

拉勾教育

— 互联网人实战大学 —

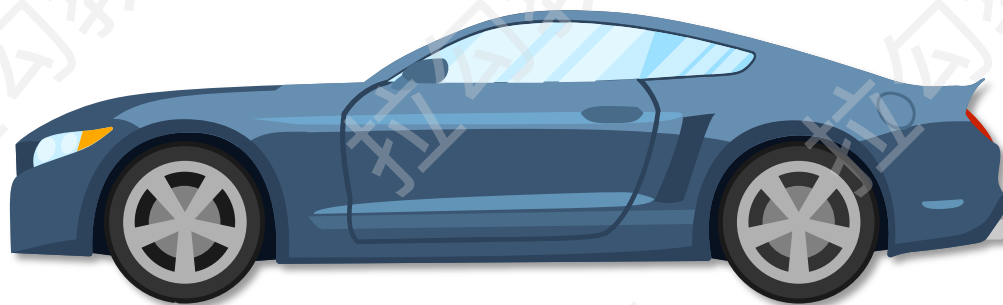
《Java性能优化与面试21讲》

李国

— 拉勾教育出品 —

第01讲：理论分析： 性能优化，有哪些衡量指标？需要注意什么？

指标是衡量很多事物，以及做出行为决策的重要参考



前言

拉勾教育

— 互联网人实战大学 —

动力性

通过性

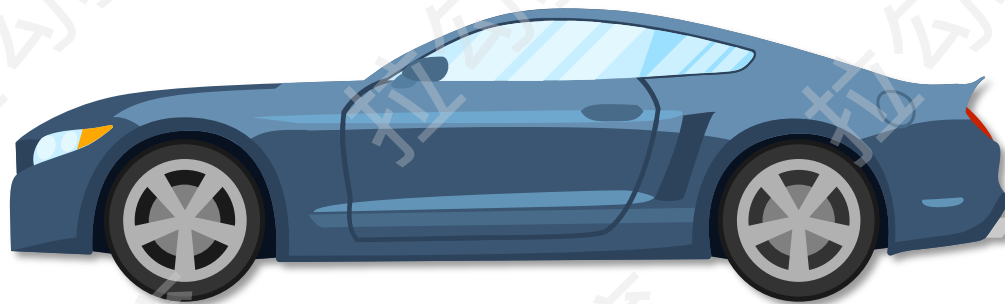
制动性

燃油经济性

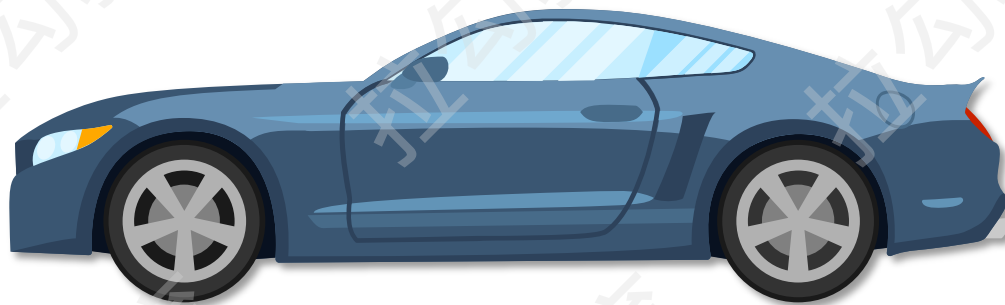
平顺性

操纵稳定性

排放与噪声



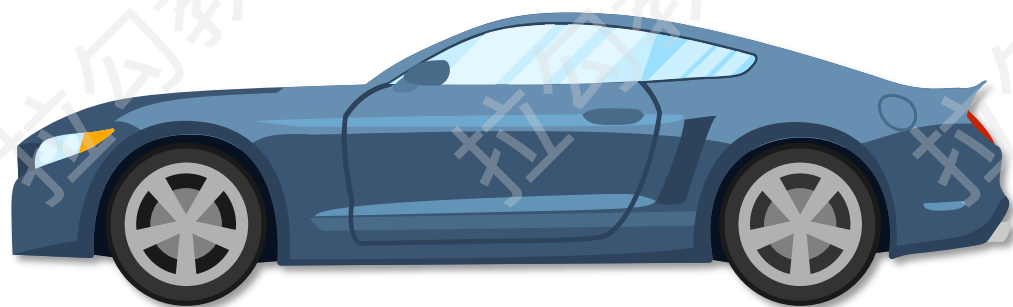
性能优化



缺乏理论依据



- 在衡量一项优化是否能达到目的之时，不能仅靠感觉
它同样有一系列的指标来衡量你的改进
- **如果在改动之后**，性能不升反降，那就不叫性能优化





- **性能**，使用有限的资源、在有限的时间内完成工作

