> 不依赖工具，但要重视工具。压测工具是压测理论的产物，学习好工具，可以帮助更加理解性能测试的基础知识。

性能测试，简而言之，就是模拟大量用户同时访问，测试服务是否满足性能要求。

> 示例脚本见：链接: https://pan.baidu.com/s/17naPnSvQXj5NIDu9HEFO0A 密码: sb4e

# 1、选择合适的工具

选择工具之前，首先了解一下，我们平常做性能测试，对性能工具的需求有哪些：

- 经常测试的对象：Http(s)协议、WebSocket (网络通信协议)、TCP/IP(测试Mysql、Redis)
- 扩展性好：支持分布式部署
- 使用成本低：文档多、支持UI界面操作、插件全

下面，我们从工具功能和性能两个角度，来横向对比。

性能测试工具功能对比

| 工具名 | 开发 | 是否 | 操作模式 | 脚本开发 | 支持系统 | 功能 |
|---|---|---|---|---|---|---|

综上，考虑到免费、开源、支持多协议、分布式扩展、脚本开发方式、平台支持和性能表现，选择Jmeter作为我们主要的性能测试工具。

# 2、了解Jmeter

## (1) 安装

- 解压后，运行Jmeter.bat或Jmeter.sh即可

- 启动后界面

这里的线程组，可以理解为一个用户模型。

## （2）线程组

线程组可以理解为用户模型。

线程组控制面板包括：

- 线程组名称
- 线程数（正在测试的用户数）
- 加速(Ramp-up)时间：从0增加到线程数的时间
- 循环计数：循环测试的次数
- 调度器：
  - 持续时间：表示脚本持续运行的时间，以秒为单位，比如如果你要让用户持续不断登录1个小时，你可以在文本框中填写3600。这样就会在1小时内循环执行。
  - 启动延迟：表示脚本延迟启动的时间，在点击启动后，如果启动时间已经到达，但是还没有到启动延迟的时间，那么，启动延迟将会覆盖启动时间，等到启动延迟的时间到达后，再运行系统。

## （3）测试Http服务

下面以开发一个Http脚本为例，来熟悉性能测试中常用的功能和插件。

### A. http脚本开发

如图中所示，界面中一目了然，协议、域名、Method、Url、参数分别填到对应位置即可。

如果请求需要header，可以添加Http信息头管理器，添加header信息。

### B. 参数化

总所周知，性能测试接口必须进行参数化，如果是固定数据，可能导致所有请求全部访问了缓存，这样就无法评估服务真实性能。

Jmeter参数化有3种常用方法，用户自定义变量、csv数据文件设置、BeanShell预处理变量。

#### 用户自定义变量

可以设置一些常量变量。

#### csv数据文件设置

可设置参数化数量较多常量。

#### BeanShell预处理变量

当有些变量需要加解密处理时，就需要BeanShell预处理。

注意：Jmeter通过vars.put("变量名", "变量值")来声明Jmeter变量。

```

import org.apache.commons.net.util.Base64;

// base64加密
String source = "哈利波特";
byte[] encodedBytes = Base64.encodeBase64(source.getBytes("UTF-8"));
String encoded = new String(encodedBytes);
vars.put("b64", encoded);

```

### C. 断言

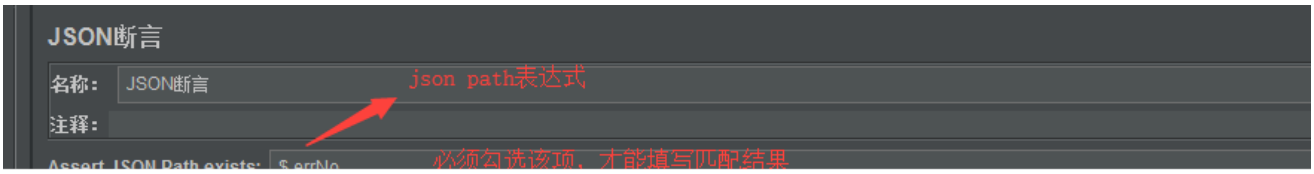
断言是用于检查Http 请求Response是否满足预期的手段。

Jmeter常用的断言方法有3种：响应断言、Json断言、Beanshell断言。

#### 响应断言

#### JSON断言

当Response Body为Json格式时，可以通过Json断言插件，精确断言响应内容。




#### BeanShell断言

> 参考：JMeter中使用BeanShell断言详解

BeanShell是jmeter的解释型脚本语言，和java语法大同小异，并有自己的内置对象和方法可供使用。

vars:操作jmeter的变量：vars.get(String parmStr) 获取jmeter的变量值；vars.put(String key,String value) 把数据存到Jmeter变量中；

