

01-项目性能优化

刘亚雄

极客时间-Java 学科讲师



今日目标

1. 了解应用性能问题分析方法论
2. 掌握压力测试基础概念
3. 掌握压力测试：线程组配置、结果分析，插件使用
4. 理解性能关键的指标
5. 掌握压测监控平台搭建

一、应用性能问题分析方法论

1.1 性能优化的终极目标是什么？

用户体验 = 产品设计(非技术) + 系统性能 \approx 系统性能 = 快？



快是一种什么样的体验？

1.2 快是一种什么样的体验？

3秒，不能更久了



怎么让系统快起来呢？

1.3 应用性能调优是个大工程

- **前端工程师：**首屏时间、白屏时间、可交互时间、完全加载时间
- **移动端（Android、IOS）工程师：**端到端响应时间、Crash率、内存使用率、FPS
- **后端工程师：**RT、TPS、并发数
 - 影响因素01：数据库读写、RPC、网络IO、逻辑计算复杂度、缓存
 - 影响因素02：JVM【Throughput、Footprint、Latency】

1.4 影响性能的关键要素

- **产品设计：产品逻辑、功能交互、动态效果、页面元素**
- **基础网络：网络 = 连接介质 + 计算终端**
 - 连接介质：电缆、双绞线、光纤、微波、载波或通信卫星。
 - 计算终端：PC、手机、可穿戴设备、家具家电...
 - 基础网络设施，互联网，局域网(LAN)、城域网(MAN)、广域网(WAN)
- **代码质量&架构**
 - 架构不合理
 - 研发功底和经验不足
 - 没有性能意识：只实现了业务功能不注意代码性能，当业务上量后系统出现连锁反应，导致性能问题叠加。
 - 数据库：慢查询、过多查询、索引使用不当、数据库服务器瓶颈...
- **移动端环境：设备类型&性能、系统版本、网络(WiFi、2G、3G、4G)**
- **硬件及云服务：服务器硬件，内存、磁盘、网卡...**



怎么找到短板在哪里呢？

二、压力测试

2.1 压力测试灵魂三问

➤ 01-什么是压力测试？

- 压力测试（英语：Stress testing）是针对特定系统或是组件，为要确认其稳定性而特意进行的严格测试。会让系统在超过正常使用条件下运作，然后再确认其结果。
- 说白了就是对系统不断施加压力，来预估系统**负载能力**的一种测试。

➤ 02-为什么要对系统进行压测呢？有必要么？

- 一般而言，只有在系统基础功能测试验证完成、系统趋于稳定的情况下，才会进行压力测试。

➤ 03-压测目的是什么？

- 1. 当负载逐渐增加时，观察**系统各项性能指标的变化情况**是否有异常
- 2. 发现系统的**性能短板**，进行针对性的性能优化
- 3. 判断系统在高并发情况下**是否会报错**，进程是否会挂掉
- 4. 测试在系统某个方面达到瓶颈时，粗略估计系统性能上限

2.2 压力测试的指标

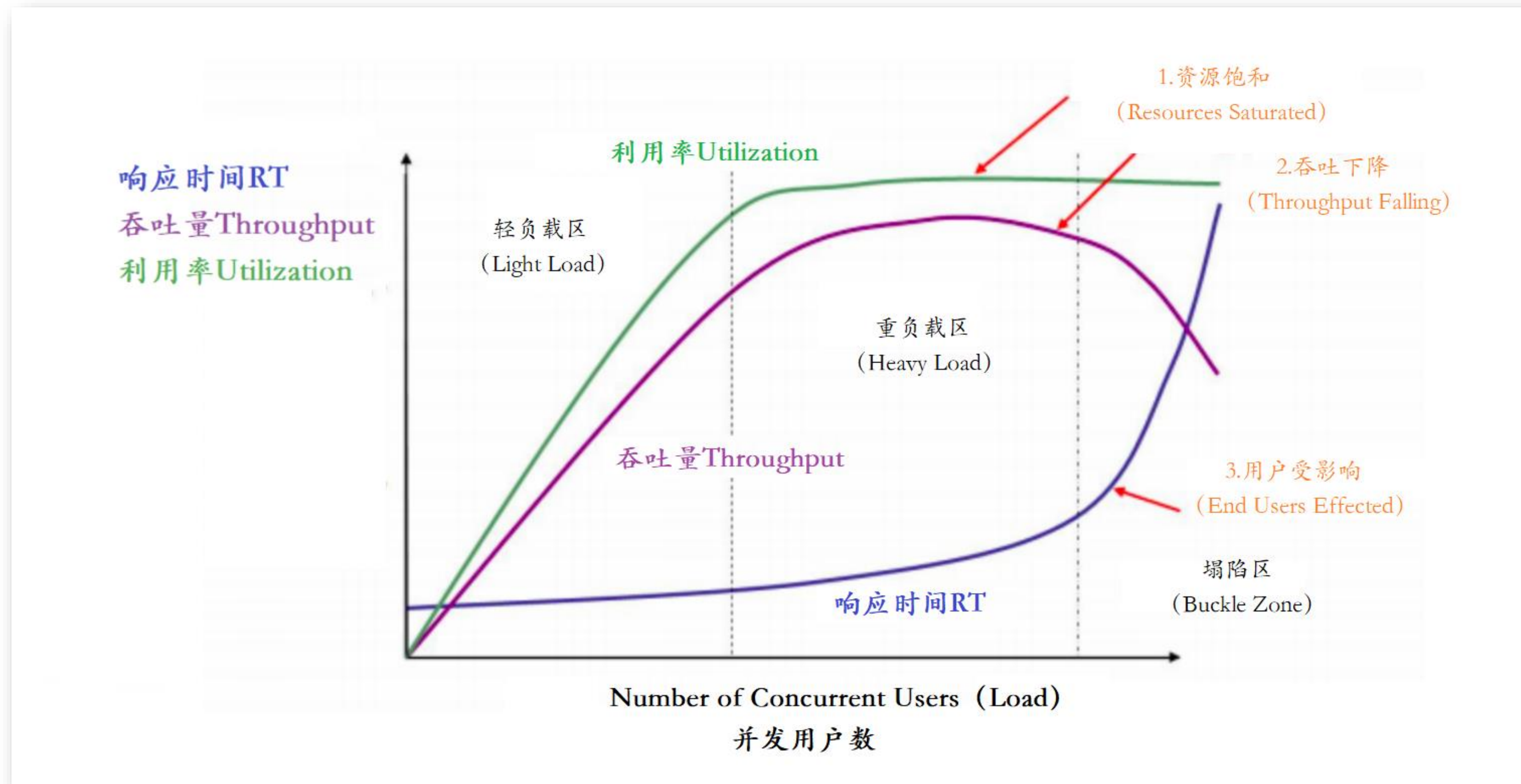
指标	含义
响应时间（RT）	是指系统对请求作出响应的平均时间，对于单用户的系统，响应时间可以很好地度量系统的性能。
吞吐量（Throughput）	是指系统在单位时间内处理请求的数量， 每秒事务数TPS 也算是吞吐量的一种
资源利用率	CPU占用率、内存使用率、系统负载、网络I/O
并发用户数	是指系统可以同时承载的正常使用系统功能的用户的数量，用户不同的使用模式会导致不同用户在单位时间发出不同数量的请求。
错误率	失败请求占比，在测试时添加响应断言，验证不通过即记为错误；若不添加，响应码非200 即为错误。