最主要的衡量因素是时间，很多衡量指标，都可以把时间作为横轴



- **性能**，使用有限的资源、在有限的时间内完成工作

最主要的衡量因素是时间，很多衡量指标，都可以把时间作为横轴

- 加载缓慢的网站，会受到搜索排名算法的惩罚，从而导致网站排名下降

衡量指标有哪些？

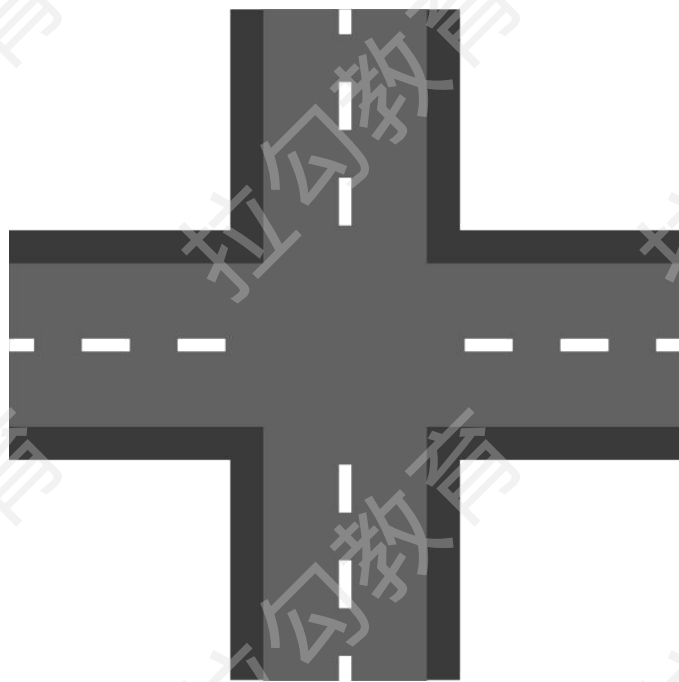
拉勾教育

— 互联网人实战大学 —



分布式的高并发应用并不能把单次请求作为判断依据，它往往是一个统计结果

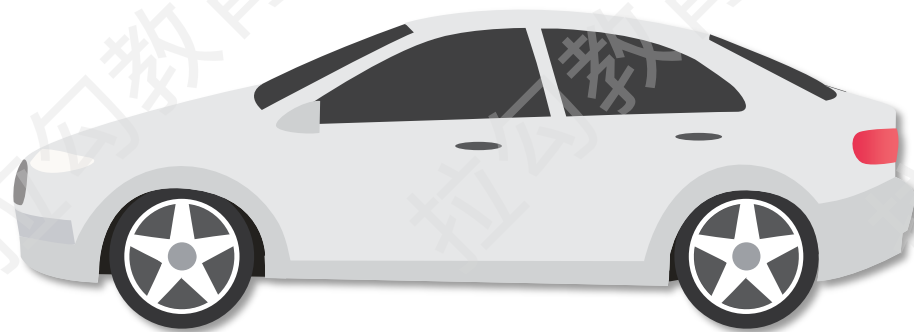
其中最常用的衡量指标就是吞吐量和响应速度



吞吐量 and 响应速度

拉勾教育

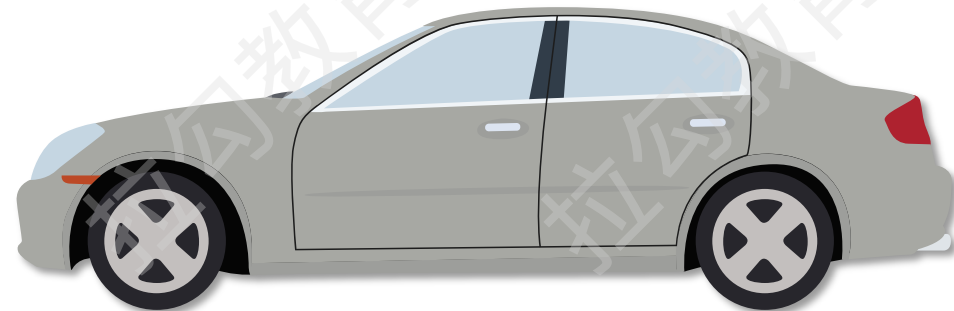
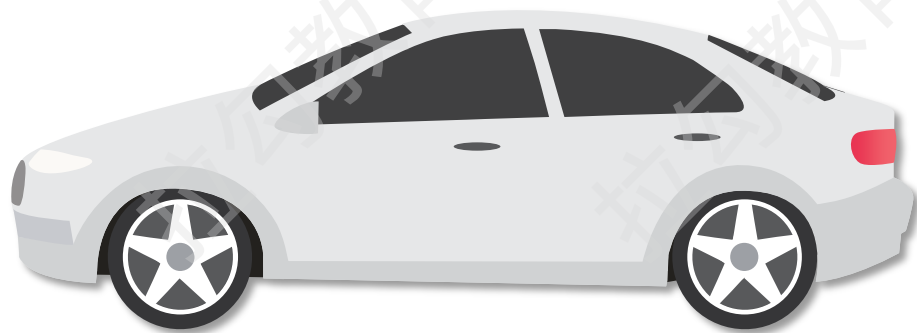
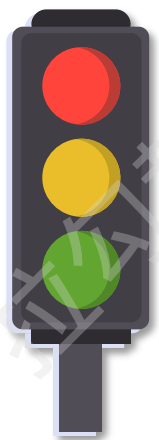
— 互联网人实战大学 —



