云压测

常见问题

腾讯云

版权所有：腾讯云计算（北京）有限责任公司

【版权声明】

©2013-2024 腾讯云版权所有

本文档（含所有文字、数据、图片等内容）完整的著作权归腾讯云计算（北京）有限责任公司单独所有，未经腾讯云事先明确书面许可，任何主体不得以任何形式复制、修改、使用、抄袭、传播本文档全部或部分内容。前述行为构成对腾讯云著作权的侵犯，腾讯云将依法采取措施追究法律责任。

【商标声明】

及其它腾讯云服务相关的商标均为腾讯云计算（北京）有限责任公司及其关联公司所有。本文档涉及的第三方主体的商标，依法由权利人所有。未经腾讯云及有关权利人书面许可，任何主体不得以任何方式对前述商标进行使用、复制、修改、传播、抄录等行为，否则将构成对腾讯云及有关权利人商标权的侵犯，腾讯云将依法采取措施追究法律责任。

【服务声明】

本文档意在向您介绍腾讯云全部或部分产品、服务的当时的相关概况，部分产品、服务的内容可能不时有所调整。您所购买的腾讯云产品、服务的种类、服务标准等应由您与腾讯云之间的商业合同约定，除非双方另有约定，否则，腾讯云对本文档内容不做任何明示或默示的承诺或保证。

【联系我们】

我们致力于为您提供个性化的售前购买咨询服务，及相应的技术售后服务，任何问题请联系 4009100100或95716。

常见问题

最近更新时间：2024-10-25 17:32:52

常用术语

什么是 VU（并发用户数）？

VU（Virtual User）：虚拟用户数。用来模拟真实场景中，在同时执行操作的用户数量，所以也叫“并发用户数”。

- VU 代表了施压端向被压端施压的能力。
- 压测系统通常用一个线程实现一个 VU，每个 VU 重复执行压测脚本。因此，当多线程/多 VU 并发时，就能模拟真实场景中，多个用户同时执行操作的情形。
- 每个 VU 执行脚本的次数：一般靠压测时长和迭代次数来规定，任一参数达到上限即停止。例如：压测时长为 1 小时，则每个 VU 在1小时内持续反复执行脚本，直到1小时结束。（在 PTS 里，支持配置时长，暂不支持配置迭代次数。）
- VU 跟真实用户的区别：一个 VU 执行完一次脚本，会继续重复执行。其关注点不在于代表某个固定的真实用户，而在于跟其他 VU 一起，在每个时刻模拟出足够的并发用户数量。也即，施压端会按照施压配置，在相应的时刻，保证满足所配置的 VU 数量、对被压端产生足够压力。
- VU 跟在线用户的区别：在线用户不一定在做操作；而 VU 一定在做脚本里的相关操作，持续不断地给被压端造成压力。

在 PTS 中，VU 的数值是在场景的施压配置中提前设置好的。

- 并发模式下：直接设置 VU，可按时间梯度递增。
- RPS 模式下：1 个压测资源 = 500VU。

什么是 RT（响应时间）？

从客户端发出请求，到客户端完全接收服务器响应的时间消耗。
为了衡量 RT 指标，施压端会采集一个时间窗口内的所有请求从发出到收到响应的耗时，再聚合计算这批数据，得到多种维度的特征值，例如：平均值、最大值、最小值、分位值（50/90/95/99 百分位）。
在 PTS 压测报告中：

- 概览里的响应时间，是以压测任务的整个时长为时间窗口、以平均值为特征值，计算整个压测任务期间所有请求的平均响应时间。
- 各个实时曲线里的响应时间，是以一个很小的时间窗口随着时间轴移动、以平均值为特征值，模拟计算压测任务期间的各个时刻，实时的平均响应时间。