评判系统性能，主要考虑三个性能指标【RT、TPS、资源利用率】



它们的相关性咱们来看图

2.2 压力测试的指标



怎么看这张图？

2333

两个点，三条线，三个区，三个状态

- **两个点：**最优用户并发数，最大并发用户数
- **三条线：**RT、Throughput、资源利用率
- **三个区：**轻负载、重负载，塌陷区
- **三种状态：**资源饱和，吞吐量下降，用户受影响

2.3 压力测试工具JMeter

JMeter是Apache组织开发的基于Java的压力测试工具。用于对软件做压力测试，它最初被设计用于Web应用测试，但后来扩展到其他测试领域。它可以用于测试**静态和动态资源**，例如**静态文件、Java 小服务程序、CGI 脚本、Java 对象、数据库、FTP 服务器，等等**。

其他常用压测工具：

- ① **Apache JMeter 可视化的测试工具**
- ② LoadRunner 预测系统行为和性能的负载测试工具
- ③ Apache的ab压力测试
- ④ nGrinder 韩国研发一款性能测试工具
- ⑤ PAS 阿里测试工具等等



压测目标简单来说有4个：

- 负载上升各项指标是否正常
- 发现性能短板
- 高并发下系统是否稳定
- 预估系统最大负载



SpringBoot服务处理并发请求的理论的上线是多少？

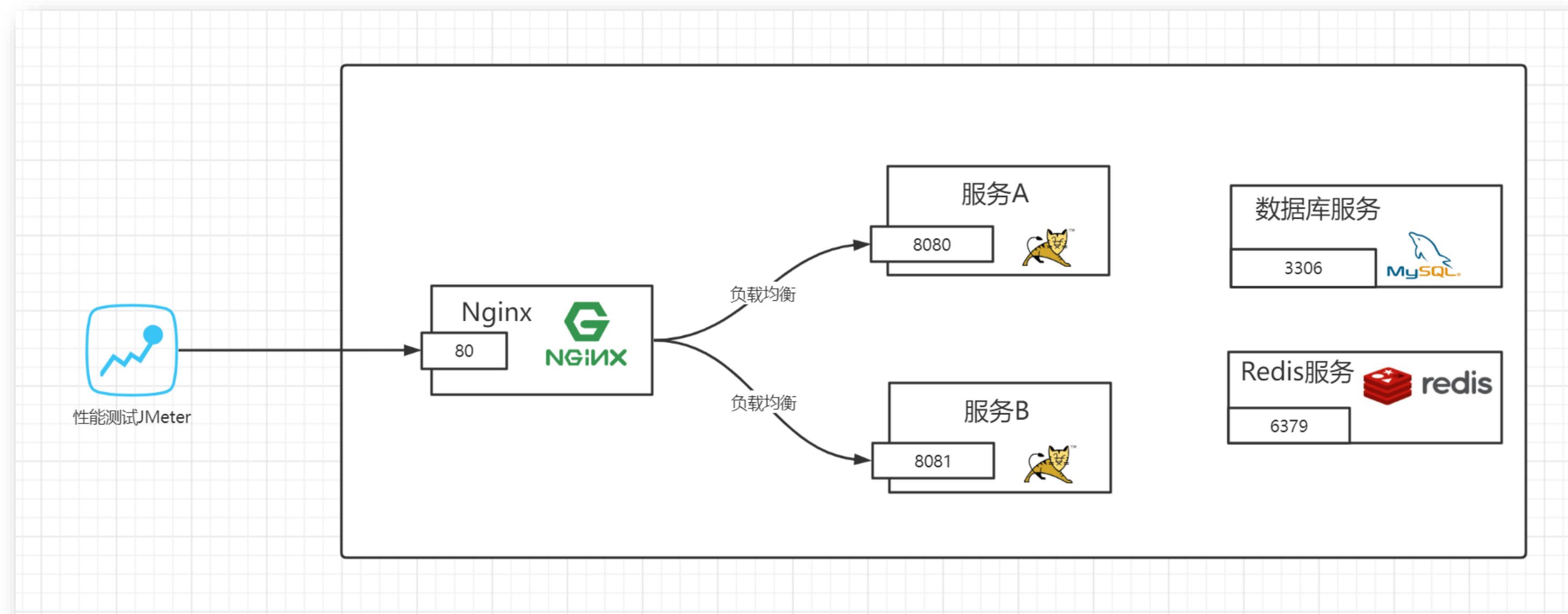
2.4 压力测试案例

目标：完成压测案例，评测SpringBoot项目的吞吐量上限。

步骤：

- ① 创建测试计划
- ② 配置线程组、http请求、断言、结果监听器
- ③ 执行测试
- ④ 查看测试结果，分析测试结果

压测环境架构图



2.5 压测结果分析

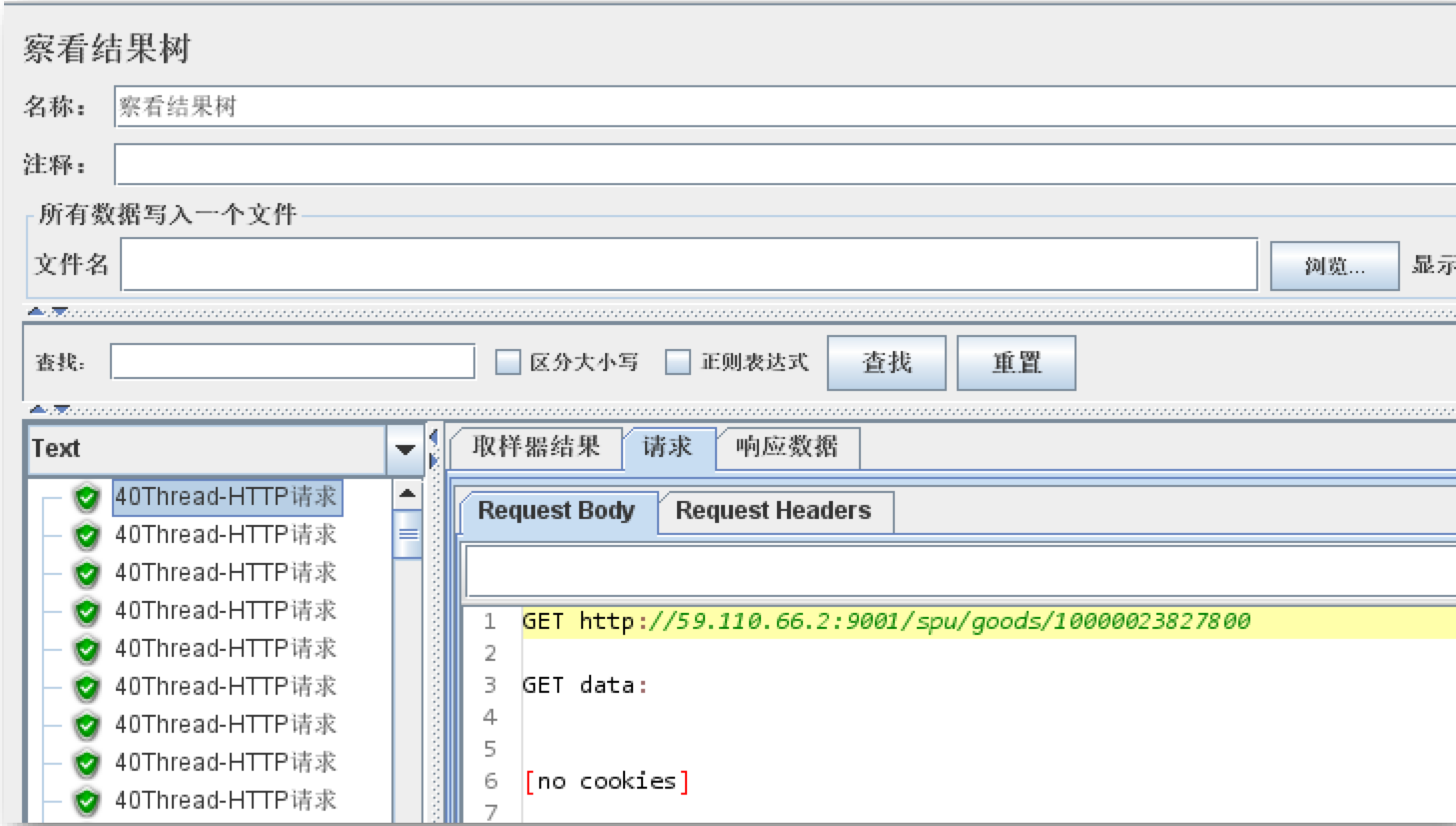
01-聚合报告

- 样本（Sample）：发送请求的总样本数量
- 响应时间（RT）：平均值，P90，P95，P99，Min，Max
- 异常（Error）：出现错误的百分比
- 吞吐量（Throughput）：被测试接口的吞吐能力
- 发送与接收数据量：KB/sec

02-查看结果树：记录了样本中每一次的请求

03-汇总报告（类似于聚合报告）

- 样本（Sample）
- 响应时间（RT）
- 异常（Error）
- 吞吐量（Throughput）
- 发送与接收数据量：KB/sec



聚合报告

名称：

聚合报告

注释：

所有数据写入一个文件

文件名

浏览...