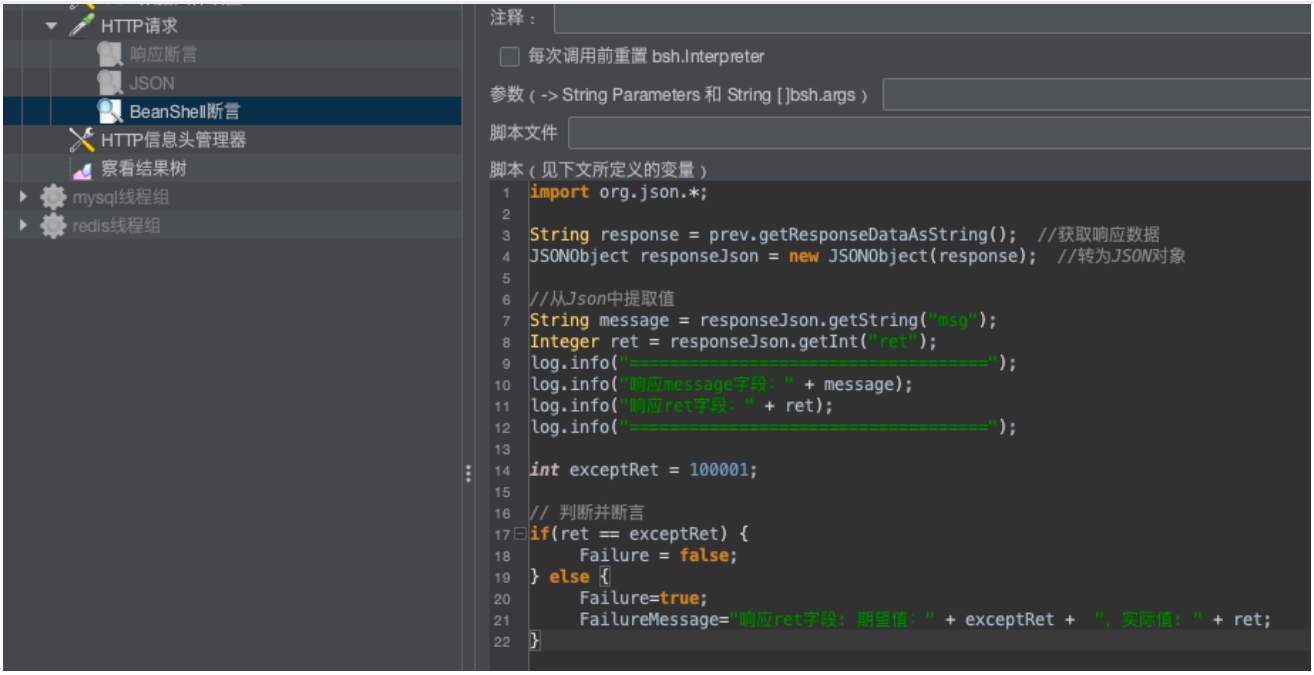
prev:获取sample返回的信息，prev.getResponseDataAsString() 获取响应信息；prev.getResponseCode() 获取响应状态码；

```
Failure = false;-----表示断言成功，

FailureMessage = "......"; ----自定义的成功信息

Failure = true;-----表示断言失败，

FailureMessage = "......";-----自定义的失败信息。
```




java

```java

import org.json.*;

String response = prev.getResponseDataAsString();  //获取响应数据
JSONObject responseJson = new JSONObject(response);  //转为JSON对象

//从Json中提取值
Integer code = responseJson.getInt("code");
String result = responseJson.getString("result");
log.info("====================================");
log.info("响应code字段：" + code);
log.info("响应result字段：" + result);
log.info("====================================");

int exceptCode = 200;

// 判断并断言
if(code == exceptCode) {
    Failure = false;
} else {
    Failure=true;
    FailureMessage="响应code字段：期望值：" + exceptCode +  "，实际值：" + code
}

