



下载APP



23 | 监控分析：你的性能调优工具足够有效吗？

2021-07-08 尉刚强

《性能优化高手课》

课程介绍 >



讲述：尉刚强

时长 16:11 大小 14.82M



你好，我是尉刚强，今天我们来聊聊如何选择合适的性能调优工具。

实际上，选择合适的性能调优工具，就像医生会借助一些医疗设备来诊断病因一样，主要是为了验证性能问题根因分析的准确性，以此更好地支撑性能调整与优化。

可是一方面，由于现在业务复杂度的不断提升，软件的技术架构越来越复杂，我们对软件系统运行的监控分析也越来越困难了。因此，如果仍继续使用单一功能的 OS 或硬件级的监控工具来获取监控数据的话，其实很难有效地支撑性能问题分析工作。



而另一方面，正是由于业务复杂度的提升，导致监控工具也越来越复杂化，我们需要获取监控调优相关的数据种类也越来越多了。所以手动分析这些数据的效率，就成为了性能调优工作的重中之重。

也就是说，我们需要**建立一套系统的监控分析工具体系**，这样既能够帮助获取多个层级的监控数据，也可以通过体系化的工具选择和使用，避免一些繁琐的重复性工作，来有效提升分析效率。

所以今天这节课，我就来给你介绍下目前常用的一些观测技术手段，让你明白它们在监控分析中起到的作用和价值。然后，我会带你学习如何针对软件系统去设计整个监控分析的工具体系，以此获取并分析相关的监控数据，从而帮助你更加高效地进行性能调优。

常用的观测技术手段

那么首先，我们来了解下目前业界比较常用的观测技术手段，主要有四种，分别是计数器、跟踪、剖析和监视。

计数器

计数器是针对特定事件的一种统计手段，这是一种成本非常低的观测手段，所以通常你可以一直开启它来监测特定事件发生的次数。

在 OS 级别与硬件级别有很多计数器可以使用，比如使用命令 `iostat`，来获取与 io 相关的统计信息，如下所示：

```
k6 iostat
```

disk0			disk2			cpu			load average		
KB/t	tps	MB/s	KB/t	tps	MB/s	us	sy	id	1m	5m	15m
45.79	11	0.50	21.86	0	0.00	2	3	95	1.82	1.57	1.67

```
k6
```

跟踪

这是通过收集一系列操作或事件内容来跟踪分析业务流程的观测手段，比如日志打印就是一种跟踪手段。不过通常来说，跟踪是存在一定的开销的，所以一般不建议全部开启。以日志打印为例，在高性能软件系统中，一般只有错误或告警级别才会默认开启。

剖析

这是通过对目标进行采样或者快照来归纳目标特征的观测手段，但是受制于采样周期的影响，往往会存在采样反馈特征不准确或者有偏差的问题。因此，一般只有在分析软件代码函数调用栈概率分布的时候，我们才会使用这种方式。举个简单的例子，使用 JProfile 来分析 Java 程序的热点代码，就是采用的这种方式。

监视

这是一种主动触发控制的观测手段，它会通过显式触发获取特定周期内的信息，来支持监控数据获取的过程，比如操作系统级别的 Sar 命令等。

其实对于操作系统内核和硬件来说，已经有很多基于这四种观测手段开发的监控工具，来实现对其内部进行深入的性能监控分析。比如说，使用 iostat 命令获取 IO 统计、使用 top 命令获取进程使用统计等（如果你想更深入地学习这部分知识，可以参考学习 Brendan Gregg 《性能之巅》）。

但是，对于实际开发的软件系统来说，仅仅基于操作系统内核和硬件级的监控观察手段，来分析复杂的业务性能问题其实还远远不够。

我举个例子。假设你的公司研发了一个演唱会购票交易网站，然后因为某个演唱会非常火热，在卖票期间出现整个网页卡死的性能问题。那么为了更好地监控分析这个性能问题，你就需要在业务中增加很多观测手段，比如获取网页在线人数、同时提交购票请求人数、购票交易平均时长等很多观测数据，来协助进行性能分析。