吞吐量 and 响应速度

拉勾教育

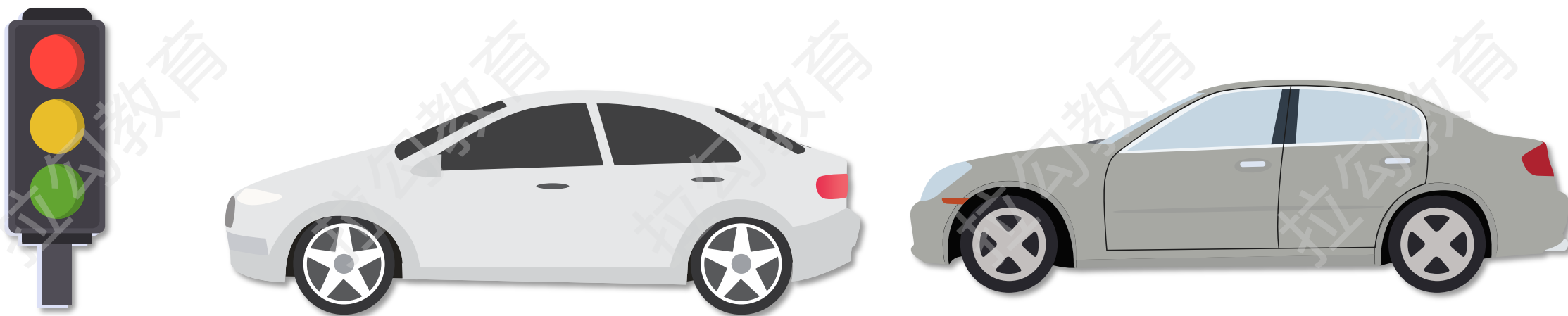
— 互联网人实战大学 —



吞吐量 and 响应速度

拉勾教育

— 互联网人实战大学 —



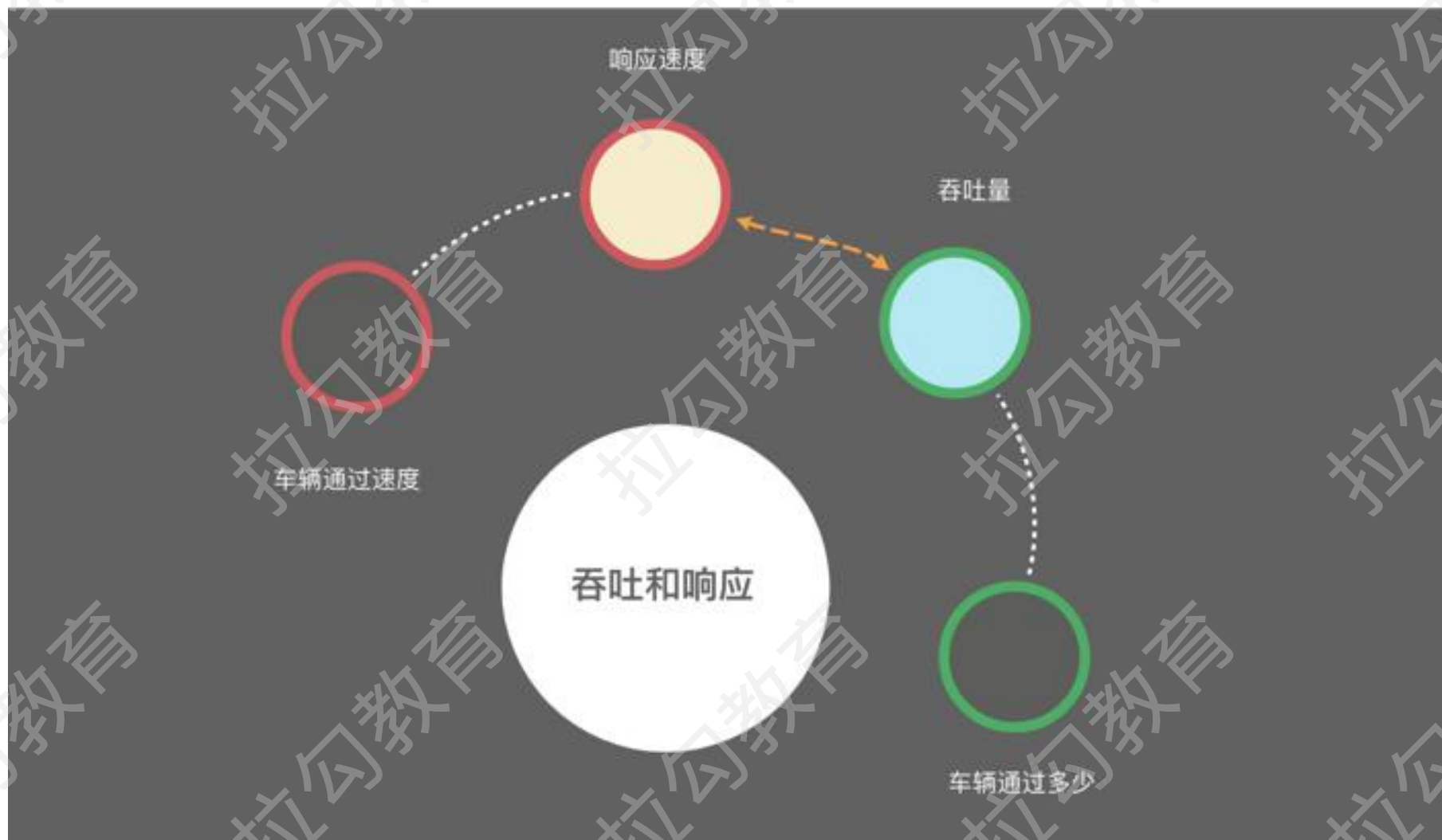
响应时间

L / A / G / O / U

吞吐量 and 响应速度

拉勾教育

— 互联网人实战大学 —



吞吐量 and 响应速度

拉勾教育

— 互联网人实战大学 —

QPS

每秒查询的数量

TPS

每秒事务的数量

HPS

每秒 HTTP 请求数量

