ALIBABA CLOUD

阿里云

性能测试

常见问题

文档版本：20220707

阿里云

法律声明

阿里云提醒您在阅读或使用本文档之前仔细阅读、充分理解本法律声明各条款的内容。如果您阅读或使用本文档，您的阅读或使用行为将被视为对本声明全部内容的认可。

1. 您应当通过阿里云网站或阿里云提供的其他授权通道下载、获取本文档，且仅能用于自身的合法合规的业务活动。本文档的内容视为阿里云的保密信息，您应当严格遵守保密义务；未经阿里云事先书面同意，您不得向任何第三方披露本手册内容或提供给任何第三方使用。

2. 未经阿里云事先书面许可，任何单位、公司或个人不得擅自摘抄、翻译、复制本文档内容的部分或全部，不得以任何方式或途径进行传播和宣传。

3. 由于产品版本升级、调整或其他原因，本文档内容有可能变更。阿里云保留在没有任何通知或者提示下对本文档的内容进行修改的权利，并在阿里云授权通道中不时发布更新后的用户文档。您应当实时关注用户文档的版本变更并通过阿里云授权渠道下载、获取最新版的用户文档。

4. 本文档仅作为用户使用阿里云产品及服务的参考性指引，阿里云以产品及服务的“现状”、“有缺陷”和“当前功能”的状态提供本文档。阿里云在现有技术的基础上尽最大努力提供相应的介绍及操作指引，但阿里云在此明确声明对本文档内容的准确性、完整性、适用性、可靠性等不作任何明示或暗示的保证。任何单位、公司或个人因为下载、使用或信赖本文档而发生任何差错或经济损失的，阿里云不承担任何法律责任。在任何情况下，阿里云均不对任何间接性、后果性、惩戒性、偶然性、特殊性或刑罚性的损害，包括用户使用或信赖本文档而遭受的利润损失，承担责任（即使阿里云已被告知该等损失的可能性）。

5. 阿里云网站上所有内容，包括但不限于著作、产品、图片、档案、资讯、资料、网站架构、网站画面的安排、网页设计，均由阿里云和/或其关联公司依法拥有其知识产权，包括但不限于商标权、专利权、著作权、商业秘密等。非经阿里云和/或其关联公司书面同意，任何人不得擅自使用、修改、复制、公开传播、改变、散布、发行或公开发表阿里云网站、产品程序或内容。此外，未经阿里云事先书面同意，任何人不得为了任何营销、广告、促销或其他目的使用、公布或复制阿里云的名称（包括但不限于单独为或以组合形式包含“阿里云”、“Aliyun”、“万网”等阿里云和/或其关联公司品牌，上述品牌的附属标志及图案或任何类似公司名称、商号、商标、产品或服务名称、域名、图案标示、标志、标识或通过特定描述使第三方能够识别阿里云和/或其关联公司）。

6. 如若发现本文档存在任何错误，请与阿里云取得直接联系。

通用约定

格式	说明	样例
危险	该类警示信息将导致系统重大变更甚至故障，或者导致人身伤害等结果。	危险 重置操作将丢失用户配置数据。
警告	该类警示信息可能会导致系统重大变更甚至故障，或者导致人身伤害等结果。	警告 重启操作将导致业务中断，恢复业务时间约十分钟。
注意	用于警示信息、补充说明等，是用户必须了解的内容。	注意 权重设置为0，该服务器不会再接受新请求。
说明	用于补充说明、最佳实践、窍门等，不是用户必须了解的内容。	说明 您也可以通过按Ctrl+A选中全部文件。
>	多级菜单递进。	单击**设置**> **网络**> **设置网络类型**。
粗体	表示按键、菜单、页面名称等UI元素。	在**结果确认**页面，单击**确定**。
Courier字体	命令或代码。	执行 `cd /d C:/window` 命令，进入Windows系统文件夹。
斜体	表示参数、变量。	`bae log list --instanceid` *Instance_ID*
[] 或者 [a|b]	表示可选项，至多选择一个。	`ipconfig [-all|-t]`
{} 或者 {a|b}	表示必选项，至多选择一个。	`switch {active|stand}`

目录

1.并发和RPS

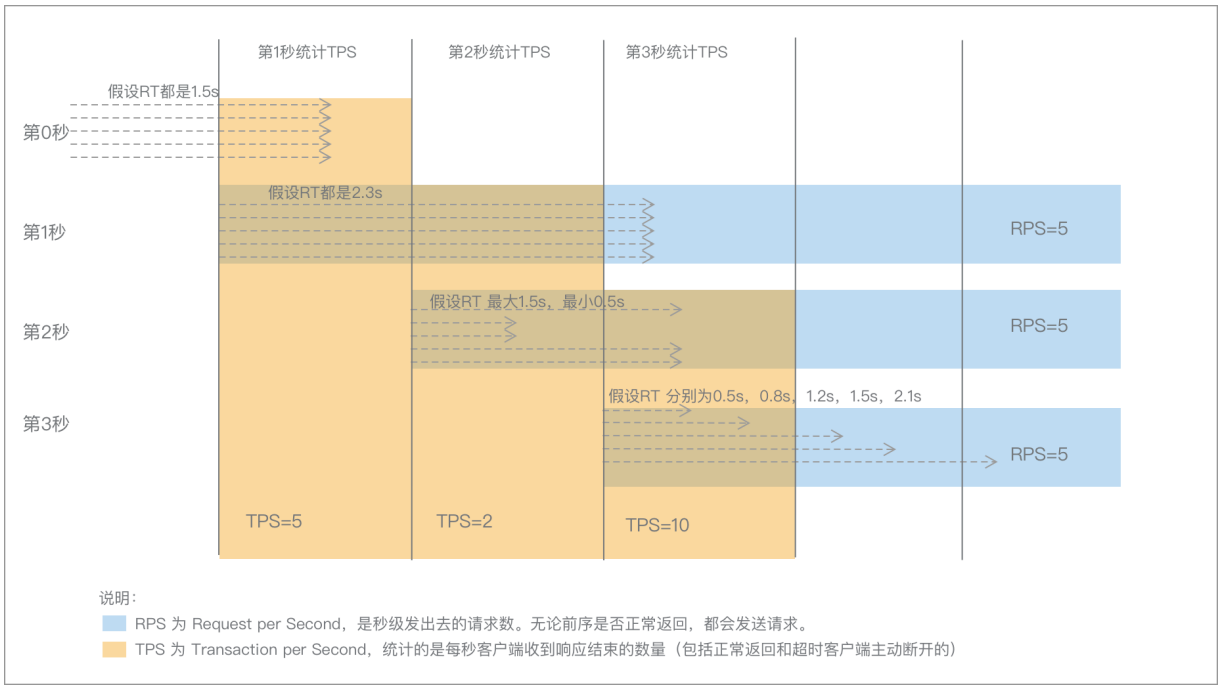
1.1. RPS模式下，为何统计的当前TPS比设置的RPS要多？

如下图所示，压测中上限的TPS为150（即设置的RPS），但压测时显示当前TPS是192，高于RPS设置值150。

这是因为每个请求的响应时间RT（Response Time）是不同的，客户端统计的TPS数据，是过去1秒内客户端一共收到了多少个返回的请求事务信息。示例中表示至少42个请求是较早前未能及时（1秒内）返回的。

可参考以下示意图，假设RPS设置为5。

- 假设第0秒发出的5个请求RT都是1.5秒，第1秒发出的5个请求RT都是2.3秒，则第1秒统计的TPS为5，表示第0秒发出的5个请求响应结束。
- 假设第2秒发出的5个请求中3个RT是1.5秒，2个RT是0.5秒，则第2秒统计的TPS为2，表示第2秒发出的RT为0.5秒的2个请求响应结束。
- 假设第3秒发出的5个请求RT分别为0.5秒、0.8秒、1.2秒、1.5秒、2.1秒，则第3秒统计的TPS为10，包括第1秒发的5个RT为2.3秒的请求、加上第2秒发的3个RT为1.5秒的请求、加上第3秒发的2个RT为0.5秒和0.8秒的请求。



1.2. 为何设置了300并发，实际压测过程中并未能达到？

本文介绍为何设置了300并发，实际压测过程中并未能达到的解决方案。

问题描述

在并发模式下，设置了最大300并发，但压测中并发并没有达到设置的值。

- **并发模式**：即虚拟用户模式。详细定义，请参见并发用户、RPS、TPS的解读。
- **并发用户数**（以下简称“并发”）：现实系统中同时操作业务的用户数量，在性能测试工具中一般称为虚拟用户（Virutal User）。最大并发用户数，即在压测中最多多少用户同时对压测业务产生压力。
- **RPS**：Request Per Second，每秒请求数。RPS是从服务端的视角衡量系统每秒处理的请求数，一个并发用户可能产生多个请求。

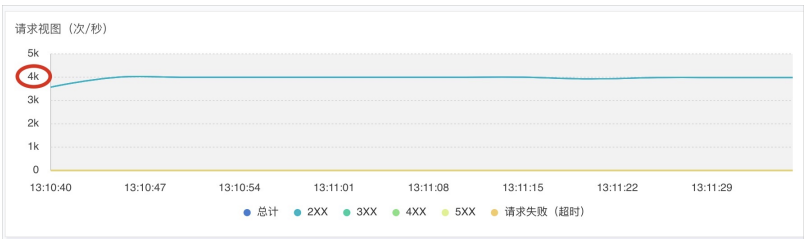
> **说明** 并发是基于串联链路的，并发数是串联链路中所有API并发之和。假设串联链路的并发是200，包含了7个API，发起压测时，PTS将按顺序对7个API进行压测，7个API的总并发为200。由于每个API压测时响应时间不一致导致每个API并发不一样，API的响应时间少则并发小。

原因分析

由于压测的施压机计算资源有限，同时为了保证压测流量更稳定，在施压时对RPS有一定的限制（资源包规格中不同并发有相应的RPS上限）。RPS如果到达上限，即不会增加更大的压力。

例如，将最大并发设置为300时，在施压配置页面您可以看到对应的RPS限制为4000：

以下面压测报告为例，平均RPS（总请求数/总压测时长）已经达到4001（4001是采集周期内计算的合理误差），在压测趋势图中已经达到了4000的RPS上限，所以并发不会再继续往上涨了。



解决方法

推荐您在施压配置中，配置更高的目标并发，再次进行压测，请参见压力模型。或者使用IP扩展的功能增加IP个数，也可以将两种办法结合一起使用获得更高的RPS上限。

1.3. 并发模式下，为什么同一个并发下TPS会不同？

在并发模式下，同样的虚拟用户数（并发）下TPS取决于被压测服务的处理能力，也就是平均的RT（响应时间）。

TPS=并发/RT（秒）

如果平均RT变长，TPS就会变小。例如，并发是1000时，RT平均从1秒变长到2秒，那么TPS也从1000下降到了500。

1.4. 场景中如何设置起步和目标量级？

本文介绍如何在场景中设置起步和目标量级。

无论并发模式还是TPS模式，场景就是一个压测模型，压测模型中有串行的事务，如添加购物车+购物车下单+付款，也有并行的接口，如在不同串联链路中的压测API，最终组成一个复杂或者简单的场景。然后根据新业务上线的目标、或者日常峰值的等比例目标、或者重大业务活动的预估支撑能力去设置每个API的目标能力。

TPS是一步到位按照吞吐能力设置的，推荐您使用TPS模式。例如添加购物车+购物车下单+付款这种流程就是一个漏斗模型，TPS设置为逐渐变小的模型即可，当然也可以在初期的测试中将目标量级设置得整体低一点，当最终能力达到之后建议您可以调整原定目标量级到120%或者150%，验证限流准入和高可用基础设施的抗压能力。

目标量级即当前压测场景中这个压测API的施压上限。而起步量级可以从5%或者10%开始，过程中视业务指标数据和被压测端的整体负载临时调整，PTS铂金版提供了全局百分比调整的功能。