什么是 RPS（每秒请求数）？

RPS（Requests per Second），每秒请求数，也叫“吞吐量”。在 PTS 中，有两处用到了 RPS：

- 施压配置 > RPS 模式下的调速参数，用于配置 PTS 每秒发出请求的上限；
- 压测报告里的 RPS 性能指标，用于体现 PTS 收到服务端响应的速度。

版权所有：腾讯云计算（北京）有限责任公司

详细解释如下：

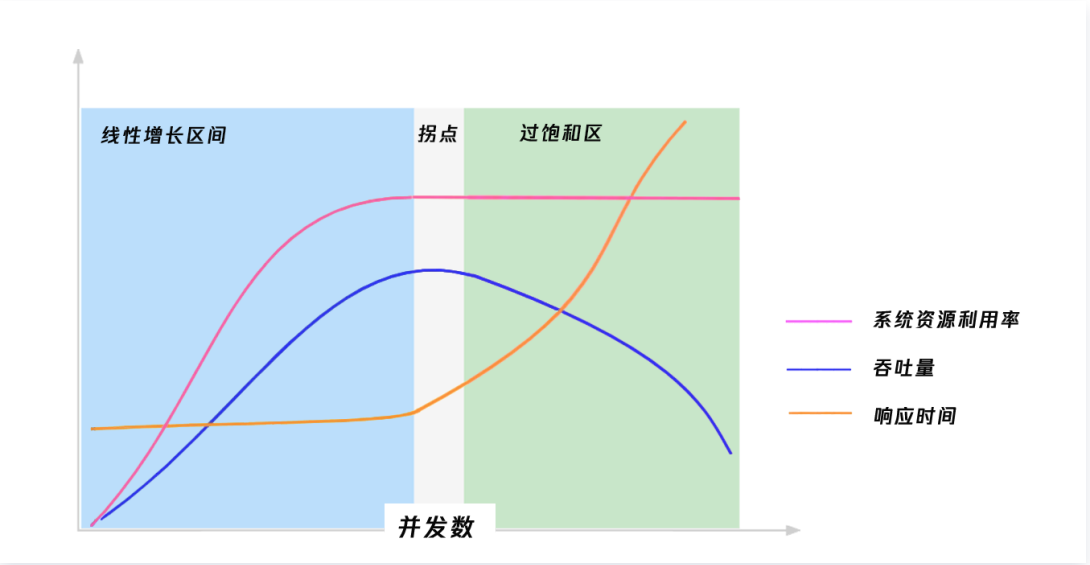
- 在绝大多数情况下，压测领域的 RPS 是指收响应的速度，用作性能指标。（若将“发请求+收响应”定义为一个“事务”，则也可将该指标称作 TPS/Transaction Per Second/每秒事务数。）
  - 在施压端统计每秒收到响应的请求数，来反映被压系统的处理能力。
  - 每秒响应结束的请求数，包括施压端作为客户端正常收到服务端响应的请求、以及主动结束的请求。
  - 在 PTS 中，您可观察压测报告里的 RPS 指标，得知 RPS 的概览值和实时值：
    - 概览里的 RPS，是以压测任务的整个时长为时间窗口，计算整个压测任务期间的 RPS。
    - 各个实时曲线里的 RPS，是以一个很小的时间窗口随着时间轴移动，模拟计算压测任务期间的各个时刻，实时的 RPS。
  - RPS 的值跟 VU 和 RT 密切相关，详见下文对三者关系的描述。
- 在 PTS 中，除了上述反映被压系统处理能力的 RPS 指标，还存在一个控制施压端每秒发出请求数的 RPS 调速参数。
  - 在 RPS 模式的施压配置中，起始 RPS/最大 RPS/动态调速 RPS，都属于 RPS 调速参数。
  - PTS 实现该 RPS 调速参数的方式，是在发请求时配置了限流。所以该参数本质上是每秒发出请求数的上限。
  - 在配置 RPS 调速参数时，PTS 会自动调整压测资源数（1 压测资源 = 500 VU），来保证施压端有能力在每秒发出足够的请求。

受限于被压系统的处理能力是否平稳、网络状况是否平稳、带宽资源是否充足等条件，客户端发请求的速度上限，不一定等于客户端收到服务端响应的速度。因此，施压时配置的 RPS 调速参数，不一定等于呈现在压测报告里的 RPS 性能指标。

VU、RPS、RT 有什么关系?

VU = RPS × RT（也即：并发数 = 吞吐量 × 响应时间）
此公式基于 Little 定律得出。Little 定律的完整表述是：在系统的稳定状态下（尚未达到系统资源过载的拐点、响应时间基本稳定、到达系统的 RPS = 离开系统的 RPS），系统中平均同时服务的用户数量 = 用户请求到达系统的速度 × 每个用户请求平均在系统中等待的时间。
例如：假设系统某接口的响应时间为100ms，那么在 1 秒内，施压端的 1 个VU 能连发10个请求并获得响应，反映了被压端的系统吞吐量是10个请求每秒（RPS 为 10）；那么，当同时做操作的用户数翻了100倍，也即100个 VU 同时并发施压，如果被压接口的响应耗时仍为100ms，则在 1 秒内，每个 VU 都能发送10次请求并获得响应，反映了被压端在 1 秒内处理了 1000 个请求，也即 RPS 达到 1000。
以上换算建立在被压系统表现稳定、响应时间保持不变的理想状况下。然而实际上，随着并发数增大、系统负载升高，被压接口的响应时间不一定能保持在100 ms，而可能呈现以下的增大趋势：

