

讲堂 > 持续交付36讲 > 文章详情

## 29 | 计算资源也是交付的内容

2018-09-08 王潇俊



### 29 | 计算资源也是交付的内容

朗读人：王潇俊 11'07" | 5.10M

你好，我是王潇俊。今天我和你分享的主题是：计算资源也是交付的内容。

在传统的持续交付中，我们虽然一直深受环境和计算资源的困扰，但却很少去考虑如何去彻底解决这个问题。归根结底，原因有两方面：

1. 这个问题解决起来难度较大；
2. 这个问题也不太算是持续交付的范畴。

但是，随着 DevOps 深入人心，以及云计算的发展，我们已经有了解决这些问题的思路和方法。所以今天我要和你聊聊如何解决这方面的问题，从而获得更优的环境管理能力。

计算资源包括什么内容？

通常情况下，我们所说的计算资源包括：CPU、内存、本地磁盘、外部存储、网络等。

为了提高计算资源的利用率，传统做法往往是按需申请和分配资源。但是计划往往赶不上变化，整个申请和分配过程冗长，缺乏快速弹性的能力，最终影响了持续交付的效率。

## 计算资源是导致持续交付的反模式的原因

从实际情况来看，计算资源是导致持续交付反模式的主要原因。

在《持续交付：发布可靠软件的系统方法》一书中，作者给我们列举了几个反模式，比如：

1. 手工部署软件，即由详尽的文档描述一个部署过程，部署需要手工操作及验证；
2. 开发完之后才向类生产环境部署，即开发完成后才第一次向类生产环境部署，如果是全新的应用，第一次部署往往会失败；
3. 生产环境需要手工配置管理，即有专门的团队负责生产环境的配置变更，修改配置时，需要这个专门的团队手工登录到服务器进行操作。

你可以按照我在发布及监控这个系列分享的内容，通过合理打造一套发布系统，解决“手工部署软件”这个反模式的问题。

但是，“开发完之后才向类生产环境部署”和“生产环境需要手工配置管理”这两个反模式，我们总是难以克服。究其原因，我们是在计算资源的交付上出了问题。为什么呢？

- 产生“开发完之后才向类生产环境部署”反模式的原因是，开发测试环境和生产环境的差异。导致这个差异产生的原因，绝大多数是因为，生产环境的模拟成本太大，线下难以模拟，归根结底是计算资源的问题。
- 产生“生产环境需要手工配置管理”反模式的原因是，环境需要长期使用，而且需要不断更新，不能即取即用。说到底，这还是计算资源问题。

这时，你可能会说，计算资源的问题也可以通过一些方法解决。比如，准备一个计算资源缓冲池，从这个缓冲池中获取计算资源就很方便、快捷了。

这确实是一个解决方案，但是这个解决方案带来的成本浪费是巨大的。

然后，云计算出现了，正好可以彻底解决这些问题。现在，我就和你聊聊云计算是如何解决这些问题的。

## 云计算带来的划时代变革

其实，云计算带来的变革，主要体现在以下三个方面。

1. 云计算的弹性能力，使得计算资源的提取成为持续交付过程的一个自然步骤。  
使用云计算之前，获取计算资源的过程往往耗时很长、结果不可预知，因此往往不能作为持续交付流水线的一部分，而是要独立出来，或者采用异步方式进行处理。

然而，云计算的弹性能力，使得我们获取计算资源的速度和数量有了质的飞跃，而且保证了结果的可控性。所以，计算资源的提取，也就可以与其他环节一样作为持续交付中的一个自然步骤了。

## 2. 云计算的 Immutable，可以保证计算资源的生命周期与持续交付过程一致。

以前，为了降低计算资源回收交替的成本，我们往往会采用复用计算资源的方式。但是，如果遇到不同的应用，而这些应用又需要不同的配置，或安装一些不同的软件的话，我们就需要对这些被复用的机器进行人工维护。

所以，这个复用方式，就使得一个计算资源的生命周期变得不再清晰，我们也再次掉进了“生产环境需要手工配置管理”的陷阱。

但是，在云计算中，计算资源的生命周期可以被严格定义。所以，计算资源就可以做到专事专用，进而保证与持续交付过程的一致性。

## 3. 云计算本身不区分环境，这样可以获取到与生产环境几乎一样配置的资源。

以前，测试所用的计算资源和生产环境所用的计算资源，往往不一致。最核心的原因是，我们没有制订计算资源的交付标准，或者制订的标准在资源交付后，由于脱离了控制而被轻易打破。

而云计算则会通过系统对计算资源的抽象、标准化，以及管控，可以很容易地获取到与生产环境更相近的测试环境。

目前，各公有云都提供了非常完备的持续交付平台，越来越多的企业正在或正将把生产环境搬迁到公有云上。如果你的应用符合云原生的特性，那恭喜你，你可以尽情地享受云计算带来的红利了。

