

拉勾教育

— 互联网人实战大学 —

《Java 性能优化实战 21 讲》

李国 前京东、陌陌高级架构师

— 拉勾教育出品 —

21 | 性能优化的过程方法与求职面经总结

即使熟悉了开发中的各项技术和优化技巧

但在真正的性能优化场景下，自己依旧很难开展优化任务

即使熟悉了开发中的各项技术和优化技巧

但在真正的性能优化场景下，自己依旧很难开展优化任务

本课时

总结一下性能优化的过程方法

性能优化需要多方面权衡

拉勾教育

— 互联网人实战大学 —

业务需求层面

有一个报表业务，查询非常缓慢，有时候甚至会造成内存溢出
经过分析，发现是查询时间跨度范围太大造成的

硬件层面

有一个定时任务，每次都把 CPU 用得满满的。由于系统有架构上的硬伤，无法做到横向扩容
技术经过评估，如果改造成按照数据分片执行的模式，则需要耗费长达 1 个月的工时

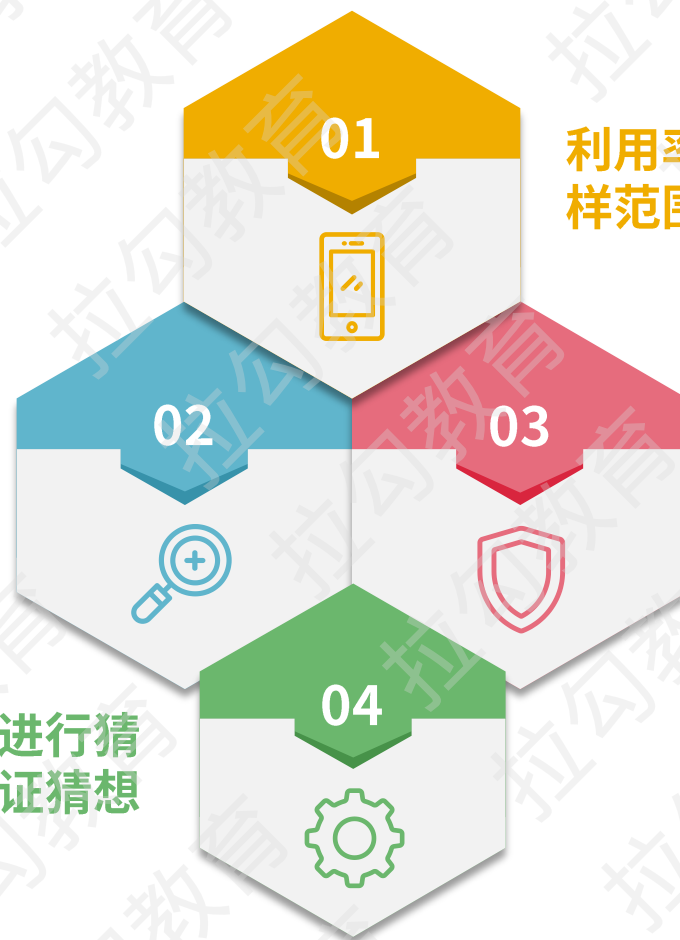
如何找到优化目标?

拉勾教育

— 互联网人实战大学 —

饱和度：一般指资源是否被合理利用，能否用分担更多的工作

联想信息：对引起的原因进行猜测，并用更多的工具验证猜想



利用率：一般是瞬时值，属于采样范围，用来判断有没有峰值

错误信息：错误一般发生在问题严重的情况下，需要特别关注

如何找到优化目标?

拉勾教育

— 互联网人实战大学 —

CPU

查看 CPU 使用可以使用 top 命令，尤其注意它的负载（load）和使用率
vmstat 命令也可以看到系统的一些运行状况

内存

可以使用 free 命令查看，尤其关注剩余内存的大小（free）
对于 Linux 系统，启动之后由于各种缓存和缓冲区的原因，系统内存会被迅速占满
top 命令的 RES 列，显示的就是进程实际占用的物理内存

如何找到优化目标?

拉勾教育

— 互联网人实战大学 —

网络

iostat 可以看到占用网络流量最高的进程

通过 netstat 命令或者 ss 命令，能够看到当前机器上的网络连接汇总

I/O

通过 iostat 命令，可以查看磁盘 I/O 的使用情况

iostat 可以查看占用 I/O 最多的进程

如何找到优化目标?

