Git总结

Git是一个开源的分布式版本控制系统，可以有效、高速地处理从很小到非常大的项目版本管理。Git 是 Linus Torvalds 为了帮助管理 Linux 内核开发而开发的一个开放源码的版本控制软件。

特点：分布式相比于集中式的最大区别在于开发者可以提交到本地，每个开发者通过克隆（git clone），在本地机器上拷贝一个完整的Git仓库。

Git的功能特性：

**从一般开发者的角度来看，git有以下功能：**

1、从服务器上克隆完整的Git仓库（包括代码和版本信息）到单机上。

2、在自己的机器上根据不同的开发目的，创建分支，修改代码。

3、在单机上自己创建的分支上提交代码。

4、在单机上合并分支。

5、把服务器上最新版的代码fetch下来，然后跟自己的主分支合并。

6、生成补丁（patch），把补丁发送给主开发者。

7、看主开发者的反馈，如果主开发者发现两个一般开发者之间有冲突（他们之间可以合作解决的冲突），就会要求他们先解决冲突，然后再由其中一个人提交。如果主开发者可以自己解决，或者没有冲突，就通过。

8、一般开发者之间解决冲突的方法，开发者之间可以使用pull 命令解决冲突，解决完冲突之后再向主开发者提交补丁。

**从主开发者的角度（假设主开发者不用开发代码）看，git有以下功能：**

1、查看邮件或者通过其它方式查看一般开发者的提交状态。

2、打上补丁，解决冲突（可以自己解决，也可以要求开发者之间解决以后再重新提交，如果是开源项目，还要决定哪些补丁有用，哪些不用）。

3、向公共服务器提交结果，然后通知所有开发人员。

优点：

适合分布式开发，强调个体。

公共服务器压力和数据量都不会太大。

速度快、灵活。

任意两个开发者之间可以很容易的解决冲突。

离线工作。

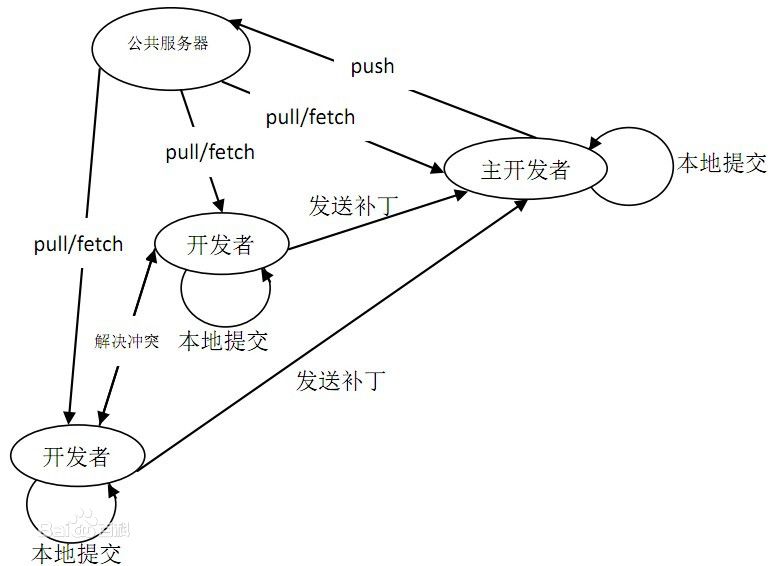
缺点：

资料少（起码中文资料很少）。

学习周期相对而言比较长。

不符合常规思维。

代码保密性差，一旦开发者把整个库克隆下来就可以完全公开所有代码和版本信息。



个人理解：

所有的概念都是基于一台本地的计算机上的：

1. 一个项目对应了1个本地仓库，1个或多个远程仓库。

2.一个仓库对应了多个分支，每个分支相当于一个独立的项目副本。每个时刻，git 只有一个当前分支，用来表示当前显示出来的项目副本。

3.添加，修改，删除过的文件必须手动的加载到某个分支中去（如当前分支），一个目录上不同的文件可能有三种状态：没有被分支收录，即将被分支收录，已经被分支收录。git 对于分支间的操作甚至仓库间的操作都是对于已经收录在分支中的文件而言的。

4.本地的多个分支可以合并，分之合并后并不会消失，分支以指针的形式值相同一份数据。

5.对于远程的 Git 项目，先使用 git fetch 命令，将远端的数据获取并存储在本地。这个分支是游离（HEAD detached）于本地分支的，因此无法 Commit。

6.通过 git merge命令可以将不同的分支合并到一处，包括暂存的远程分支。这时就获取到了远程数据 + 本地数据的综合项目。

7.通过 git push命令将项目更新到远端的某个分支。