版权所有：腾讯云计算（北京）有限责任公司



1. 刚开始为 “线性增长区”，此时响应时间（RT）基本稳定，吞吐量（RPS）随着并发用户数（VU）的增加而增加，三者关系符合 Little 定律：VU = RPS × RT。
2. 随着 VU 增大、系统的资源利用率饱和，系统到达 “拐点”，若继续增大 VU，响应时间开始增大，RPS 开始下降。
3. 继续增加 VU，系统超负荷、进入过饱和区，此时响应时间急剧增大、RPS 急剧下降。

失败率的定义是什么?

一批请求中结果出错的请求所占比例，以校验响应结果是否符合期望。（不同系统对错误率的要求不同，但一般不超出千分之六，即成功率不低于99.4%。）

- PTS 通过统计一批请求中失败响应码所占比例，来计算请求失败率。响应码大于或等于 400，视为请求失败。（其中包含 PTS 端认为被压端不可达而主动取消请求的情况，相关响应码详见 错误代码手册 。）
- 请求失败率不包含检查点断言失败的情况（检查点情况参见检查点明细）。

在 PTS 压测报告中：

- 概览里的失败率，是以压测任务的整个时长为时间窗口，计算整个压测任务期间所有请求的失败率。
- 各个实时曲线里的失败率，是以一个很小的时间窗口随着时间轴移动，模拟计算压测任务期间的各个时刻，实时的失败率。

