



下载APP



21 | 性能CI：性能测试也可以集成到Pipeline中吗？

2021-07-03 尉刚强

《性能优化高手课》

课程介绍 >



讲述：尉刚强

时长 14:35 大小 13.36M



你好，我是尉刚强。这节课，我们来讲讲如何利用 Pipeline 来实现更好的性能测试效果。

如果你用过开源数据库 MongoDB，那你可能会遇到或者是听说过一个比较典型的性能问题，也就是 **N+1 性能问题**。

这个问题描述是：本来业务实现中需要查询 N 条数据项，因此最佳的性能实现方式，当然是通过 1 条查询语句返回所有数据。但是，如果编码人员对 MongoDB 客户端的 API 接口不太熟悉，或者是编码过程中不小心，都有可能导致最后实现的查询代码，执行了 N+1 次数据库查询请求，从而造成性能浪费。而如果 N 的数字比较大，可能还会对软件性能造成更严重的影响。

那么针对这类性能问题，有没有什么好的解决办法呢？

当然是有的，我们可以**把组件或者微服务级的性能测试集成到 Pipeline（流水线）上**，让它成为 CI（持续集成）中的一部分，就可以很好地解决这类问题。

而至于具体的原因，今天这节课我就会先带你一起探究下。然后，我还会针对不同种类的性能测试，给你分享一些实用的集成到 Pipeline 中的策略和思路。你可以根据今天学习的内容，将自己产品中的一些关键性能测试也集成到 Pipeline 上，来帮助团队更早地发现性能问题，从而提升研发效率。

好，下面我们就先来了解下 Pipeline 的工作原理，看看为什么可以把性能测试集成到 Pipeline 上。

为什么性能测试可以集成到 Pipeline 上？

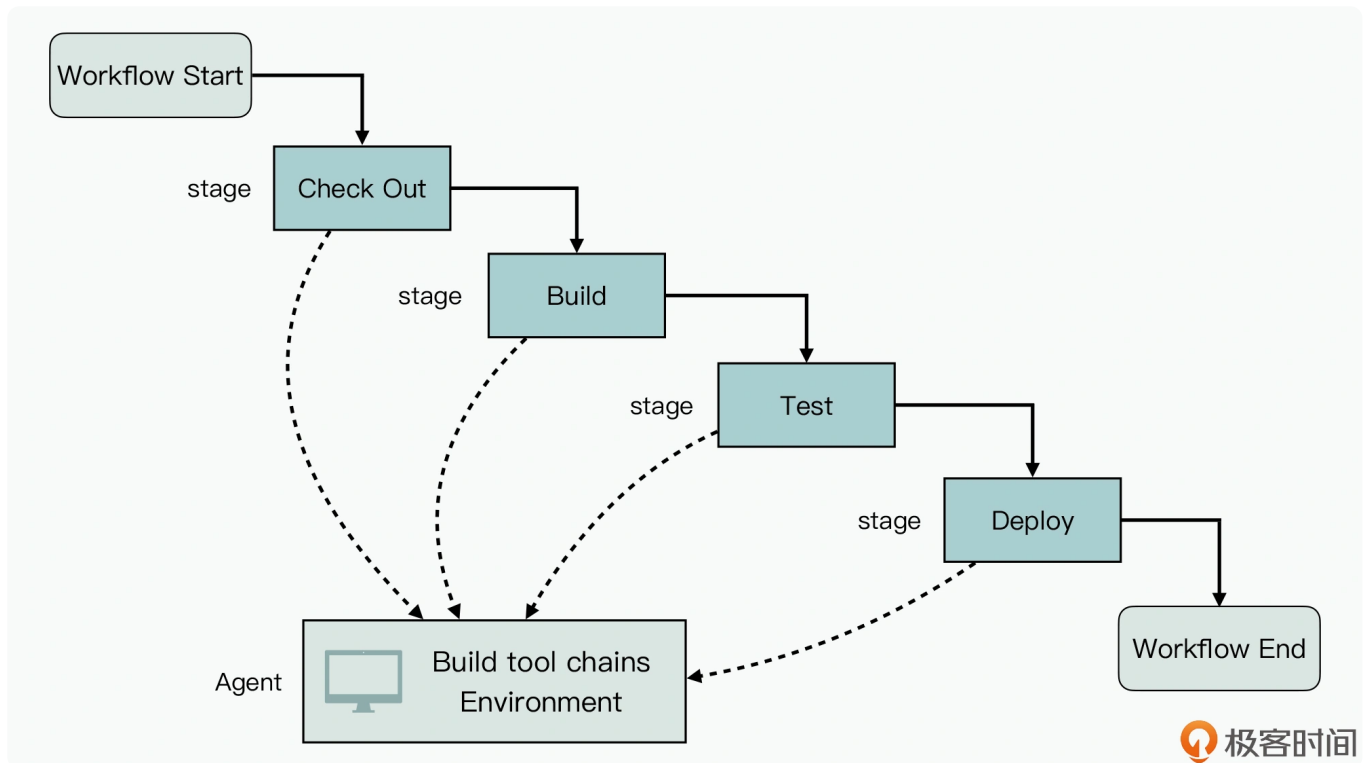
首先，我这里所说的 Pipeline，其实指的是 DevOps 软件开发方法中提出 CI/CD 的一种技术实现手段。