1.5. RPS模式的并发量是如何计算的？

RPS模式以吞吐量作为目标，例如1000 RPS表示一秒内发出1000个请求。在施压过程中，根据被压测接口的RT表现不同，施压引擎为了达到您指定的吞吐量，会自适应调整虚拟用户数（即并发量）。

虚拟用户数（并发量）如何计算

计算公式：RPS模式下的虚拟用户数 = RPS x RT（秒）。

> **说明** 上述公式是基于压测过程中的瞬时RPS、RT，计算出瞬时的虚拟用户数；并非某一时间段的平均值。

示例：

当RPS设置为1000，

- 如果被压测服务RT为0.1秒，则虚拟用户数为100；
- 如果被压测服务RT为2秒，则虚拟用户数为2000；
- 以此类推。

服务异常怎么办？

在服务异常时，请及时停止压测。

当被压测服务异常时，您在PTS控制台会看到出现大量的RT变高，甚至出现请求失败超时。由于PTS无法感知被压测端的整体情况，同时为了达到您设置的RPS值，PTS触发的并发会越来越高，而且在API的超时时间内累积。此时，继续压测并无意义，您需要及时停止压测。

为避免上述问题，建议您：

- 在创建压测场景的模型时，设置合理的目标RPS。
- 将起步RPS设置得较低，压测过程中手动逐步调高RPS，进行观察监控。

1.6. 如何设置目标并发或目标RPS？

本文帮助您了解什么是并发用户、TPS，以及它们之间的关系。

基本概念

- 并发用户：指的是现实系统中同时操作业务的用户，在性能测试工具中一般称为虚拟用户（Virtual User）。并发用户这个概念一般是从客户侧评估的角度出发，但是不便于服务端的一些容量评估和高可用评估。

  并发用户与注册用户、在线用户不同。注册用户一般指的是数据库中存在的用户。在线用户只是“挂”在系统上，对服务器不产生压力。但并发用户一定会对服务器产生压力。

- TPS：Transaction Per Second， 每秒事务数，是衡量系统性能的一个重要指标。在PTS中，为了直接评估TPS，也可以采用RPS（Request Per Second，每秒请求数）设置压测流量的大小。RPS模式更适合容量规划和作为限流管控的参考依据。

  示例：

  假如1个虚拟用户在1秒内完成1个事务，那么TPS就是1。要想达到1000 TPS至少需要1000个用户。如果某个业务响应时间是1毫秒，那么1个用户在1秒内能完成1000个事务，TPS也是1000。

  因此1个用户可以产生1000 TPS，1000个用户也可以产生1000 TPS，主要看响应时间的快慢。

设置目标并发和TPS

评估并发用户数：

- 线上系统。

  在线上系统的高峰时刻，选取一定周期内使用系统的人数，这些人数可以认为是在线用户数。而并发用户数则取在线用户数的10%。例如，在1小时内使用系统的用户数为10000，则建议取1000作为并发用户数。

- 未上线系统或新上线系统。

  由于没有历史数据可供参考，因此只能通过业务发展趋势来预判各项指标。

评估TPS（RPS）：

- 线上系统。
  - 已有系统：可选取高峰时刻，在一定时间内（如3分钟~10分钟），获取系统总业务量，计算单位时间（秒）内完成的笔数，乘以2~5倍作为峰值的TPS，例如峰值3分钟内处理订单18万笔，平均TPS是1000，峰值TPS可以是2000~5000。
  - 新系统：没有历史数据作参考，建议通过业务部门进行评估。
- 未上线系统或新上线系统。

  由于没有历史数据可供参考，因此只能通过业务发展趋势来预判各项指标。

1.7. 怎么理解SLB的并发连接数和PTS里的并发？

首先为了便于理解，假设PTS中的压测场景是一个串联链路A，里面包含一个压测API 1。

SLB的并发连接数的定义。

指标	说明
并发连接数	所有建立的TCP连接数量。
活跃连接数	所有ESTABLISHED状态的TCP连接。因为如果您采用的是长连接的情况，一个连接会同时传输多个文件请求。
非活跃连接数	表示指除ESTABLISHED状态的其它所有状态的TCP连接数。Windows和Linux服务器都可以使用netstat -an命令查看。

当压测进行时，PTS上的API 1的并发数基本会和SLB的并发连接数接近或者更小，因为SLB上可能还有正常业务流量。但由于PTS上的施压端监控数据是秒级采集的，而SLB上是15秒或者分钟级数据，所以PTS的API 1并发数和SLB并发数会有一定的误差。此外还需要注意一种情况，如果被压测服务端支持Keep-Alive，当串联链路中放入了思考时间且时间比较大（大于Keep-Alive的timeout），则会出现SLB上的并发连接数特别是活跃连接数下降的情况。

2.JMeter

2.1. 如何开通兼容JMeter原生压测的功能?

PTS支持原生JMeter压测，免维护。

操作步骤

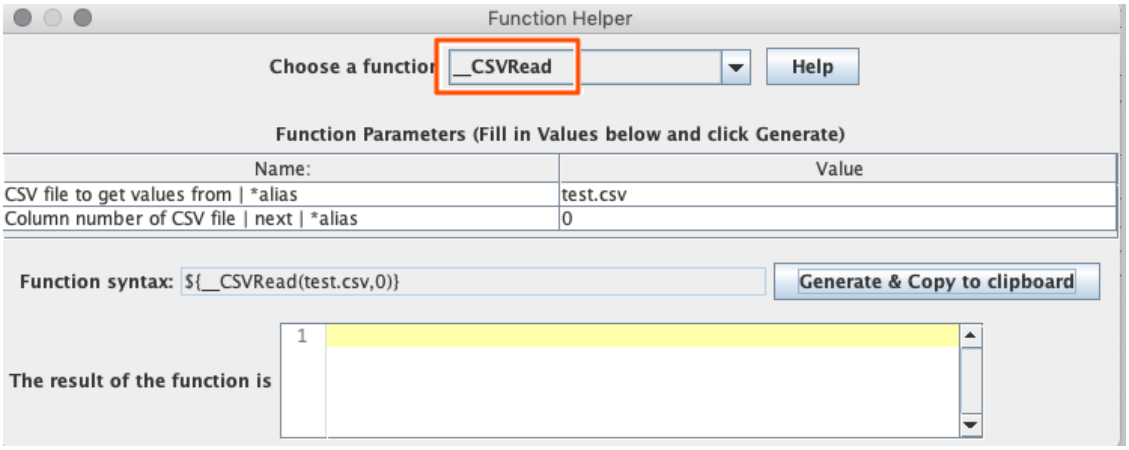
使用此功能前，您需要完成如下操作。

1. 开通PTS服务。
2. 购买高虚拟用户（VU）5000及以上的资源包（即278元及以上的资源包）。详情请参见资源包规格说明。

2.2. 为什么__CSVRead函数读取不到值?

Condition

在JMeter的脚本中有使用到 __CSVRead函数，但是实际压测时通过采样日志未发现CSV文件中的值。

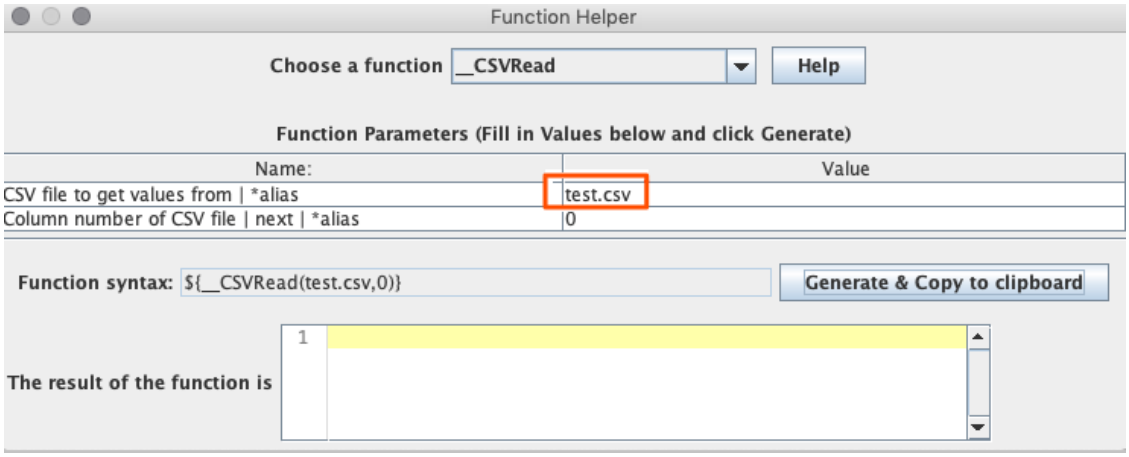


Remedy

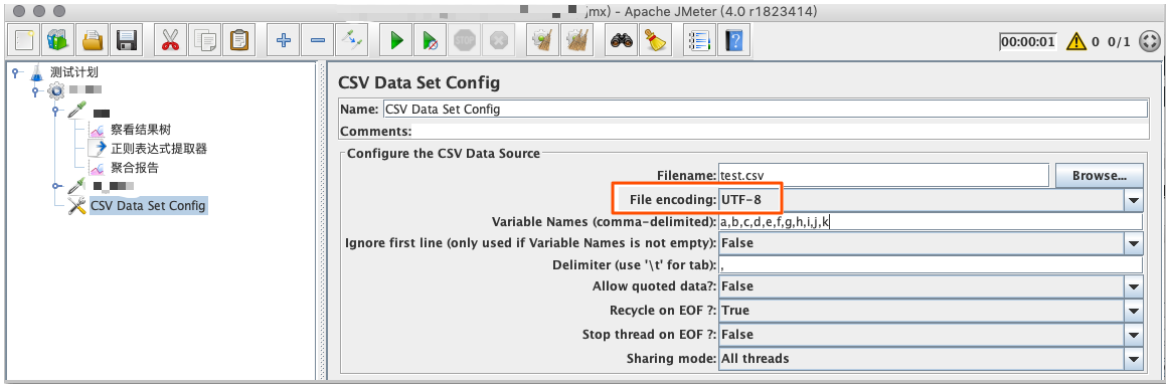
操作步骤

1. 确认脚本使用到的CSV文件都已经上传到PTS。

2. 路径设置使用相对路径。请务必将CSV文件的value修改为文件名而不是文件路径。



3. 如果数据文件中有包含中文的情况，编码设置需要调整为UTF-8。
4. 最后，即使修复了问题，还是建议使用CSV Data Set Config，可以直接设置编码格式，同时相较于__CSVRead函数有更好的维护性。