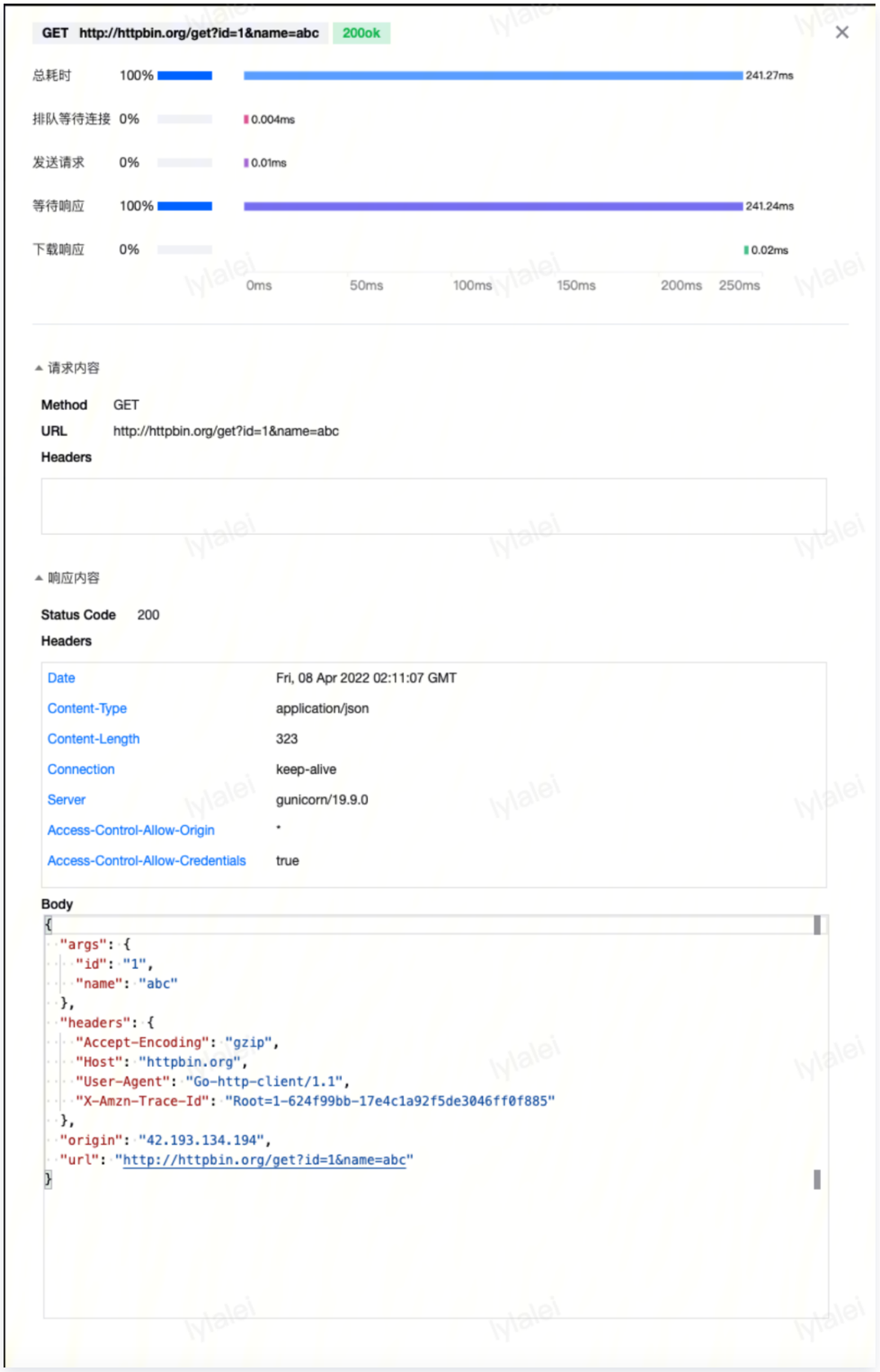
常见问题

PTS 调试失败/没有日志，如何去定位问题?

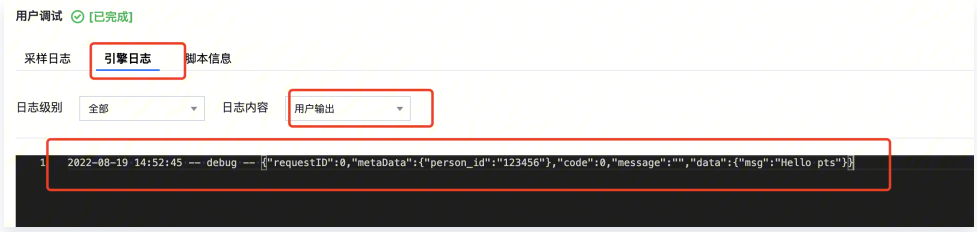
PTS 提供了全面的日志定位手段，分别为引擎日志/用户日志/请求日志，主要有三个阶段：

- **引擎日志**：JavaScript 脚本编写出现语法错误、空指针错误等问题，导致引擎解析脚本失败，可以按照日志提示的报错点进行修复。

- **请求日志**：已经产生网络 I/O 请求，对应请求包已经发到服务端，但服务端解析失败。可以根据请求采样中、或调试功能中的请求/响应包体的详细信息，定位哪个协议字段存在问题。



- **用户日志：**您在脚本中通过 console.log 打印相关变量，可以在用户日志中查看，调试相关变量的结构体/返回值定义。



测试报表会在云压测中保存多久?

测试报表包含指标数据及日志数据，默认保留45天，45天后将自动清理过期数据。在过期前，用户可下载测试报表，在本地进行保存。用户也可将测试报表设置为基线报表，基线报表将永久保存。

VPC 内压测 VU 数量上限?

VPC 私有网络压测，PTS 需要在用户 VPC 侧创建弹性网卡，并绑定到施压机上。受限于用户 VPC 能够创建弹性网卡数量限制，最大支持10W VU 并发。

如何保护被压端服务，防止被压端服务异常影响业务可用性?

当被压端服务异常时，通过实时测试报表，您可以看到请求 RT 变高，甚至出现请求失败。
为了防止服务异常，您可以在测试场景编排中，设置被压服务 SLA（服务可用性指标），例如：限制响应 RT < 100ms，请求失败率 < 0.1%。当压测指标触发被压服务 SLA 水位线时，可通过告警通知到您，也可根据设置自动停止压测任务。
另外为避免服务异常，也建议您：

- 设置合理的压力模型。
- 将起步压力设置较低，通过梯度模型或者手动逐步调高压力，观察服务整体可用性。

PTS 支持 JMeter 压测吗?

用户只需要在场景编排中导入 jmx 文件，即可以原生方式运行 JMeter 压测。PTS 支持以分布式方式运行 JMeter 引擎，提供便捷的横向扩容能力和实时测试报表。

HTTP 服务请求失败率高，返回大量的 `net/http: request canceled` 错误信息?

在压测报告可以看到详细的错误率，用户可以在采样日志看具体的耗时分布，如果是服务端返回超时，可自定义配置全局 option http timeout 参数，默认为 10s。

```
export const option = {
  http: {
    // 单位ms
    timeout: 10000,
  }
}
```

HTTP 请求出现 `x509: cannot validate certificate` 返回错误?