显示日志内容：

☐ 仅错误日志

☐ 仅成功日志

配置

Label	# 样本	平均值	中位数	90% 百分位	95% 百分位	99% 百分位	最小值	最大值	异常 %	吞吐量	接收 KB/sec	发送 KB/sec
5Thread-HT...	25000	13	12	16	18	26	8	52	0.00%	376.0/sec	398.97	0.00
10Thread-H...	50000	14	14	18	21	29	8	325	0.00%	678.4/sec	719.94	0.00
15Thread-H...	75000	15	15	20	23	33	8	1033	0.00%	921.3/sec	977.79	0.00
20Thread-H...	100000	16	15	21	25	34	8	317	0.00%	1194.2/sec	1267.52	0.00
25Thread-H...	125000	18	18	26	29	39	9	258	0.00%	1295.7/sec	1375.24	0.00
30Thread-H...	150000	21	20	29	33	45	9	1045	0.00%	1383.4/sec	1468.30	0.00
35Thread-H...	175000	26	25	37	42	56	9	339	0.00%	1312.8/sec	1393.43	0.00
40Thread-H...	200000	30	28	44	50	66	9	1048	0.00%	1303.5/sec	1383.54	0.00
总体	900000	22	20	34	40	55	8	1048	0.00%	1129.2/sec	1198.54	0.00

2.6 线程组配置解释

01-线程数：用来发送http请求的线程的数量

- 线程组常用来模拟一组用户访问系统资源（API接口）。假如客户机没有足够的能力来模拟较重的负载，可以使用 **JMeter分布式压力测试**，通过一个JMeter的Master来远程控制多个JMeter的Slave完成测试。

02-循环次数：循环执行多少次操作

- 循环次数直接决定整个测试单个线程的执行时间，和整体测试执行时间。
- **单线程执行时间** = 单请求平均响应时间 × 循环次数
- **整个测试耗时** = 单线程执行时间 + (Ramp-Up - Ramp-Up / 线程数)