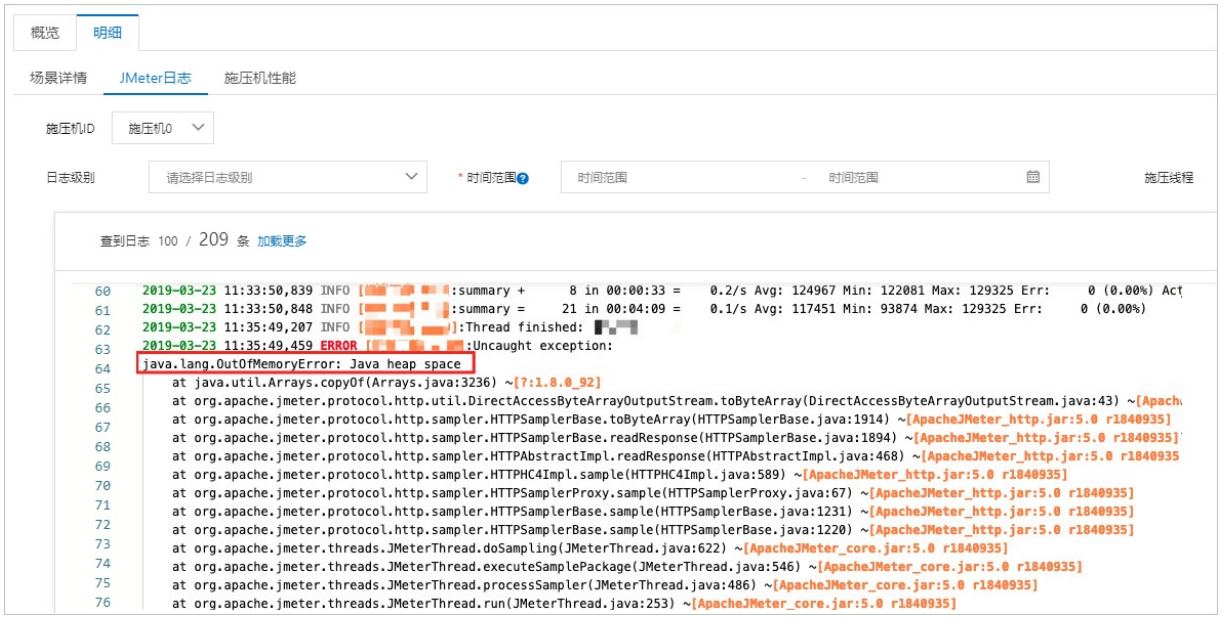
2.3. 为何JMeter模拟下载的时候，运行一段时间就自行报错退出了？

Condition

JMeter模拟下载时，运行一段时间就自行报错退出。

Cause

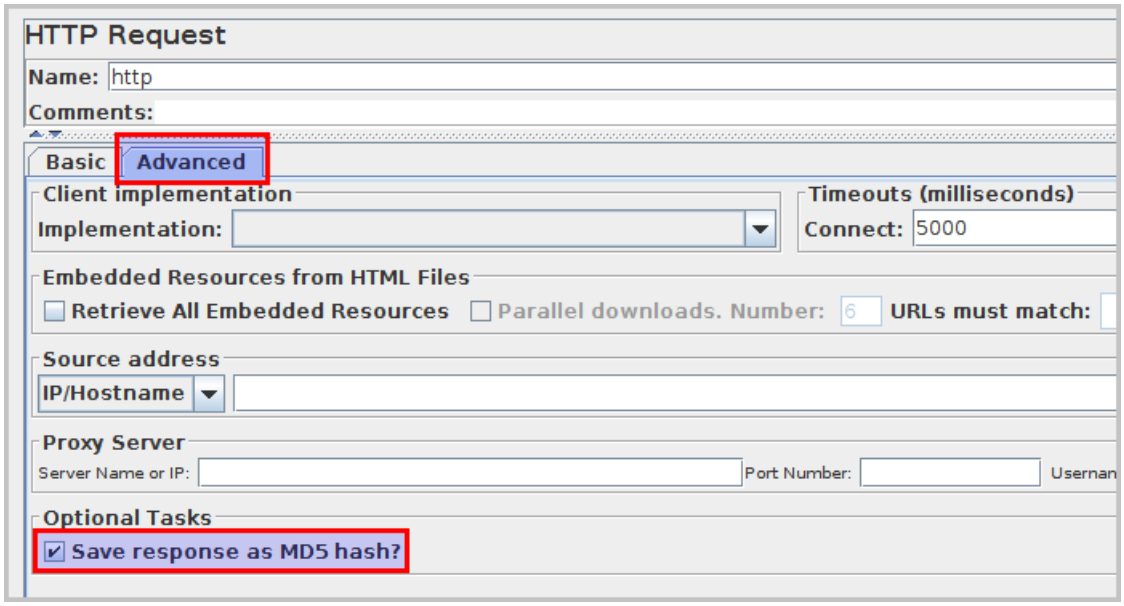
施压机类似于被压测服务的请求用户侧，当进行下载压测时，会将download的文件流存于内存中，当并发较高、文件较大时，内存就会出现异常，JMeter日志中报如下错误：



Remedy

操作步骤

1. 解决此问题，需要在JMeter脚本配置时，对应的下载请求Sampler的**Advanced**选项中，选中**Save response as MD5 hash?**。选中该选项表示，仅保存结果的MD5值，不保存原始信息。这样就不会保存较大的下载原始文件。



2.4. JMeter集成压测中的数据是如何统计的？

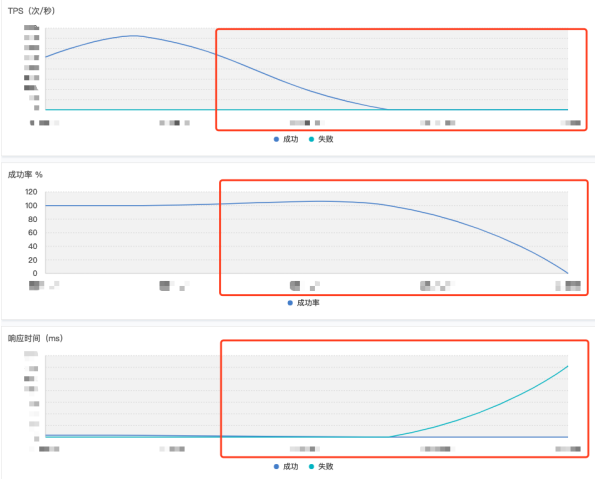
本文介绍JMeter集成压测中的数据是如何统计的。

PTS的JMeter集成压测使用原生JMeter引擎，其中的监控数据采集部分的数据来源也是基于Backend Listener做了实现进行了简单的聚合计算，请放心查看。

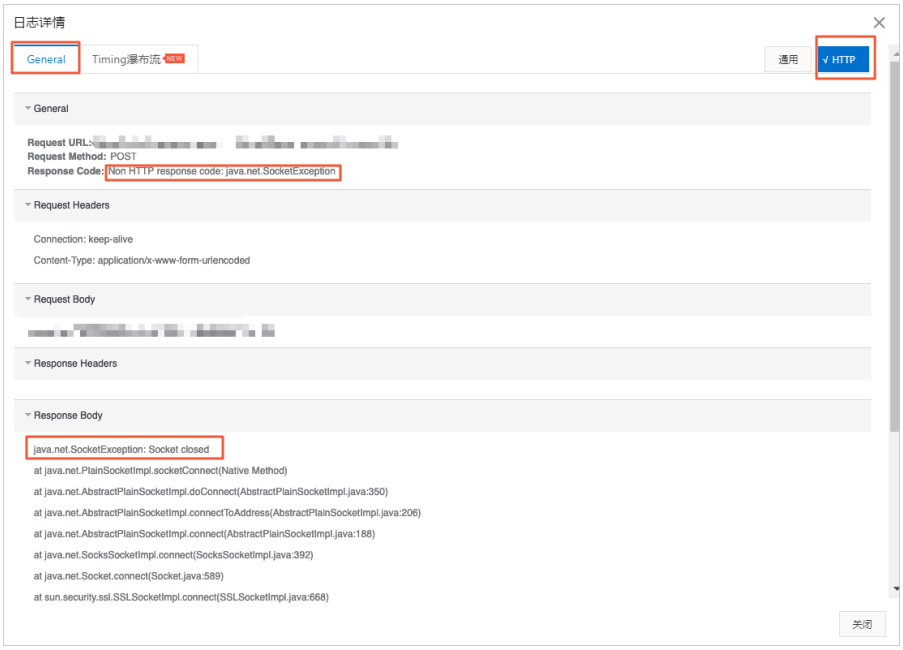
2.5. 为什么采样日志中报java.net.SocketException: Socket closed错误？

Condition

如果压测的是HTTPS接口，同时在压测进行过程中出现RT（响应时间）逐渐变高，TPS和成功率都有跌零或者相应的趋势。



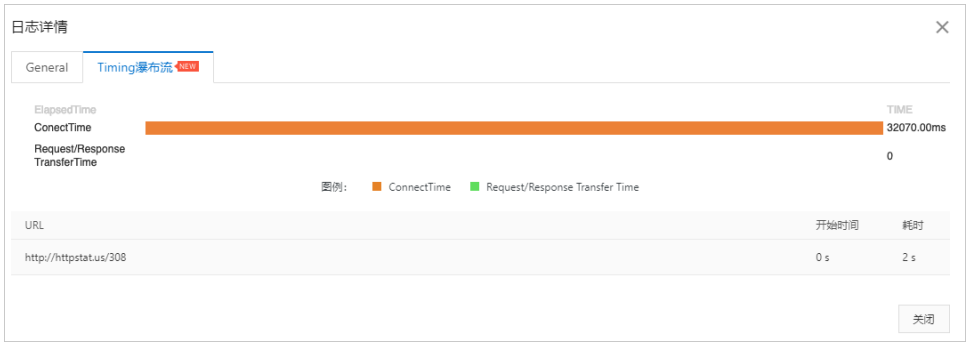
这时，单击**压测报告**右上角的**查看采样日志**，单击**操作**列的**点击查看详情**，打开采样日志看到的报错如下（选择HTTP模板更清晰）。



详细的文本参考如下。

```
java.net.SocketException: Socket closed
at java.net.PlainSocketImpl.socketConnect(Native Method)
at java.net.AbstractPlainSocketImpl.doConnect(AbstractPlainSocketImpl.java:350)
at java.net.AbstractPlainSocketImpl.connectToAddress(AbstractPlainSocketImpl.java:206)
at java.net.AbstractPlainSocketImpl.connect(AbstractPlainSocketImpl.java:188)
at java.net.SocksSocketImpl.connect(SocksSocketImpl.java:392)
at java.net.Socket.connect(Socket.java:589)
at sun.security.ssl.SSLSocketImpl.connect(SSLSocketImpl.java:668)
at org.apache.http.conn.ssl.SSLSocketFactory.connectSocket(SSLSocketFactory.java:542)
at org.apache.http.conn.ssl.SSLSocketFactory.connectSocket(SSLSocketFactory.java:414)
at org.apache.jmeter.protocol.http.sampler.LazySchemeSocketFactory.connectSocket(LazyScheme
SocketFactory.java:97)
at org.apache.http.impl.conn.DefaultClientConnectionOperator.openConnection(DefaultClientCo
nnectionOperator.java:180)
at org.apache.jmeter.protocol.http.sampler.hc.ManagedClientConnectionImpl.open(ManagedClien
tConnectionImpl.java:318)
at org.apache.jmeter.protocol.http.sampler.MeasuringConnectionManager$MeasuredConnection.op
en(MeasuringConnectionManager.java:114)
at org.apache.http.impl.client.DefaultRequestDirector.tryConnect(DefaultRequestDirector.jav
a:610)
at org.apache.http.impl.client.DefaultRequestDirector.execute(DefaultRequestDirector.java:4
45)
at org.apache.http.impl.client.AbstractHttpClient.doExecute(AbstractHttpClient.java:835)
at org.apache.http.impl.client.CloseableHttpClient.execute(CloseableHttpClient.java:83)
at org.apache.jmeter.protocol.http.sampler.HTTPHC4Impl.executeRequest(HTTPHC4Impl.java:695)
at org.apache.jmeter.protocol.http.sampler.HTTPHC4Impl.sample(HTTPHC4Impl.java:454)
at org.apache.jmeter.protocol.http.sampler.HTTPSamplerProxy.sample(HTTPSamplerProxy.java:74
)
at org.apache.jmeter.protocol.http.sampler.HTTPSamplerBase.sample(HTTPSamplerBase.java:1189
)
at org.apache.jmeter.protocol.http.sampler.HTTPSamplerBase.sample(HTTPSamplerBase.java:1178
)
at org.apache.jmeter.threads.JMeterThread.executeSamplePackage(JMeterThread.java:498)
at org.apache.jmeter.threads.JMeterThread.processSampler(JMeterThread.java:424)
at org.apache.jmeter.threads.JMeterThread.run(JMeterThread.java:255)
at java.lang.Thread.run(Thread.java:766)
```

查看采样日志的Timing页签，可以看到时间都耗费在连接建立阶段。



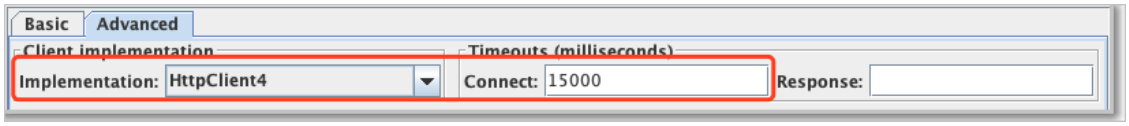
Cause

引起java.net.SocketException: Socket closed错误的原因通常是未设置连接的超时时间。

Remedy

操作步骤

1. 如果在HTTP Request Sampler的**Basic**里选中了**Use KeepAlive**，则建议您在**Advanced**页签下设置如下参数：
   - 选择**Implementation**为**HttpClient4**。
   - **Connect**设置一个10秒~60秒的值，作为连接空闲超时时间，避免由于没收到被压测端返回的Keep-Alive的Header而导致连接断开。



更多信息，请参见JMeter Socket相关资料。

2.6. 为什么JMeter的压测发起之后很快停止了？

Condition

JMeter压测发起之后很快停止了。

Cause

最有可能的原因是在JMeter测试计划中使用了Loop Controller，而且设置了Loop Count为1。

Remedy

操作步骤

1. 不建议设置Loop Count。建议您在施压配置中设置好循环次数，具体操作，请参见施压配置。

2.7. JMeter设置了集合点为什么没生效？

Condition

JMeter脚本中设置了集合点，但是在PTS中压测时，发现设置的集合点未生效。

Cause

JMeter的集合点只在单台施压机（JVM）内生效，具体操作，请参见Synchronizing Timer。PTS中的JMete可能自动分配多台施压机，因此建议不要使用Synchronizing Timer。若需使用请注意以下几点：

- Synchronizing Timer只在单台施压机内生效，不能全局同步。
- **Number of Simultaneous Users to Group by**设置值不能超过单机并发数，否则永远不能满足条件，建议设置为0，即自动取单机最大并发数。

- 合理设置超时时间 Timeout in milliseconds，例如设置为5000，即使条件不满足，超时后也会继续执行。

2.8. 为何设置了Synchronizing Timer在本地生效，在PTS中不生效?