拉勾教育

— 互联网人实战大学 —

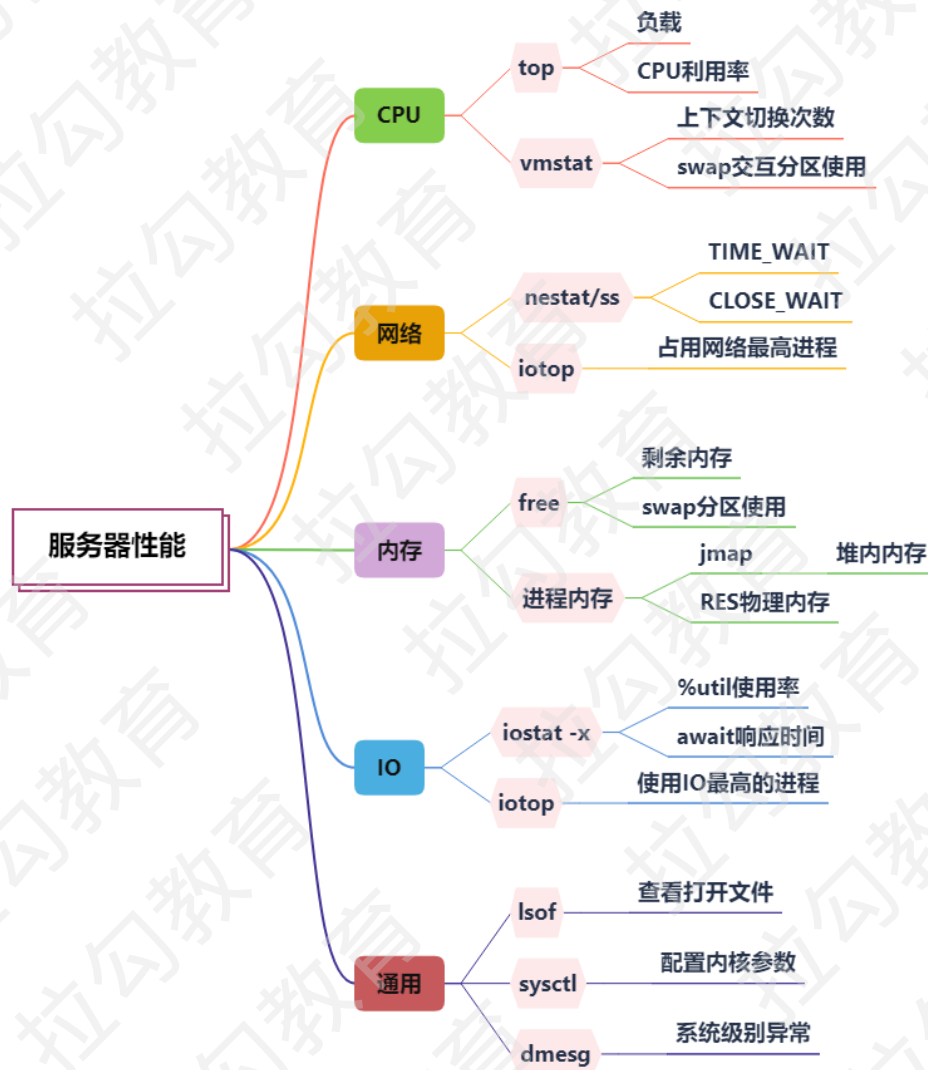
通用

- **lsdf 命令**可以查看当前进程所关联的所有资源
- **sysctl 命令**可以查看当前系统内核的配置参数
- **dmesg 命令**可以显示系统级别的一些信息

如何找到优化目标?