补充：DevOps 是一种软件开发方法，它将持续开发、持续测试、持续集成、持续部署和持续监控贯穿于软件开发的整个生命周期，基于这种方法可以帮助提升团队的开发效率，加快产品的交付。

在一个 Pipeline 中，你定义多个步骤或阶段的执行操作（编译、构建、测试、部署等）时，都可以在代码提交后自动触发执行。而如果中间某个阶段出现了错误，就会导致整个 Pipeline 失败，然后你就可以从中找到引入问题的代码合入节点。

其实，在不同的平台或工具上，支持 DevOps 的 CI/CD 的流水线技术都是不一样的。比如说，在 Jenkins 中主要是基于 Pipeline 的插件来完成的；而在 GitHub 中，你可以通过基于 action 的 Workflow 来实现。

但是，它们解决问题的原理与思路是比较相似的。所以下面，我们就通过 Jenkins 中的 Pipeline 插件，来了解下 Pipeline 的工作原理。



如上图所示，在 Jenkins 中，每个 Pipeline 其实是定义了一个工作流，中间由多个 Stage（阶段）组成，它们分别完成不同的流水线功能，比如编译、测试等。但是在每个 Stage 执行时，都需要一个运行时环境，这个在 Jenkins 中是使用 Agent 来表示的。

一般情况下，Pipeline 会基于代码仓提交来触发执行，但是软件的编译、构建等相关功能使用的工具链，通常是比较稳定的，我们不需要在代码仓中管理。所以更有效的实现方式，就是将这些工具链都安装到 Pipeline 中的执行 Agent 内，来实现复用。

事实上，不同的软件产品构建 Agent 运行环境的差异非常大。比如说，在很多嵌入式内核与驱动的软件构建流水线中，很可能在 Agent 中安装 Linux 源码或框架，还有一堆交叉编译工具链。所以你其实可以认为，**Agent 就是目前封装 Pipeline 的基础设施的重要手段。**

那么，对于性能测试工具来说，你就可以把它安装到 Agent 这个 Pipeline 的基础设施中。而再进一步，既然性能测试工具可以集成到 Pipeline 的基础设施上，自然性能测试也就可以集成到 Pipeline 中了。

好，在明确了 Pipeline 的原理之后，接下来我们还需要解决的问题就是：如何才能将性能测试集成到 Pipeline 当中呢？

性能测试集成 Pipeline 的策略思路

在第 17 和 18 讲中，我把性能基准测试划分成了两类，分别是微基准测试和宏基准测试。所以实际上，这两类性能基准测试集成到 Pipeline 中，所碰到的问题与解决思路是有些差异的，下面我们就来分别讨论下。