Condition

在JMeter中设置了Synchronizing Timer且在本地生效，但是在PTS中运行时不生效。

Cause

Synchronizing Timer只在单台施压机（JVM）内生效。使用PTS压测时，可能自动分配多台施压机，因此不生效。PTS现已支持JMeter分布式适配设置，可选择脚本中的设置值是全局生效或单机生效。详情请参见设置同步定时器。

如果有使用，请注意如下事项：

- Number of Simultaneous Users to Group by数值不能超过单机并发数，否则不能满足条件，建议设置为0，即自动取单机最大并发数。
- 合理设置Timeout in milliseconds（超时时间）。例如设置Timeout in milliseconds为5000，即使条件不满足，超时后也会继续执行。

2.9. 常数吞吐量分布式使用示例

常数吞吐量定时器（Constant Throughtput Timer）通常用于控制吞吐量，您可以根据压测脚本的业务目的，来选择不同的计算模式。若叠加上分布式施压源，您需要考虑脚本中的配置值及配置模式，以便匹配不同的压测目标模型。本文将从应用场景的角度介绍2种常见的使用模式，以及分布式适配不同计算模式的效果。

背景信息

通过以下示例您可以了解到不同的分布式适配方式。假设使用到2台施压IP，并发为100，脚本上仅1个线程组，其吞吐量目标为每分钟100，计算模式为当前线程（this thread only）。

- 全局生效即为脚本中设置值为集群整体阈值，施压机会根据使用到的IP数来拆分到单机吞吐量目标值上（即单施压机阈值为脚本中值/IP数），此时场景的总目标吞吐量为2×（100并发/2 ）×（100/2）= 5000（每分钟）。
- 单机生效即为脚本中设置值为单台施压机的目标值，不会替换脚本内容，需要注意并发量级与配置值是否匹配。该模式下，场景的总目标吞吐量为2×（100并发/2 ）×100=10000（每分钟）。

常见应用场景

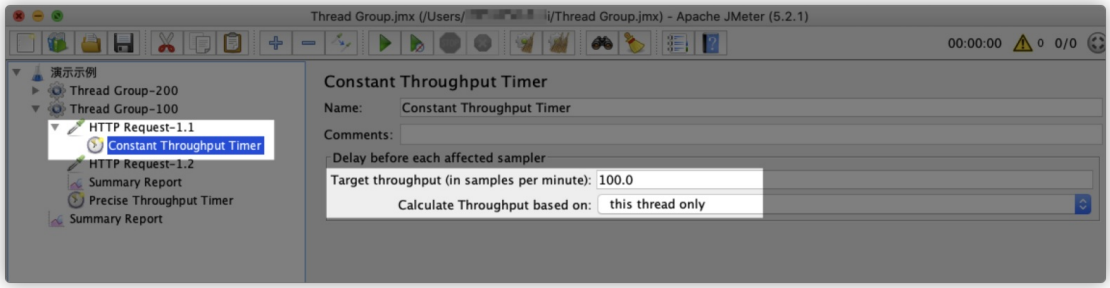
通常配置吞吐量控制的时候，您可以选择Sampler维度或线程组维度（多线程组即全场景维度）即可，不推荐使用其他过于复杂的场景。以下示例中使用了2台施压机IP，线程组总并发100，目标吞吐量每分钟100。

- Sampler吞吐量控制的实现
  i. 登录PTS控制台，在左侧导航栏选择**性能测试 > 创建场景**，然后单击**JMeter压测**。
  ii. 选择**场景配置**页签，在**常数吞吐量定时器**中选择**单机生效**。

> **说明** 只有上传的脚本中设置了定时器，**场景配置**中才会出现**分布式适配设置**的配置框。

iii. JMeter脚本中将该组件（Constant Throughtput Timer）置于Sampler下面。

iv. 设置**Target throughtput**为 `100` ，单位为分钟。**计算模式**选择 `this thread only` ，如下图所示。



- 线程组吞吐量控制的实现

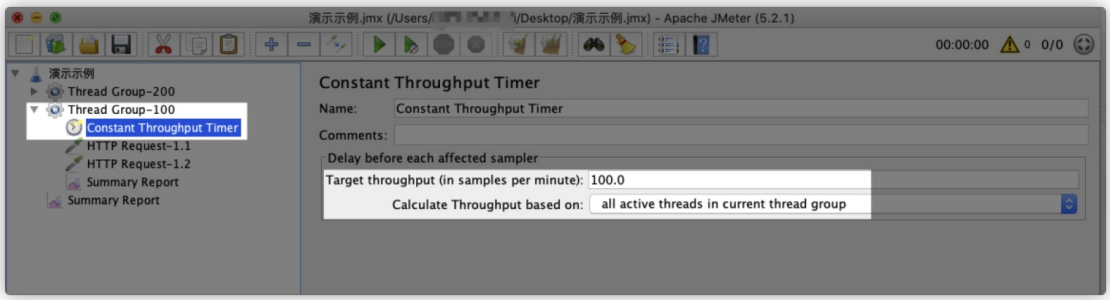
i. 登录PTS控制台，在左侧导航栏选择**性能测试 > 创建场景**，然后单击**JMeter压测**。

ii. 选择**场景配置**页签，在**常数吞吐量定时器**中选择**全局生效**。

> **说明** 只有上传的脚本中设置了定时器，**场景配置**中才会出现**分布式适配设置**的配置框。

iii. JMeter脚本中将该组件（Constant Throughtput Timer）置于Sampler下面。

iv. 设置**Target throughtput**为 `100` ，单位为分钟。**计算模式**选择 `All active threads in current thread group` ，如下图所示。



5种计算模式

示例：假设使用单台施压源，2个线程组，即分别含有2个Sampler，分别是100并发和200并发，**Target throughtput** 均是每分钟100。

> **说明** 以下示例仅供参考，具体请以官方文档为准，请参见官方文档。

基于的计算方式	说明	其他信息
this thread only	每个线程单独计算，那么总的吞吐量就是并发 × Target throughtput。	非活跃的线程，没有对应的吞吐量，按照上述示例（每分钟）： • 线程组1：每个线程每分钟发出请求100次，即Sampler1.1和Sampler1.2各自的吞吐量都是100线程×100次/60秒=83左右。 • 线程组2：每个线程每分钟发出请求100次，即Sampler2.1和Sampler2.2各自的吞吐量都是200线程×100次/60秒=160左右。 • 配置到Sampler上即可控制到单个Sampler的吞吐量。
All active threads	将设置的吞吐量，分配到活跃的线程上（所有线程组的所有线程），然后这个线程在上次运行自行结束之后，等待合理时间（为了控制吞吐量）再次运行。	多线程组的时候，要求其他线程组也有一样的配置按照上述示例： • 因为线程是一开始就发出去，所以在执行第二个Sampler的时候，吞吐量就会非常低。因为是从线程组维度控制的，每个线程吞吐量分别是每分钟为1和2，所以到第二个Sampler的时候，次数会减少。 • 线程组的Throughtput是每分钟为100，全局的吞吐量是每分钟为200。 • 需要运行一段时间之后（线程组内多个Sampler发出去的请求数一致=并发数），每个线程组内Sampler的吞吐量才会趋于均衡。
All active threads in current thread group	将设置的吞吐量，分配到活跃的线程上（当前线程组的活跃线程），然后这个线程在上次运行自行结束之后，等待合理时间（为了控制吞吐量）再次运行。	单线程组的时候，与上一个相同。按照上述示例： • 因两个Timer的配置一致，故和上述情况一致。 • 当把第二个Timer的吞吐量改为每分钟为200时，每个线程组可单独控制。

基于的计算方式	说明	其他信息
All active threads （shared ）	将设置的吞吐量，分配到活跃的线程上（所有线程组的所有线程），所有线程都会在所有活跃线程运行结束之后即所有线程结束之后，等待合理时间（为了控制吞吐量）再次运行。	类似全场景的多次延迟释放，因为需要等待所有活跃线程运行结束，按照上述示例分析如下： • 在压测过程中，可以看到不会有多个Sampler的数据同时变化。 • 已经发起过请求的线程会被暂停在第一个Sampler上。 • 全场景的Target throughtput是为每分钟100，且多个timer的配置保持一致。
All cative threads in current thread group （shared）	将设置的吞吐量，分配到活跃的线程上（当前线程组的活跃线程），所有线程都会在所有活跃线程运行结束之后即所有活跃线程结束之后，等待合理时间（为了控制吞吐量）再次运行。	当前线程组中，活跃线程组中线程结束之后，再等待一点时间运行。 • 线程组的Target throughtput为每分钟100，但每个线程会考虑全局其他活跃线程。 • 线程组的吞吐量可以单独配置和控制。

其他计算模式的分布式适配情况

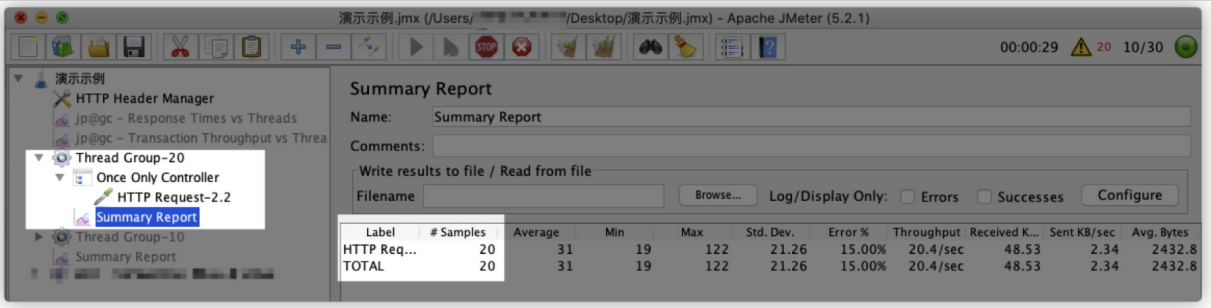
基于的计算方式	应用场景	全局生效	单机生效
All active threads	全场景吞吐量控制驱动：多个线程组的业务模型类似，且全场景的吞吐量固定，即可拆分到单个线程组的均匀吞吐量。	单机并发线程是50，Target/2替换为脚本中的值，即单机上的吞吐量变成脚本中的一半为每分钟50。全局的吞吐量不变化仍然为每分钟100。当有多个线程组的时候，还会多个线程组累加。	单机并发线程是50，单机上的吞吐量为每分钟100，全局吞吐量变为脚本中的值×IP数，即为每分钟200。当有多个线程组的时候，还会多个线程组累加。
All active threads （shared ）	全场景吞吐量不区分线程组，略低于多线程组配置值叠加。	线程组1配置是100，线程组2配置是200，则全局为150，即两个线程组的均值。	线程组1配置是100，线程组2配置是200，则单机为（100+200）/2，全局为300，但是单机多个Sampler达到均衡的时间更长。
All cative threads in current thread group （shared）	线程组的吞吐量考虑全局活跃，略低于多线程组配置值叠加。	线程组1配置是100，线程组2配置是200，则全局为300，是线程组1和2的配置数累计之和。	线程组1配置是100，线程组2配置是200，则单机为100+200，全局为600，但是单机多个Sampler达到均衡的时间更短。

