支持的协议较多，尤其在视频领域支持的协议非常多，具有独特的优势。官网地址：<http://www.70testing.com>



## 2、LoadRunner（商用版）

是一款C/S架构的商业版性能测试工具，在国内存在的时间较早，在国内在使用较广泛，知名度较高。该工具免费开放了50个虚拟用户，可供学习和使用。在国内的网站上有破解版本，但是到了最新的12版本，不再有破解版。破解版的仿真度较差，HTTP的瀑布图是按两个两个并发（与浏览器行为不一样），最新版本的仿真相对提高很多，对于HTTP静态请求相似度请求提高到80%，而动态请求就要差很多。由于loadRunner起步早，支持的协议很多，很多不常用的协议都支持，如电子邮件相关协议都支持。但对一些视频协议支持的不太好。loadrunner目前对浏览器最新版支持的不太好，最新的loadrunner12只支持ie11，对其它浏览器如谷歌、火狐等支持的较差。在脚本录制方面有些网站录制不了。价格上比较昂贵，一般小企业无力承担。

LoadRunner官网：<https://saas.hpe.com/zh-cn/software/loadrunner>

## 3、Apache JMeter（开源免费）

---

JMeter是一款开源免费的压测产品，最初被设计用于Web应用功能测试使用，如今JMeter被国内企业用于性能测试。对于WEB服务器（支持浏览器访问），不建议使用Jmeter,因为jmeter的线程组都是线性执行的，与浏览器相差很大，测试结果不具有参考性。对于纯接口的部分场景（对接口调用顺序无严格要求）测试可以使用，但是要注意使用技巧，才能达到理想结果。

官网链接：<http://jmeter.apache.org/>

## 4、NeoLoad（商用版）

---

NeoLoad是Neotys出品的一种负载和性能测试工具，可真实地模拟用户活动并监视基础架构运行状态，从而消除所有Web和移动应用程序中的瓶颈。NeoLoad通过使用无脚本GUI和一系列自动化功能，可让测试设计速度提高5-10倍，并将维护的脚本维持在原始设计时间的10%，同时帮助用户使用持续集成系统自动进行测试。

NeoLoad支持WebSocket、HTTP1/2、GWT、HTML5、AngularJS、Oracle Forms等技术协议，能够监控包括操作系统，应用服务器，Web服务器，数据库和网络设备在内的各种IT基础设施，同时可以通过Neotys云平台发起外部压力。

官网链接：<http://www.neotys.com/product/overview-neoload.html>

## 5、WebLOAD（商用版）

---

WebLOAD是来自Radview公司的负载测试工具，它可被用以测试系统性能和弹性，也可被用于正确性验证（验证返回结果的正确性）。其测试脚本是用Javascript（和集成的COM/Java对象）编写的，并支持多种协议，如Web（包括AJAX在内的REST/HTTP）、SOAP/XML及其他可从脚本调用的协议如FTP、SMTP等，因而可从所有层面对应用程序进行测试。

WebLOAD存在免费和专业两个版本，免费版本支持50个虚拟用户，专业版还提供更多的报告和协议供用户选择。WebLOAD通常用作QA团队的独立运行工具，在开发周期的验证阶段，被测系统（System Under Test, SUT）投入实用之前，在模拟环境中对被测系统进行测试。

官网链接：<http://www.radview.com/>

## 6、Loadster（商用版本）

---

是一款商用负载测试软件，用于测试高负载下网站、Web应用、Web服务的性能表现，支持Linux，Mac和Windows等运行环境。

Loadster能够对Web应用/服务的Cookies、线程、头文件、动态表格等元素发起测试，获得Web在压力下的性能、弹性、稳定性和可扩展性等方面的表现。

官网链接：<http://www.loadsterperformance.com/>

## 7、Loadstorm（商用版本）

---

Loadstorm是一款针对Web应用的云端负载测试工具，通过模拟海量点击来测试Web应用在大负载下的性能表现。由于采用了云资源，所以Loadstorm的测试成本非常低，用户可以在云端选择创建自己的测试计划，测试标准和测试场景。

