



下载APP

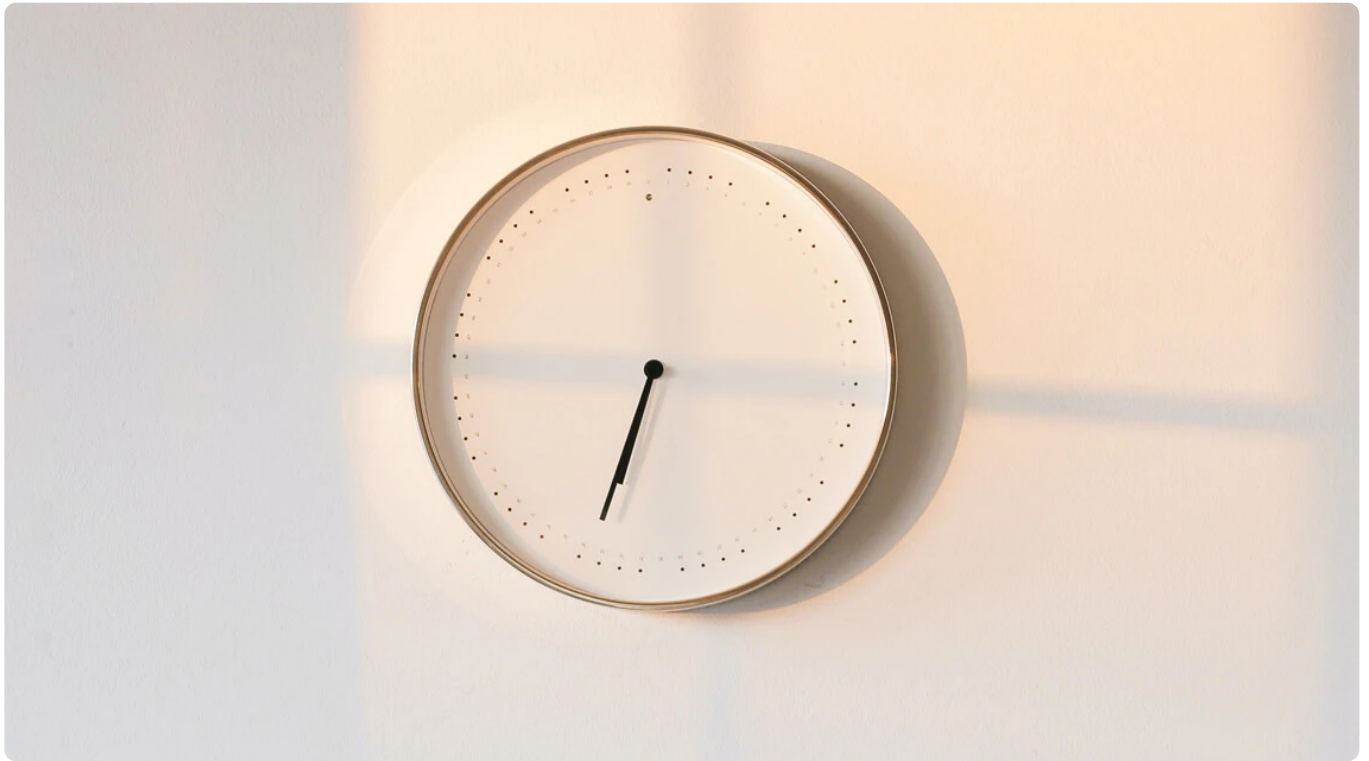


22 | 性能调优方法论：如何科学高效地定位性能问题？

2021-07-06 尉刚强

《性能优化高手课》

课程介绍 >



讲述：尉刚强

时长 20:49 大小 19.07M



你好，我是尉刚强。

一提起软件系统中的性能问题，可能你首先会想到的是 CPU 使用率过高，或是内存占用率太大，导致程序执行速度变慢，然后关注点只停留在软件实现层面的性能调优。

实际上，这种以最小化资源占用为导向的性能优化，它的核心目标是**降低成本**。但对于产品而言，最关键的其实是客户视角所关注的业务性能，比如用户的平均响应时延、99% 的用户响应时延分布等，而**这类性能优化的核心目标是提升用户体验，提升产品的行业竞争力**。



可是问题在于，在互联网服务领域，从客户视角去定位分析业务的性能问题，会有一定的挑战，主要原因有两点：

首先，针对一些存在业务性能问题的场景，其系统内的服务或组件，并不一定处于饱和状态，所以你不能直接从系统资源级监控中识别出问题；

其次，如果只是个别服务或组件处于饱和状态，往往也可以通过弹性扩展来更快地提升性能体验，所以这种也并不是最棘手的性能问题。

我们再来看看嵌入式领域，就拿无线通信领域来说，它也存在很多业务性能优化非常苛刻的场景，比如说 FTP 下载速率优化、速度平稳没有毛刺优化等。

那么在这类系统的实现过程中，导致出现业务性能问题的原因会更多更复杂，其问题可能发生在链路上任何一个网元设备内，或具体设备内的任何一个软件组件或硬件单元上。因此，定位分析业务层性能问题的挑战会更大。