拉勾教育

— 互联网人实战大学 —



信息收集

nmon 是一个可以输出系统整体性能数据的命令行工具，需要开启 JMX 端口才能够被远程连接

监控

目前最流行的组合是 prometheus + grafana + telegraf，可以搭功能强大的监控平台

压测工具

- wrk 是一个命令行工具，可以对 HTTP 接口进行压测
- jmeter 是较为专业的压测工具，可以生成压测报告

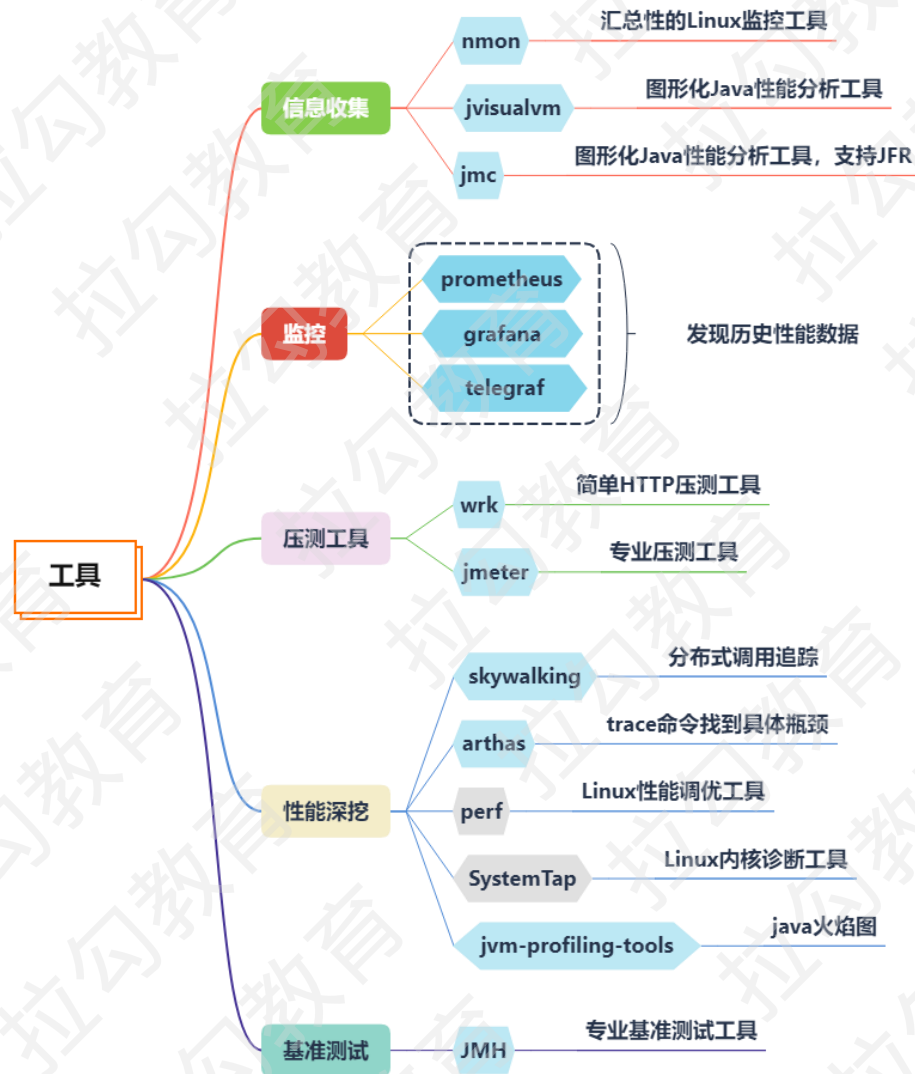
性能深挖

- skywalking 可以用来分析分布式环境下的调用链问题，可以详细地看到每一步执行的耗时
- 使用命令行工具 arthas 对方法进行 trace，最终也能够深挖找到具体的慢逻辑
- jvm-profiling-tools，可以生成火焰图，辅助我们分析问题
- 针对操作系统的性能测评和调优工具，还有 perf 和 SystemTap

