

对于易于本地增加分支,图中Git本地和服务器端结构都很灵活,所有版本都存储在一个目录中,你只需要进行分支的切换即可达到在某个分支工作的效果。而 SVN则完全不同,如果你需要在本地试验一些自己的代码,只能本地维护多个不同的拷贝,每个拷贝对应一个SVN服务器地址。 以下是一个例子:

以前我所在的小组使用SVN作为版本控制工具,当我正在试图增强一个模块,工作做到一半,由于会改变原模块的行为导致代码服务器上许多测试的失败,所以并没有提交代码。这时候上级对我说,现在有一个很紧急的Bug需要处理,必须在两个小时内完成。我只好将本地的所有修改diff,并输出成为一个patch文件,然后回滚有关当前任务的所有代码,再开始修改Bug的任务,等到修改好后,在将patch应用回来。前前后后要完成多个繁琐的步骤,这还不计中间代码发生冲突所要进行的工作量。可是如果使用Git,我们只需要开一个分支或者转回到主分支上,就可以随时开始Bug修改的任务,完成之后,只要切换到原来的分支就可以优雅的继续以前的任务。只要你愿意,每一个新的任务都可以开一个分支,完成后,再将它合并到主分支上,轻松而优雅。

分布式对于Git而言,你可以本地提交代码,所以在上面的图中,Git有利于将一个大任务分解,进行本地的多次提交,而SVN只能在本地进行大量的一次性更改,导致将来合并到主干上造成巨大的风险。Git的代码日志是在本地的,可以随时查看。SVN的日志在服务器上的,每次查看日志需要先从服务器上下载下来。

II

GIT不仅仅是个版本控制系统,它也是个内容管理系统(CMS),工作管理系统等。如果你是一个具有使用SVN背景的人,你需要做一定的思想转换,来适应GIT提供的一些概念和特征。所以,这篇文章的主要目的就是通过介绍GIT能做什么、它和SVN在深层次上究竟有什么不同来帮助你认识它。那好,这就开始吧...

1. GIT是分布式的, SVN不是:

这是GIT和其它非分布式的版本控制系统,例如SVN,CVS等,最核心的区别。如果你能理解这个概念,那么你就已经上手一半了。需要做一点声明,GIT并不是目前第一个或唯一的分布式版本控制系统。还有一些系统,例如Bitkeeper,Mercurial等,也是运行在分布式模式上的。但GIT在这方面做的更好,而且有更多强大的功能特征。GIT跟SVN一样有自己的集中式版本库或服务器。但,GIT更倾向于被使用于分布式模式,也就是每个开发人员从中心版本库/服务器上chect out代码后会在自己的机器上克隆一个自己的版本库。可以这样说,如果你被困在一个不能连接网络的地方时,就像在飞机上,地下室,电梯里等,你仍然能够提交文件,查看历史版本记录,创建项目分支,等。对一些人来说,这好像没多大用处,但当你突然遇到没有网络的环境时,这个将解决你的大麻烦。同样,这种分布式的操作模式对于开源软件社区的开发来说也是个巨大的恩赐,你不必再像以前那样做出补丁包,通过email方式发送出去,你只需要创建一个分支,向项目团队发送一个推请求。这能让你的代码保持最新,而且不会在传输过程中丢失。GitHub.com就是一个这样的优秀案例。有些谣言传出来说subversion将来的版本也会基于分布式模式。但至少目前还看不出来。

2. GIT把内容按元数据方式存储,而SVN是按文件:

所有的资源控制系统都是把文件的元信息隐藏在一个类似.svn,.cvs等的文件夹里。如果你把.git目录的体积大小跟.svn比较,你会发现它们差距很大。因为,.git目录是处于你的机器上的一个克隆版的版本库,它拥有中心版本库上所有的东西,例如标签,分支,版本记录等。

3. GIT分支和SVN的分支不同:



分支在SVN中一点不特别,就是版本库中的另外的一个目录。如果你想知道是否合并了一个分支,你需要手工运行像这样 的命令svn propget svn:mergeinfo,来确认代码是否被合并。感谢Ben同学指出这个特征。所以,经常会发生有些分支被 遗漏的情况。 然而,处理GIT的分支却是相当的简单和有趣。你可以从同一个工作目录下快速的在几个分支间切换。你很 容易发现未被合并的分支,你能简单而快捷的合并这些文件。

4. GIT没有一个全局的版本号,而SVN有:

目前为止这是跟SVN相比GIT缺少的最大的一个特征。你也知道,SVN的版本号实际是任何一个相应时间的源代码快照。我 认为它是从CVS进化到 SVN的最大的一个突破。因为GIT和SVN从概念上就不同,我不知道GIT里是什么特征与之对应。如 果你有任何的线索,请在评论里奉献出来与大家共享。 更新:有些读者指出,我们可以使用GIT的SHA-1来唯一的标识一 个代码快照。这个并不能完全的代替SVN里容易阅读的数字版本号。但,用途应该是相同的。

5. GIT的内容完整性要优于SVN:

GIT的内容存储使用的是SHA-1哈希算法。这能确保代码内容的完整性,确保在遇到磁盘故障和网络问题时降低对版本库的 破坏。这里有一个很好的关于GIT内容完整性的讨论 - http://stackoverflow.com/questions/964331/git-file-integrity

返回



Never despair...

该日志由 Bob.B 于2011年07月01日发表在SCM分类下, 你可以发表评论,并在保留原文地址及作者的情况 下引用到你的网站或博客。

转载请注明: SVN和Git的比较 | 扛一肩记忆 +复制链接

关键字: Git, SCM, SVN

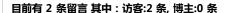
【上篇】kickstart自动全新安装ubuntu server

【下篇】18888km留恋

您可能还会对这些文章感兴趣!

- 1. How to use Subversion with Eclipse
- 2. Some Useful Tech Pictures
- 3. Code Static Check Tool CCCC
- 4. How ElectricMake guarantees reliable parallel build
- 5. How to use Subversion with Eclipse
- 6. What's new in ElectricAccelerator 5.4.0
- 7. Bash completion for ElectricAccelerator







学习!

blackold **W 2** : 2011年07月04日11:04:56 @回复

2楼

1楼

支持(0)反对(0)

回复

回复

确实都很强大!

支持(0)反对(0)

不留下点什么? 吹个牛也行的^_^





昵称*(必填,大侠留个名吧)

