Galing压力测试

大家接触过形形色色的压力测试工具，例如lr,jmeter各有各的优点，那么最近在做接口测试中涉及到压力测试，小弟就看到一个好用的工具俗称“加特林”英文Gatling,就简单研究一番，今天为大家分享一下。

Gatling是一款基于Scala 开发的高性能服务器性能测试工具，它主要用于对服务器进行负载等测试，并分析和测量服务器的各种性能指标。目前仅支持http协议，可以用来测试web应用程序和RESTful服务

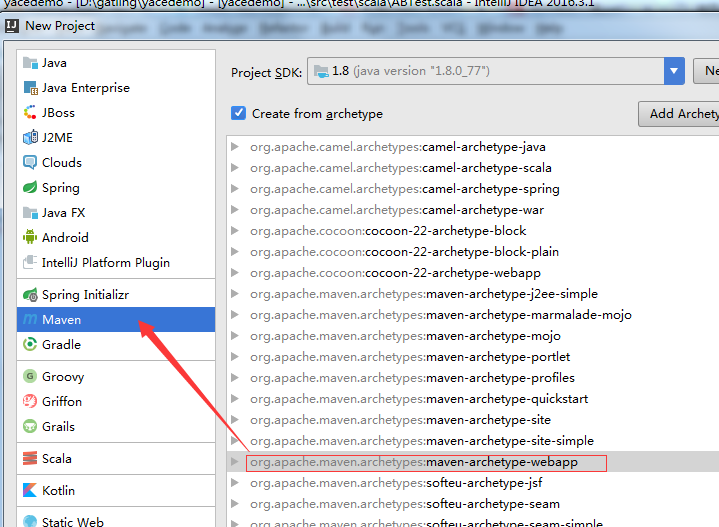
官网：https://gatling.io/

优点：

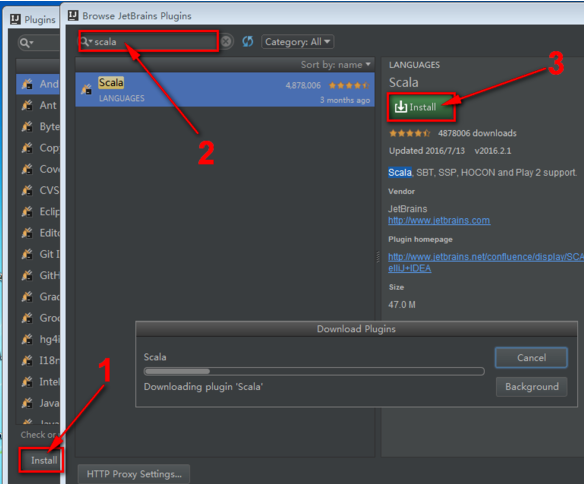
1. gatling和其他压力工具相比有个好处是放在同一内网环境下linux服务器上，这样避免其他压力使用办公机使用共有网络，网络情况对压力测试的影响。  
     2 生成比较详细的压力测试报告。  
     3 能够更灵活的编写压力脚本。  
     4 可以jenkins集合，将压力测试整合自动化持续集成中

上面都是干货，下面为大家直接实战，我是通过java+maven+scala搭建的一个环境以及Demo

1. 创建一个Maven项目

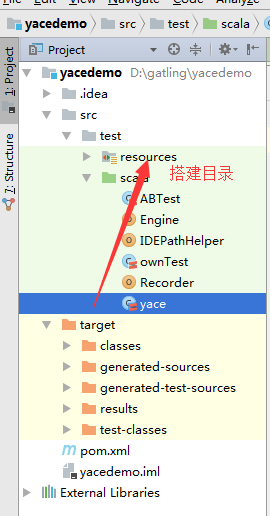


1. 安装Saala插件



1. pom添加核心依赖包

<**dependency**>  
 <**groupId**>io.gatling</**groupId**>  
 <**artifactId**>gatling-app</**artifactId**>  
 <**version**>${gatling.version}</**version**>  
</**dependency**>  
<**dependency**>  
 <**groupId**>io.gatling</**groupId**>  
 <**artifactId**>gatling-recorder</**artifactId**>  
 <**version**>${gatling.version}</**version**>  
</**dependency**>  
<**dependency**>  
 <**groupId**>io.gatling.highcharts</**groupId**>  
 <**artifactId**>gatling-charts-highcharts</**artifactId**>  
 <**version**>${gatling.version}</**version**>  
</**dependency**>-----太多我就不一一贴出，有兴趣微信公众号联系我

