

# 什么是DVCS?

## 关于版本控制系统 (Version Control Systems) 的基本介绍

要理解什么是DVCS (分布式版本控制系统), 首先理解版本控制系统 (VCS)很重要。Subversion (SVN)可能是您遇到过的一个VCS例子。

假设您有一个5个开发人员组成的团队在开发一项很棒的产品。一天, 您的两个开发人员同时完成同一批文件。通过SVN, 他们处理的文件放在一个 "central repository" (中央库) 中。如果要继续处理这些文件, 那么他们必须一起取出这些文件、编辑它们, 然后将它放回到这个中央库中。

这样, Bob将这个文件的一份副本下载到他的计算机中, 然后开始修改, 同时Alice也在做同样的工作。当他们一起完成时, 他们'commit' (提交) 他们的变更并将它们放回到中央库中。但是通过SVN, 这个步骤引起了一个大问题。这个中央库将存储它们的新版本, 但是它无法决定首先发布那个变更。而且, 如果Bob和Alice一起修改了相同的代码行, 这个中央库将无法决定保留哪个变更。在SVN中, 一个日志将被更新, 从而保留这个文件以前所有的历史版本, 但是如果两个人同时提交, 那么这个日志将无法解决首先记录哪次编辑的问题。

这样, 哪个版本将成为最新版本? 以及其他人的变更又会怎样?在最初的VCS中, 为了解决发生在这些情况下的所有合并问题, 需要由用户人工修改。因此为了改进这个过程, 就创建了DVCS。

## 解决方案: 分布式版本控制系统

分布式VCS (DVCS)是VCS概念的一种新改进版本。怎样的? DVCS给每个人一份这个库的完整副本, 放在他们自己本地的计算机上 (包括历史日志文件), 这使用户无需网络连接就能更快地访问历史或提交变更。DVCS也解决了VCS的内部问题: 中央库的合并冲突。

在上述相同场景下, 我们有两个用户同时提交到中央库, VCS将无法阻止任何冲突。相反, DVCS力图通过追踪用户对文件的修改历史并使用历史巧妙地将变更合并回父文件来迅速解决合并冲突。

除了中央库, DVCS也使用户可以从其它用户的本地服务器访问文件, 增加协作机会。

Subversion是VCS的一个范例工具, 而Git和Mercurial是受欢迎的DVCS工具。

## 因此, 简而言之...

- **整个库的本地盘副本=速度&多功能:** 在DVCS中, 每个开发人员将整个库的一个副本提取到自己的本地计算机上。这个本地副本使开发人员可以离线提交。他们不必依赖一个服务器连接, 因此这个过程超快。而且, 无论有或没有网络连接, 开发人员可以从任何地方处理他们的代码。
- **无忧提交:** 由于变更被提交到本地库, 开发人员可以用它快速分支代码和实验, 不必担心将他们未定义的代码发布给其他的开发人员。这项功能鼓励您的开发人员进行试验和创新。此外, 开发人员可以为他们的工作提供“核对基准点”并将不完整的代码变更提交到他们的本地库, 而不必担心干扰其他的开发人员。

- **避免合并冲突:** Subversion最关键的问题之一是它如何处理同时提交。如果您使用Subversion，那么您肯定遇到过这个问题。这些问题会随着开发人员 and 提交数量的增加而增加。VCS避免了这些问题，因为开发人员提交到他们自己的库中，当他们准备好，他们可以将他们的库推到中央库。然后，通过一个集成决定是否拉入这些变更。

DVCS可以帮助企业降低协同工作的摩擦。在任何给定的时间内，任何企业自然有许多开发人员在一个文件集上工作，而且Stash可以帮助您的团队实现平滑和安全的协作开发。