而且事实上，定位分析业务的性能问题，也是很多程序员都很头疼的问题。它需要你具备很高的业务能力，包括对业务流程的熟悉度、对软件架构及软件内实现逻辑的理解程度，甚至是对 OS 和硬件原理都要有深入的理解。

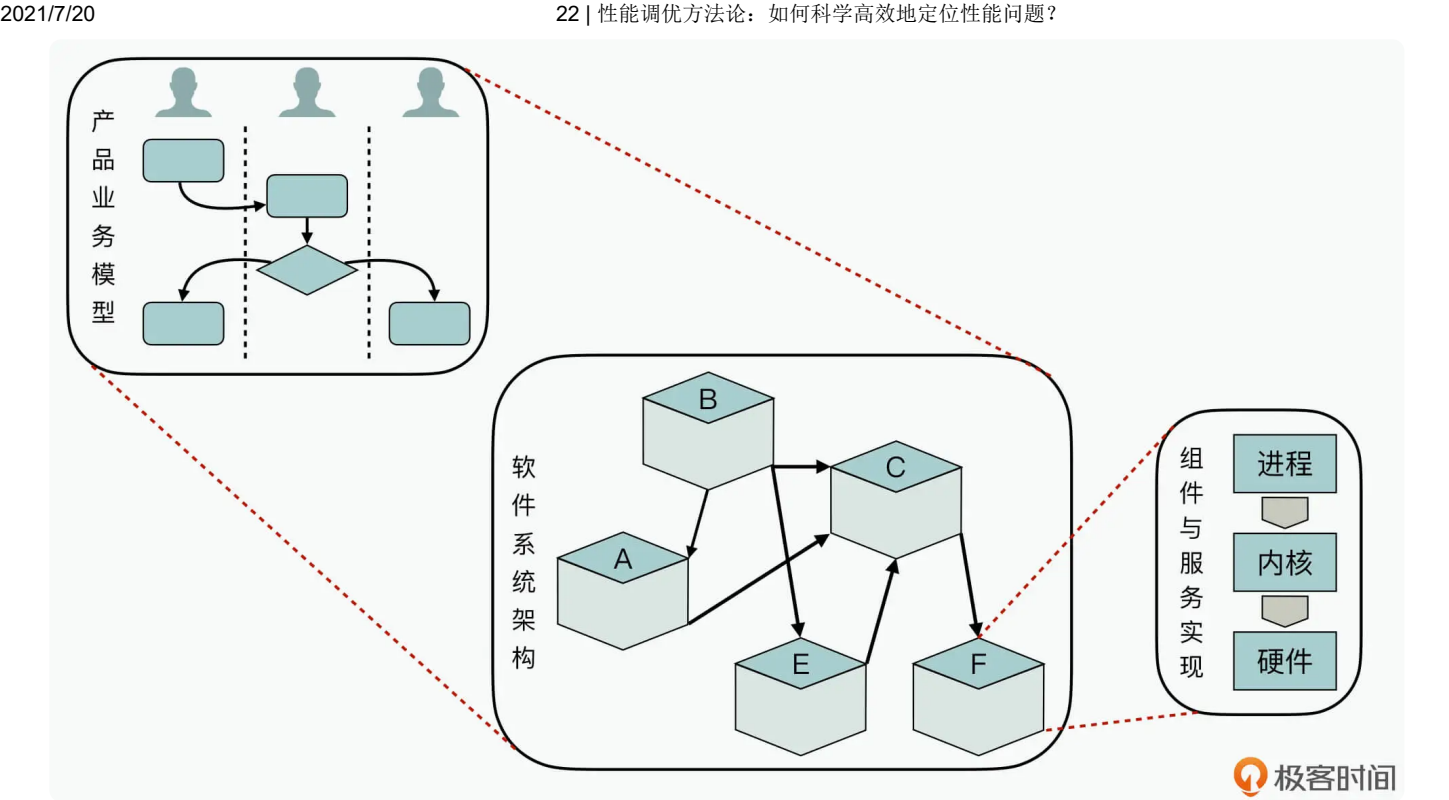
不过就我的实践经验来看，即使是掌握了这些信息，如果没有系统的定位分析方法的指导，那你依旧很难定位出性能问题。