```

## （4）测试Mysql

> 参考：JMeter进行MySQL接口性能测试实战：数据库服务器测试

**配置mysql连接（JDBC Connection Configuration）**

需要设置一些重要的字段，这些字段将决定数据库和JMeter之间的正确连接。 这些字段包括：

- 上半部分：
  - ariables Name for created pool：变量名，在JDBC Request的时候会用同样的名字确定是连接的那个库和进行的配置
  - MaxNumber of Connection：数据库最大链接数，通常该值设置为0
  - Max wait：最大的等待时间 ms毫秒，超出后会抛一个错误
  - Time Between Eviction Runs (ms)：数据库空闲连接的回收时间间隔
  - Auto Commit：自动提交。有三个选项，true、false、编辑，选择true后， 每条sql语句就是一个事务，执行结束后会自动提交；false、编辑则不会提交，需要自己手动提交
  - Transaction Isolation： 数据库事务隔离的级别设置
    - TRANSACTION_NONE：不支持的事务
    - TRANSACTION_READ_UNCOMMITTED ： 事务读取未提交内容
    - TRANSACTION_READ_COMMITTED： 事务读取已提交读内容
    - TRANSACTION_SERIALIZABLE： 事务序列化（一个事务读时，其他事务只能读，不能写）
    - DEFAULT：默认
    - TRANSACTION_REPEATABLE_READ ： 事务重复读（一个事务修改数据对另一个事务不会造成影响）
- 下半部分：
  - 绑定到池的变量名称 - 它唯一地标识配置。 JDBC Sampler将进一步使用此名称来标识要使用的配置。这里将其命名为test。
  - 数据库URL：jdbc:mysql://127.0.0.1:3306/AutoTest?useSSL=true
  - JDBC驱动程序类 - com.mysql.jdbc.Driver。
  - 用户名 - root。
  - 密码 - root用户的密码。