微基准性能测试

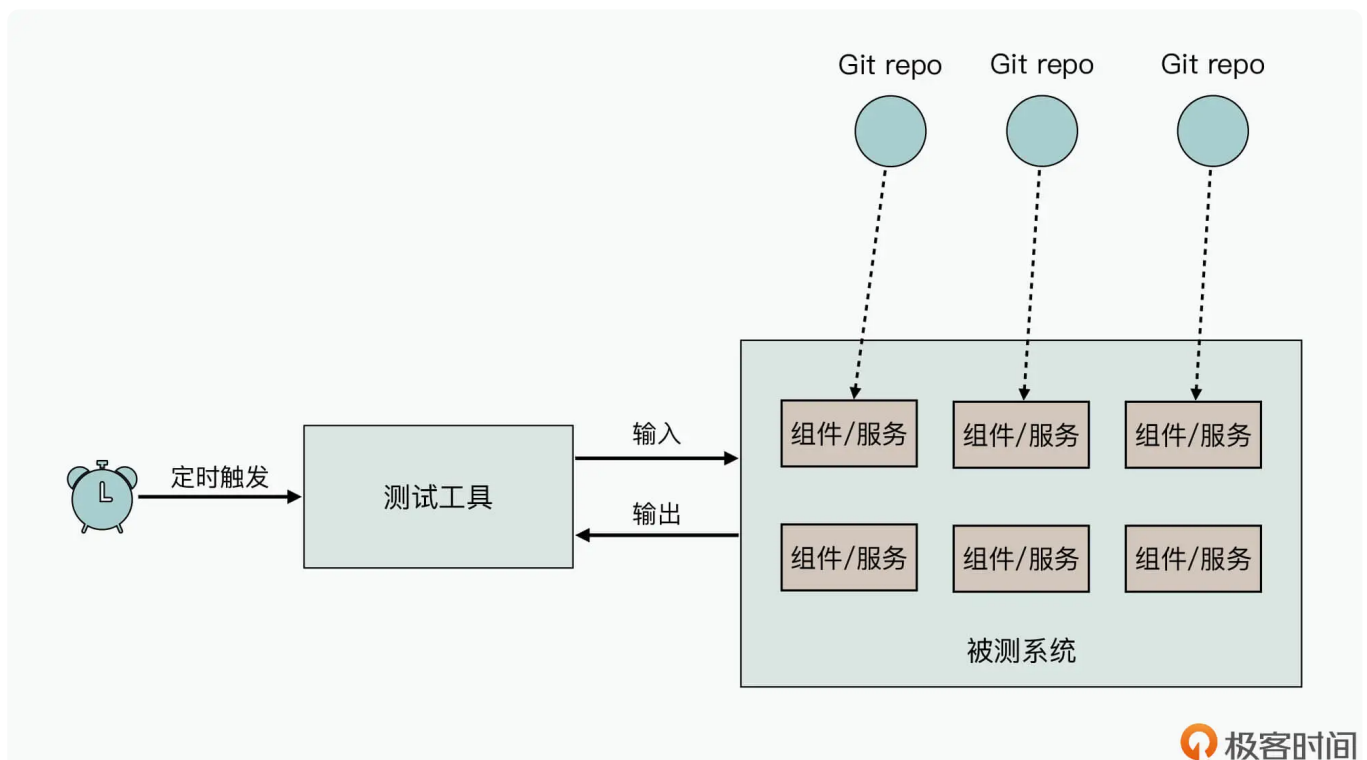
首先，对于微基准测试而言，其实它的测试运行过程和普通的单元测试比较相似，所以它可以比较方便地集成到 Pipeline 中。但这里你需要注意的是，微基准测试与单元测试存在一个比较大的差异，就是**微基准测试对执行时间比较敏感**。

因为执行时长会直接影响测试结果的准确性，而一般的单元测试并不会，所以针对微基准性能测试集成到 Pipeline 时，你需要确保它在执行期间使用的资源配置是确定的，才能保证最后的执行结果是有意义的。

宏基准性能测试

其次，对于宏基准测试来说，这里我们可以将其进一步划分为**全系统端到端的性能测试**和**组件 / 服务级的性能测试**。

我们先来看下，全系统级的性能测试的原理模型是什么样的：



那么具体我们应该怎么做呢？其实，针对这种类型的性能测试，我们可以**定义独立的流水线**。因为它和代码仓是独立的，所以你可以使用**定时器触发机制**来触发性能测试流水线，并生成性能测试报告。

对于云服务化的性能测试工具来说，其实很多都提供了定时触发机制，比如阿里云的 PTS 等。但假如你使用的是单机版的性能测试工具，那你就需要基于流水线来实现相似的能力。

在上节课，我也给你介绍了将系统级的性能测试，拆分成组件 / 服务级的性能测试的各种好处，但我之前讲得并不全面。其实，这种拆分方式还有一个明显的好处，就是你可以更容易地把拆分后的性能测试添加 Pipeline 中。

那么，如果我们将组件 / 服务级基线的性能测试集成到 Pipeline 后，它对应的工作流程图就如下图所示：



我们对比之前的 Pipeline 工作原理模型图，可以发现将组件 / 服务级性能测试集成到 Pipeline 后，它可以在原来的 Deploy 阶段执行，从而实现将组件 / 服务实例部署到被测运行环境中的目的。当然，这个测试运行环境的系统资源与规格应该是确定的，比如说 CPU 核数、内存等。

然后，我们就可以在 Deploy 阶段后，添加一个**自动化性能测试阶段**（Performance Test），来执行对被测系统的自动化性能测试，并生成性能测试报告。

另外从上图中，你还可以发现，在自动化性能测试阶段运行的 Agent，是一个包含了被测工具的运行时环境，那么这个 Agent 应该如何添加呢？

其实，这一步并不难。现在主流的 Pipeline 工具基本都支持容器化的运行环境，所以你其实只需要**将被测相关工具，通过 Dockerfile 方式构建成镜像，并上传到特定的容器镜像仓**之后（比如 docker.io 上），你就可以在 Pipeline 中选择这个镜像作为 Agent 了。