所以今天这节课，我会给你分享一套性能调优方法论，带你理解企业应用系统架构的整体逻辑和工作流程，然后在此基础上去了解都有哪些引起性能问题的潜在软硬件瓶颈点，最后会介绍系统分析定位的方法。基于这个方法，当你再碰到比较棘手的业务的性能问题时，就可以做到有的放矢，使用这套系统定位分析方法去解决真实的性能问题。

好了，下面我们就来了解下系统的整体架构。

系统架构视图

为了能更好地分析与定位复杂的性能问题，你首先需要理解整个系统架构视图，其核心主要包含这三层：产品业务模型、软件系统架构、组件或服务实现，如下图中所示：



下面我来介绍下不同层次模块的具体组成结构。

产品业务模型

这是系统对外提供的业务功能与逻辑。就拿购物流程来举例，它对外提供的功能逻辑是：浏览商品—> 添加购物车—> 查看库存—> 支付—> 发快递等，这其实是站在用户视角所感知到的与系统发生交互的过程。这个业务模型是软件业务领域建模阶段的重要产物，也是每个系统级产品都应该包含的模型。

软件系统架构

软件系统架构，它表示的是整个软件系统到具体组件或服务之间拆分逻辑，以及组件或服务间的交互关系，如图中 A、B、C、E、F 等。这些软件运行单元可以是一个微服务实例，也可以是一个组件实例，它们会通过通信与交互支撑实现复杂的业务功能。

实际上，不管是嵌入式领域或者互联网领域，软件系统架构设计本质上都是对业务功能拆解的一个过程，通过拆分形成具有独立清晰边界的、更小的软件运行单元。

组件或服务实现

这是组件或服务的内部代码实现，它会依托于进程、内核、硬件的协作来完成核心的业务功能。我们知道，系统的业务性能是由硬件、OS、软件实现以及之上的业务流程综合决定的，当我们碰到业务性能问题时，会基于系统的架构视图从上到下进行分析，最后总能找到具体的制约性能的瓶颈点。

那么，如果你可以提前认识软硬件中常见的一些性能瓶颈点，了解它们对系统性能的影响，就能够帮助你更加准确地分析业务性能问题。但是，很多程序员在定位性能问题时，喜欢只聚焦到某个点上持续优化，而这个点可能并不是性能瓶颈，所以最后很难有比较好的效果。

那么，系统潜在的性能瓶颈点都有哪些呢？接下来，我们就一起来看看。

潜在的性能瓶颈点

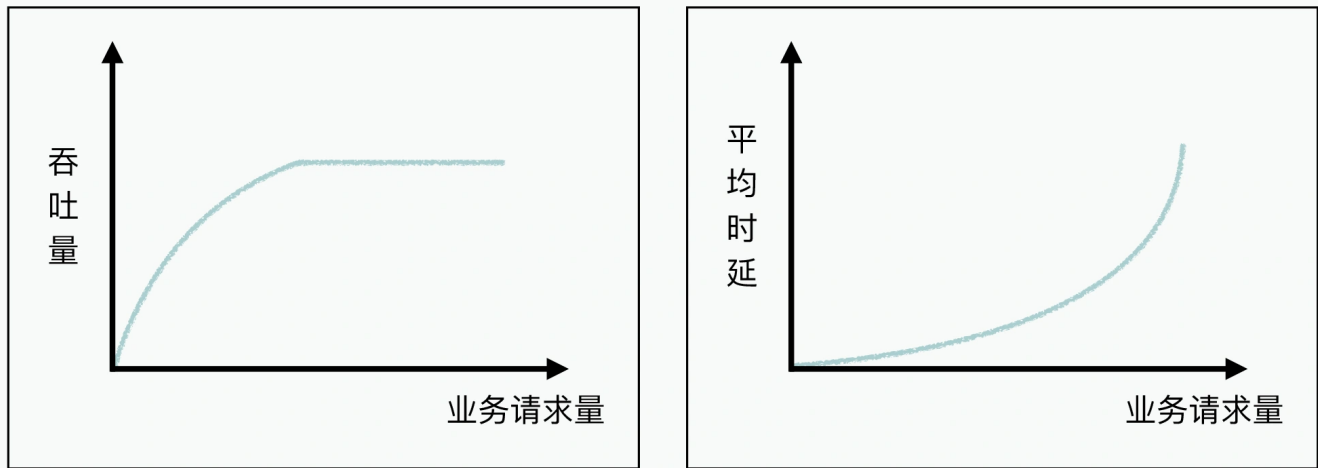
首先一般来说，性能瓶颈都是由一些关键资源使用过度而引起的，而且不同资源使用到达瓶颈（饱和）状态对性能产生的影响和规律是不一样的。在计算机系统中，每种硬件资源如 CPU、Cache、内存、磁盘、网络接口、总线等，都可能成为性能瓶颈。这是因为当硬件处理到达饱和状态时，会直接引起硬件上的软件运行性能下降，这样最终就会导致业务性能受到影响。