2.10. 为什么脚本中设置了Once Only Controller，压测的时候还是会重复发请求？

本文介绍为什么脚本中设置了Once Only Controller，压测的时候还是会重复发请求。

问题现象

如果您在本地JMeter脚本中配置了一个线程组，并在其内部配置了仅一次控制器（Once Only Controller），但是将脚本上传至PTS的JMeter压测后，线程组的请求还是会循环重复执行。如下图所示。

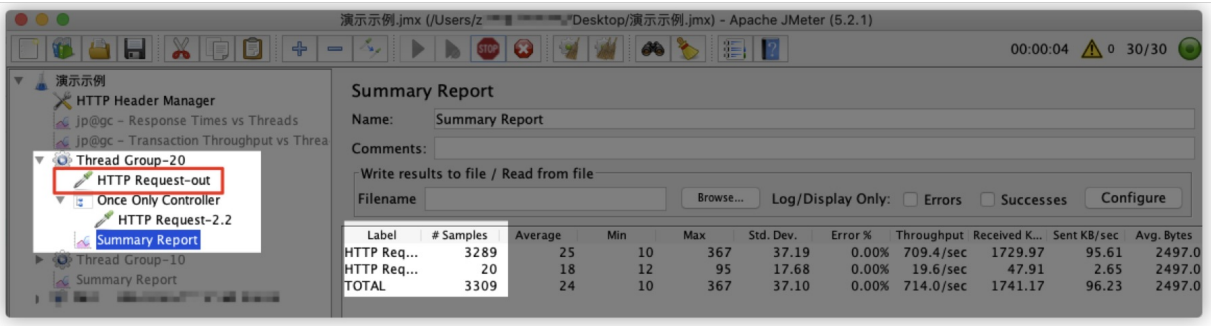


可能原因

因为PTS中增加了可调速及设置非固定量级施压的功能，会改变原生的Thread Group设置，使得JMeter脚本中Once Only Controller的配置不生效。此外，如果JMeter脚本中只有一个可执行的Once Only Controller，则用PTS进行压测和脚本在本地调试并无区别，达不到性能测试的目的。

解决办法

请在同一个线程组中，保证除Once Only Controller外，仍有其他生效的Sampler，这样Once Only Controller中的内容只会执行一次，而其余的Sampler会根据配置循环执行。



3.参数化和函数

3.1. PTS在并发和RPS模式下读取多文件参数的方式

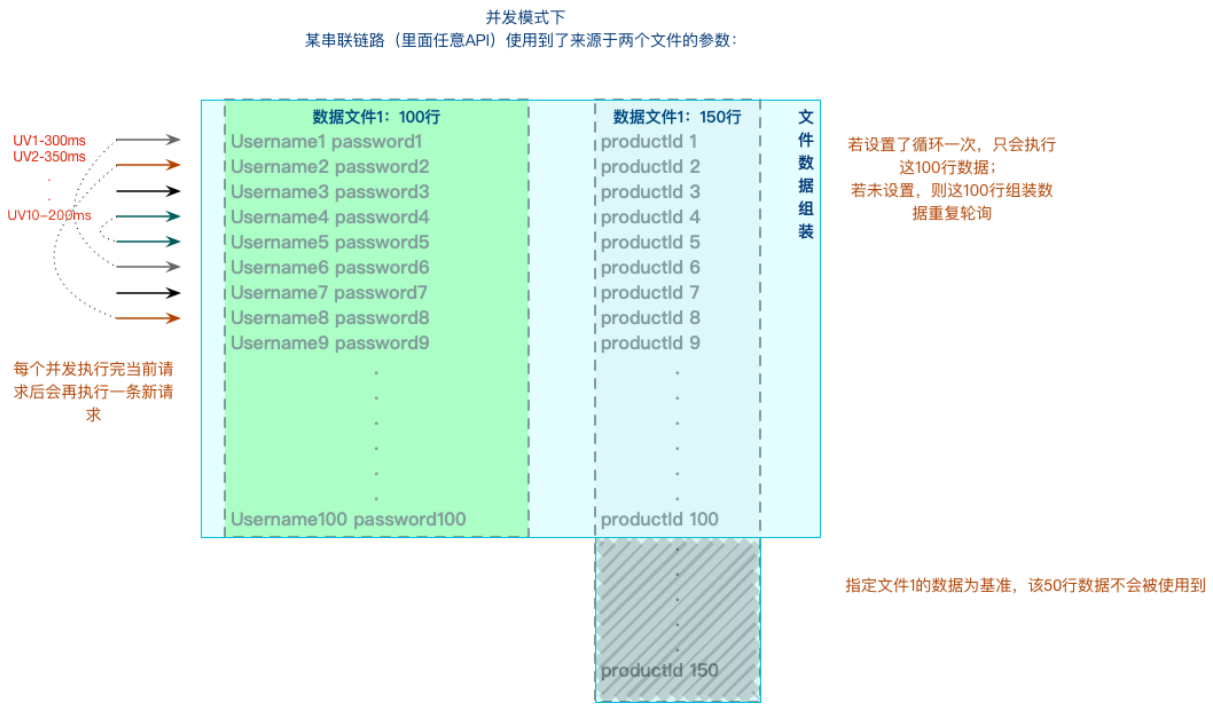
本文介绍在并发模式和RPS模式下，PTS读取多文件参数的方式。

并发模式下某API文件读取示例

说明如下：

- 若使用的参数来源于多个文件时，可以指定基准参数，PTS会先按照基准参数的行数进行组装。
- 若为某参数设置轮询一次后，则以该参数为基准。若先设置了基准参数，还可设置是否轮询一次。
- 并发模式下，每个并发当前请求处理完成（含收到响应或者超时）后，会再去读取新的数据，发出一条新请求。

指定行数较少参数为基准文件



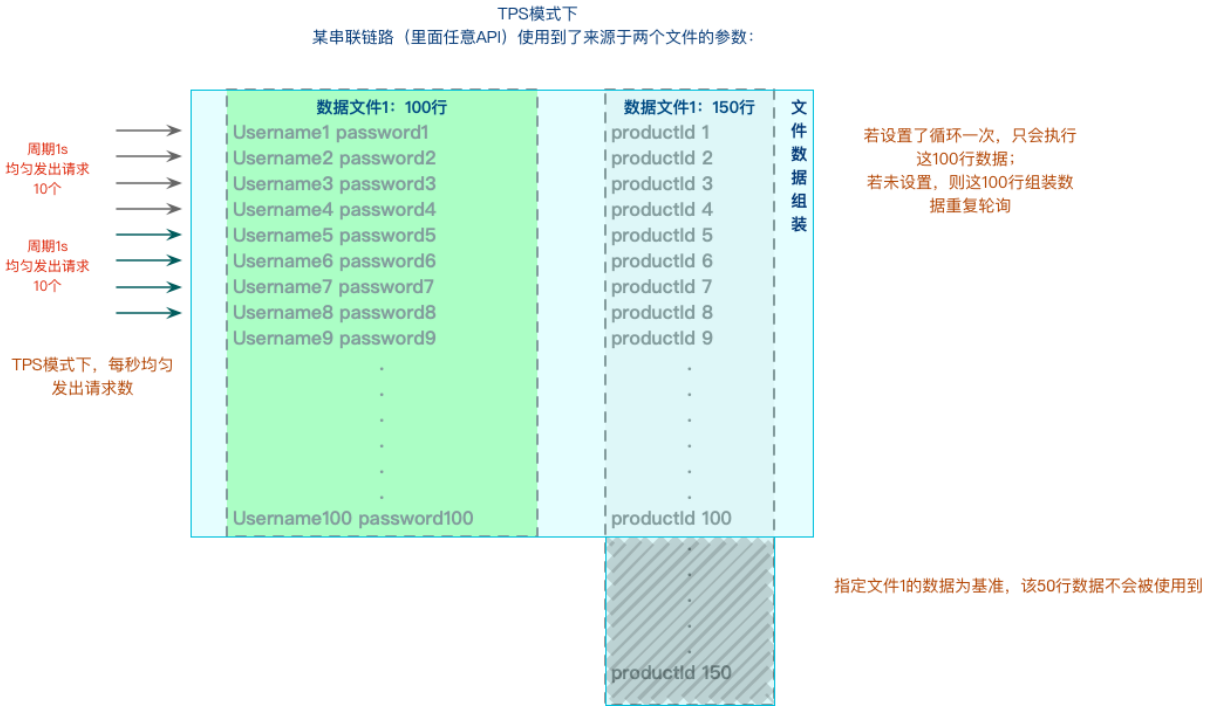
指定行数较多参数为基准文件

RPS模式下某API文件读取示例

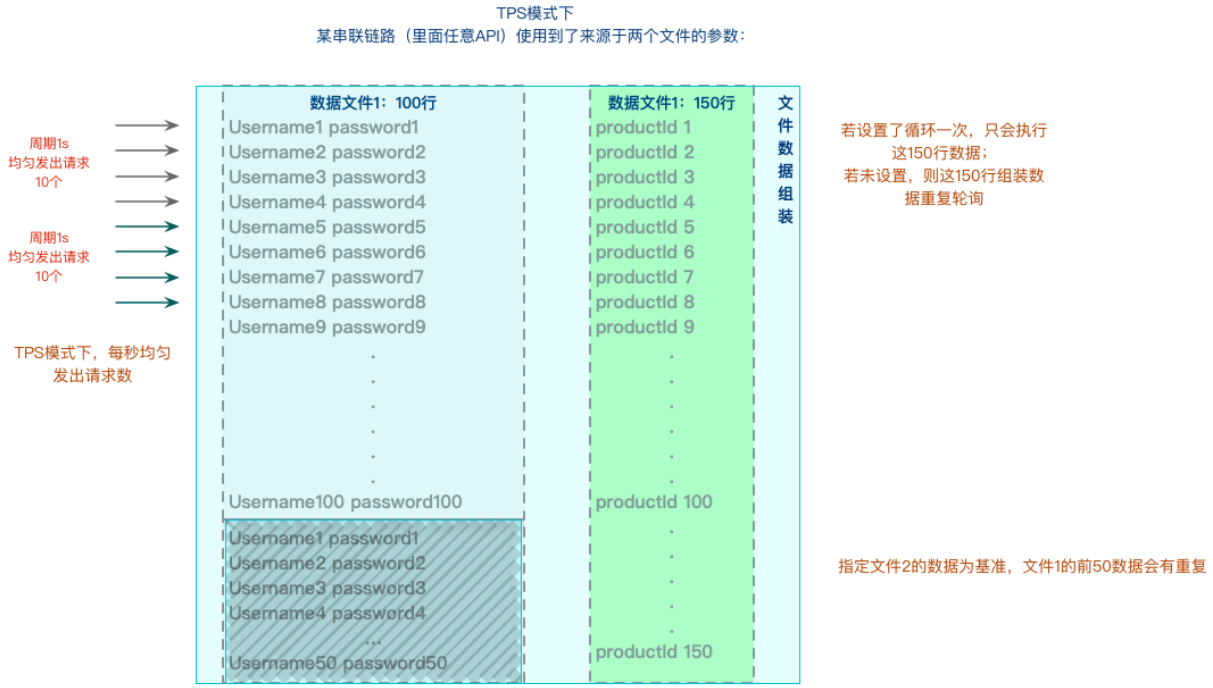
说明如下：

- 若使用的参数来源于多个文件时，可以指定基准参数，PTS会先按照基准参数的行数进行组装。
- 若为某参数设置轮询一次后，则以该参数为基准。若先设置了基准参数，还可设置是否轮询一次。
- RPS模式下，每秒均匀发出固定请求数（当前施压值）。如果请求处理快，则只需要较少的并发即可；如果处理慢，则需要更多的并发。