在性能优化的时候，我们要搞清楚优化的目标，到底是吞吐量还是响应速度



- **响应速度**是串行执行的优化，通过优化执行步骤解决问题
- **吞吐量**是并行执行的优化，通过合理利用计算资源达到目标

吞吐量和响应速度

拉勾教育

— 互联网人实战大学 —

高并发的互联网应用

响应速度

吞吐量

吞吐量和响应速度

拉勾教育

— 互联网人实战大学 —

高并发的互联网应用

响应速度

吞吐量

使用有限的硬件资源，从中找到一个平衡点

平均响应时间

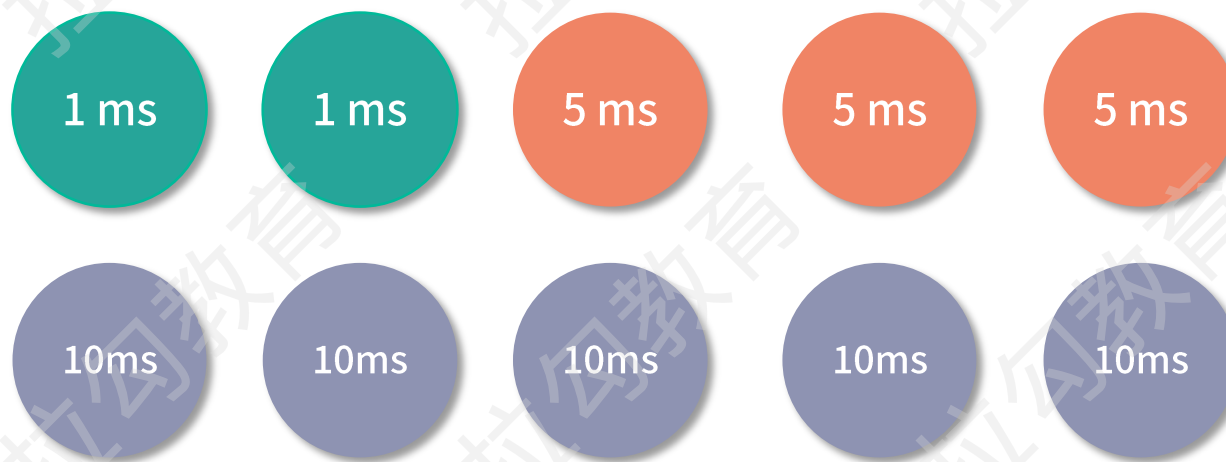
最常用的指标，指标能够体现服务接口的平均处理能力