但是在通常情况下，监控工具对硬件资源的使用状态监测比较成熟，而且当资源使用到达饱和状态时，对性能的影响规律也比较明确，所以这部分内容就不在咱们今天课程的讲解范围之内了。

那么相比较而言，软件实现所导致的性能问题及影响，其实很容易被忽略。所以今天，我会重点介绍下由于软件实现而引起业务性能问题的瓶颈点，来帮助你在分析业务性能问题的过程中，更好地去识别和发现这些问题。

串行资源受限

第一类典型的瓶颈问题就是串行资源（或互斥资源），这是一个可能触发业务性能问题的软件实现。下图是串行资源的扩展对性能影响的模型图，表示随着业务规模的扩大，对性能造成的影响。



极客时间

如图中左侧所示，因为串行资源是有限的，随着业务请求量增加，当资源使用饱和后，会导致请求处理吞吐量到达峰值后就不能再提升了；同时，如图的右侧所示，当串行资源使用饱和后，平均处理时延也会因为排队或者阻塞而不断拉长。

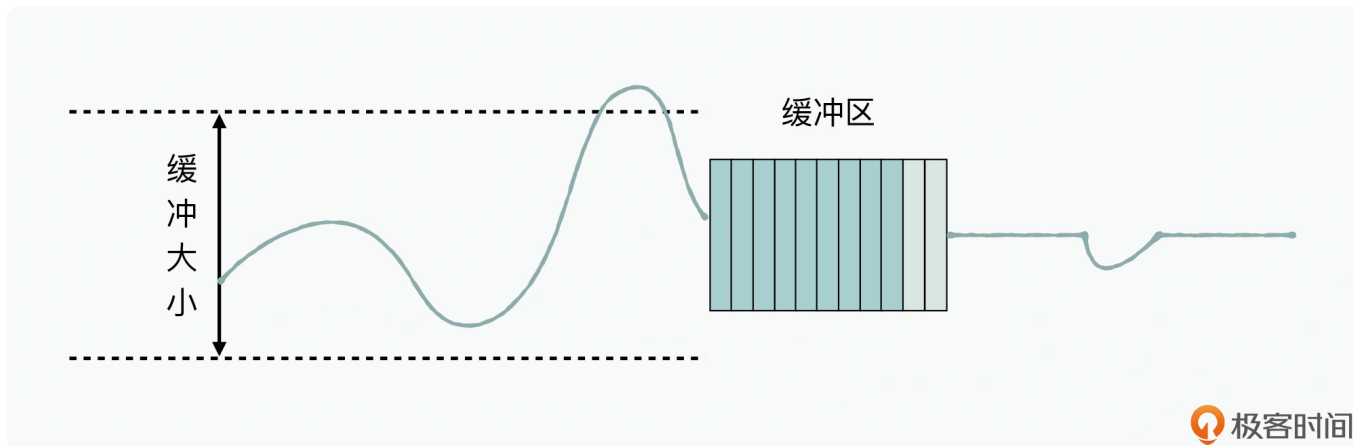
在软件实现中，对性能影响比较大的串行资源种类有很多，粒度从小到大可以包含：互斥锁、并行设计中的串行部分、系统依赖的不可扩展的服务或接口等，都有相似的影响。

但你也要注意，在业务场景中，还有不少资源数目是大于等于 2 的，比如线程池、数据库连接池、其他不能无限扩展的服务或接口。这种有限软件资源的场景，它们对性能的影响与串行资源对性能的影响也是比较类似的，所以你同样可以参考串行资源对性能影响的分析视图。

缓冲类资源消息溢出

第二种容易引发性能问题的软件实现是缓冲类资源，比如缓冲区、消息队列、消息中间件等都属于缓冲类资源。缓冲技术可以实现削峰填谷的机制，可以平滑上游和下游处理速度之间的差异。