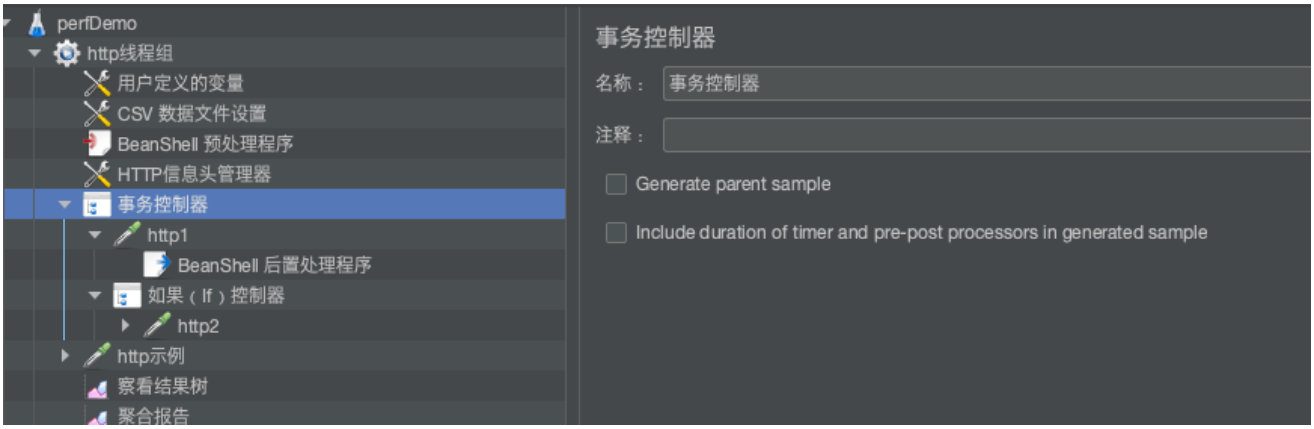
JDBC Request

## （5）事务定义

将多个接口定义为事务，以便**从业务的角度，来评估性能。**

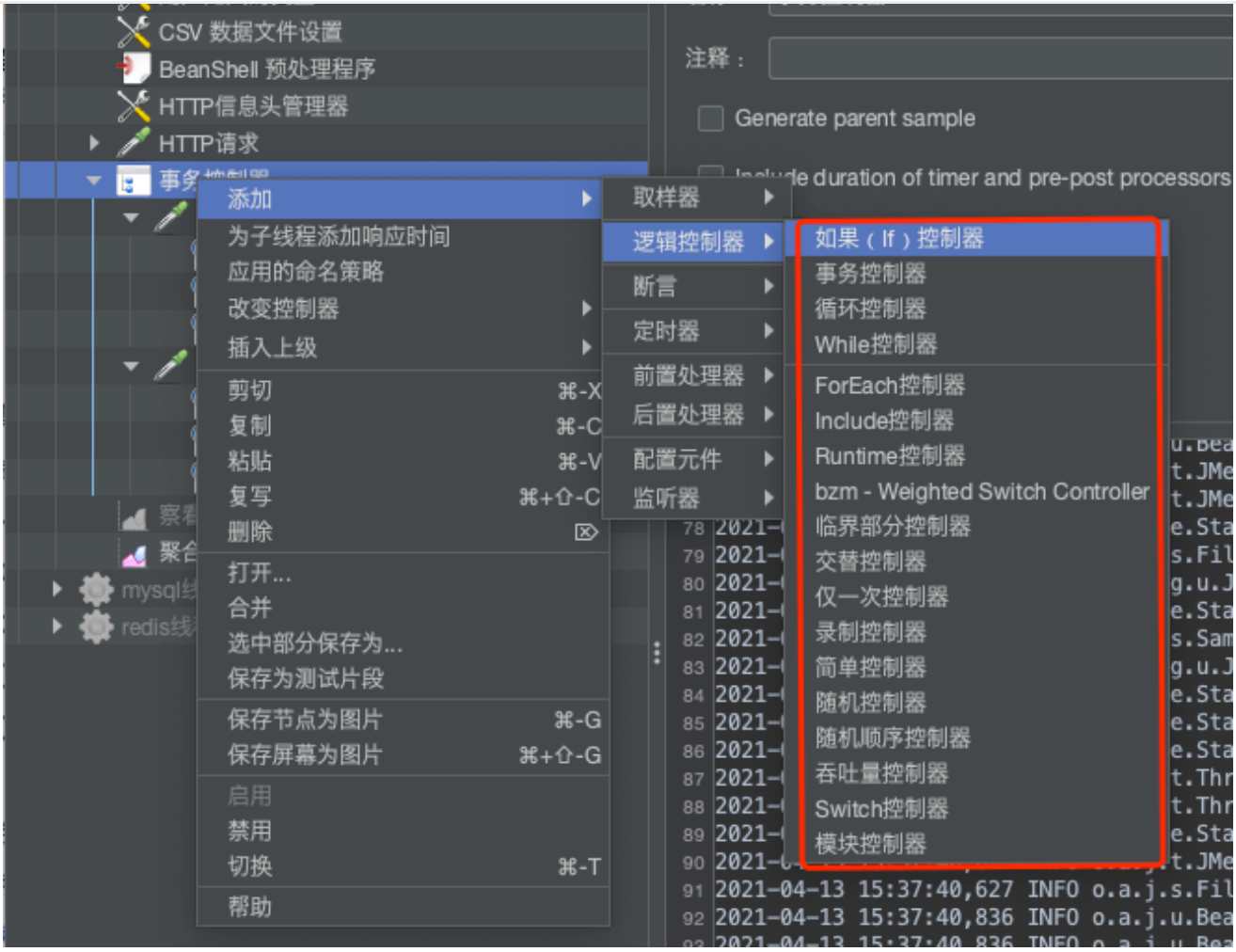
比如，完整的手机验证码登录流程，一般包含：（1）获取手机验证码；（2）用手机号和验证码进行登录；