指定行数较少参数为基准文件



指定行数较多参数为基准文件



3.2. 四则运算的使用

本文介绍四则运算的使用方法。

目前四则运算一共包含：加法、减法、乘法、除法（基础）、除法（高级）、舍入模式六种运算法则。

其中加法、减法、乘法、除法（基础）计算结果涉及到小数位的时，默认四舍五入保留2位小数。如：1.234 * 5 = 6.17，1/5 = 0.20。

除法（高级）中后两位参数即与舍入模式一致，详见以下舍入模式的介绍。

舍入模式一共有七种模式，每种模式均需要指定保留的小数位数（即scale (1,2) 中的1所在参数位置，1表示保留1位小数，2表示round_mode参数为2），round_mode详解如下：

> **说明** 假设均为指定保留2位小数。

round_mode参数值	参数释义	详细说明
0	舍入远离0模式	例如，1.234舍入结果为1.24，-1.234结果为 -1.24
1	舍入接近0模式	例如，1.234舍入结果为1.23，-1.234结果为 -1.23
2	接近正无穷大的模式	例如，1.234舍入结果为1.24，-1.234结果为 -1.23
3	接近负无穷大的模式	例如，1.234舍入结果为1.23，-1.234结果为 -1.24

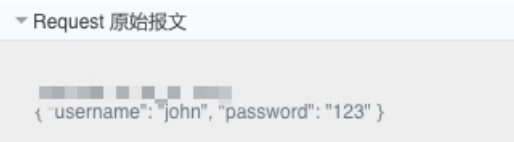
round_mode参数值	参数释义	详细说明
4	向最接近的数字舍入，若两个相邻数字距离相等，则舍入接近0	该模式为我们熟知的“四舍五入”模式。距离相等时，同模式0， 距离不等时，同模式1。例如，1.235舍入为1.24，-1.235舍入为 -1.24。
5	向最接近的数字舍入，若两个相邻数字距离相等，则舍入远离0	即“五舍六入模式”。 距离相等时，同模式1， 距离不等时，同模式0。例如，1.235舍入为1.23，-1.235舍入为 -1.23。
6	向最接近的数字舍入，若两个相邻数字距离相等，则向偶数舍入	即“银行家算法” ：四舍六入，五分为两种算法（如果前一位为奇数，则入位，否则舍去）。例如，1.235舍入结果为1.24，1.245舍入结果为1.24。

3.3. 为什么调试和压测时Body中JSON的格式和设置的不同（如空格、回车）？

本文介绍调试和压测时Body中JSON的格式和设置的不同（如空格、回车）的原因。

通常在请求体编辑的时候，为了可读性会将JSON进行适当地换行、缩进，如下图。

而PTS实际在处理请求构建时会去除键和值以外的空白、换行字符，如下图。



即使在请求API编辑的时候没有设置过缩进和换行，PTS在调试时为了查看的便捷也会在Request Body结构化里进行适当地编排便于阅读，但是实际请求报文还是参考Request原始报文。

3.4. 系统函数及字符串如何组合嵌套使用？

本文介绍系统函数及字符串如何组合嵌套使用。

如果参数化的部分（等号右侧）需要系统函数搭配字符串或者另一个系统函数，规则是直接拼接即可，不需要额外的连接字符。参考以下的例子：

- 在函数前加字符串：

```
=abc${sys.random(1,20)}
```

```
=-234${sys.random(1,20)}
```

- 两个函数拼接：

```
=${sys.random(1,20)}${sys.select("a","b","c")}
```

如果系统函数的参数中有单双引号，直接使用单引号，双引号需要在前面再加一个双引号。例如需要给一个字符串做MD5加密（系统函数是 `${sys.md5("")}` ），以字符串包含单引号和双引号来分别举例参数化（等号右侧）写法：

- 需加密字符串是 `leo say 'hi'` 时：

```
=${sys.md5("leo say 'hi'")}
```

- 需加密字符串是 `leo say "hi"` 时：

```
=${sys.md5("leo say ""hi""")}
```

如果系统函数间需要嵌套使用时，不需要额外加双引号，但是函数的参数部分用到了文件参数、自定义参数（如通过字符串定义的和用系统函数定义的）或有拼接的情况才需要加双引号。参考以下例子：

- substring的字符串是直接拼接的情况，需要带双引号：

```
=${sys.substring("abc${sys.random(1,20)}", 0, 1)}
```

- substring的字符串是函数，所以不需要双引号，但是嵌套的MD5里面有拼接就需要双引号：

```
=${sys.substring(${sys.md5("${input1}${input2}")},2,5)}
```

- 需要给来源于数据文件的参数num2做一个base64的时候，需要加个双引号：

```
=${sys.base64("${num2}")}
```

- 更复杂的一种情况，函数的参数是字符串，这个字符串是一个JSON的情况，JSON里也带了文件参数：

需要传入函数中的JSON原文是： `{"username":"${username}"}` **则函数的嵌套使用如下：**

```
=${sys.select("{""username"":""${username}""}","blue","green")}
```

4.错误信息

4.1. 请求出现405错误的原因是什么？

405错误一般指请求 `method not allowed` 错误，本文列出了出现该错误的可能原因。

出现405错误的可能原因有：

- POST类请求出现302跳转，302跳转的时候会更改请求方法此时服务端可能不能识别，则报405错误。
- 请求服务端直接校验Method，对应Response Header中会有 `Allow =GET` 的信息字样。
- 负载均衡或者Web Server上做转发的时候，修改了请求Method导致后端无法识别。

> **注意** 若以上信息未帮助您解决问题，请提工单联系阿里云技术支持进行排查。

4.2. 为何调试或压测时出现403但浏览器访问正常？

Condition

浏览器访问正常，在PTS对请求进行调试或压测的时候却出现403错误。

Cause

服务端网关有强校验Header中的UA（User-Agent），对带有不合法的UA的请求返回无权限的信息。PTS发起的请求中默认的UA会带有特殊字样，为部分业务来区分统计流量和限流规则。

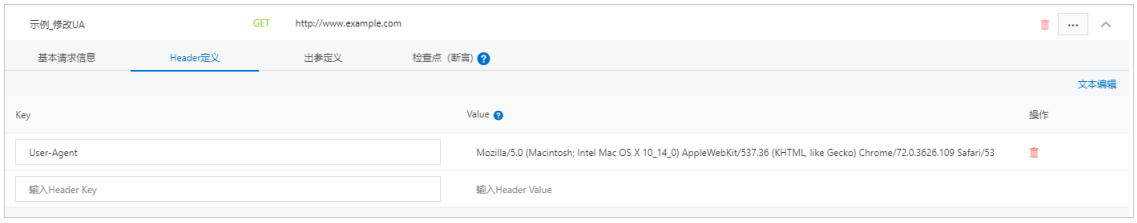
Remedy

操作步骤

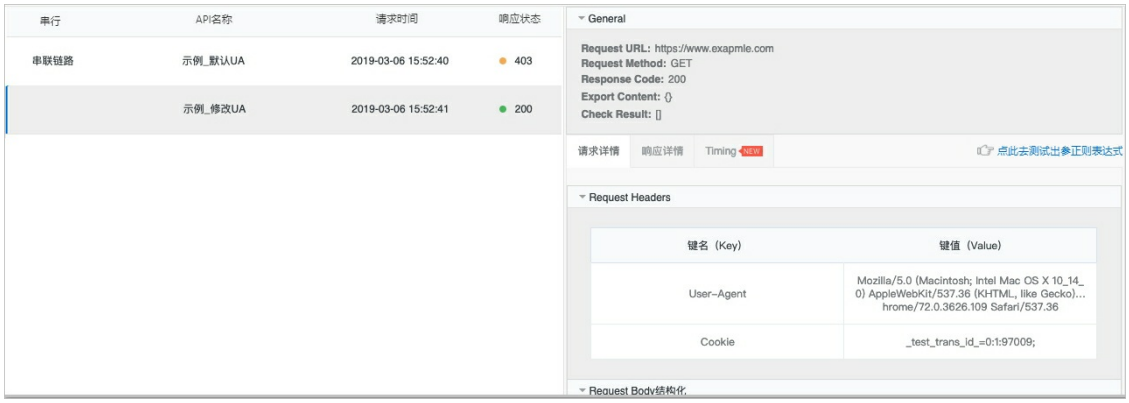
1. 返回压测场景中API的编辑页面，在**Header定义**中添加一个通用的UA Header。

   例如添加：

   - Key：`User-Agent`
   - Value：`Mozilla/5.0 (Macintosh; Intel Mac OS X 10_14_0) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/72.0.3626.109 Safari/537.36`

2. 再次进行压测场景调试，观察请求是否正常。如修改UA后请求正常，则可以判定是UA校验引起的，您可以通过修改UA来继续压测。



Cause

被WAF拦截（这种情况可能性会比较小）。

Remedy

操作步骤

1. 如果需要有WAF白名单限制，请根据文档如何避免PTS的压测流量被Web应用防火墙拦截设置允许PTS流量通过的规则。

Cause

压测域名未备案，或域名解析到了未备案的其他域名。

您可以根据返回结果判断域名是否备案。若出现403，且返回如下HTML信息，则是因为域名未备案导致的报错。

```
<html>
<head>
<meta http-equiv="Content-Type" content="textml;charset=UTF-8" />
   <style>body{background-color:#FFFFFF}</style>
<title>TestPage184</title>
  <script language="javascript" type="text/javascript">
         window.onload = function () {
           document.getElementById("mainFrame").src= "http://batit.aliyun.com/alww.html";
            }
</script>
</head>
  <body>
    <iframe style="width:860px; height:500px;position:absolute;margin-left:-430px;margin-top:-250px;top:50%;left:50%;" id="mainFrame" src="" frameborder="0" scrolling="no"></iframe>
    </body>
</html>
```

Remedy

操作步骤

1. 对域名进行备案，具体操作，请参见ICP备案流程概述。
   备案成功后，再次压测此域名即可。

4.3. 调试或者压测时出现406是什么原因？

Condition

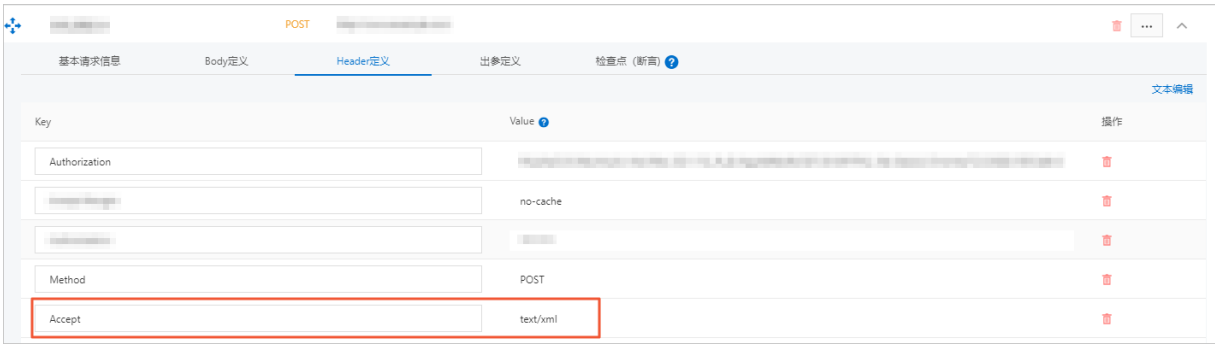
调试或者压测时出现406。

Cause

在调试时，请求出现406报错，一般是请求构建的Header中的Accept字段设置错误导致的。

Accept代表发送端（在这里表示PTS）希望接受的数据类型。

Content-Type代表发送端（在这里表示PTS）发送的实体数据的数据类型。在PTS里Body中设置的Content-Type会自动同步到Header中。如果Header中的Accept不符合事先约定的内容，就会返回406错误。