Loadstorm最多可以生成多达50000个并发用户，通过数以千计的云服务器发起访问。使用Loadstorm不需要任何脚本知识，同时提供多样化的测试图表和报告模版，用于准确测量Web应用的各项性能指标，如错误率，平均响应时间和用户数量等。Loadstorm可以申请免费试用，但更多压力和功能需要开通高级帐户。

官网链接：<http://loadstorm.com/>

## 8、Load impact（免费使用）

---

是一个在线可以免费测试网站负载能力，它就可以满足你的基本要求，当然成为他的付费用户测试的项目将会更多。

Load impact是一款服务于DevOps的性能测试工具，支持各种平台的网站、Web应用、移动应用和API测试。Loadimpact可以帮助用户了解应用的最高在线用户访问量，通过模拟测试不同在线人数下网站的响应时间，估算出服务器的最大负载。

Load impact的使用非常简单，只需要输入网址进行测试，便可统计出加载网站的一些详细数据。包括整体加载和站内图片，javascript, CSS等代码载入。可以在右侧列表选择不同文件来同时对比最多三个对象的加载数据，并生成图表显示，方便网站设计者来分析。测试完成之后，网站还可以存储测试过的统计数据。

官网链接：<http://loadimpact.com/>

特别提醒：

- 1.由于此测试数据发生在国外，一般反映出的是国外用户访问您要测试的网站的速度参数，具有一定的参考性；
- 2.对于国内主机，并且主要客户来自于国内的用户来说请慎重将此数据作为依据。

## 9.locust(开源免费)

---

Locust 完全基本 Python 编程语言，采用 Pure Python 描述测试脚本，并且 HTTP 请求完全基于 Requests 库。除了 HTTP/HTTPS 协议，Locust 也可以测试其它协议的系统，只需要采用Python调用对应的库进行请求描述即可。但是需要手工编写脚本，有一定的难度

## 10.OpenSTA(开源免费)

---

OpenSTA是一个免费的、开放源代码的web性能测试工具，能录制功能非常强大的脚本过程，执行性能测试。例如虚拟多个不同的用户同时登陆被测试网站。其还能对录制的测试脚本进行,按指定的语法进行编辑。在录制完测试脚本后，可以对测试脚本进行编辑，以便进行特定的性能指标分析。其较为丰富的图形化测试结果大大提高了测试报告的可阅读性。OpenSTA 基于CORBA 的结构体系，它通过虚拟一个 proxy，使用其专用的脚本控制语言，记录通过proxy 的一切HTTP/S traffic。通过分析OpenSTA的性能指标收集器收集的各项性能指标，以及HTTP 数据，对系统的性能进行分析。

# 总结

商用性能工具在易用性（脚本生成）、并发模型、统计指标上要比开源免费软件要好很多，可以大大提高工作效率，降低使用难度，在统计指标上要丰富的多。

免费开源软件的优点就是免费，不用花钱。但使用难度大，统计指标少，在仿真能力上比较弱。

如何选择主要看使用者的经济能力和使用频度。如果是使用频繁，免费的软件实际使用成本比商用软件还要贵。

	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
测试脚本形式	C/Java	GUI	Python	C
报告与分析	完美	简单图表	简简单单图表	简单结果
授权方式	有社区版，完整版需要商业收费	开源免费	开源免费	开源免费
资源监控	支持	不支持	不支持	不支持