当定义为登录事务时，性能测试的时候，我们就可以更宏观的关注服务端对登录事务的性能评估，而不是只关注到接口层。




## （6）逻辑控制器

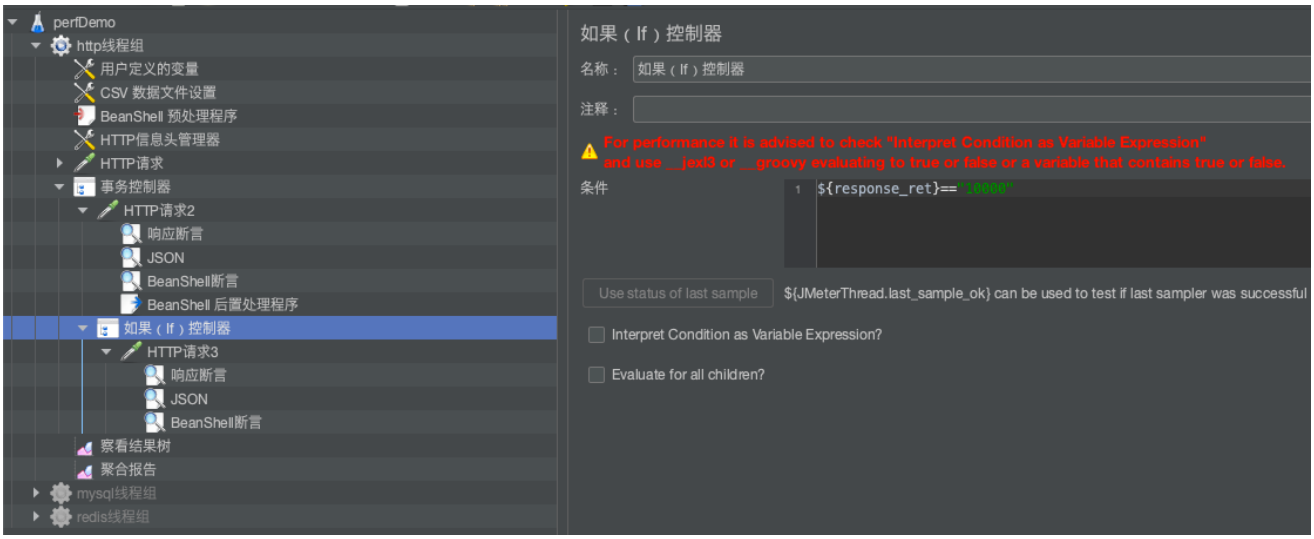
当面对复杂业务时，比如涉及到逻辑判断的，就需要添加逻辑控制器。




比如，if控制器：

> 参考：Jmeter中if控制器的使用

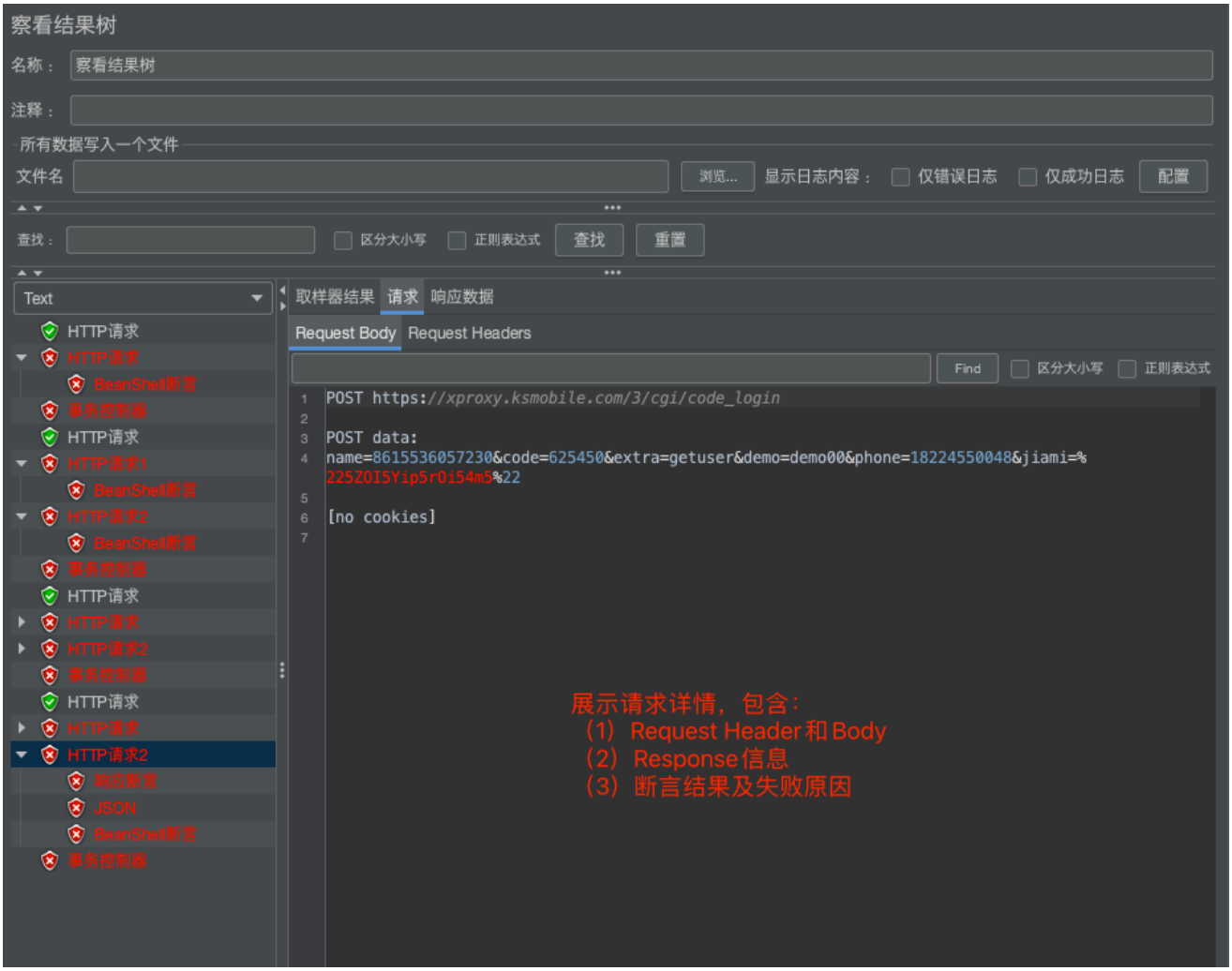
利用接口1的返回信息，进行逻辑判断，决定是否执行接口2。比如根据用户状态信息，判断用户是否是VIP用户，如果不是，则不能购买某商品。