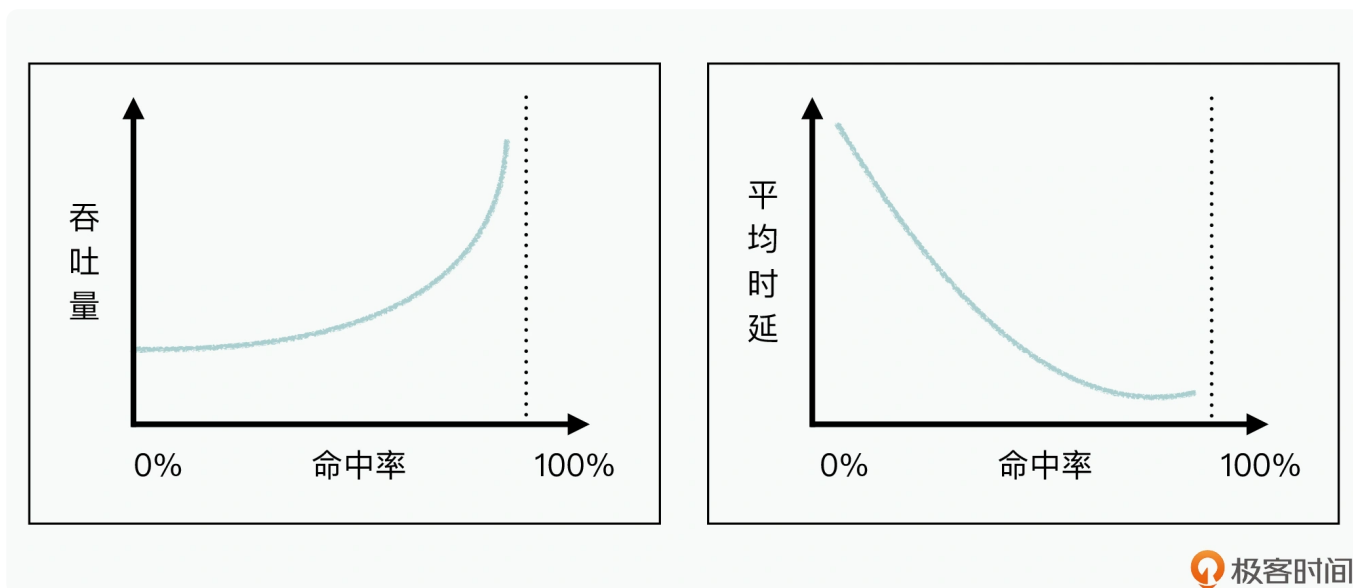
如下图所示，如果缓冲区设置过小，当上游请求到达峰值时，可能会造成部分请求被阻塞或丢弃，影响到业务性能。如果缓冲区设置越大，所实现削峰填谷的能力则会更强。



但如果缓冲区设置得过大，也会造成内存资源的浪费，所以缓冲区大小设置对性能的影响也很关键。

缓存命中率过低

缓存技术是软件实现层做性能优化的一个重要手段，同时也是影响业务性能的潜在因素之一。我们知道，缓存使用有一个很重要的指标：**缓存命中率 = 缓存命中个数 / (缓存命中个数 + 缓存未命中个数)**，这个指标对业务性能的影响如下图所示：



可以看到，当命中率提高时，由于命中之后的请求处理时延变短，就导致了业务的平均处理时延降低，并且吞吐量也会随之提高；反之，当缓存命中率降低时则性能下降会比较明显。

软件 Bug

最后，在软件实现中还有一个对性能影响非常大的因素，就是软件 Bug。

在以往定位分析业务性能问题的过程中，由于软件 Bug 而引起的性能问题不占少数。比如，代码中本来是批处理操作，但是由于处理错误而退化成循环调用等；或是在使用 Cache 技术时，由于 Cache Key 的构造错误，导致缓存永远不可能命中，引起业务性能突然下降等。

不过看到这里，你可能要问了：是不是所有的业务性能问题，都是由于软硬件层的资源性能瓶颈触发而引起的呢？

我认为并不是，有些性能问题可能是由于业务模型或者流程本身的问题，软件和硬件只是载体而已。比如说，因为业务中的一些限速策略，就导致所处理的性能受限。

另外在有些更复杂的系统业务设计中，也可能会引入性能问题，这就像一些设计不合理的十字路口红绿灯一样，会导致拥堵严重。不过，针对这种因业务模型等而引起的性能问题，我们其实可以通过调整红绿灯时间，也就是通过调整软件不同模块的业务职责和接口实现，来显著改善拥堵现象。