操作步骤

1. 确认服务端能验证通过的Accept类型。
2. 尝试设置不同的Accept类型Value。

   Accept参考信息如下：

   以下是Accept的格式类型和匹配顺序，供参考。

   - text/html：HTML格式
   - text/plain：纯文本格式
   - text/xml：XML格式
   - image/gif：GIF图片格式
   - image/jpeg：JPG图片格式
   - image/png：PNG图片格式
   - application/xhtml+xml：XHTML格式
   - application/xml：XML数据格式
   - application/atom+xml：Atom XML聚合格式

- application/json：JSON数据格式
- application/pdf：PDF格式
- application/msword：Word文档格式
- application/octet-stream：二进制流数据，例如常见的文件下载
- application/x-www-form-urlencoded：`<form encType="">`中默认的encType，form表单数据被编码为key/value格式发送到服务器（表单默认的提交数据的格式）。

Accept应用规则如下：

- 当Accept头有`application/xml`、`text/html`、`application/json`，将按照如下顺序进行produces的匹配：

  `application/xml` > `text/html` > `application/json`

- 当Accept头有`application/xml;q=0.3`、`application/json;q=0.8`、`text/html`，将按照如下顺序进行produces的匹配：

  `text/html` > `application/json` > `application/xml`

  参数为媒体类型的质量因子，数字越大则优先权越高（从0到1）。

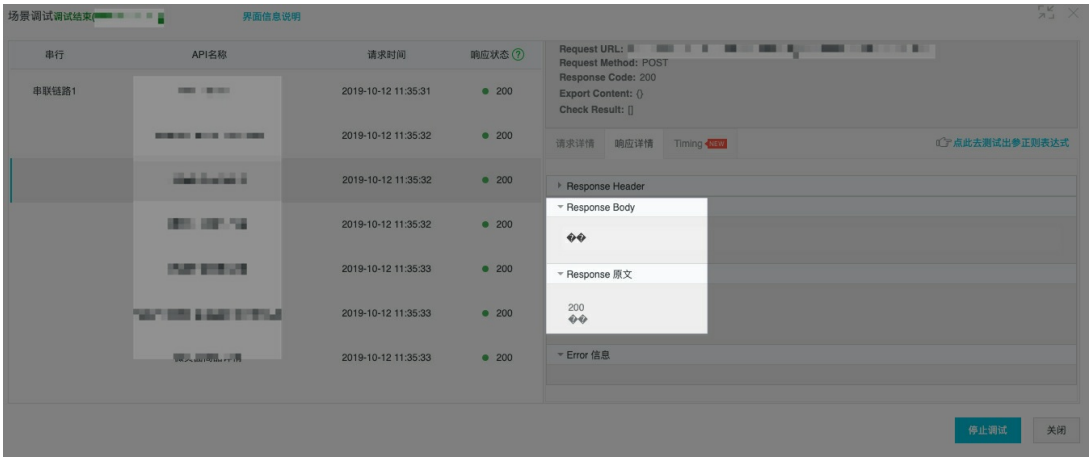
- 当Accept头有`*/*`、`text/*`、`text/html`，将按照如下顺序进行produces的匹配：

  `text/html` > `text/*` > `*/*`

4.4. 为什么调试/采样日志中响应Body是乱码？

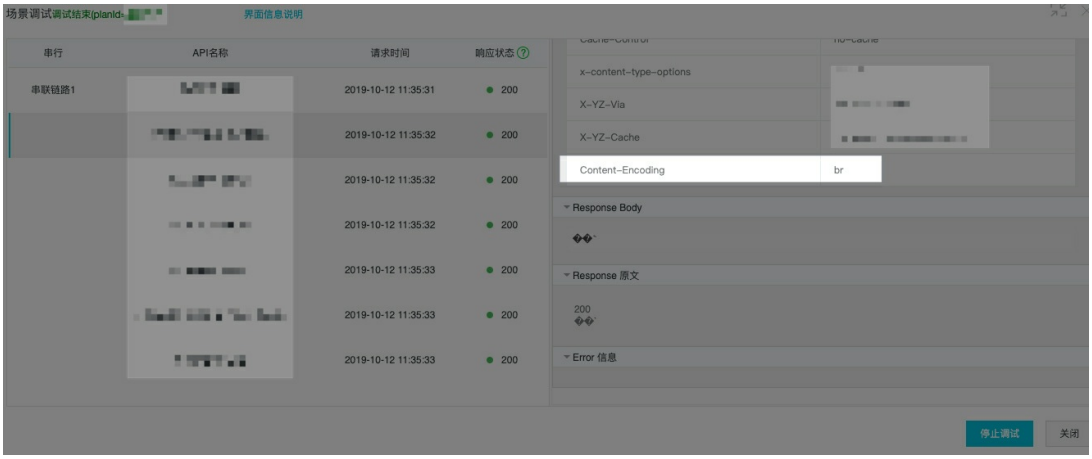
Condition

使用PTS压测的场景调试时，调试结果的Body显示是乱码，但是实际线上业务（PC页面或者APP）是正常的。形态如下：



在Response Header中配置了压缩方法，导致返回的Body内容被压缩。压缩方法一般有两种：

- Content Encoding：gzip
- Content Encoding：br



4.5. 为什么后端压力不大但压测时报错或超时?

Condition

后端压力不大，但是压测时已经大量报错或者超时。

Cause

一般这种情况，往往是服务的最外层，即网络接入层，有瓶颈或者触发了某些阈值导致的。

Remedy

请务必对已经使用网络接入层（网络入口）的产品进行监控观察。常见的产品类型和响应的注意事项如下：

操作步骤

- 使用了SLB的业务

  请结合购买的产品计费类型，关注规格限制（最大连接数、CPS每秒新建连接数、QPS）和带宽限制。

  当SLB是服务最外层，且接口是HTTPS或者开启了七层会话保持功能，压测中出现一些503错误，而且后端并没有相关流量和日志，可能出现了SLB单IP地址限流问题。

- 使用了**高防IP和WAF**的业务

  由于压测的某些特征符合CC和DDoS的行为，很容易触发对应产品防护策略，导致压测数据不准确、压测失败等异常情况。建议您在评估业务影响的前提下，做一些暂时关闭的调整，或者设置放行PTS的压测流量。

- 使用了**CDN或者全站加速**的业务

  如果接近或者超出已有业务峰值时，请提前提交工单报备给这两个产品。

>

4.6. 为什么PTS的采样日志上有大量的503，但后端服务器上没有相关信息?

Condition

PTS的采样日志上有大量的503，但是后端服务器上确没有相关信息。

Cause

如果压测的接口错误中有很多503，同时满足以下的现象，则说明该报错是SLB抛出的。因为压测场景中发起压力的源IP有限，单IP触发了SLB集群的单Proxy限流；或者因为请求客户端的Connection会默认长连，IP较少时，会导致其不能完全做到负载均衡。

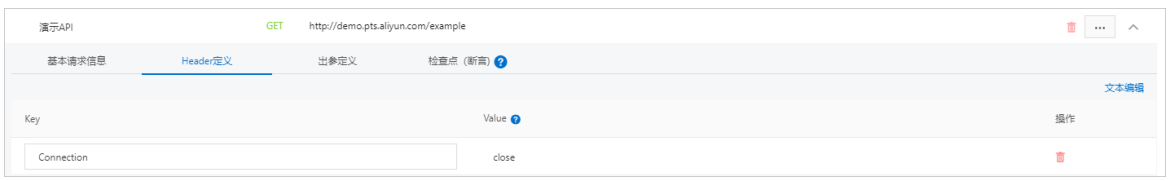
- 接口是HTTP/HTTPS
- 压测环境入口的是SLB，无论是公网还是内网SLB
- 后端服务并没有抛出503，甚至没有任何记录
- 503报错信息与下面内容相同：

```
<!DOCTYPE HTML PUBLIC "-//IETF//DTD HTML 2.0//EN">
<html>
<head><title>503 Service Temporarily Unavailable</title></head>
<body bgcolor="white">
<h1>503 Service Temporarily Unavailable</h1>
<p>The server is temporarily unable to service your request due to maintenance downtime o
r capacity problems. Please try again later.</body>
</html>
```

Remedy

操作步骤

1. 升级PTS资源包。PTS 9.9元资源包由于成本原因只有一个发起源IP，只要是628及以上资源包即可享有最大70个及以上源IP。
2. 使用IP扩展功能。
3. 设置更高的目标并发或者RPS。
4. 在请求header中设置Connection:close的配置。新建API会默认配置，修改的时候可按需使用。



4.7. 压测和调试日志中常见的Error信息有哪些？分别表示什么意思？

本文介绍压测和调试日志中常见的Error信息。

以下为压测和调试日志中，常见的Error信息：

- class java.net.SocketTimeoutException:null

  表示请求在等待响应或者读取中途（idle）超时。请检查服务端健康状况或者PTS的压测API超时时间的设置是否合理，另外还有可能是服务端处理能力出现瓶颈。

- class java.net.ConnectException:null

  表示请求在与远端（被压测端）建立TCP连接时就出现失败或者被远端拒绝。请检查服务端健康状况，或者是网络连接层是否有瓶颈。

- class java.util.concurrent.TimeoutException:null

  表示请求在与远端（被压测端）建立TCP连接时就出现失败或者被远端拒绝。请检查服务端健康状况，或者是网络连接层是否有瓶颈。

- class org.apache.http.ConnectionClosedException:Connection closed

  表示连接异常关闭，服务端主动关闭了连接。

- class java.io.IOException:Connection reset by peer

  表示连接被重置。若使用了SLB，请查看SLB的配置是否有问题。

- class org.apache.http.ConnectionClosedException:Connection closed unexpectedly

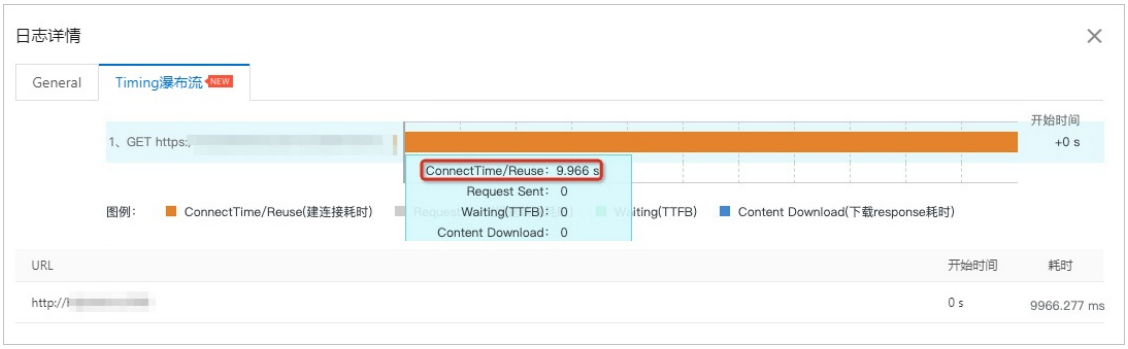
  表示数据尚未接收完毕，连接就已关闭。可能服务端未及时响应或者提前终止调试或压测。