而这样一来，为了能更深入地对业务性能进行监控分析，你还需要在实际软件系统的开发过程中，单独定制化实现以上不同种类的观测手段。

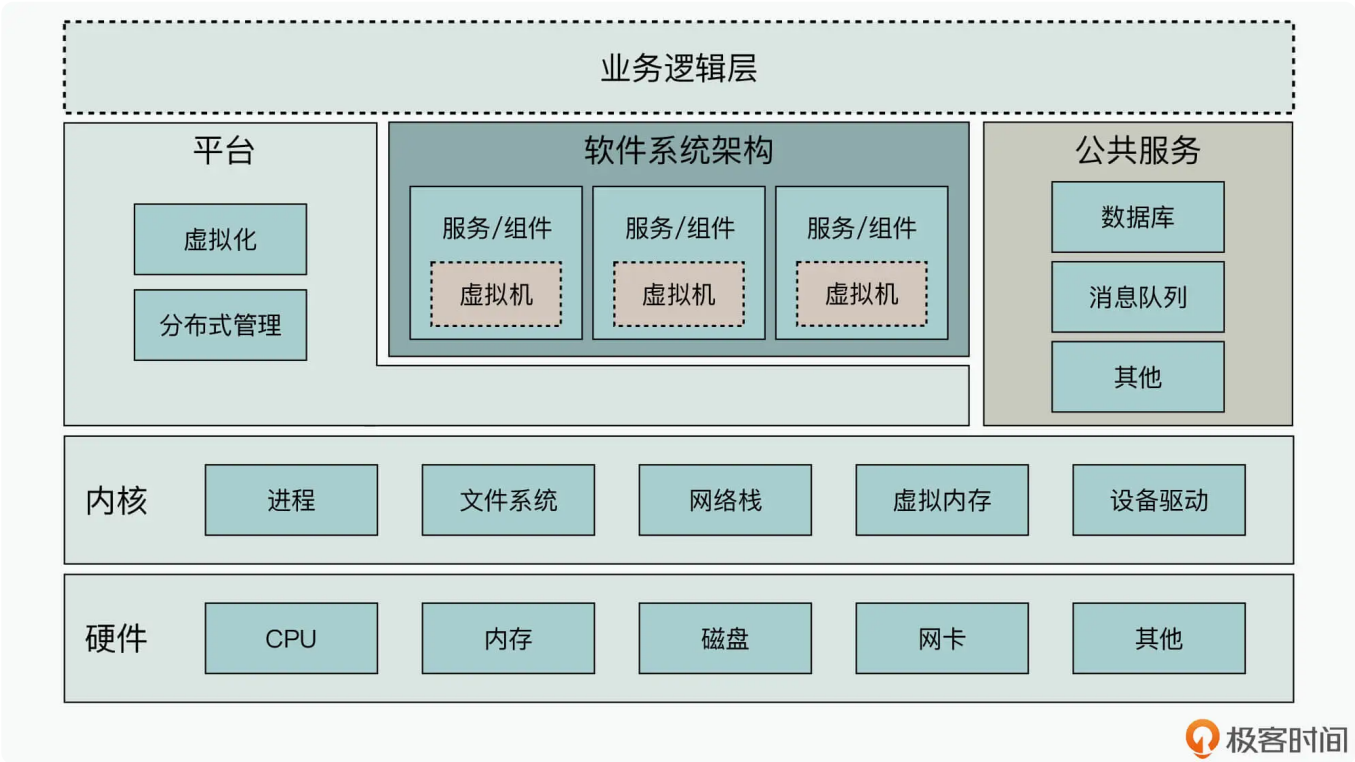
不仅如此，由于软件系统都是运行在操作系统 OS 和具体硬件芯片之上，所以针对一些复杂的业务性能问题，你还需要统筹好操作系统与软硬件定制开发的观测手段，一起配合，这样才能更加有效地支撑性能问题的分析定位。

可是要想实现以上目标，就需要我们对软件系统的性能监控观测节点和观测手段有个全面的了解和把握。所以接下来，我们就通过一个通用的软件系统观测视图，来了解下如何才能更有效地支撑性能问题的监控分析工作。