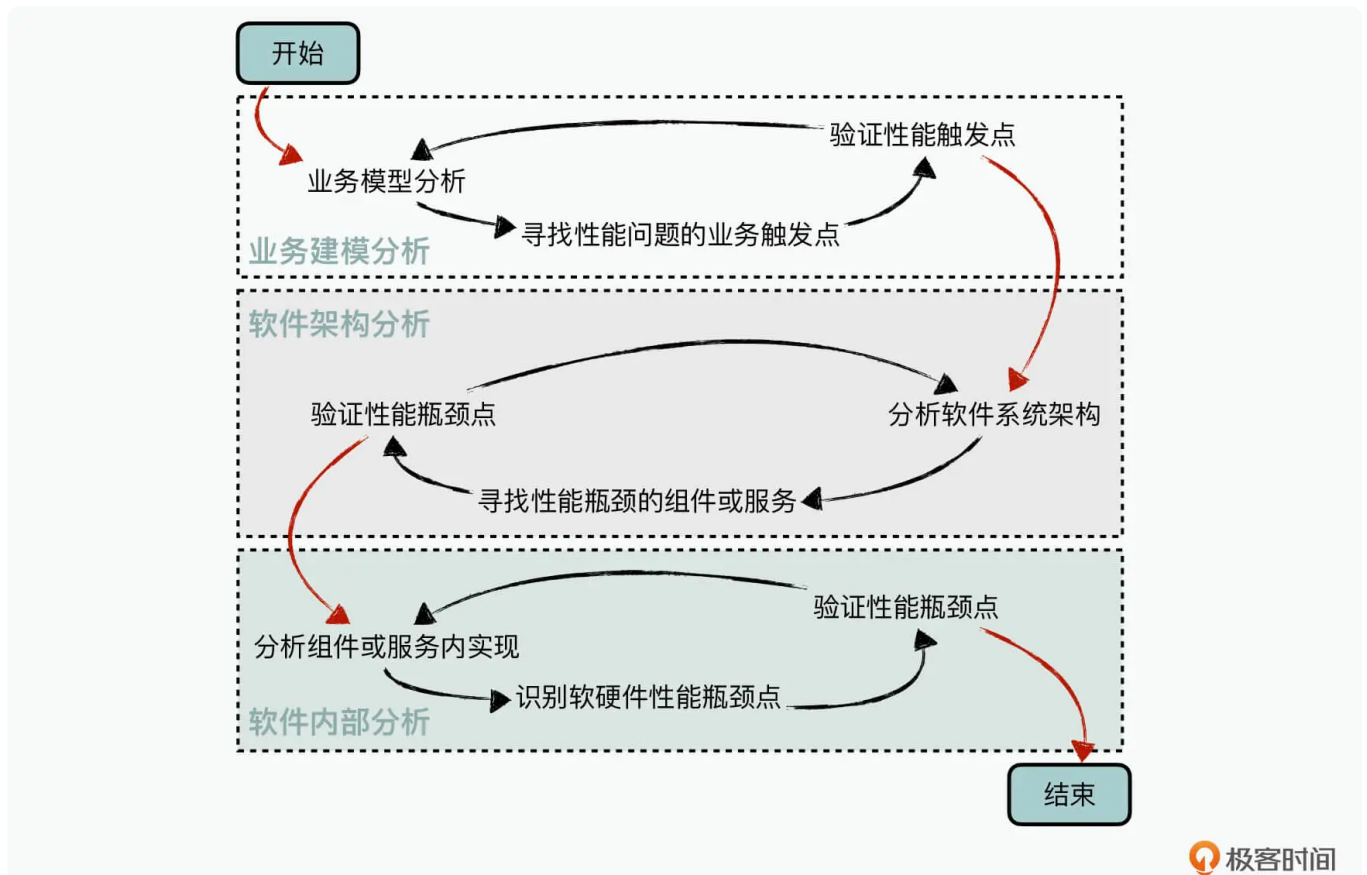
好了，总之在软件实现当中，可能引起潜在性能问题的因素非常多。既然如此，我们就需要系统的分析定位方法指导，否则会很难准确识别出系统中所有的性能瓶颈点。

所以接下来，我们就一起探讨下这个系统分析定位方法吧。

系统分析定位法

实际上，分析定位性能问题的方法有很多种，比如 USE 方法（即从检查每个资源、使用率饱和的视角寻找可能的性能瓶颈）、从下向上逐步分析系统中资源使用指标法来寻找性能瓶颈。其他的还有随机变动讹方法（这是一种随机猜测然后验证的方法）、科学实验法等。针对特定的性能问题时，可能每种定位分析法的效果也不一样。

不过，今天我要给你介绍这种方法，是我根据以前分析各种性能问题积累的经验，而总结出的一种方法思路，它是一种相对系统且比较高效的方法，你在碰到各种业务领域的性能问题时，都可以使用。具体如下图所示：



整个定位分析方法从上到下一共分为三层，分别是：业务模型分析、软件架构分析、组件或服务实现分析。下面我来给你一一介绍下。

业务模型分析

首先，业务模型分析的目标是**寻找到引起业务性能问题的业务触发点，并验证分析的正确性**。越复杂的业务模型，导致出现业务性能问题的根因可能会越隐蔽，如果你在业务模型分析时，发现的根因不对，那么后续花费更多力气去进行深入分析也都将是徒劳。

比如，分析 TCP 流量下降的性能问题，可能是下行链路的问题，也可能是上行链路的问题。如果因为上行链路延迟增大，而导致 ACK 反馈不及时触发拥塞控制，导致 TCP 流量不高，那么你花费再大的精力，在下行链路上分析也都是无用的。