两种解决方案，第一种全局配置参数 insecureSkipVerify:true，第二种上传单独 TLS 证书。

```
// 通过全局配置，跳过证书验证
export const options = {
    tlsConfig: {
      // localhost 为示例服务名，需根据实际情况替换
      'localhost': {
        insecureSkipVerify: true
      }
    }
}
```

PTS 支持哪些扩展方法，具体的参数定义去哪里查看?

- PTS 脚本示例，包括 HTTP、WebSocket 等常用协议。
- PTS JavaScript API 文档。
- PTS 常用工具函数，包括随机数/base64 编解码/math 函数等。
- PTS 支持完整 ES6 语法，还支持 第三方包引用（如 crypto.js 这种 PTS 暂时没有集成的能力）。

PTS 如何从测试文件读取数据?

PTS 支持 dataset 读取测试数据，用户在压测场景完成文件上传，引擎会解析 csv 文件并按行轮询进行读取，具体的语法如下：

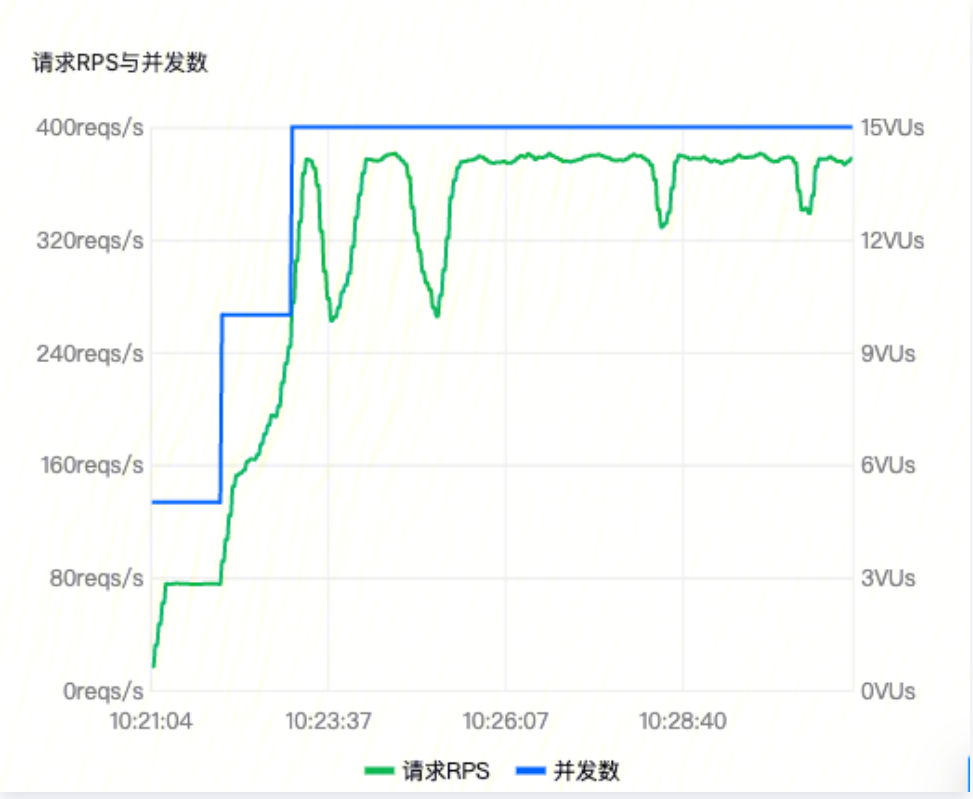
```
import dataset from 'pts/dataset';

export default function () {
    const value = dataset.get("MyKey")
    //@ts-ignore   忽略校验
    const postResponse =
http.post("http://mockhttpbin.pts.svc.cluster.local/post", {data:
value});
    console.log(postResponse)
};
```

> **说明**
> 详细使用指引可参见 使用参数文件。

PTS 报表显示 VU = 0，或者跟施压配置的值对应不上?

VU=0：PTS 报表显示请求的 VU（并发用户数）为瞬时指标，当处于任务结束时，其瞬时值有可能为 0。
跟施压配置的值对应不上：由于大部分用户设置的是梯度发压模型，VU 值会随时间梯度变化，其瞬时值应以图表显示的 VU 曲线变化为准。



导入的 csv 出现乱码，如何解决?