它的本质是把所有的请求耗时加起来，然后除以请求的次数

平均响应时间

拉勾教育

— 互联网人实战大学 —

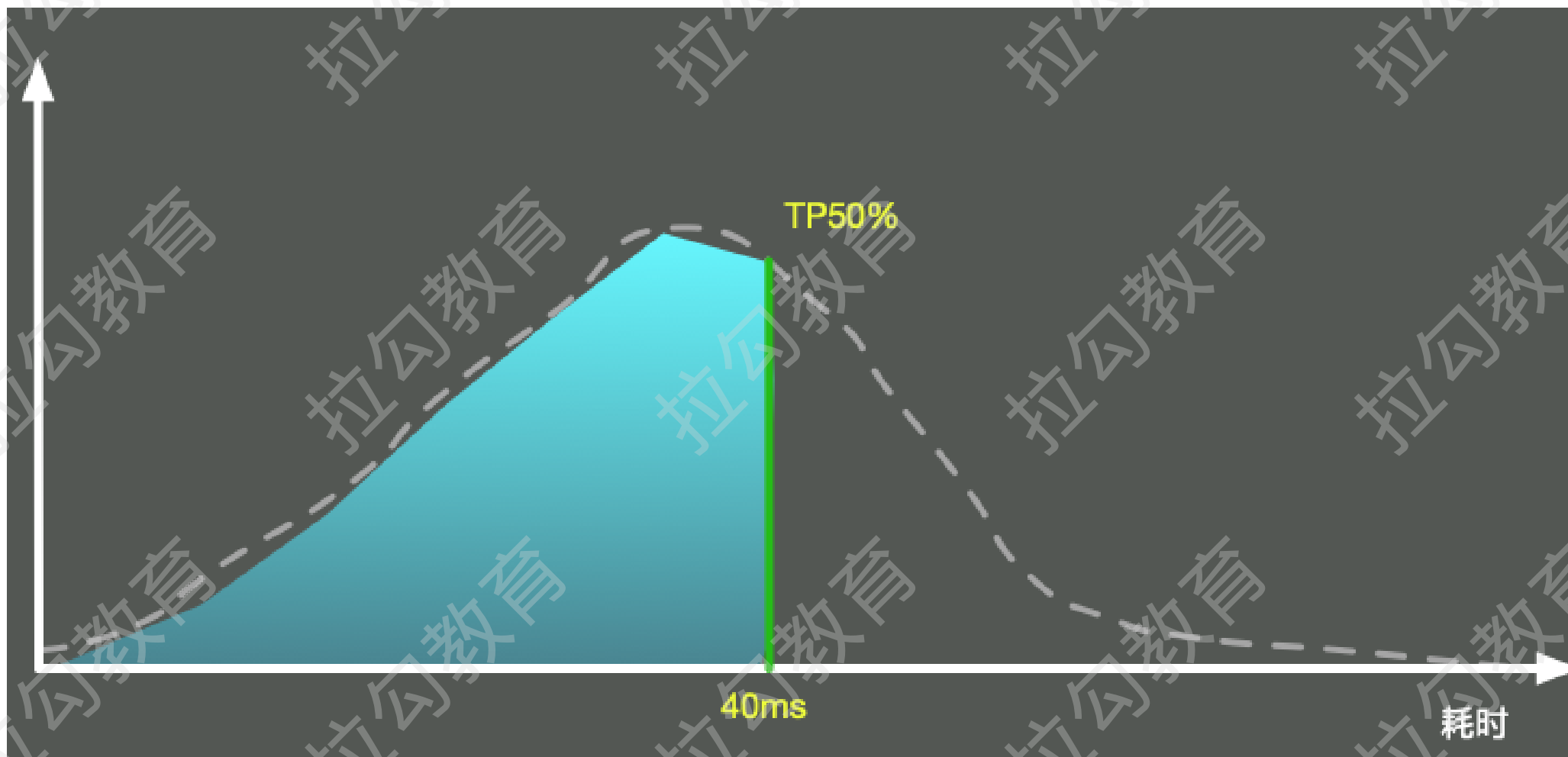


平均耗时: $(2*1+3*5+5*10) / 10 = 6.7ms$

百分位数 (Percentile)

拉勾教育

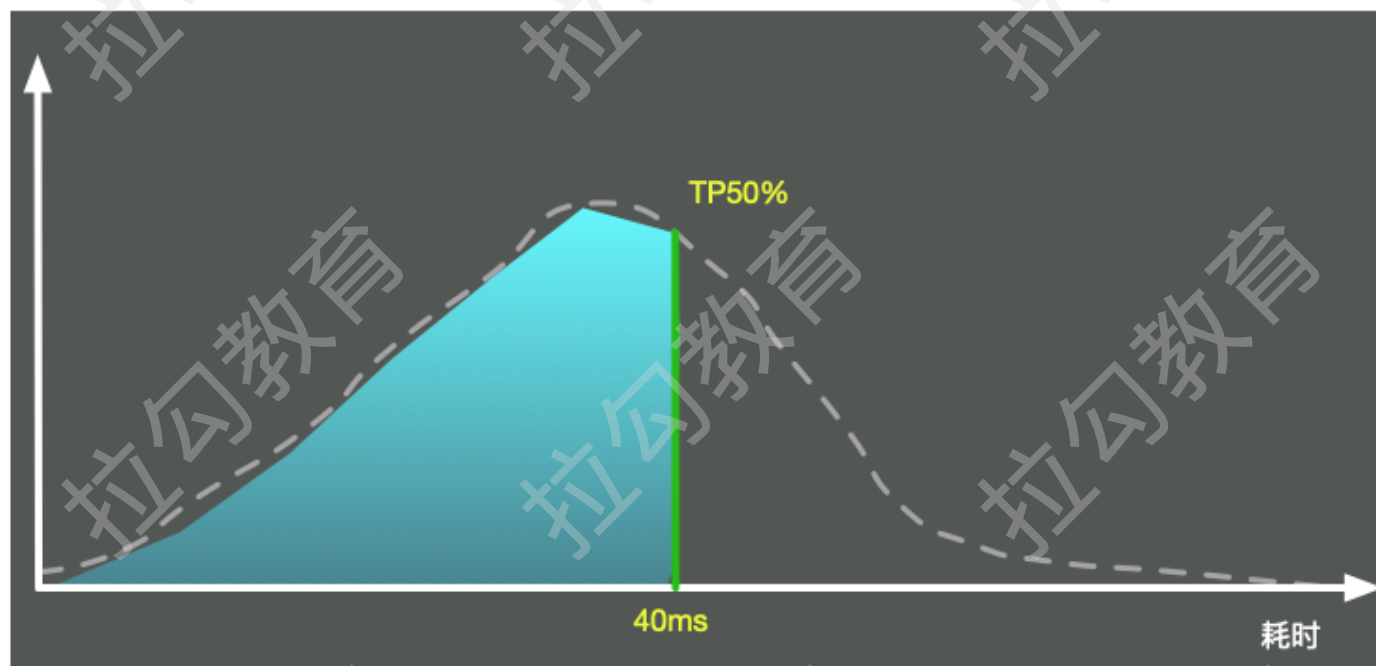
— 互联网人实战大学 —



百分位数 (Percentile)

拉勾教育

— 互联网人实战大学 —



这个指标是**非常重要**的，它能够反映出应用接口的整体响应情况

百分位数 (Percentile)