03-Ramp-Up(in-seconds)：建立全部线程耗时

- 代表隔多长时间执行，默认值是0，0代表同时并发，用于告知JMeter 要在多长时间内建立全部的线程。

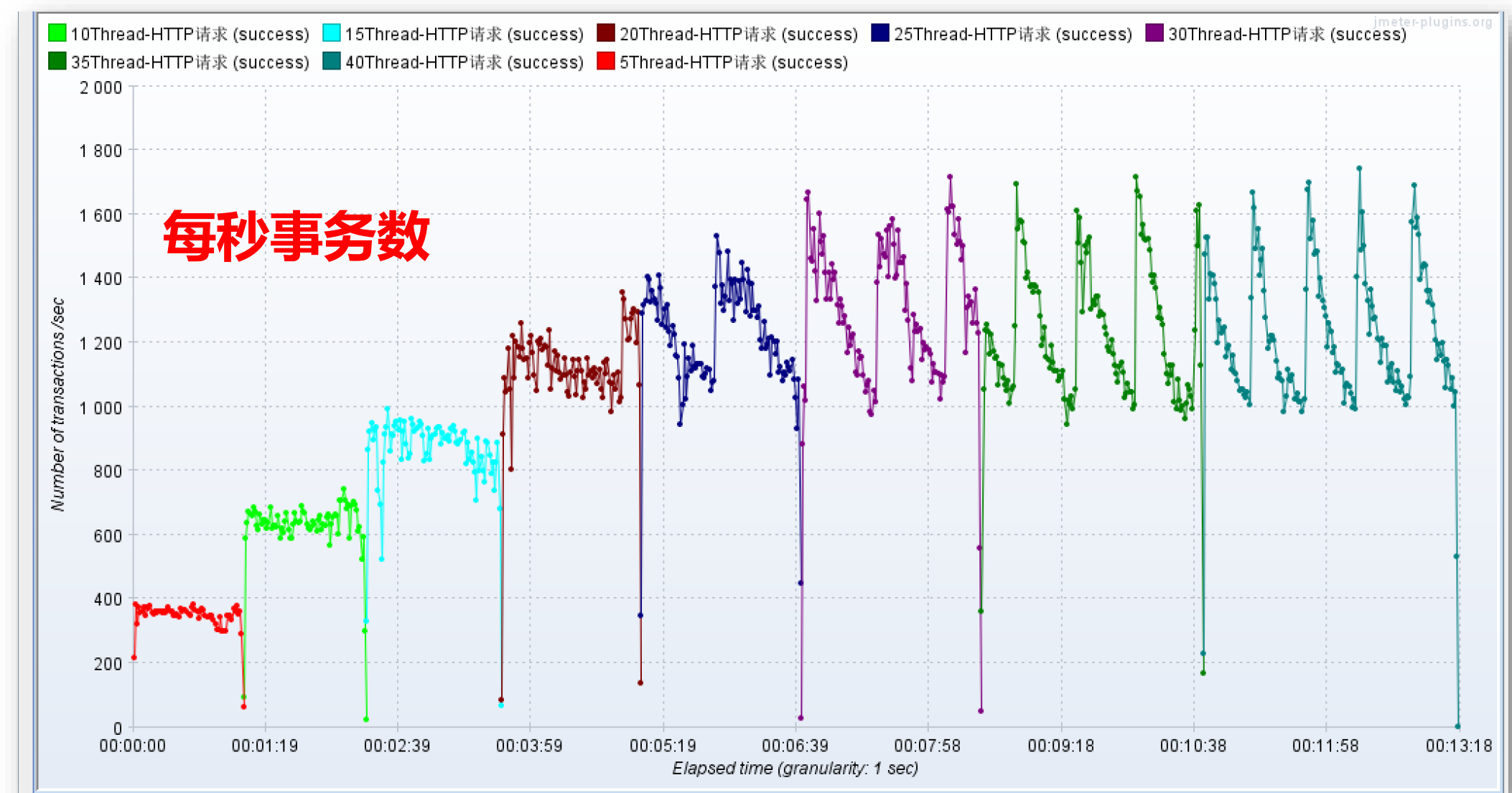
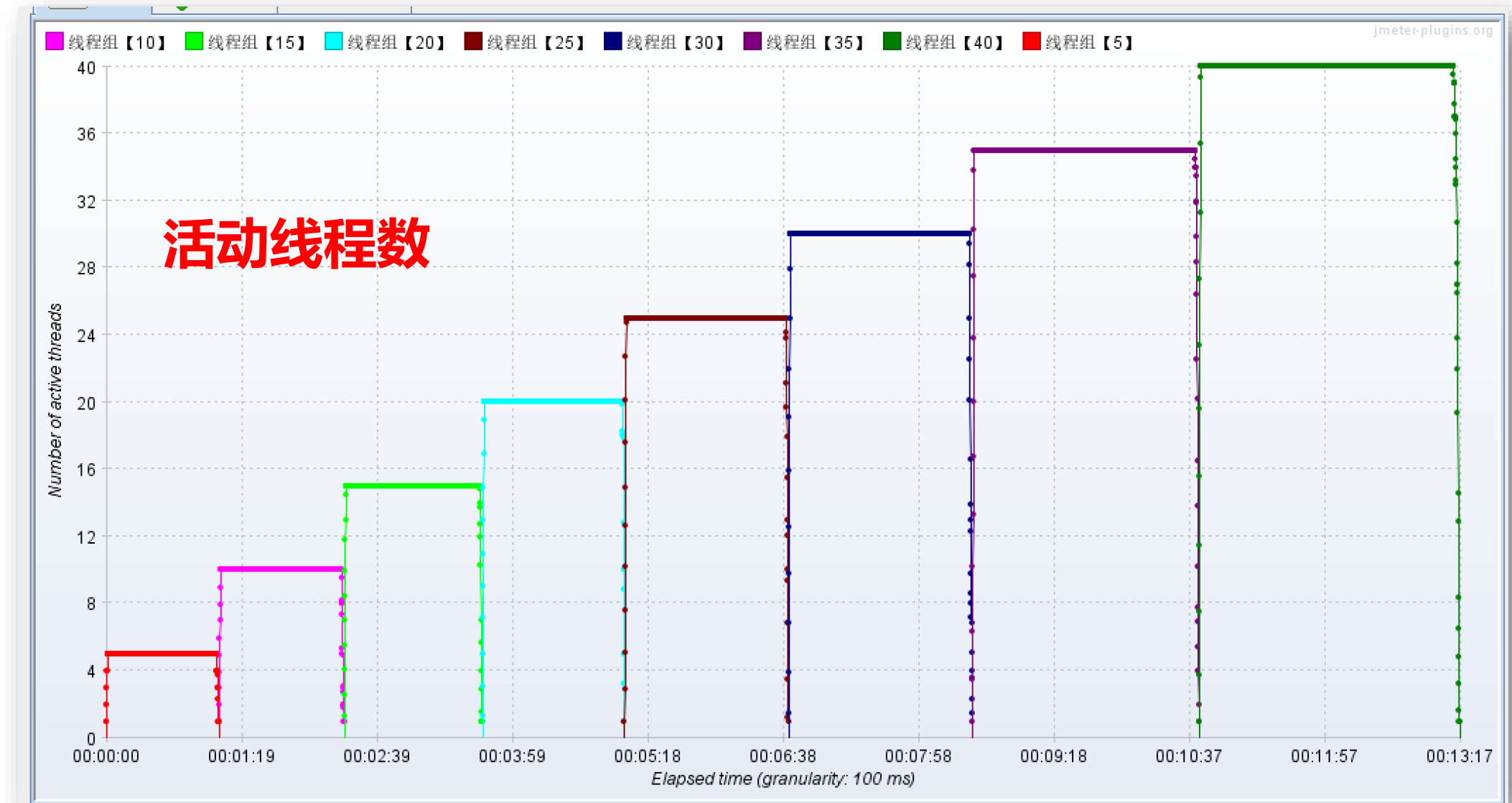
2.7 JMeter插件

Basic Graphs主要显示

- Average Response Time 平均响应时间
- Active Threads 活动线程数
- Successful/Failed Transactions 成功/失败 事务数