因此，验证业务层性能问题根因分析的结论是否正确，是很有必要的。这里我建议你可以**通过打桩方式**将根因触发点暂时规避掉，来观察性能问题是否改善，以此作为有效的验证手段。

当然，不同系统间的业务模型差异非常大，所以业务模型没有通用的方法或规律可循，你只能靠对业务逻辑的深入理解，并根据业务层实现的运行状态监控信息来进行分析。当做

好了业务模型分析后，你会发现导致性能问题的原因，可能是由某个具体业务流程引起的，然后你就可以针对这个具体业务流程，深入到软件架构中进一步分析。

软件架构分析

在软件架构分析中，我们主要依赖软件架构中的组件或服务的接口交互关系来进行分析，通常对大的系统级产品而言，组件或服务间接口的交互信息一定可以被监控获取分析的。这样，根据获取的接口交互监控信息，我们可以寻找到触发业务性能问题的具体组件或服务。

另外，与业务模型分析一样，我们在软件架构分析阶段识别出的性能瓶颈的组件或者服务，也需要做进一步验证，在确保分析结果正确之后，再进行下一阶段的深入分析。

我以前在通信领域从事性能问题分析时，每个子系统组件都是由不同团队开发维护的，而当面对复杂的业务性能问题时，我就需要沿着子系统组件的接口边界逐步排查分析，将性能问题交接给下一个子系统组件，并且还要提供充足的分析与验证信息。但即使是在这样的场景下，也依旧会存在分析出错，导致返工的情况。所以这一点，你一定要注意。

软件内部分析

但事实上，在互联网业务领域分析性能问题时，不管是业务模型分析还是软件架构分析的阶段，验证分析结论正确性的过程，其实都很容易被我们所忽视，从而就会导致性能问题分析定位的返工或者出错。

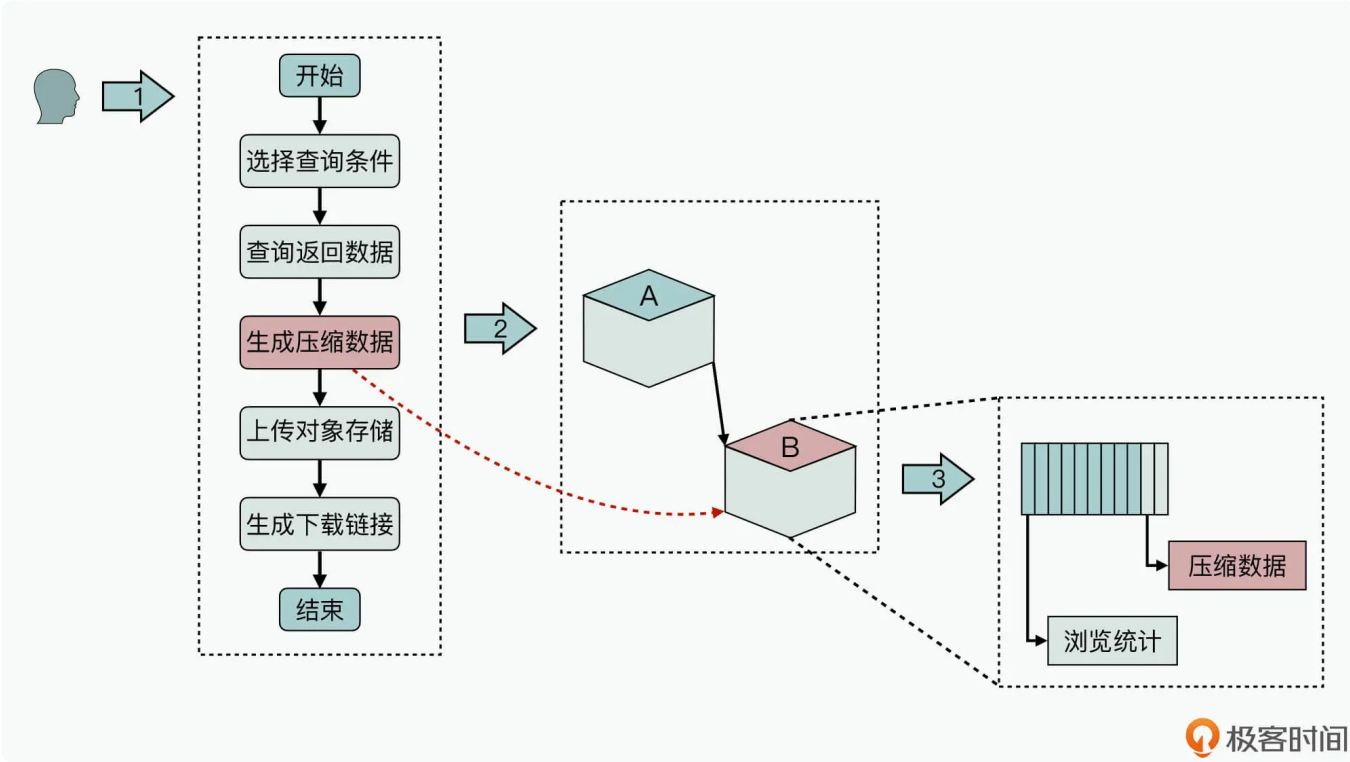
所以就我的实践经验来看，当定位到具体的某个组件性能瓶颈时，你就可以深入到组件或服务内实现来进行分析了，然后根据我前面介绍的资源性能瓶颈点以及对性能的影响规律，来准确识别发现导致业务性能问题的软硬件资源。