拉勾教育

— 互联网人实战大学 —

- 在这些高稳定性系统中，目标就是要**干掉严重影响系统的长尾请求**

这部分接口性能数据的收集，采用更加详细的日志记录方式，而不仅仅靠指标

比如，将某个接口，耗时超过 1s 的入参及执行步骤，详细地输出在日志系统中



- **并发量**是指系统同时能处理的请求数量，这个指标反应了系统的负载能力
- 在高并发应用中，仅仅高吞吐是不够的，它还必须**同时能为多个用户提供服务**

并发高时，会导致很严重的共享资源争用问题，需要减少资源冲突，以及长时间占用资源的行为

- 针对于响应时间进行设计，一般来说是万能的
- 因为响应时间减少，同一时间能够处理的请求必然会增加



— 互联网人实战大学 —



APP 启动速度

秒开率

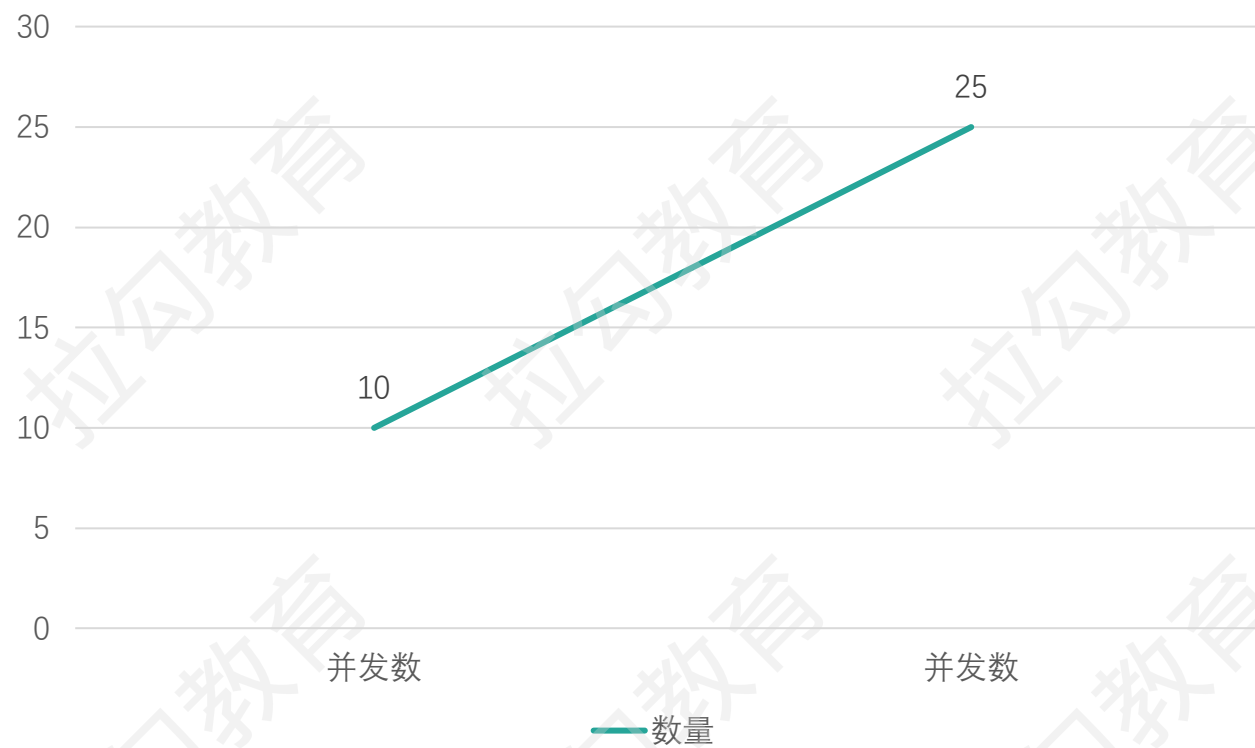
拉勾教育

— 互联网人实战大学 —

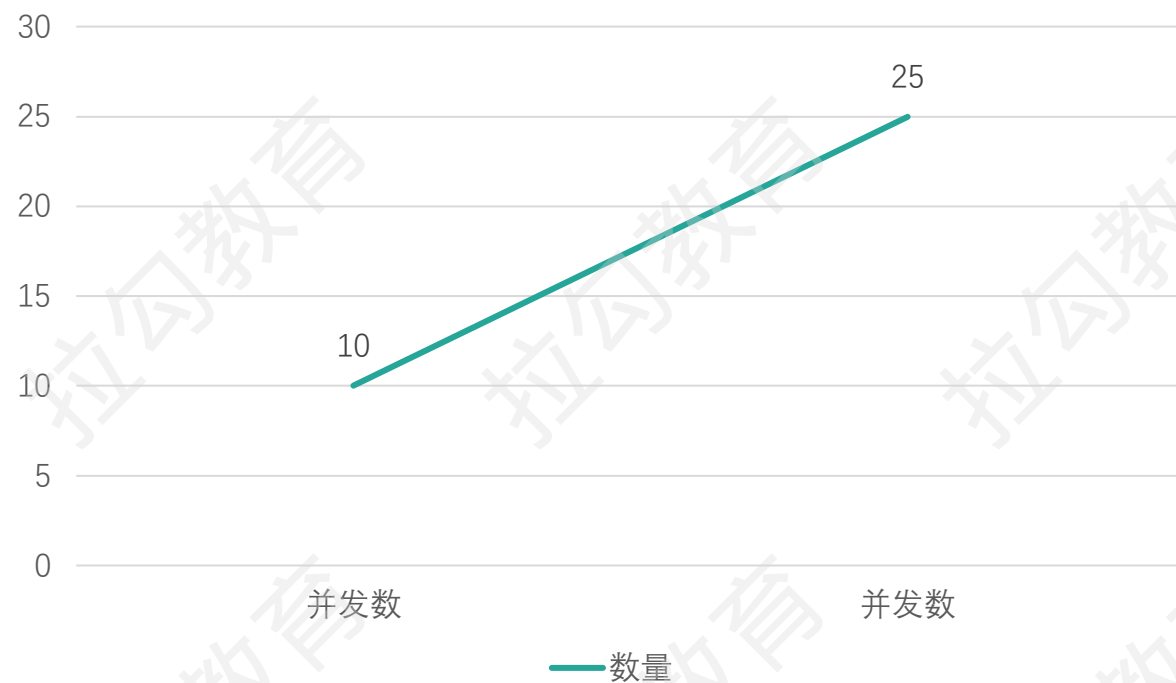


APP 启动速度

接口测试



接口测试



出现事故原因

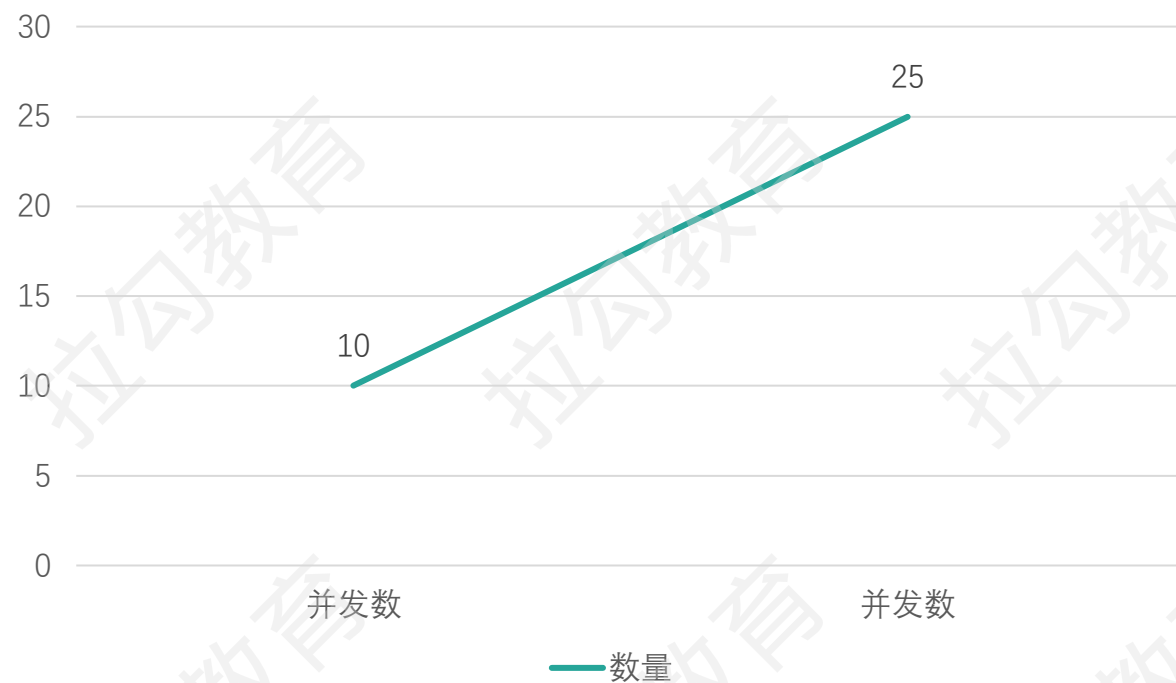
因为接口返回的都是无法使用的数据

正确性 ★

拉勾教育

— 互联网人实战大学 —

接口测试



出现事故原因

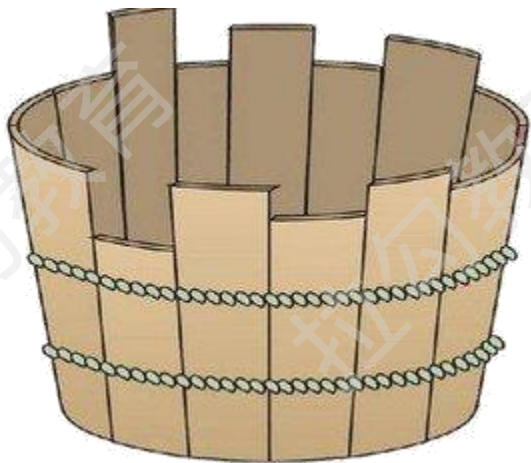
项目中使用了熔断

性能优化

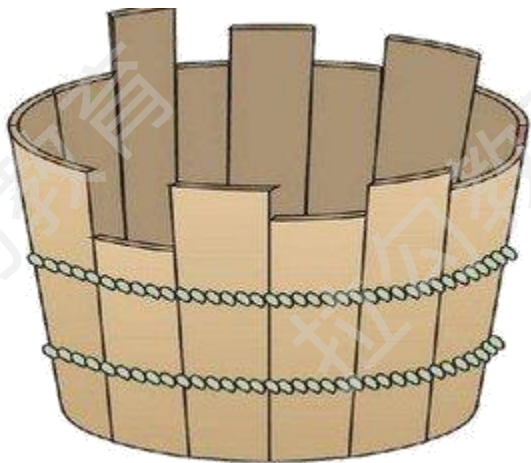
基础测试