含中文的 csv 导入后乱码的问题：
因为 Windows 默认导出的 csv 使用的是 GBK 编码，并且旧版本的 Excel 2016 前会不保存 Bom (byte order mark)。
解决方法：将 csv 导出为 utf-8 格式：

- Windows 可以使用记事本打开 csv 文件后，另存为 utf-8 格式。
- Mac 上使用 `iconv -f GBK -t UTF-8 xxx.csv > utf-8.csv` 。

状态码 999 是什么错误，如何排查?

施压端没能从被压服务端得到有效的 HTTP 响应状态码，则会将状态码置为 999 Unknown。这些请求会被视为错误请求，计入压测报告的错误率。
错误原因可能是请求本身的协议/地址等有误，或者是网络原因、服务端的 DNS/防火墙/SSL 证书/超时断连等原因，导致服务不可达。
如需排查，可参考请求采样里的错误信息、施压机日志里的报错信息，还可使用调试模式调试请求。
常见原因如下：

- 施压端没能正常发出请求。
  - 请求采样里的报错信息：`Error net/http: request canceled while waiting for connection` 。可能的原因：

    - 施压端到被压端服务端口之间的网络不通。
    - 被压端服务的 DNS/防火墙/SSL 证书等配置错误。
- 施压端已正常发出请求，但没能在超时时间内获得有效的响应状态码。
    - 例如，目前 HTTP 协议默认 10 秒钟超时，可观察压测报告里的响应时间是否已超过 10 秒、请求采样里的错误信息是否为 `Error net/http: request canceled` 。若确实是超时导致，可排查为何被压服务响应慢、优化其处理请求的能力。
    - 若需调大 HTTP 超时时间，可在脚本模式下配置，详见：配置选项 。
- 压测任务结束释放资源时，若有部分请求尚未完成，则会被施压端自动取消掉，此时请求采样里的报错信息为：`Error context deadline exceeded` 。

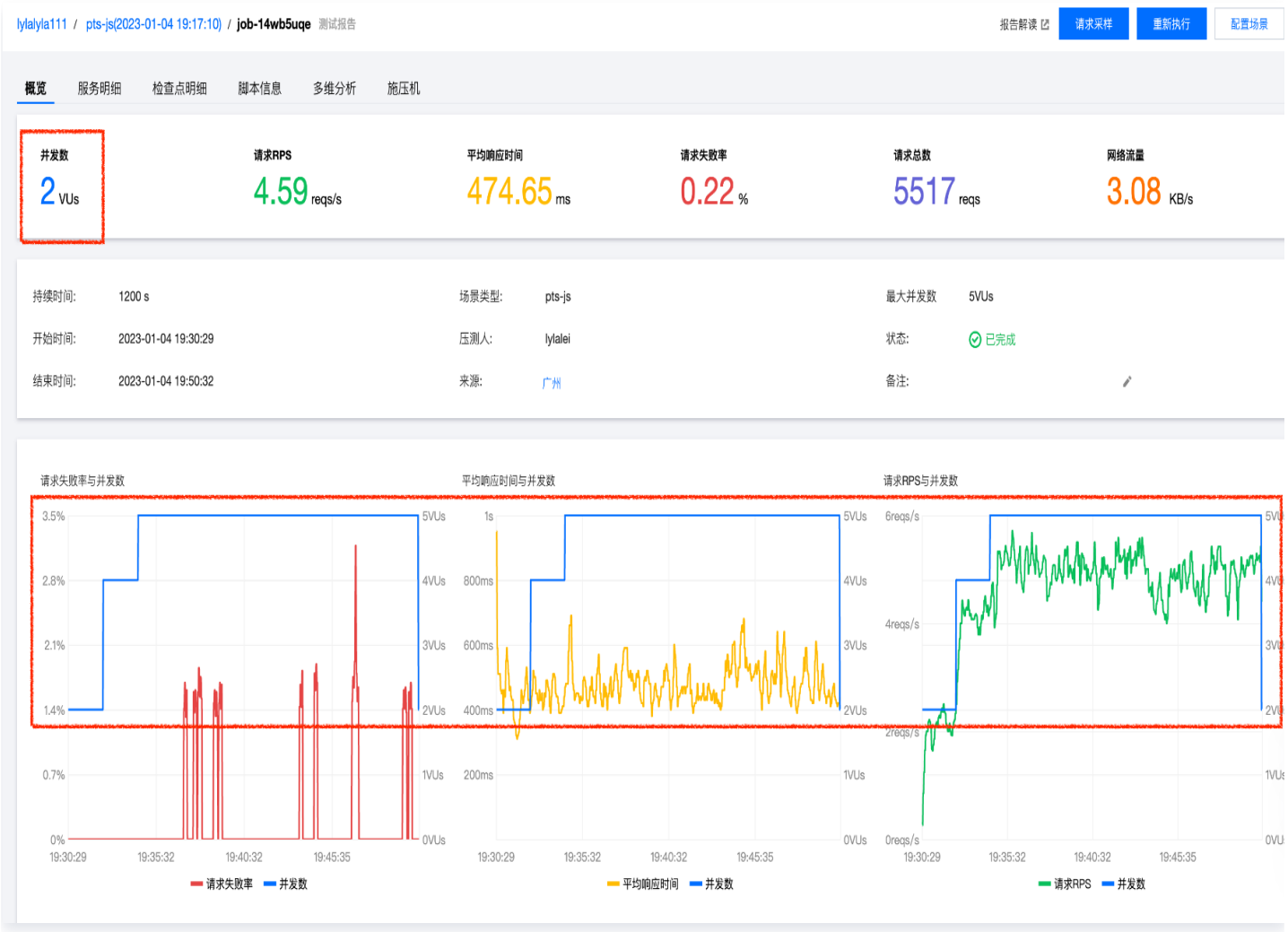
调试模式下，为什么我的请求只执行了一部分?