## （7）调试插件

### 查看结果树

调试脚本时，可以利用 查看结果树 插件来查看request和response信息，保障脚本的正确。




### Jmeter执行Log

一般是编写BeanShell或引入jar包时，脚本调试需要看log，定位脚本开发的问题。

## （8）关注的性能指标

- QPS(TPS)：每秒请求数
- 成功率：请求成功率
- 95%响应时间：

测试期间，不允许出现Full GC。

# 3、Jmeter优化建议

> 参考：Jmeter性能调优建议

## （1）调整JMeter堆栈内存大小

在默认情况下现在JMeter5版本开辟的内存空间为1G，这也是它的最大内存。在实际测试的过程中，默认JMeter内存配置情况下，开启3W个线程来处理http的静态访问求基本上就达到了极限。再往上加的可能会报OOM的错误。

有两个JAVA的参数直接影响着JMeter能够使用的系统内存为多少，一个是"Xms"（代表初始化堆栈内存的大小），一个是"Xmx（代表最大内存池可以分配的大小）"。如果你的测试机器只跑JMeter一个JAVA应用程序，那么建议Xmx和Xms保持一致。Xmx和Xms保持一致是为了减少JVM内存伸缩，减少维护伸缩带来的成本。

## （2）64位操作系统内存配置大小

JMeter内存分配尽量在32位的系统上避免分配4G以上空间，在64位的操作系统尽量避免分配32G以上的空间。

## （3）垃圾回收机制

参考：【JVM】hotSpot VM 7种垃圾收集器：主要特点 应用场景 设置参数 基本运行原理

在JAVA中有大概五类的垃圾回收机制。分为Serial收集器，ParNew收集器，Parallel收集器，Cms收集器，G1收集器。在垃圾回收机制上应该尽量减少垃圾回收器带来的内存和CPU的性能损耗。当然这些并不会对开启线程数有着决定性的影响，属于细节性的微调。这里比较推荐使用G1垃圾回收器。G1垃圾回收器的特点如下：

- 支持很大的堆，高吞吐量
- 支持多CPU垃圾回收

## （4）使用非GUI模式

GUI在一定程度上冻结并消耗资源，这样会更容易产生一些不准确的性能测试结果。 对于JMeter来说GUI存在的意义主要在于可视化输出结果，编写你的测试计划和debug你的测试计划。 jmeter命令执行：

```
jmeter -n -t /usr/local/apache-jmeter-4.0/my_threads/sfwl.jmx
```