Additional Graphs主要显示

- Response Codes
- Bytes Throughput
- Connect Times
- Latency
- Hits/s

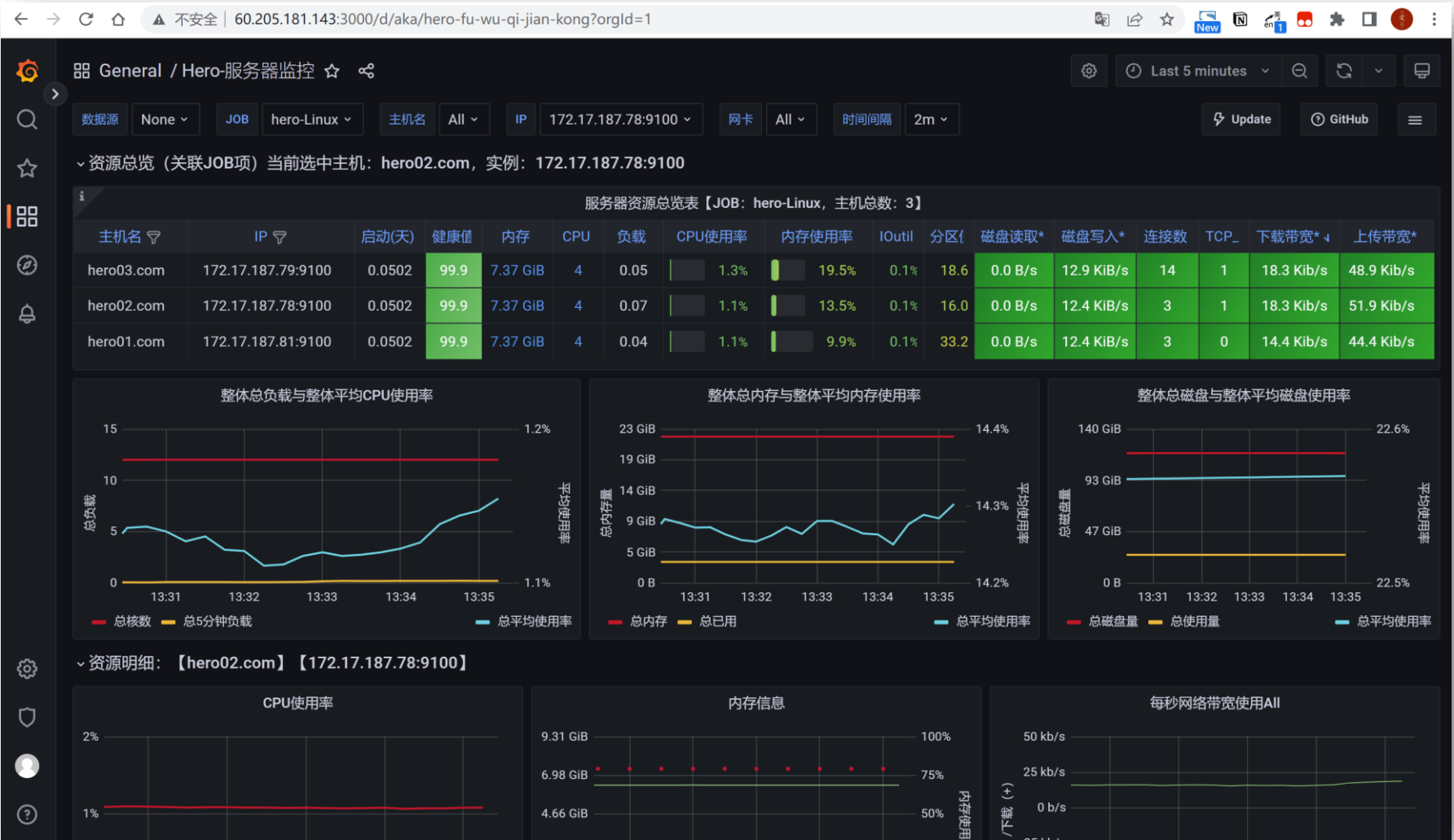