实际软件系统的观测节点和手段都有哪些？

现在的软件系统通常都不是单机系统，而是一个复杂的分布式系统或是云服务化产品。所以，针对软件系统的监控分析就不再会停留在单机上，而是需要对系统中很多的节点和元素进行观测。

下面展示的是一个软件产品或服务的系统观测视图，你可以认为图中的每一个矩形元素都是一个观测节点，它们都提供了前面四种观测手段中的一种或者几种观测手段。



那么在这个系统观测视图中，你可以发现上层的软件与服务是运行在下层的软硬件之上的，因此我们针对下层软硬件的监控分析，其实也可以支撑上层服务和组件的性能分析。

所以这里，我就从下到上来依次给你介绍下它们所提供的观测实现手段，帮助你更好地理解系统观测视图的全局。

硬件层的 CPU

首先，在硬件芯片层，每个物理设备都有对应观测方式。就拿 CPU 芯片来说，现在的 CPU 芯片架构越来越复杂，所以如何写出对 CPU 更加友好的代码，充分发挥 CPU 的硬件性能，就是一个很重要的性能调优方向。

以 CPU 观测分析为例，现在的 CPU 大多都有性能监控单元（Performance Monitoring Unit, PMU），可以统计系统中发生的特定硬件事件，比如缓存未命中（Cache Miss）或者分支预测错误（Branch Misprediction）等。所以呢，你就可以利用这些观测数据来分析，并指导优化软件的编码实现，来提升在 CPU 芯片上执行效率。

以 Intel 的 CPU 体系架构为例，其内置的计数器有几百个，你可以使用工具 perf 获取这些计数器的值，再借助工具如 pmu-tools、intel-Vtune profile，并使用 [TMAM](#)（Top-down Microarchitecture Analysis Method，自顶向下的微架构分析方法），分析出**取指令阻塞**、**取数据阻塞**、**指令分支预测效率低**等影响 CPU 性能发挥的因素，从而指导编码优化来提升 CPU 的 IPC（单周期的执行指令数），以此提升软件的性能。

OS 内核层的进程

在芯片之上是操作系统内核层，包含了进程、协议栈、文件系统等功能，每个功能也都有对应的观测工具与手段。以进程为例，我们可以通过 ps、pmap 等命令获取内存和 CPU 使用统计信息，也可以通过 strace 等命令获取系统调用的跟踪信息。

云平台

现在，我们的软件系统通常并不是只运行在一个单独的操作系统内核上，而往往是运行在一个云平台上。针对互联网应用与服务产品而言，这个平台通常就是指的是 PaaS 平台，各大云厂商如 AWS、阿里云、Azure 等，也都提供了研发 PaaS 云平台，当然你也可以使用 kubernetes 来组件一个私用的 PaaS 云平台。

但是不管选择使用哪一款 PaaS 平台，它们其实都提供了对应的监控运维工具，可以支持获取平台上的弹性资源、负载均衡等相关的监控观测数据。

公共服务

另外，对于软件系统而言，通常还会依赖很多公共服务，常见的包括数据库、消息队列或其他第三方服务等。这些公共软件与服务，通常也都有对应观测与监控分析工具，以数据库 MongoDB 为例，有基于统计的观测工具 mongostat，还可以使用 Profiler 进行查询分析，也有专业监控分析工具 MMS（MongoDB Monitoring Service）等。

软件系统架构中的服务 / 组件

在操作系统与云平台之上是软件系统架构，软件系统架构是由很多个服务或组件组成的。首先，这些服务与组件通常都可以支持获取线程使用、内存使用、CPU 使用、IO 使用等**与底层相关监控观察数据**，只是因为运行虚拟机和编程语言的差异，可以选择使用的工具不同。比如 C/C++ 语言中，常用的剖析分析工具有 gperftools、gprof、Valgrind 等；针对 Java 语言，常用的剖析分析工具有 VisualVM、JProfiler 等。