常用工具集合

拉勾教育

— 互联网人实战大学 —



CPU 问题

在 Linux 系统上，通过 top-Hp 便能容易地获取占用 CPU 最高的线程，进行针对性的优化

棘手的性能问题：**线程都阻塞在 ForkJoin 线程池上**

代码在等待耗时的 I/O 时，采用了并行流（parallelStrea）处理

但是 Java 默认的方式是所有使用并行流的地方，公用了一个通用的线程池

这个线程池的并行度只有 CPU 的两倍

内存问题

内存问题通常是 OOM 问题。如果内存资源很紧张，CPU 利用率低，可以考虑时间换空间的方式
一般在高并发的应用中，会把 SWAP 关掉，因为它很容易会引起卡顿

I/O 问题

通过调整日志级别，或者清理无用的日志代码，缓解磁盘 I/O 的压力

通过 RPC 调用一个远程的服务，期望使用 NIO 来减少一些无效的等待，或者使用并行来加快信息的获取

类似于 ES 这样的数据库应用，数据写入本身，就会造成繁重的磁盘 I/O

可以增加硬件的配置，比如换成 SSD 磁盘，或者增加新的磁盘

数据库服务本身，也会提供非常多的参数，用来调优性能

比如 ES 的 segment 块大小，translog 的刷新速度等

基本解决方式

拉勾教育

— 互联网人实战大学 —

网络问题

在 SpringBoot 的配置文件中，通过配置下面的参数开启 gzip

```
server:  
  compression:  
    enabled: true  
    min-response-size: 1024  
    mime-types: ["text/html","application/json","application/octet-stream"]
```