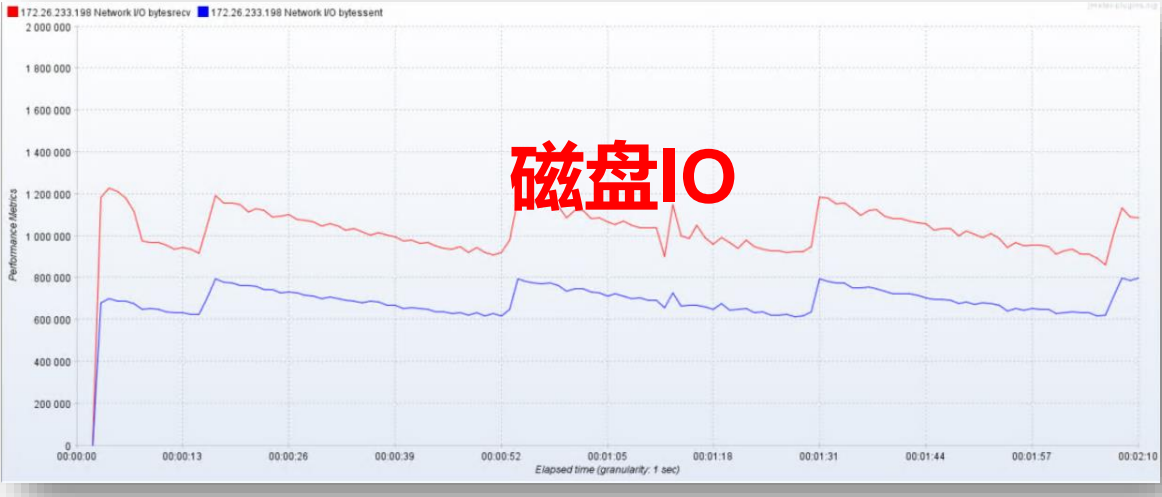
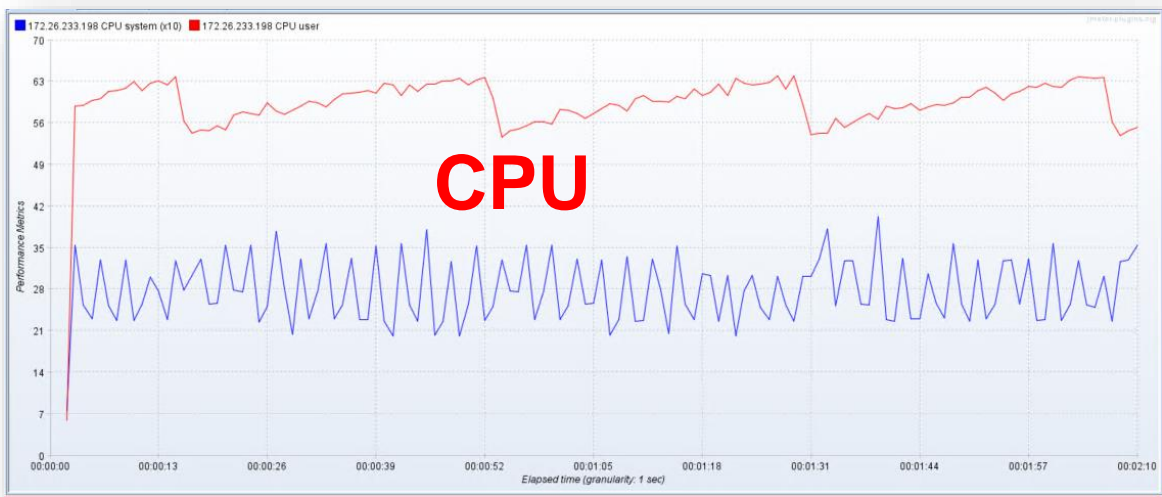
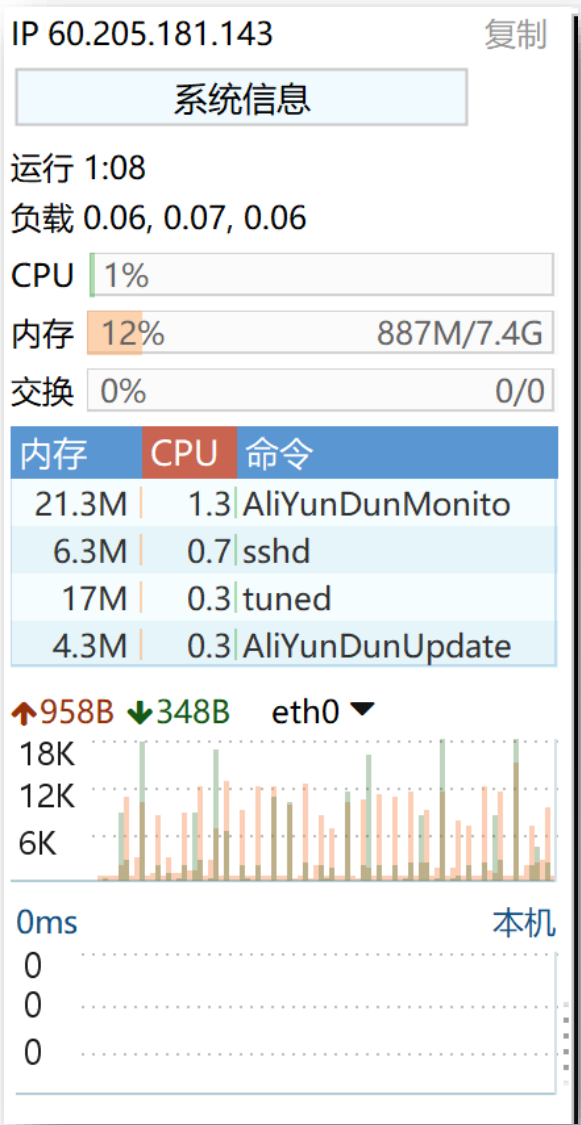