1. 安装(和一切python第三方包安装方式一致) \$ pip install locust
2. 编写一个Locust 文件(保存为 一个py文件，取任意名，比如basic.py)

```
from locust import HttpUser, TaskSet, task

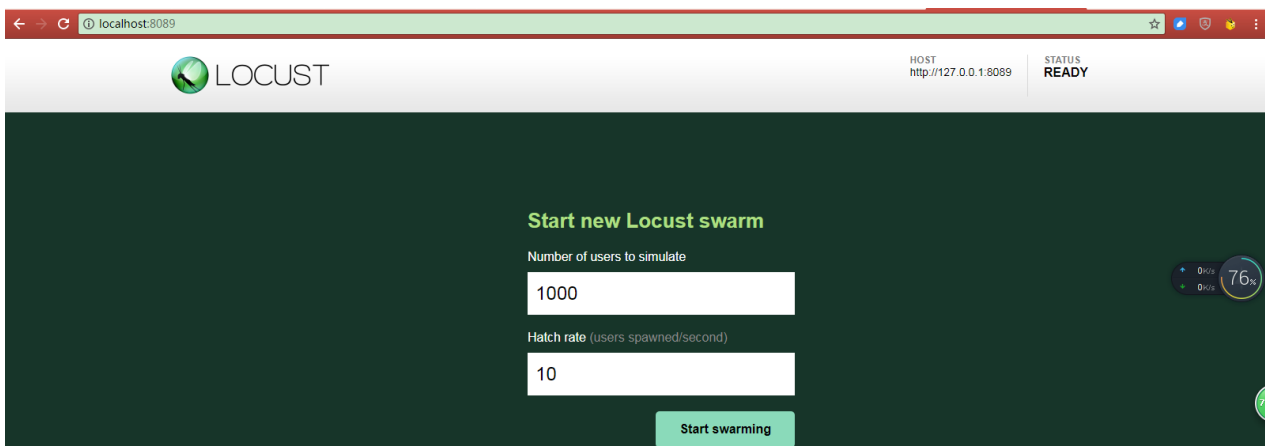
class WebsiteUser(HttpLocust):
    host = "http://127.0.0.1:8000"
    min_wait = 20
    max_wait = 50
    def index(self):
        self.client.get("/index")
```

1. 在存放文件的目录下以命令行方式运行以上py文件 \$locust -f basic.py 可见类似以下的画面：

```
D:\Python\Demo\LocustDemo>locust -f basic.py
[2017-09-28 21:16:38,913] Tracycheng-PC/INFO/locust.main: Starting web monitor a
t *:8089
[2017-09-28 21:16:38,913] Tracycheng-PC/INFO/locust.main: Starting Locust 0.8a4
```

basic.png

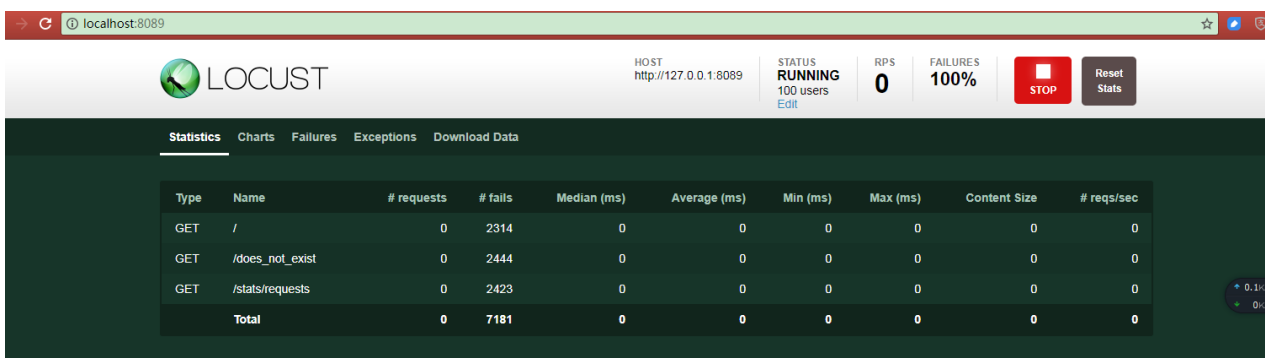
4.打开浏览器，输入IPV6地址：http://[::1]:8089,开启Locust Web操作页面