木桶理论

Amdahl 定律



- 在解释系统性能上，也非常适合
- 组成系统的组件，在速度上是良莠不齐的
- **系统的整体性能**，就取决于系统中最慢的组件



- 比如：

在数据库应用中，制约性能最严重的是**落盘的 I/O 问题**

硬盘是这个场景下的短板

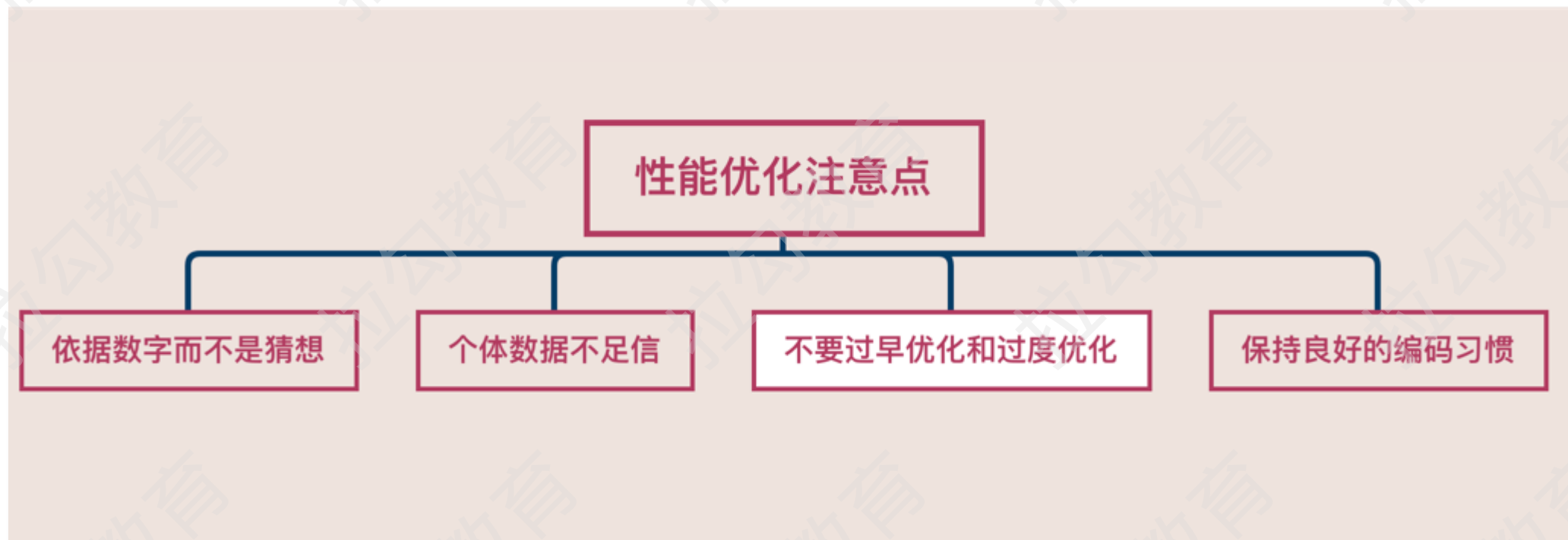
首要的任务就是补齐这个短板

基准测试 (Benchmark)

- 并不是简单的性能测试，是用来**测试某个程序的最佳性能**
- 应用接口往往在刚启动后都有短暂的超时

在测试之前，对应用进行预热，消除 JIT 编译器等因素的影响

在 Java 里就有一个组件，即 JMH，就可以消除这些差异



依据数字而不是猜想

拉勾教育

— 互联网人实战大学 —

第一位

性能分析

第二位

性能优化

辅助

直觉

依据数字而不是猜想

拉勾教育

— 互联网人实战大学 —

性能优化

根据难度和影响程度

依据数字而不是猜想

拉勾教育

— 互联网人实战大学 —

击破影响最大的点

性能优化

根据难度和影响程度

将其他影响因素逐一击破

个体数据不足信

拉勾教育

— 互联网人实战大学 —



正在加载中.....

因为个体请求的小批量数据，可参考价值并不是非常大
响应时间可能因用户的数据而异，也可能取决于**设备和网络条件**

个体数据不足信

拉勾教育

— 互联网人实战大学 —



正在加载中.....

合理的做法，是从统计数据中找到一些规律，比如上面所提到的平均响应时间、TP 值等