- class java.lang.RuntimeException:java.net.UnknownHostException

  表示域名信息无法解析。请检查域名是否已经正常注册并可以解析、未注册的域名是否已进行域名绑定。

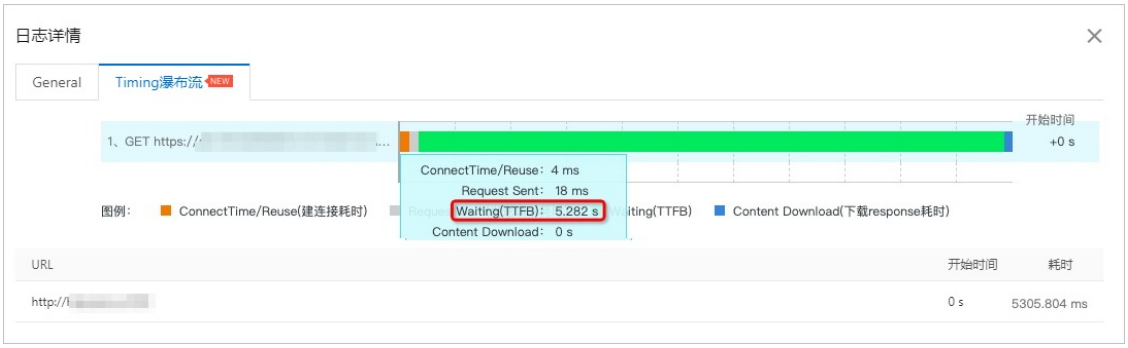
建议结合Timing瀑布模型查看，各种报错都可以体现在Timing瀑布模型中，例如：

- 建立连接环节超时。



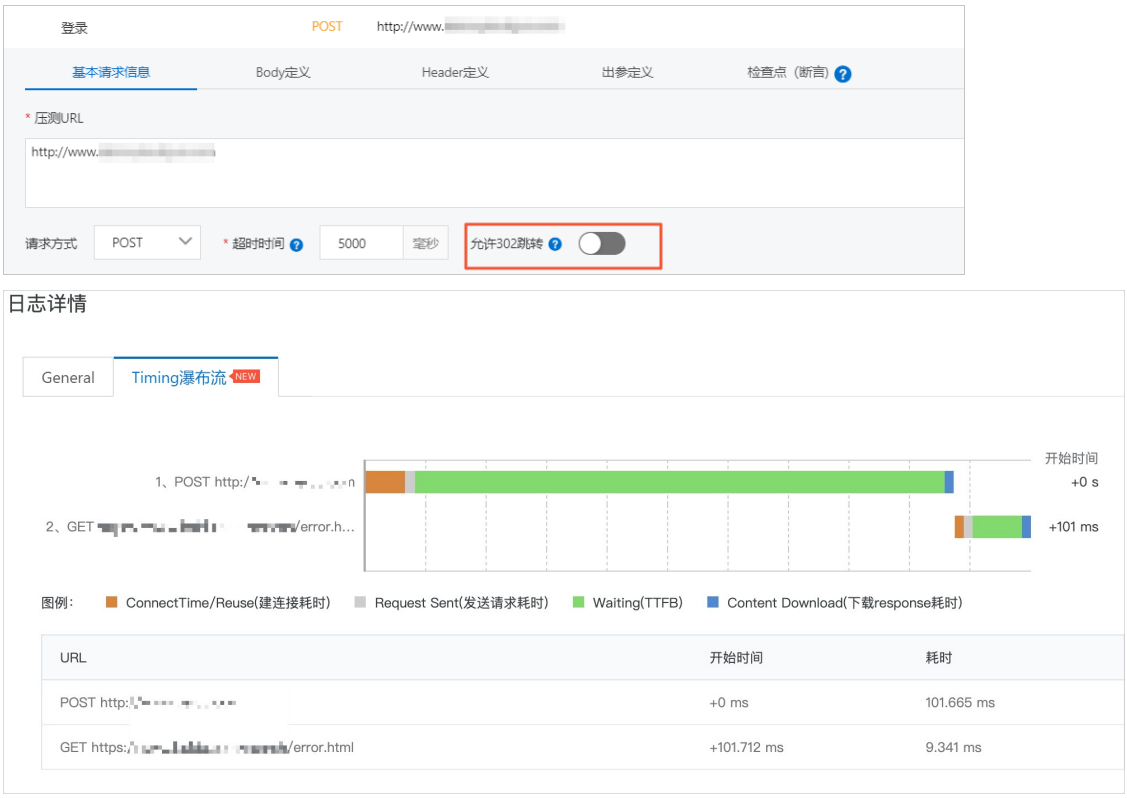
- 等待响应或者读取响应的间隔时间超时。

  Waiting（TTFB）表示等待第一个响应字节的时间。这个时间包括一次完整的往返和延迟，同时包括了服务器应对响应所用的时间。



- class org.apache.http.client.CircularRedirectException

  表示请求出现了循环重定向的情况（A -> B -> C -> A），或者跳转超过了10次（A1 -> A2 -> A3... -> A10 ->A11）。建议取消302跳转配置后压测查看原始请求信息，并可结合Timing瀑布流查看跳转具体路径。



- class org.apache.hc.core5.http.ProtocolException:Header 'key: value' is illegal for HTTP/2 messages

  表示在服务端优先使用HTTP2协议的情况下，场景配置了HTTP2协议不支持的Header，请移除相应Header后重试。HTTP2不支持的常见Header有：Connection、Keep-Alive、Proxy-Connection、Transfer-Encoding、Host、Upgrade。

4.8. PTS的压测流量被Web应用防火墙拦截，怎么办？

Condition

PTS的压测流量被Web应用防火墙拦截。

Cause

如果您使用PTS进行压测的服务同时使用到了Web应用防火墙（Web Application Firewall，简称WAF），可能会因为压测流量较大而触发一些拦截、阻断和清洗，进而影响压测的顺利进行。

Remedy

解决的方案有两种：

- 绕开WAF，例如直接压测后面的SLB公网IP或者其他类型的公网IP。
- 设置WAF网站白名单，配置一个基于Header的网站白名单规则，放行所有PTS的压测请求，具体操作如下。

操作步骤

1. 登录Web应用防火墙控制台。
2. 在顶部菜单栏，选择Web应用防火墙实例的资源组和地域（**中国内地**、**非中国内地**）。
3. 在左侧导航栏，选择**防护配置 > 网站防护**。
4. 在**网站防护**页面上方，切换到要设置的域名。

5. 单击页面右上角的**网站白名单**。

6. 新建网站白名单规则，并设置匹配条件为 `Header包含x-pts-test` 。

   PTS发送的HTTP请求中，会带有x-pts-test的Header，因此需新建一个网站白名单规则并将其匹配条件设置为 `Header包含x-pts-test` 。更多有关网站白名单设置的信息，请参见设置网站白名单。

   > **说明**
   > - 建议不选中任何后续安全策略，避免请求继续被拦截。
   > - 请注意WAF自身的规则匹配顺序，避免设置的以上规则失效。
   > - 出于安全考虑，压测完成之后请删除该放行规则。

5.其他

5.1. 如何模拟实现验证码的操作？

本文介绍模拟实现验证码操作的方法。

在网站登录页面，通常还需要输入验证码。若您在压测时需要模拟多用户登录的场景，您可以在业务代码中配置一个万能验证码，并且在PTS配置API时带上万能验证码，然后通过在压测场景中导入包含用户名和密码的参数文件，构造压测API时关联导入的参数来实现。

5.2. 为什么URL中的井号（#）及后续内容保存之后自动省略了？

本文介绍URL中的井号（#）及后续内容保存之后自动省略的原因。

一般而言，混合（Native + Web）框架的App常使用单页面，一个单页面的URL中会包括一个井号（#），用于指定网页中的一个位置，其右侧的字符就是该位置的标识符。如http://www.example.com/index.html#abc就代表网页index.html的abc位置。浏览器读取这个URL后，会自动将abc位置滚动至可视区域。所以，井号（#）是用来指导浏览器动作的，对服务器端完全无用。PTS在保存URL的过程中也会略去井号（#）和后续的内容。

5.3. 压测的请求带宽和响应带宽是如何统计的？

本文介绍压测的请求带宽和响应带宽的统计方法。

首先，PTS中看到的请求带宽和响应带宽都和SLB或者ECS中看到的带宽不同。具体的计算逻辑如下：

- PTS中统计的请求带宽是基于发送的实际的HTTP请求体大小。
- PTS中统计的响应带宽的计算方式是：HTTP响应头字节数 + Content - Length（没有该值则统计响应的BODY实际字节数）。

最后，基于上面的逻辑最终的计算方式为：每台施压机器5s为周期计算周期内的平均值（响应大小或请求大小，会拉平峰值），用户看到的统计信息为施压机的计算值之和（高并发下会分配多台施压机进行施压）。

5.4. 文件和输入框的最大限制是多少？

本文介绍文件和输入框的最大限制。

主要的输入框及文件部分的最大限制情况如下。

文件

文件（CSV）最大支持60 MB。

单行最大支持2 W个字符。

界面输入框

URL输入框最大支持2083个字符。

Body输入框最大支持65535个字符。

5.5. 压测报告中的分位值是什么含义？

压测报告中，概览统计会出现分位值。本文介绍分位值的意义和计算方法。

1. 分位值的意义是什么？

分位值即把所有的数值从小到大排序，取前N%位置的值，即为该分位的值。

- 一般用分位值来观察大部分用户数据，平均值会“削峰填谷”消减毛刺，同时高分位的稳定性可以忽略掉少量的长尾数据。
- 高分位数据不适用于全部的业务场景，例如金融支付行业，可能就会要求100%成功。

2. 分位值是如何计算的？

以95分位值为例： 将采集到的100个数据，从小到大排列，95分位值就是取出第95个用户的数据做统计。

同理，50分位值就是第50个人的数据。

5.6. PTS是否可以压测微信小程序的场景？

PTS支持HTTP/HTTPS协议的压测，无论客户端是自研的App、移动端网页、PC端网页、微信小程序，还是C/S结构的软件，只要协议是HTTP/HTTPS，PTS就能支持压测。

但是压测需要知道请求构造的细节，例如请求体、请求头的情况，哪些部分需要做参数化等，然后再基于PTS进行设置即可。当然，如果请求体都不清楚，也可以通过指定的PTS资源包提供的**云端录制**功能进行请求的抓取，再基于此进行调试和压测。

5.7. 性能测试不能对阿里云外的站点进行压测吗？

阿里云性能测试已全网开放，支持全网站点性能测试。

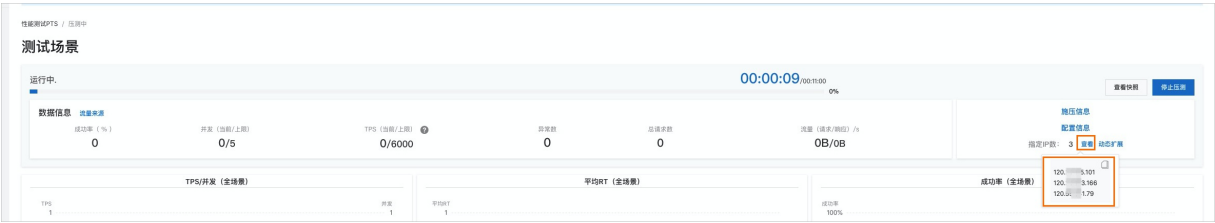
PTS支持阿里云和非阿里云站点的性能测试（只要被压测的服务有公网IP或者公网可访问）。

5.8. 如何获取PTS施压机的IP

方式一

获取施压机IP之前，您需要先创建PTS压测场景并启动压测。具体操作，请参见创建压测场景。若您已有创建好的压测场景，您可以在控制台的左侧导航栏选择**性能测试 > 场景列表**，然后在**场景列表**页面，单击目标压测场景**操作**列的**启动**，启动压测。

开始压测后，在**压测中**页面，单击**查看**，您可直接查看到施压机IP。



您还可以在**压测中**页面，单击**流量来源**，然后在右侧弹出的**流量来源**面板中，您可以查看施压机的地域以及IP等的具体信息。

方式二

您可以在压测开始之前来获取施压机IP，使用PTS提供的独占资源池功能，申请资源池后，施压端IP会提前透出。具体操作，请参见定制资源池。

后续操作

获得PTS施压机的IP后，您可以将其添加到白名单供针对性开启策略和压测使用。

添加Web应用防火墙表名单，避免压测流量被拦截。具体操作，请参见设置白名单。