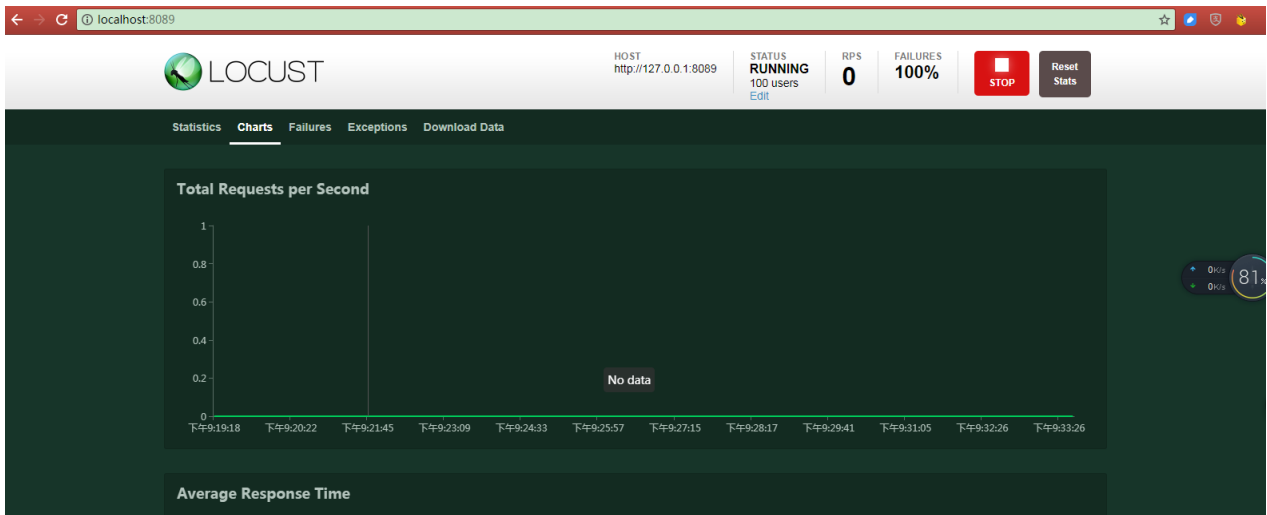
- 第一个输入框：想并发的人数
- 第二个输入框：虚拟用户初始化的比例 比如上图中的意思就是想测试1000个虚拟用户对系统的压测，刚开始的时候是以10人/秒的速度开始递增到1000人。

点击“Start Swarming”后你就可以开始压测你想压测的系统了。

1. 查看执行结果（上一步点击后页面会自动刷新到结果页面，但是需要手动停止）



也可以切换到Charts看图形化结果



目前只有每秒请求数，平均响应时间，用户的增长曲线 三个图可看。

当然，你也通过 Download Data链接下载刚才的结果。

再回到刚开始的问题，什么是Locust呢？

locust是一个易于使用的，分布式的，用户负载测试工具。用于web站点（或其他系统）的负载测试，然后算出系统能够处理多少并发用户。locust的思想是：在测试期间，一大群"蝗虫"会攻击你的网站，每一个"蝗虫"的行为都是由你自己定义的，同时，可以在一个web界面上实时的监控这群进程。这会帮助你更好的"进行战斗"，在真正的用户进入之前，就找出代码中的瓶颈。locust完全是事件驱动的，因此它能够在单机支持数以千计的并发用户，相比许多其他的基于事件的应用，locust不使用回调函数。它使用轻量进程---gevent。每一个访问你的网站的locust实际上都在它自己的进程内部运行（准确地说，是greenlet），也就是我们通常说的协程。这允许你在不使用带回调函数的复杂代码的情形下，使用python写出非常具有表现力的脚本。