甚至是响应时间分布的直方图，帮助我们评估性能质量

个体数据不足信

拉勾教育

— 互联网人实战大学 —

“过早的优化是万恶之源”

——Donald Knuth

- 如果一项改进并不能产生明显的价值，那我们为什么还要花大力气耗在上面呢？
- 比如，某个应用已经满足了用户的吞吐量需求和响应需求

但有的同学热衷于 JVM 的调优，依然花很大力气在参数测试上，这种优化就属于**过度优化**



个体数据不足信

拉勾教育

— 互联网人实战大学 —

- 时间要花在刀刃上，需找到最迫切需要解决的**性能点**，然后将其击破
- 比如，一个系统主要是慢在了数据库查询上，结果你却花了很大的精力去优化 Java 编码规范

这就是偏离目标的典型情况





正确的做法是，项目开发和性能优化，应该作为两个独立的步骤进行
要做性能优化，要等到整个项目的架构和功能大体进入稳定状态时再进行

保持良好的编码习惯

拉勾教育

— 互联网人实战大学 —

保持好的编码规范

使用合适的设计模式

小结

拉勾教育

— 互联网人实战大学 —

本课时

了解了衡量性能的一些指标，比如常见的吞吐量和响应速度

探讨了一些其他的影响因素，比如并发量、秒开率、容错率等

谈到了木桶理论和基准测试等两种过程方法，并对性能测试中的一些误区和注意点进行了介绍

在平常的工作中，也应该尽量使用专业术语，这样才能对系统性能进行正确评估

Next: 第02讲 《性能优化都有哪些考量点》

拉勾教育

— 互联网人实战大学 —



下载「拉勾教育App」
获取更多内容