其次，这些组件与服务还需要根据特定的业务场景，开发实现一些**自定义的观察手段或者工具**，支持从业务视角进行性能分析。比如说，增加业务相关的计数器统计，或添加动态 Trace 机制等。实际上这部分的观察手段与能力，也是很多业务性能问题分析的关键，所以是需要重点考虑并建设的部分。

再进一步，云平台、软件系统、公共服务一起协作实现了整个产品的业务功能，而对一个复杂的业务系统而言，也应该有对应的观测分析工具。所以，对于很多大型业务系统来说，通常都在后台管理工具中集成了各种观测数据分析工具，这样你就可以比较方便地进行性能问题分析和调优。

但针对一些不成熟的软件系统，它们提供的观测分析手段就可能会比较薄弱，从而就导致了对软件系统进行性能调优非常困难。

所以根据以上的介绍分析，你可能会发现，对一个业务软件系统而言，从上到下的可观测节点数目会非常多，而且每个观测节点提供的观测分析工具种类也很多。

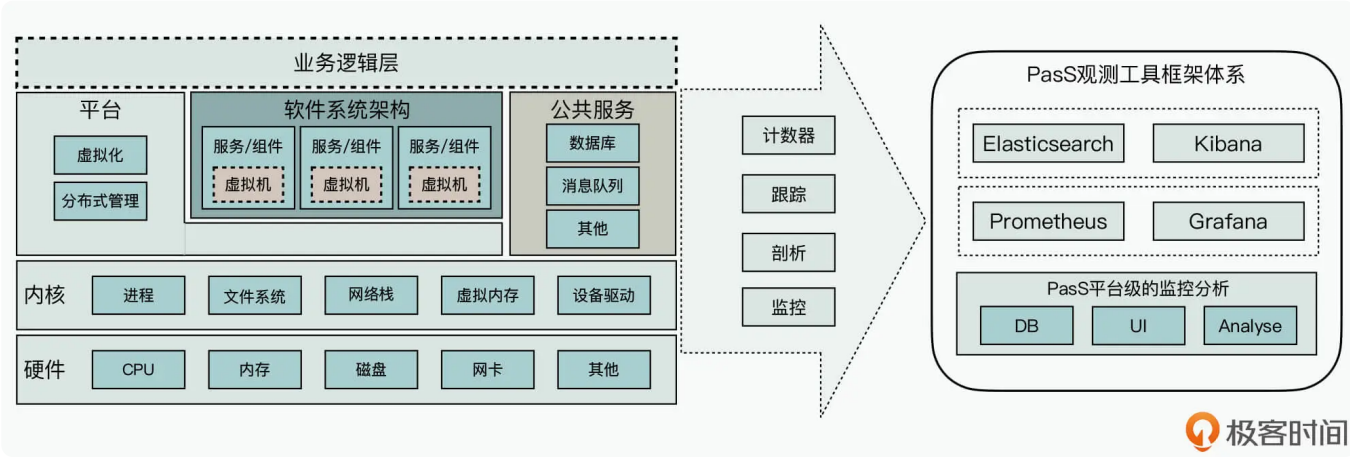
那么在对软件系统的性能进行监控分析时，如何寻找定位到待观测分析的节点，并使用对应的观测分析工具获取监控数据，这个过程通常是非常耗时的。从我的实践经验来看，很多研发人员进行性能调优时，绝大部分时间都花费在了学习和选择观测分析工具上，效率就非常低。

因此，构筑一套观测工具框架体系，来支撑对系统的观察分析，就是提升性能调优效率的关键。那么具体要如何做呢？下面我就来给你分享下我的构建思路。

如何构筑软件系统观测工具体系？

首先，为了更好地使用观察工具来支撑对系统的性能调优，我们应该从种类繁多的监控工具中解脱出来，**尽量使用统一的监控平台与工具来支撑性能调优。**

如下图右侧所示，我们可以尽量将系统中的观察分析数据对接到统一的分析平台上，来减少寻找使用各种工具的成本。



在通常情况下，各种云厂商提供的 PaaS 平台其实都携带了监控运维工具，并且也都自动集成了操作系统层、硬件层常用的观测数据，所以很多时候，你并不需要再深入到具体观测工具的使用方法当中。