PTS 在调试模式下，压测引擎最多执行 10 秒，之后会自动退出。
如果您场景里编排的请求无法在 10 秒内全部完成，则会表现为只执行了部分请求。
建议直接以较小的 VU 数运行压测任务，来代替调试模式，避免其 10 秒无法执行全部请求的问题。

压测结束时，概览里的并发数（VUs）为什么突然下降了?

压测运行时，在报告页的概览栏里，并发数（VUs）的数值是实时值，与图表里代表并发数的蓝色梯度线在每个时刻的值是一致的。
压测结束时，PTS 会将资源回收，所以实时 VU 可能表现为瞬间下降，这是符合预期的正常的行为。
您可参考图表里的蓝色线，观察并发数（VUs）随时间轴的变化，可以发现它是在将您配置的梯度发压如期完成后，在压测结束时刻才下降的。

版权所有：腾讯云计算（北京）有限责任公司



采样日志的采样策略是什么样的，采样比例是多少?

PTS 使用首次采样与比例采样结合的方法来对用户请求进行采样。

- 首次采样策略

  我们将请求[service, method, status, result] 四个维度组合起来作为请求特征， 如果一组请求特征没有被记录过，那么这样的请求会被采样记录下来。

- 比例采样策略

  按千分之一的比例采样用户请求。首次采样策略命中的请求不计入该比例中。比例采样策略简化后：采样第1个请求，采样第1001个请求，以此类推。

以压测 http get 请求 `https://mockhttpbin.pts.svc.cluster.local/get` 请求为例:

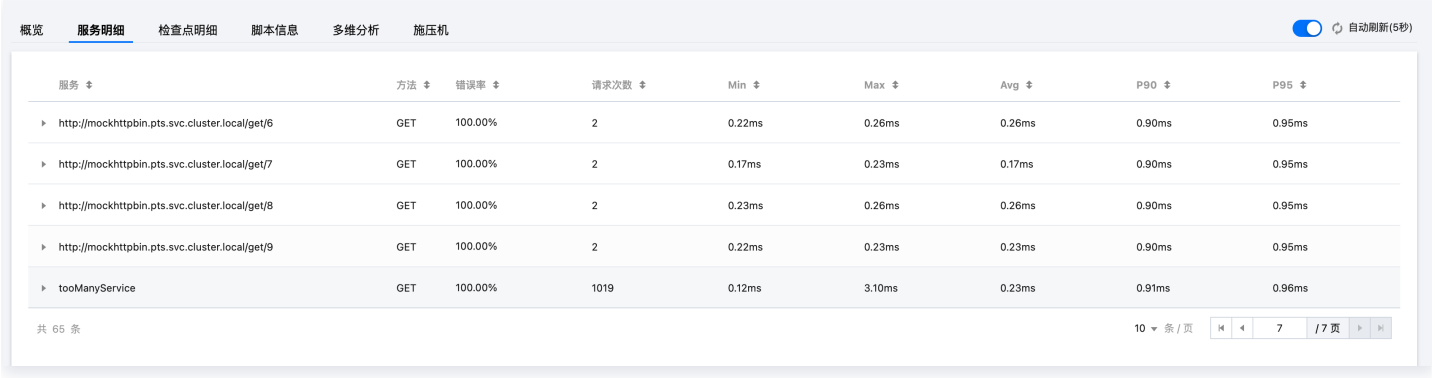
- 第1个请求状态码返回200， 请求特征["https://mockhttpbin.pts.svc.cluster.local/get", "get", "200", "ok"]，这个特征首次出现，请求将被采样。
- 第2个请求返回status 200，比例采样策略命中该请求，该请求被记录下来。
- 第10个请求时候，出现了500错误，请求特征["https://mockhttpbin.pts.svc.cluster.local/get", "get", "500", "internal error"]，首次采样策略观测到这是首次出现的特征，该请求也会被采样。