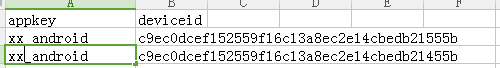


1. 下面就是Demo

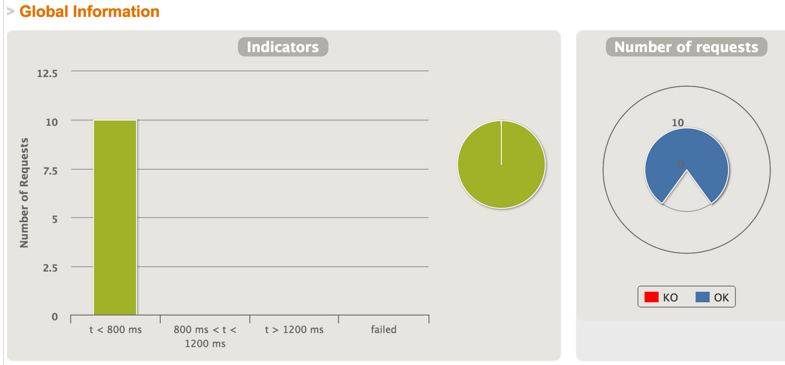
*/\*\*  
 \* Created by Administrator on 2017/12/10.  
 \*/***import** io.gatling.core.Predef.\_  
**import** io.gatling.http.Predef.\_  
**class** ABTest **extends** Simulation {  
 *//1.主要实现模拟不同用户重复100次执行结果* **object** Search {  
 *//val feeder = csv("D:\\gatling-2.2.5\\gatling-charts-highcharts-bundle-2.2.5\\user-files\\data\\appdata.csv")* **val** *search* =repeat(100) {*//重复次数100次  
 //读取csv数据* **val** feeder = csv(**"D:\\gatling-2.2.5\\gatling-charts-highcharts-bundle-2.2.5\\user-files\\data\\appdata.csv"**).random  
 feed(feeder)  
 .exec(*http*(**"ABTest"**)  
 .get(**"/abTest?appkey=${appkey}&deviceid=${deviceid}"**))  
 .pause(1)  
 *//exec()里的参数就是我们的执行动作，http("本次请求的名称").get("本次http get请求的地址")* }  
 }  
 **val** *httpConf* = *http* .baseURL(**"http://10.168.99.209:8080/abapi"**) *////设置请求的根路径* **val** *users* = *scenario*(**"Users"**).exec(Search.*search*) *// 声明Scenario，指定我们的请求动作* setUp(  
 *users*.inject(atOnceUsers(400)).protocols(*httpConf*))  
 *//atOnceUsers立马启动的用户数，可以理解为并发数*}

读取Excl数据500个用户

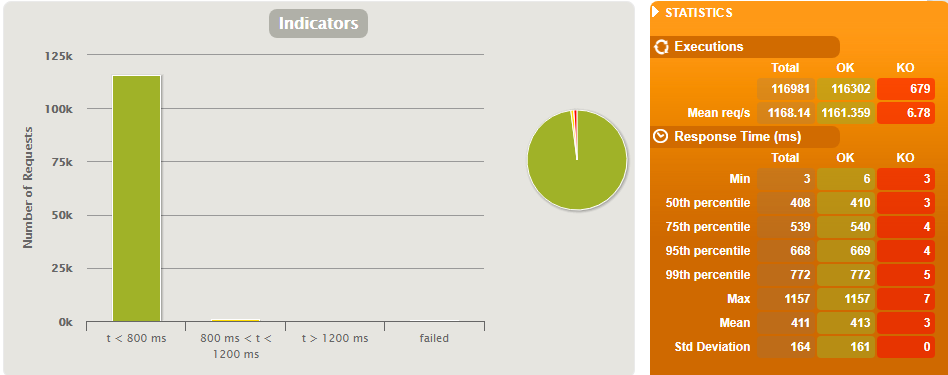
**D:\\gatling-2.2.5\\gatling-charts-highcharts-bundle-2.2.5\\user-files\\data\\appdata.csv**

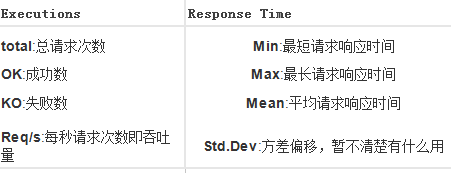


自动生成的测试报告：



概览800ms下响应的请求次数10，右边的饼状图显示成功请求10个，失败0个。注意在Global页面里的图都是整体的统计（一个测试计划可包含多个http请求），如果需要看某个请求的情况可点击到Details页面，并选取对应的请求





Response Time Distribution:响应时间分布

Response Time Percentiles over Time: 百分比响应时间分布

Number of responses per second: 每秒的响应数

Number of requests per second: 每秒的请求数