链滴

版本控制工具小结

作者：tianxin

原文链接：https://ld246.com/article/1466966990945

来源网站：链滴

Git 基础介绍

[TOC]

概要

根据黑格尔那句：存在即合理（尽管这是一个误解），一门工具的诞生一定有它的存在性，如果一门被大家都认同并且广泛使用的工具，那么一定存在着它的合理性。优秀的工具一定是简化了人们精力的付出（会有更多的精力去干 + + 什么）那么Git的诞生到底是处于什么目的呢？方便了谁？又什么作用？

版本控制

那是同一份功能，多份莫名其妙的代码在硬盘里乱飞的时代，那是同一份功能多人协作开发后，后还是由自己一个人默默重撸一遍的时代。

尽管在很早之前已经存在 diff 与 patch两个工具分别比较代码的差异和解决代码间的冲突。

这真是一个头痛的问题。

然而，有一天 Dick Grune 教授在和两个学生共同开发一个项目的时候，终于，终于受不了文件冲突，缺失，意外失踪（肯定是这两个熊孩子惹的祸），设计出第一个被大规模使用的版本控制工具 —— RVS （Concurrent Versions System）

最早的版本控制 SCCS

然而最早的版本控制并不是 CVS，而是为了更好的“产出”Unix时的 SCCS Source Code Control System。它实现的核心思路是：每个源文件都对应着一份 SCCS 格式的文件，SCCS file format 存了对应源文件的版本变化内容，而存储方法的实现核心是 Interleaved deltas (然而 并不知道这个词怎么翻译)。思想是：如果

在文件的版本一中

```
foo
bar
```

在文件的版本二中

```
bar
baz
```

则对应的SCCS格式则是

```
^AI 1
^AD 2
foo
^AE 2
bar
^AI 2
baz
```

原文链接：版本控制工具小结

^AE 2
^AE 1

Interleaved deltas 是以变动的单位是行（lines），**即如果该行中有且只有1个字符改动了，么控制版本记录的是先删掉整行内容，再将改动后的新行添加到原位置中。**其中 ^A 表示控制指令。I 表示插入，D表示删除，每个I （插入指令）和D（删除指令）都有应的结束指令，即 E （End）。面的数字代表版本号。

- 在版本一中增加 foo，并在版本二中删除 foo。
- 在版本一中增加 bar
- 在版本二中增加 baz

从 SCCS 到 RVS

RVS 相对于 SCCS 而言，更加方便程序员开发。可以自动存储，检索，日志记录，识别和修订的并版本控制的。（PS：没有使用过我的 并不知道 方便到哪里？ 哪里不会点哪里。 So easy ~ ）

但是此时的版本控制工具依然停留在只是对单个文件的控制。如果是整个项目工程，则控制起来分麻烦。而且版本控制都是在本机上。如果回家时还撸码怎么办。（有激情 ~ + -）。

版本控制工具的历史变革：集中式之 CVS

为了解决这些问题，版本控制工具终于迎来了历史性的变革 —— CVS工具诞生。CVS不仅支持对整个项目的管理（提出了仓库的概念），而且基于 客户端 - 服务端 可以使得多用户在不同地点，同时间的开发。（麻麻再也不用担心我的学习 - -）

当我在服务器中初始化好代码仓库后，程序员们在自己的电脑上导入该库。这也是集中式管理的鼻祖将所有代码统一放到服务器中管理）。然而CVS也存在着一系列的缺点，所有后面会有更有优势的 SV的诞生。不过不论是SVN还是CVS，二者都是属于集中式版本控制 （即代码的版本管理是放在服务器的，而自己本地只是作为一个开发环境，和版本管理没有任何关系）

现代主流的版本控制工具：分布式之Git

而集中式最大的弊端就是：如果想同步代码，必须将本机和服务器相连接。如果在没有网的情况，则集中式没有任何卵用。而且 ，如果作为服务端的电脑故障掉了，那么之前以往的代码版本信息将部消失（一死机成千古恨啊 - -）。

而分布式的设计思想是：每个人的电脑都是代码仓库，拥着着代码版本管理功能。这样即使没有络的情况下依旧可以进行代码版本的控制。我们会统一约定一个电脑（主机）作为这次项目开发的主库，并且将本机代码仓库与约定好的代码仓库（github）进行同步推送和更新。这样即使 github 主上的代码仓库出现意外，自己本机上仍然有着属于自己的代码仓库。

当然，分布式版本控制 与 集中式版本控制的对比与优缺点不可能仅仅只有这几项。希望能够和家多多交流，多多讨论。

总结

版本控制工具经历了 SCCS → RVS → 以CVS为开端的集中式 → 以Git为主流的分布式

工具的迭代和流行一定存在着合理的原因，希望自己能够不只是单单掌握工具的使用，而且可以

握住整个工具变迁的过程，体会开发者们设计的思想。 ^ ^

本篇文章大多是参考官方文档后加上自己的理解总结出来的内容。 如果有哪个点错误，欢迎指正 留讨论噢。

参考资料

SCCS 论文

Wiki 差分编码 Delta encoding

Wiki : Interleaved deltas

Wiki : SCCS

Oracle : SCCS

Wiki ：RVS

blog ：Git 诞生记