版权所有：腾讯云计算（北京）有限责任公司

- 第1002个请求返回 status 200，比例采样策略命中该请求，该请求也被记录下来。

报告中服务明细内出现 tooManyService 是什么意思，该如何处理?

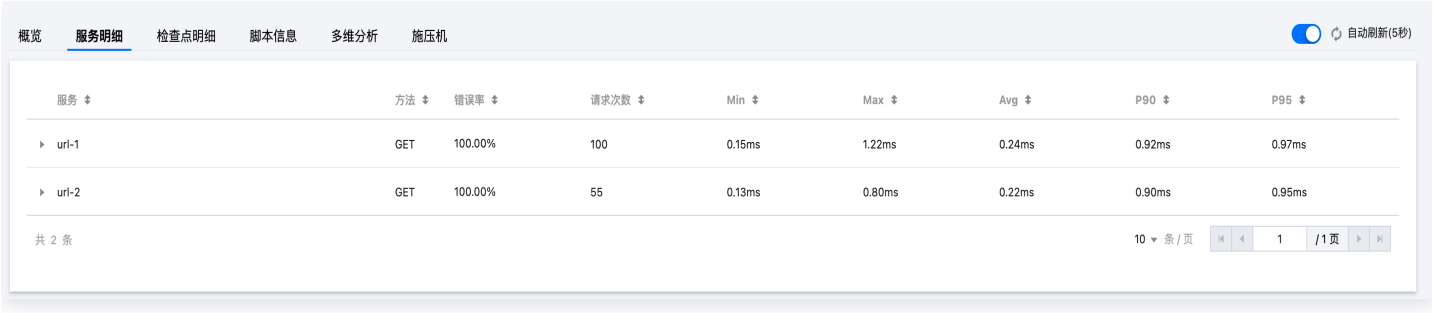
PTS 压测报告的服务明细中，默认将每个 URL 归类为一个“**服务 service**”，展示压测期间发送的所有请求的明细信息。如果 URL 数量较多，控制台仅展示 64 个不同的服务 service，更多的会以 **tooManyService** 标签汇总起来。



服务 service 数量过大，不对 URL 进行分类，对于报告的解读和分析有不利的影响；因此，把同一类的请求 service 汇总起来，对于统计分析具有较大的帮助。通过设置请求中 Request 的 service 字段，可以把同一类的请求分到同一个服务明细的展示行里面；如何配置，可以参考 Request 。

```
// Send a http get request
import http from 'pts/http';
import { check, sleep } from 'pts';


export default function () {
  const resp1 = http.get("http://mockhttpbin.pts.svc.cluster.local/get",
{
    service: "url-1",
  });
  const resp2 =
http.post("http://mockhttpbin.pts.svc.cluster.local/post", "", {
    service: "url-2",
  });
}
```



PTS 使用的是同步压测还是异步压测？为什么？

PTS 使用的是同步压测，即 PTS 的一个 VU（并发用户）会在发送请求后等待响应，直到接收到响应后才会发送下一个请求。
而异步压测是指一个 VU（并发用户）会在发送一个请求后，立即继续发送下一个请求，而不等待前一个请求的响应。
同步压测具有以下优点：

- 简单易懂；
- 更接近真实用户的行为（用户通常会等待请求的响应）；
- 易于调试（由于每个请求都是顺序执行的，调试和分析问题相对容易）。

异步压测缺点：

- 实现和调试相对复杂；
- 单个 VU（并发用户）可能占用过多资源；
- 无法准确模拟真实用户的行为（尤其是在用户需要等待响应的场景中）。

综上，PTS 平台选择同步压测的方式，以更好地模拟真实用户的行为。

版权所有：腾讯云计算（北京）有限责任公司