基本解决方式

拉勾教育

— 互联网人实战大学 —

网络问题

通过替换 Feign 的底层网络工具为 OkHTTP，使用 OkHTTP 的透明压缩（默认开启 gzip）

即可完成服务间调用的信息压缩

基本解决方式

拉勾教育

— 互联网人实战大学 —

网络问题

通过将结果集合并，使用批量的方式，可以显著增加性能

使用 netstat 命令，或者 lsof 命令，可以获取进程所关联的

TIME_WAIT 和 CLOSE_WAIT 网络状态的数量

基本解决方式

拉勾教育

— 互联网人实战大学 —



解决方式： 加入一个中间层，有缓冲 / 缓存，以及池化这三种形态
以牺牲信息的时效性为代价，加快信息的处理速度

缓冲

缓冲能够消除两方的速度差异，以批量的方式，来减少性能损耗

缓存

有堆内缓存和分布式缓存之分，有些对性能要求非常高的场景，甚至会有多级缓存的组合形态

池化

对资源进行集中管控，以池化的思想来减少对象的创建成本



普通的事务可以通过 Spring 的 @Transactional 注解简单的实现

重构

通过重构，改变代码的组织结构

通过设计模式，可以让我们的代码逻辑更加清晰，在性能优化的时候，可以直接定位到要优化的代码

异步化

异步化多采用生产者消费者模式，来减少同步等待造成的性能损耗

但它的编程模型难度较大，需要很多额外的工作

代码层面——资源利用不足

拉勾教育

— 互联网人实战大学 —

在高并发场景下，轮转就更加重要——**属于在一定压力下系统的最优状态**

资源不能合理的利用，就是一种浪费

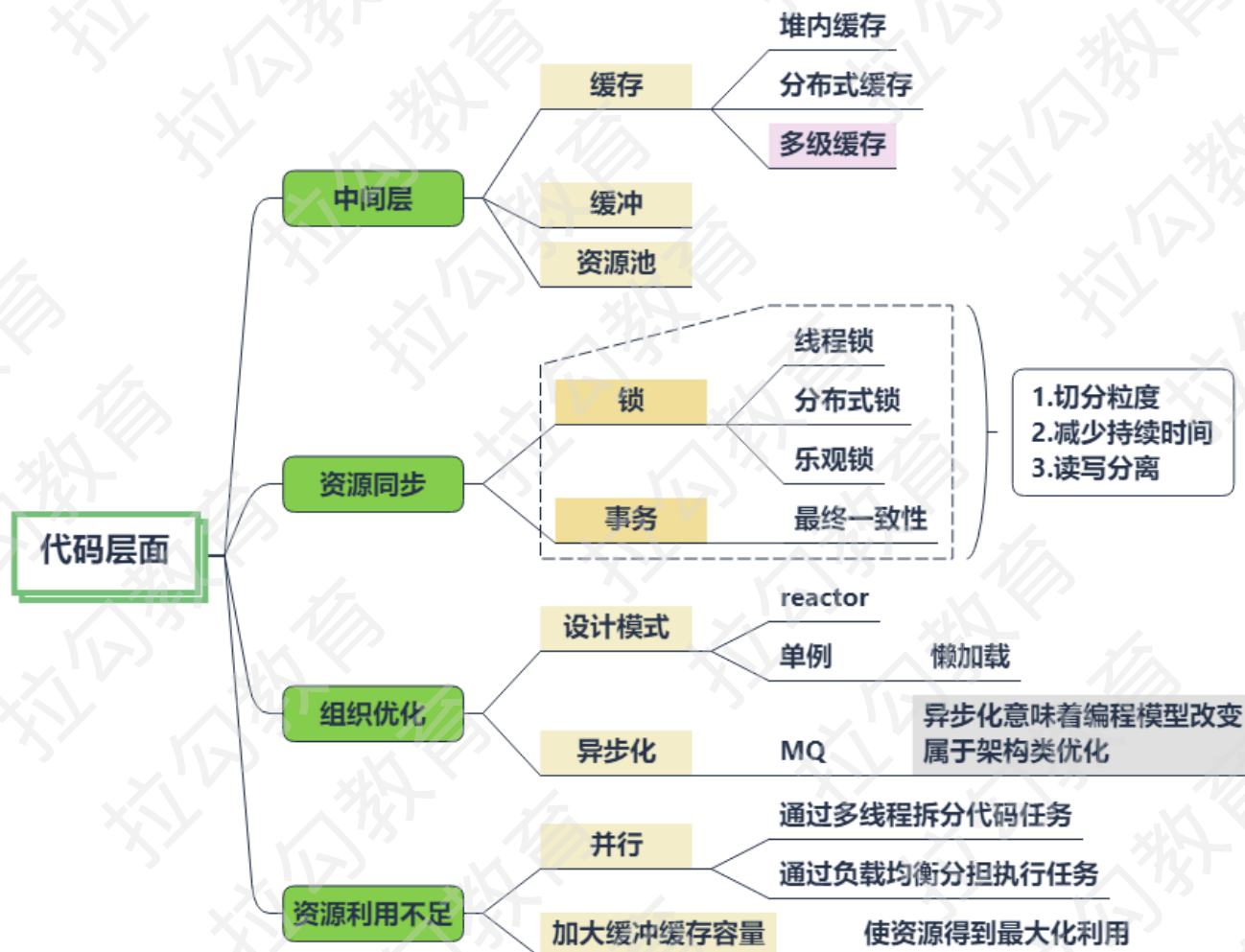
比如，业务应用多属于 I/O 密集型业务，如果让请求都阻塞在 I/O 上，就造成了 CPU 资源的浪费