参数说明：

- -h 帮助 -> 打印出有用的信息并退出
- -n 非 GUI 模式 -> 在非 GUI 模式下运行 JMeter
- -t 测试文件 -> 要运行的 JMeter 测试脚本文件
- -l 日志文件 -> 记录结果的文件
- -r 远程执行 -> 启动远程服务
- -H 代理主机 -> 设置 JMeter 使用的代理主机
- -P 代理端口 -> 设置 JMeter 使用的代理主机的端口号

注意：如果未设置Jmeter的环境变量则在执行脚本的时候需要检查当前目录是否是jmeter的bin目录下

## （5）升级JMeter与JAVA版本

尽可能使用最新版本的JMeter来进行性能测试，新版本的JMeter会使用新版的JRE和JDK，这样会一定程度上带来性能的提升。

## （6）压测过程中禁用监听器

实行压测的过程中尽可能少使用监听器，最好别用。启用监听器会导致额外的内存开销，这会消耗测试中为数不多的内存资源。比较明智的做法是使用非GUI模式，将测试结果全部保存到“.jtl”的文件中。测试完成之后将jtl格式的结果文件导入的JMeter中再进行结果分析。

## （7）谨慎使用断言

添加到测试计划的每个测试元素都将被处理。这样会占用较多的CPU和内存。这种资源的占用适用于所有的断言，尤其是比较断言。比较断言消耗大量资源和内存，所以在压力测试时慎重的使用断言和断言的数量。

## （8）利用分布式

参考：压测必经之路，解读JMeter分布式

分布式部署Jmeter，利用压力机集群进行压测，提升Jmeter的测试能力。

← 服务端性能测试入门指南

提交

## 11 评论

防水材料加盟 Firefox 98.0 Windows 10.0
2022-03-25 回复

赞一个

在路上 Chrome 99.0.4844.84 macOS 10.15.7
2022-03-29 回复

@防水材料加盟 , 这个广告可以

Anonymous Chrome 97.0.4692.71 macOS 10.15.7
2022-01-17 回复

hhhhh

Anonymous Chrome 97.0.4692.71 macOS 10.15.7
2022-01-17 回复

@Anonymous , dddd 🤣🤣🤣

Anonymous Chrome 96.0.4664.45 Linux
2021-12-29 回复

hi boy

在路上 Chrome 96.0.4664.110 macOS 10.15.7
2022-01-04 回复

**Anonymous** Chrome 96.0.4664.110 macOS 10.15.7
2021-12-24 回复

订阅新能源车使用体验 

> **Anonymous** Chrome 96.0.4664.110 macOS 10.15.7
> 2021-12-24 回复
>
> @Anonymous，哈哈，非常爽，我买的比亚迪秦-Plus-EV，冬天也还行，一直开着空调，百公里15度电。
>
> 如果不开空调，续航可以达到90%
>
> > 在路上 Chrome 96.0.4664.110 macOS 10.15.7
> > 2021-12-24 回复
> >
> > @Anonymous，可以找车试驾一下，我试驾了好猫和秦Plus，果断买了秦。
> >
> > 特斯拉驾驶体验应该是最爽的

**Anonymous** Chrome 95.0.4638.69 Windows 10.0
2021-11-26 回复

想起了张敬轩的on my way 

在路上 Edge 95.0.1020.53 macOS 10.15.7
2021-11-17 回复

欢迎大家留言😊😊

**Anonymous** Chrome 95.0.4638.69 macOS 10.15.7
2021-11-15 回复

太强了

为什么找不到您之前写的财务自由的文章了,是删了吗,还想推荐朋友看

在路上 Edge 93.0.961.38 macOS 10.15.7
2021-09-13 回复

@Anonymous，最近没时间弄，过一段时间更新出来哈。

神乐 Edge 111.0.1661.44 Windows 10.0
2023-03-24 回复

@在路上，老哥可以更新了吗，等着看~

Anonymous Chrome 92.0.4515.159 macOS 10.15.7
2021-08-21 回复

在路上 Edge 91.0.864.70 macOS 10.15.7
2021-07-21 回复

加载更多...