其次，根据业务中针对性能调优的需求，你可以将软件组件与服务、业务层、公共服务层等相关的观察数据，**通过各种适配器和服**务对接到**统一的 PasS 监控运维工具中**。这样做的好处是可以将不同系统服务间获取的监控数据关联起来分析，从而能够更好支撑性能问题的分析与定位。

在我以前参与的一个性能优化项目中，其产品是部署在 AWS 上的，为了更好地支撑性能优化，我们首先将 ClickHouse、MongoDB 等重要观察数据，都注入到了统一的 Paas 监控数据分析平台中。这样当出现性能问题时，只需要基于这个 Paas 平台上的监控数据，就可以初步确定性能问题发生在哪个服务或组件上。

而如果你参与的项目案例中，并没有这个 PasS 数据分析平台，或者平台能力并不成熟，你也可以基于相关的开源技术来构筑这部分工具能力。比如，使用 Elasticsearch 和 Kibana，来记录跟踪业务处理日志或组件与服务间的交互日志；或者是使用 Promethoeus 和 Grafana，来记录系统中相关的计数器事件等。

其实很多时候，我们在进行系统调优的过程中，也是构筑系统级监控分析工具的最佳时机。如果我们将这部分能力集成和沉淀下来，就可以帮助自己提升后续性能调优的分析工作效率。

小结

俗话说得好：工欲善其事，必先利其器。在对一个软件系统进行深入的性能分析与调优时，你需要对多层级的很多观测节点获取测量分析数据，而如何使用工具来高效获取这些测量数据，就是提升性能调优效率的关键。

在今天的课程上，我介绍了把系统软件与业务级、公共服务级相关的性能观测数据都集成到统一的 PasS 数据分析平台上的方法与思路。你需要重点关注不同层级的观察节点，都提供了哪些观察手段与能力，然后当你在碰到性能问题时，就可以基于上节课学习的性能分析方法，寻找到比较好的切入观察点，并获取观察数据来支持性能分析。

这样，你就可以借鉴今天的学习内容，构筑一套自己的系统观测工具体系，并在性能调优的过程中，逐步集成和完善监控分析工具的能力，以支持更高效的性能调优。

思考题

你认为针对互联网服务级别的软件系统，在构筑一套完整的观察工具体系的过程中，哪些观察工具的优先级是更高的呢？欢迎在留言区分享你的思路和见解，我们一起讨论。如果觉得有收获，有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。

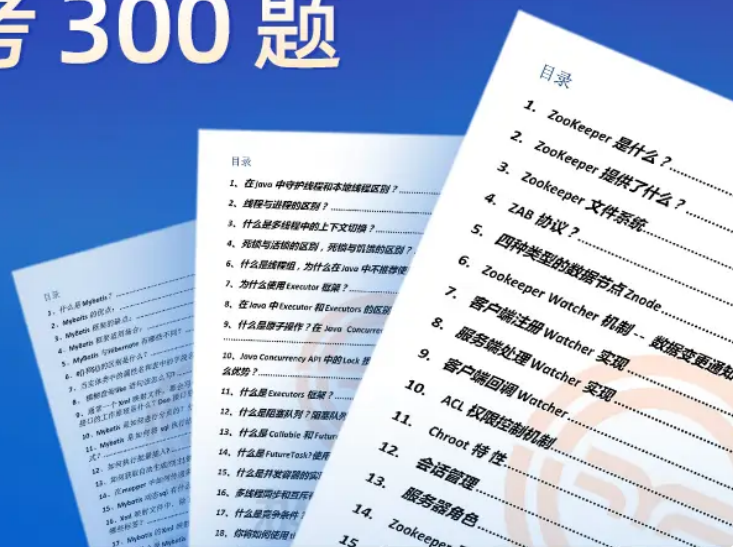
上一篇 22 | 性能调优方法论：如何科学高效地定位性能问题？

下一篇 24 | 动态监控：你的产品系统中有动态监控的能力吗？

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 (1)

 写留言

feihui

2021-07-14

本质就是分层级监控，很多地方都是用结果来说原因，颠倒逻辑，难受

