

Locust压测

- 讲师: pansir

压测组件

压力生成器

结果采集

资源监控

Locust基础

多进程 + 协程 + requests

安装: `pip install locust`

```
import logging

logging.basicConfig(level=logging.INFO, format='%(asctime)-16s %(levelname)-8s
%(message)s')

from locust import HttpLocust, TaskSet, task

user = "dcs"
passwd = 123

class WebsiteTasks(TaskSet):
    def on_start(self):
        self.client.post("/login", {
            "username": user,
            "password": 123
        })

    @task(2)
    def index(self):
        self.client.get("/index")

    @task(1)
    def about(self):
        self.client.get("/user/{}/".format(user))

class WebsiteUser(HttpLocust):
```

```
task_set = WebsiteTasks
host = "http://www.captaintests.club/login"
min_wait = 1000
max_wait = 5000
```

执行流程: on_start -> task -> wait -> task -> end

子任务描述

- 装饰器描述

```
from locust import TaskSet, task

class UserBehavior(TaskSet):
    @task(1)
    def test_job1(self):
        self.client.get('/job1')

    @task(2)
    def test_job2(self):
        self.client.get('/job2')
```

- task属性描述

```
from locust import TaskSet

def test_job1(obj):
    obj.client.get('/job1')

def test_job2(obj):
    obj.client.get('/job2')

class UserBehavior(TaskSet):
    tasks = {test_job1:1, test_job2:2}
    # tasks = [(test_job1,1), (test_job1,2)] # 两种方式等价
```

运行模式

按运行方式可分: 单进程和分布式; 按展示方式划分, 有web和no_web模式

web模式

执行 `locust -f xxx.py` 后, 打开 `http://localhost:8089` 即可

no-web模式

参数说明: `--no_web` 指定启动方式

- `-h`: 压测url地址, 若终端指定, 需要在locust子类中指定
- `-c`: 指定并发数, 即web页面: `Number of users to simulate` 参数
- `-t`: 运行时间, 单位秒
- `-r`: 每秒启动用户数, 即web页面: `hatch rate` 参数
- `-L`: 日志级别, 默认INFO
- `-P -port`: 指定web端口, 默认8089

一般作调试使用: `locust -f xxx.py --no_web -c 1 -t 1`

单进程

单个locust进程, 不能发挥压力机处理器性能, 主要用于调试脚本和小并发测试

分布式

分布式压测方式也分两种: 单台多节点, 多台多节点

单台多节点

例如4核处理器, 就可以启动一个master, 4个slave, 也可以是4*N, 不过效果差不多

启动主节点: `locust -f xxx.py --master`

启动slave节点: `locust -f xxx.py --slave` 不需要指定端口

多台多节点

多台多节点: 同上, 只是slave节点更多

启动主节点: `locust -f xxx.py --master`

启动slave节点(同机器): `locust -f xxx.py --slave`

启动slave节点(不同机器): `locust -f xxx.py --slave --master-host={主节点的ip}`

结果展示

web界面提供4个重要的指标: 并发数, RPS(TPS), 响应时间, 错误率

RPS和响应时间, 都是最近2秒请求响应数据计算结果, 可以理解为瞬时值

- statistics: 统计数据
- charts: tps, 响应时间, 用户数
- Failures: 错误统计
- Eceptions: 统计异常
- Download Data: 统计结果

脚本增强

脚本增强四大点：关联，参数化，监测点，集合点

参数关联

有多种方式进行参数解析与关联

1. re正则匹配
2. lxml解析, `etree.HTML(html).xpath`
3. jqeury解析

参数化

- 随机取值
- 循环取值, 单个用户循环取值, 构造循环索引

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

logging.basicConfig(level=logging.INFO, format='%(asctime)-16s %(levelname)-8s
%(message)s')

class UserBehavior(TaskSet):
    def on_start(self):
        self.index = 0

    @task
    def test_visit(self):
        url = self.locust.share_data[self.index]
        logging.info("[will request url]: {}".format(url))
        self.index = (self.index + 1) % len(self.locust.share_data)
        self.client.get(url)

class WebsiteUser(HttpLocust):
    host = 'https://captaintests.club'
    task_set = UserBehavior
    share_data = ['url1', 'url2', 'url3', 'url4', 'url5']
    min_wait = 2000
    max_wait = 2000
```

- 一次性取值, 所有用户不重复, 所有用户, 共用1个队列数据

```
from locust import TaskSet, task, HttpLocust
import queue
import logging
```

```

logging.basicConfig(level=logging.INFO, format='%(asctime)-16s %(levelname)-8s
%(message)s')

class UserBehavior(TaskSet):

    @task
    def test_register(self):
        try:
            data = self.locust.user_data_queue.get()
        except queue.Empty:
            logging.info("[test data empty, exit send]")
            exit(0)
        logging.info("[register user]: {} {}".format(
            data.get("username"),
            data.get("password"),
        ))
        body = {
            'username': data.get("username"),
            'password': data.get("password"),
        }
        self.client.post('/register', data=body)

class WebsiteUser(HttpLocust):
    host = 'https://www.captaintests.club'
    task_set = UserBehavior
    user_data_queue = queue.Queue()
    for index in range(10):
        data = {
            "username": "test%04d" % index,
            "password": "pwd%04d" % index,
            "email": "test%04d@tsms.com" % index,
        }
        user_data_queue.put_nowait(data)
    min_wait = 1000
    max_wait = 1000

```

- 循环取值，循环可用，共用一个队列数据，每次取完放到队尾

```

from locust import TaskSet, task, HttpLocust
import queue
import logging

logging.basicConfig(level=logging.INFO, format='%(asctime)-16s %(levelname)-8s
%(message)s')

```

```

class UserBehavior(TaskSet):

    @task
    def test_register(self):
        try:
            data = self.locust.user_data_queue.get()
        except queue.Empty:
            logging.info("[test data empty, exit send]")
            exit(0)
        logging.info("[register user]: {} {}".format(
            data.get("username"),
            data.get("password"),
        ))
        body = {
            'username': data.get("username"),
            'password': data.get("password"),
        }
        self.client.post('/register', data=body)
        self.locust.user_data_queue.put(body)


class WebsiteUser(HttpLocust):
    host = 'https://www.captaintests.club'
    task_set = UserBehavior
    user_data_queue = queue.Queue()
    for index in range(10):
        data = {
            "username": "test%04d" % index,
            "password": "pwd%04d" % index,
            "email": "test%04d@tsms.com" % index,
        }
        user_data_queue.put_nowait(data)
    min_wait = 1000
    max_wait = 1000

```