2.8 服务器硬件资源监控

压测的时候，我们需要实时了解服务器CPU、内存、网络、服务器负载情况的状态如何
哪如何监控服务器的资源占用情况呢？

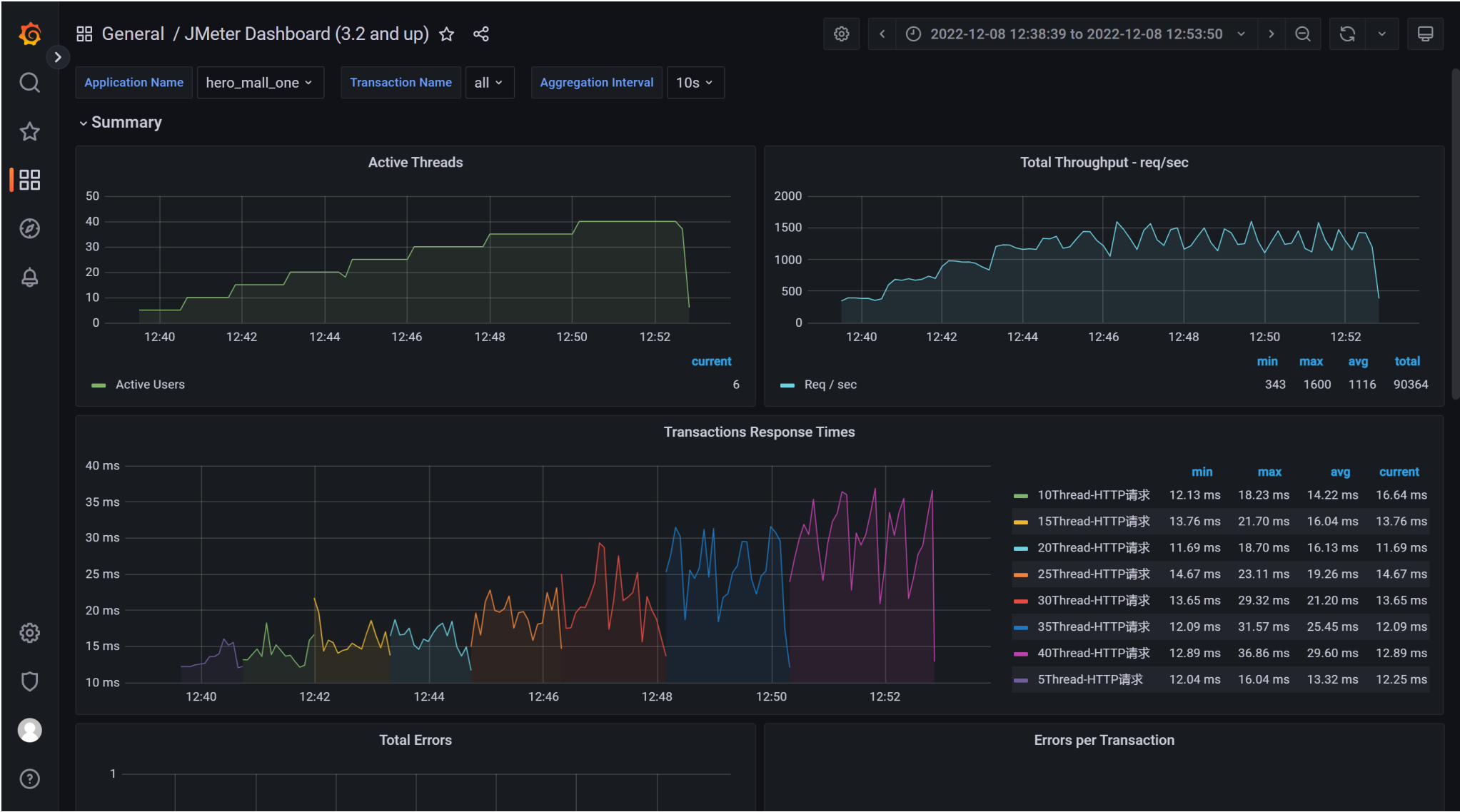
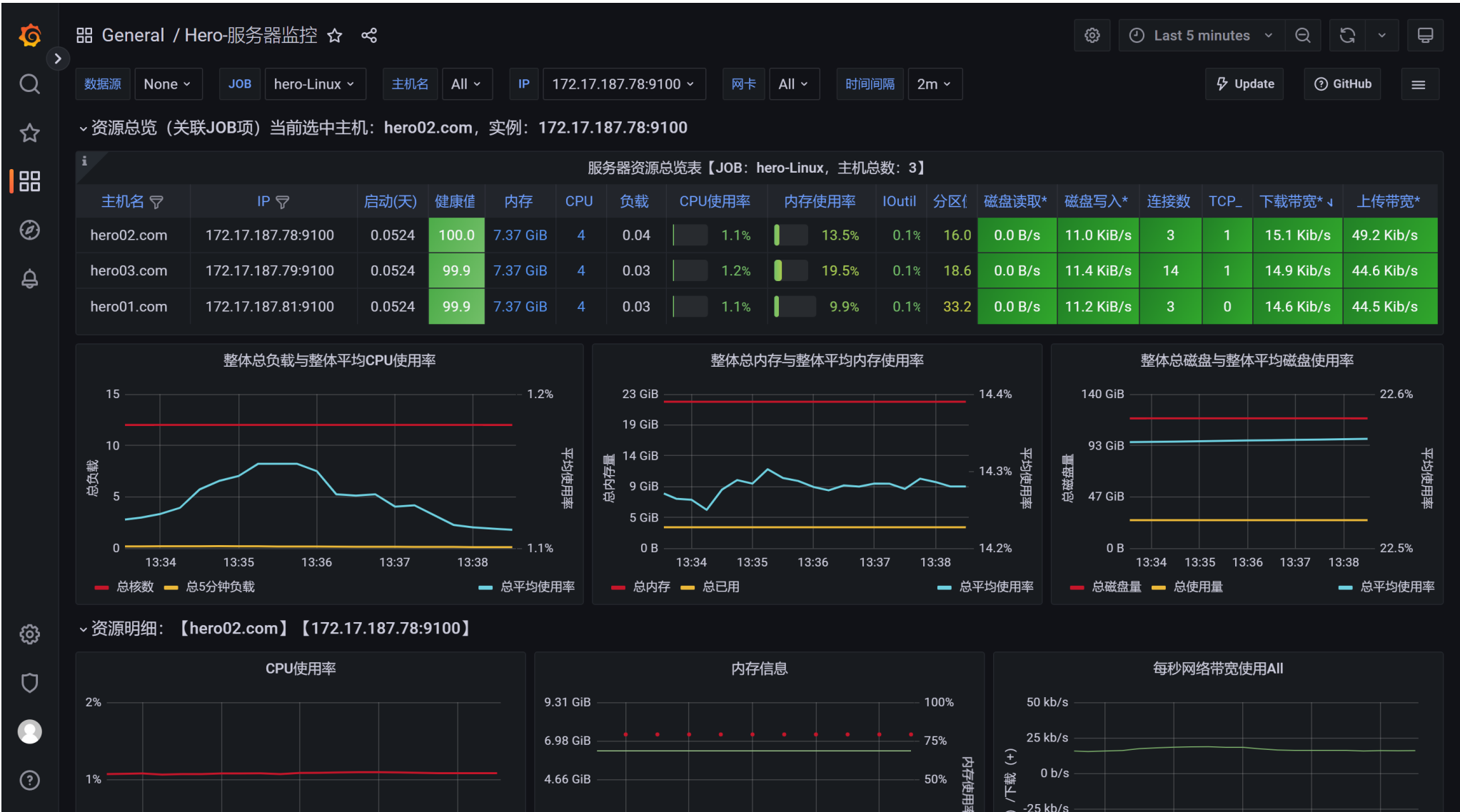
方法有很多种比如：

- 使用操作系统命令：top、vmstat、iostat、iotop、dstat、sar...
- 使用FinalShell
- 使用JMeter压测工具PerfMon
- 使用Grafana+Prometheus+node_exporter



2.9 压测监控平台搭建

- ① 配置Docker环境
- ② 安装InfluxDB
- ③ 设置Jmeter脚本后置监听器
- ④ 安装Grafana
- ⑤ 安装node_exporter
- ⑥ 安装Prometheus



THANKS

 极客时间 | 训练营

教育不是注满一桶水，而是点燃一把火