但以目前的情况来看，想要做到完全依托于公有云实现持续交付，你还需要经过大量云原生的改造。比如：需要所有的应用都无状态；应用启动过程没有特殊的额外步骤；使用公有云提供的路由方案等等。所以，绝大多数组织仍旧选择以依赖私有云或私有 IaaS 平台的形式来解决问题。

## 重塑持续交付平台的相关部分

有了云计算，或者说私有 IaaS 平台这个强大的底层支持，我们下一步要解决的就是充分发挥它的能力。所以，现在我就和你分享一下，持续交付平台的哪些部分可以利用云计算的强大能力。

首先，弹性的集成编译环境。

不同技术栈的应用需要不同的编译环境，而且要保证编译环境和运行时环境一致，否则会发生意料之外的问题。这样一来，如果组织内部同时有多个技术栈存在，或应用对环境有多种要求时，就需要有多个独立的编译环境了。

因此，如果没有云计算的话，持续交付通常要准备一个由多台不同服务器组成的编译集群，用以覆盖所有的编译需求。另外，为了达到持续交付的效果，我们可能还需要再横向地为每个独立编

译环境多准备几个实例，以便达到多个编译任务并行的目的，而不是要一直排队。

然而，这样的做法对资源的控制和利用非常不利，很有可能在编译高峰时仍旧资源不够用，而扩大资源池后，大多资源又处于空闲状态。

现在，利用云计算的弹性，你可以按需生成一个个特定的编译环境，及其所需的计算资源。这就使得你无需再提前准备一个巨大的资源池，也自然不必为如何合理配比资源池中的资源而烦恼了。

要达到这种效果，你只需要修改集成编译模块的编译调度逻辑就可以：

- 在编译调度前，生成所需要的编译环境实例；
- 在编译完成之后，保存编译日志和相关交付产物后，销毁编译实例、释放计算资源。

因此，有了云计算的支持，编译模块的效率和资源利用率都可以上一个台阶。

其次，重新定义环境管理。

云计算的弹性能力，可以帮助我们改善环境管理模块的能力，使得环境管理的方式更加灵活。

比如，原先一个测试子环境的生命周期往往与某个功能研发或是项目研发不一致，会提前准备，或是多次复用；又或者由于资源紧缺的原因，测试环境只能模拟部分实际环境；另外还会有一些环境被作为公共的资源一直保留，从不释放。这些问题都增加了环境管理的复杂度。

现在，有了云计算平台的强大能力，我们完全可以打破这些限制，将环境的生命周期设计得与项目生命周期一致，每个项目或每个功能都可以拥有自己独立的测试环境；另外，你还可以动态定义所需的任何环境，或者利用模板技术，快速复制一个已存在的环境。总之，环境管理变得越来越灵活了。

除了计算资源之外，云计算也同时提供了非常强大的网络定义能力，为环境管理插上了翅膀。

我们可以通过 VPC（专有网络），对任何环境定义网段划分、路由策略和安全策略等。这样环境与环境之间就拥有了快速处理网络隔离和相通的能力。借此，我们也可以很容易地创造沙箱环境、专用测试环境等。

有了云计算的支持，环境管理真的可以飞起来。

最后，充分利用存储。

云计算除了可以提供计算资源和网络资源的便利外，同时也可以解决资源存储的问题。分布式存储的能力，同样能给持续交付提供有利的帮助。

比如，利用共享存储，你可以在多个编译实例之间共享 Workspace，虽然性能会稍微下降一点，但可以很方便地保证不同实例之间使用相同的本地缓存，而无需再去考虑如何同步本地缓存，使其一致的问题。

又比如，利用分布式存储，你就无需再担心部署包的备份问题了。

再比如，如果你还使用公有云的存储服务的话，比如 Amazon 的 S3，云计算可以帮你很方便地把交付产物同步到全球各地，简化你异地部署的工作。

## 总结

今天我和你讨论的主题是，云计算这样的新兴技术会对持续交付会产生哪些影响。我们可以把这些影响概括为四大方面：

1. 利用云计算，能够很容易地打破持续交付反模式；
2. 利用云计算，可以使持续交付的编译模块变得更为灵活，提升利用率；
3. 利用云计算，可以自由地发挥想象力，简化环境管理的工作；
4. 利用云计算，可以使用存储服务，使持续交付工作更便捷。

总之，你会发现，计算资源是持续交付非常重要的依赖，有了云计算的支持之后，我们完全可以把计算资源的交付也纳入到持续交付过程中，更好地做到端到端的交付。

## 思考题

你将如何利用云计算的能力，优化现有的持续交付体系呢？

感谢你的收听，欢迎给我留言。



版权归极客邦科技所有，未经许可不得转载

通过留言可与作者互动