好了，现在我们知道，将组件 / 服务的性能测试集成到 Pipeline 中，可以帮助我们第一时间发现代码合入引入的性能恶化问题。

但对于很多软件系统来说，还有一点也很重要，就是我们应该**将性能基线数据持续地导出并可视化，来帮助分析软件性能的变化趋势**。因为在大部分场景下，性能的优化其实是一个逐步累积的过程，并不是一次性的。

总而言之，在做性能测试的过程中，你要有尽量将性能测试集成到流水线中的意识。但是在实践的过程中，你还需要注意的就是，当性能测试集成到 Pipeline 之后，其实还会带来一些新的问题与挑战。

所以接下来，我们就一起来看看要如何应对这些挑战。

性能测试集成 Pipeline 的问题和挑战

我们知道，原来的性能测试通常是由手动触发执行的，所以执行频率会比较低，即使出错了我们也可以进行人为的分析。但是，如果把它集成到了 Pipeline 中，首先就意味着它会频繁执行，而且还需要比较高的稳定性。因此，这就对性能测试工作提出了更高的要求。

那么这里，我就给你总结了把性能测试集成到 Pipeline 之后，可能会引入的核心变化和挑战，然后我也会给你分享更好地应对这些挑战的方法和手段，让你能在具体的业务开发过程中用得更好，少踩坑。

更低的执行成本

首先，性能测试的执行花销是不能忽视的，比如说，有些被测组件 / 服务，在业务执行期间会调用第三方服务 API，而很多都是按照请求调用次数来进行收费的（比如短信验证、云对象文件存储下载等）。另外在性能测试期间，被测组件 / 服务会频繁调用接口，如果再进一步将性能测试集成到 Pipeline 中，就势必会带来比较昂贵的成本。

那么针对这类问题，其实你可以将这些**与收费相关的依赖都打桩实现，然后在性能测试时使用桩接口即可。**

更高的稳定性要求

其次，很多系统实现性能测试的稳定性并不高，比如说，有些被测组件 / 服务，对真实的物理环境依赖比较强，尤其在嵌入式场景下非常普遍。所以针对这种情况，你就需要考虑**将性能系统中，软 / 硬件稳定性比较差的依赖项隔离出去，也可以采用打桩的方式。**

更准确的性能结果

最后，针对更准确的性能结果，这个其实是对性能测试提出了更高的要求。因为前面我也介绍过，很多性能测试结果值都是波动的，所以这样就导致我们根据测试结果，来判断性能是否劣化会比较困难。这里呢，我给你提供两个解决思路，你可以参考使用。

第一，通过优化软件设计与实现来减少性能指标抖动。比如，我在 [🔗第 10 讲](#)中提到的剥离非核心业务、引入延迟计算服务等，来解决业务处理时延的长尾效应；

第二，你可以选择一些性能抖动比较小的部分业务模块的性能指标，来替换整体业务流程性能指标，以此进行判断分析。

事实上，由于性能测试集成到 Pipeline 时会存在很多的挑战，所以大部分团队并不愿意投入很多时间和精力和成本。但是你会发现，在性能测试集成到 Pipeline 过程中所暴露的问

题，其实是帮你指出了性能测试的优化改进方向。如果你沿着这个方向去优化和设计性能测试，就可以让性能测试工作朝着更高效的方向演进。

小结

这节课，我带你了解了将性能测试集成到 Pipeline 上的优势和方法策略。其中，你要重点把握的地方，就是要根据不同的性能基准测试类型来选择具体的集成策略。另外，在将性能测试集成到 Pipeline 的实践过程中，你还需要注意去规避解决一些典型问题和挑战，比如成本的问题、稳定性的问题等。

实际上，现在很多的研发团队所做的性能基线测试，与代码提交都是脱节的，而这样就不能在第一时间发现代码提交引入的性能劣化问题。所以间接也就会导致，软件产品的性能问题解决不及时，投入的研发成本也会很大。

因此，在学完今天的课程之后，你就可以借鉴我介绍的把性能测试集成到 Pipeline 的思路方法，来改进和优化性能测试工作，然后将其集成到 Pipeline 中，助力团队更早地发现和解决软件产品中的性能问题。而且你想一想，如果每次代码提交，都可以看到软件的性能变化是不是还挺酷的？

思考题

如果性能测试工具的负载量非常大，需要部署为集群模式，那么是否也可以集成到 Pipeline 中呢？

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 1  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 20 | 性能看护：如何更好地守护产品性能？

下一篇 22 | 性能调优方法论：如何科学高效地定位性能问题？

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。