那么下面，我就通过一个具体的性能问题案例，来帮助你更深入地了解如何使用这套定位分析方法，来定位分析性能问题。

如何在真实的数据分析业务中应用这套方法论？

这是我曾经处理在线数据分析业务中碰到的真实性能问题，当时产品有部分用户投诉，在 Web 页面中的查询数据下载功能性能很差，造成用户等待时间长。而我采用的就是这套性

能定位方法，具体定位过程如下图所示：



第一阶段：通过业务模型分析，识别出性能问题的业务流程。在互联网服务领域，有很多业务的性能问题是用户帮你发现的，但在其他的业务领域内，业务性能瓶颈点则需要依赖业务领域的监控统计，并结合业务模型一起分析才可以识别。

因此对于这个案例来说，系统中可能包含了很多种业务流程。那在业务模型分析阶段中，我们找到了存在性能问题的业务流程是：查询数据下载功能。具体用户业务流程是：选择查询条件—> 查询返回数据—> 生成压缩数据—> 上传对象存储—> 生成下载链接。

然后我们就可以基于这个业务流程，来寻找定位引入性能瓶颈的组件和服务。

第二阶段：在软件架构分析阶段，我们就可以根据存在性能问题的业务流程，寻找到软件导致性能瓶颈的组件和服务。

这个阶段我主要使用系统的微服务间的接口日志进行分析，主要基于Elasticsearch+Kibana 工具，沿着业务请求接口链路寻找到具体组件或者服务，最后将性能问题定位到一个确定的微服务组件中。

第三阶段：组件或服务内实现分析，在这个阶段深入到导致性能问题的微服务，通过查看内部实现的相关监控或统计信息，发现导致性能问题的原因是：压缩数据处理与浏览统计

逻辑使用相同的计算任务队列，由于之前的突发浏览业务导致队列中积攒了很多待处理的浏览统计任务，从而引起压缩数据处理任务被排队阻塞，时延很长。

如此一来，通过仔细分析后我们发现，由于用户无法感知到浏览统计逻辑，而压缩数据处理对用户来说却比较敏感，所以这里使用一个相同的任务队列，就造成非关键业务逻辑影响到了核心业务的处理时延。

所以，我们最后的解决方案就是，将两种任务的处理队列拆分开，从而避免类似的性能问题再次发生。

小结

我们在分析处理业务性能问题的时候，其实很容易犯一些错误，比如当发现一个性能问题的根因或瓶颈后，就像抓住了救命稻草一样，将所有的问题都往上靠，最后等解决了这个根因之后，却会发现系统的性能问题并没有得到彻底解决。所以，我们的性能分析定位方法，一定要科学且严谨的分析所有可能的原因。

今天课程上介绍了当碰到业务性能问题时，可以从上到下依次进行**业务模型分析、软件架构分析、组件或服务内实现分析**，然后识别出引起业务性能问题的具体软件或硬件资源。在每个阶段的分析定位过程中，都需要进行验证工作，确保分析结果准确性和完备性。

思考题

在互联网业务流程中对数据库的操作请求非常多，如果业务中经常出现数据库查询时间比较长的情况，你一般都会怎么分析定位呢？

欢迎在留言区分享出你的思考和答案，如果觉得有收获，也欢迎你把今天的内容分享给更多的朋友。

分享给需要的人，Ta订阅后你可得 **20** 元现金奖励

👍 赞 0 💡 提建议

© 版权归极客邦科技所有，未经许可不得传播售卖。页面已增加防盗追踪，如有侵权极客邦将依法追究其法律责任。

上一篇21 | 性能CI：性能测试也可以集成到Pipeline中吗？

下一篇23 | 监控分析：你的性能调优工具足够有效吗？

更多学习推荐

Java 面试必考 300 题

最新汇总

限时免费领取



精选留言

写留言

由作者筛选后的优质留言将会公开显示，欢迎踊跃留言。