比如，监控到 JVM 的堆空闲空间，长期处于高位，可以考虑加大堆内缓存的容量，或者缓冲区的容量

代码层面——资源利用不足

拉勾教育

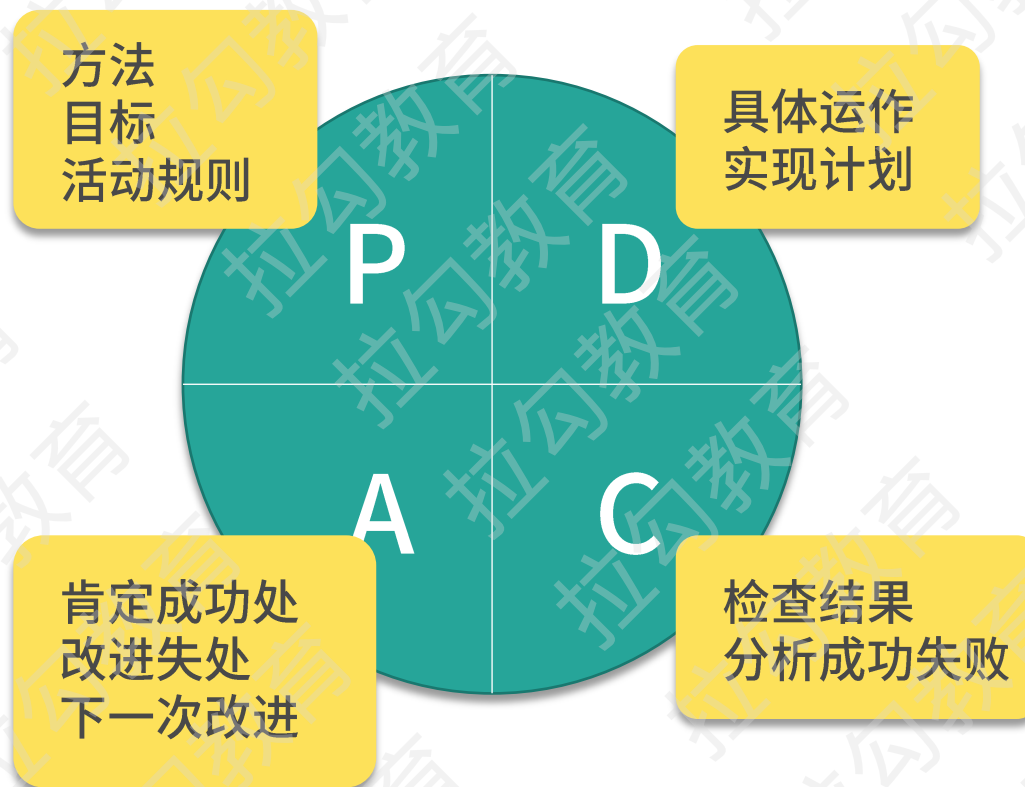
— 互联网人实战大学 —



PDCA 循环方法论

拉勾教育

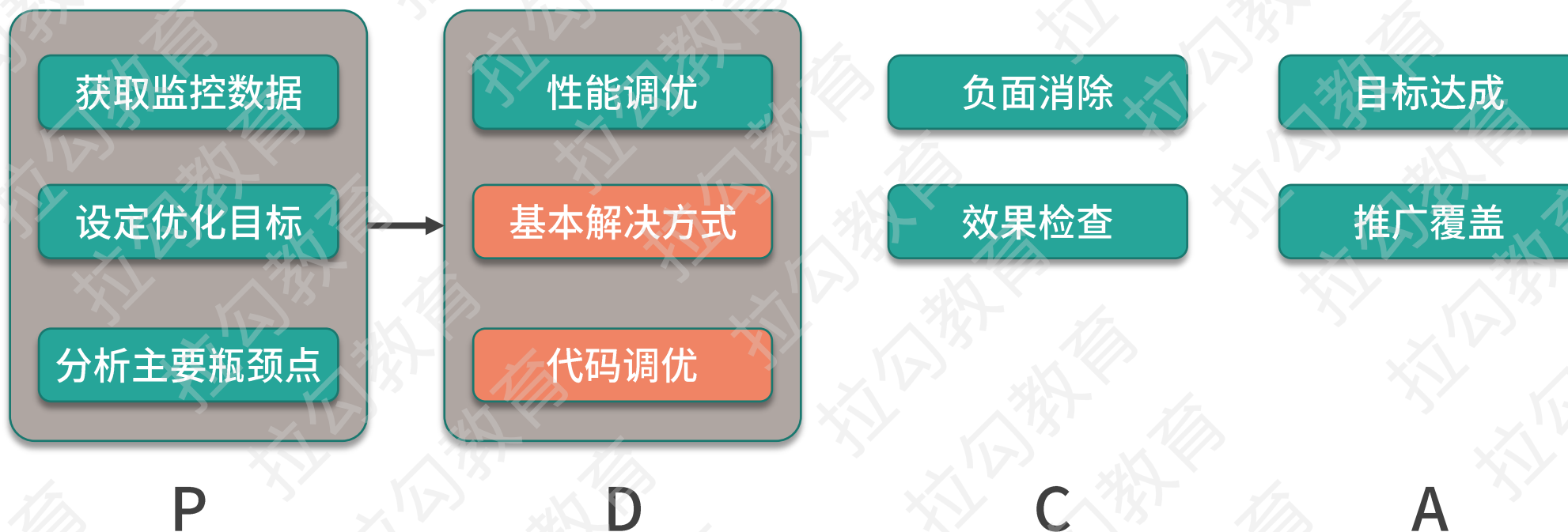
— 互联网人实战大学 —



PDCA 循环方法论

拉勾教育

— 互联网人实战大学 —



关注“性能优化”的副作用问题

性能优化的面试题，一般都是穿插在其他题目里的

你不仅需要关注“性能优化”本身，还需关注“性能优化”之后的问题

掌握好“性能优化”基础知识

你都对JVM做了那些优化，有哪些性能提升？

为什么互联网场景下通常使用乐观锁？

发散、综合性题目提前准备

你在项目中做过哪些性能优化方面的工作？

你是如何指导团队做性能优化的？



Next: 结束语 | 《实践出真知》

拉勾教育

— 互联网人实战大学 —



关注拉勾「教育公众号」
获取更多课程信息