

Git Learning

Presentation slides for developers

Press Space for next page →



What is Git?

- Git是一个免费的、开源的分布式版本控制系统，可以快速高效地处理从小型到大型的各种项目
- Git易于学习，占地面积小，性能极快。它具有廉价的本地库，方便的暂存区域和多个工作流分支等特性。其性能优于 Subversion、CVS、Perforce和 ClearCase等版本控制工具。



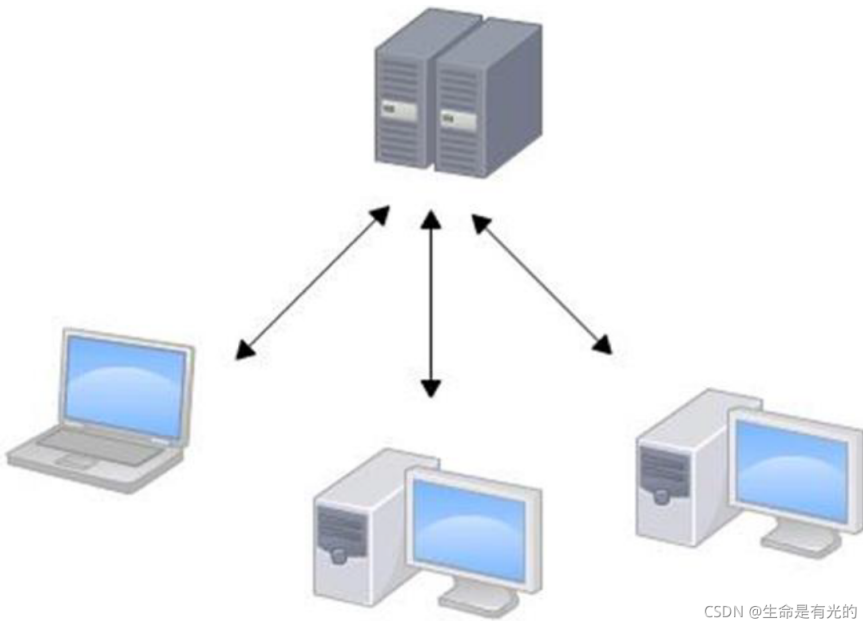
版本控制

- 版本控制是一种记录文件内容变化，以便将来查阅特定版本修订情况的系统。
- 版本控制其实最重要的是可以记录文件修改历史记录，从而让用户能够查看历史版本，方便版本切换

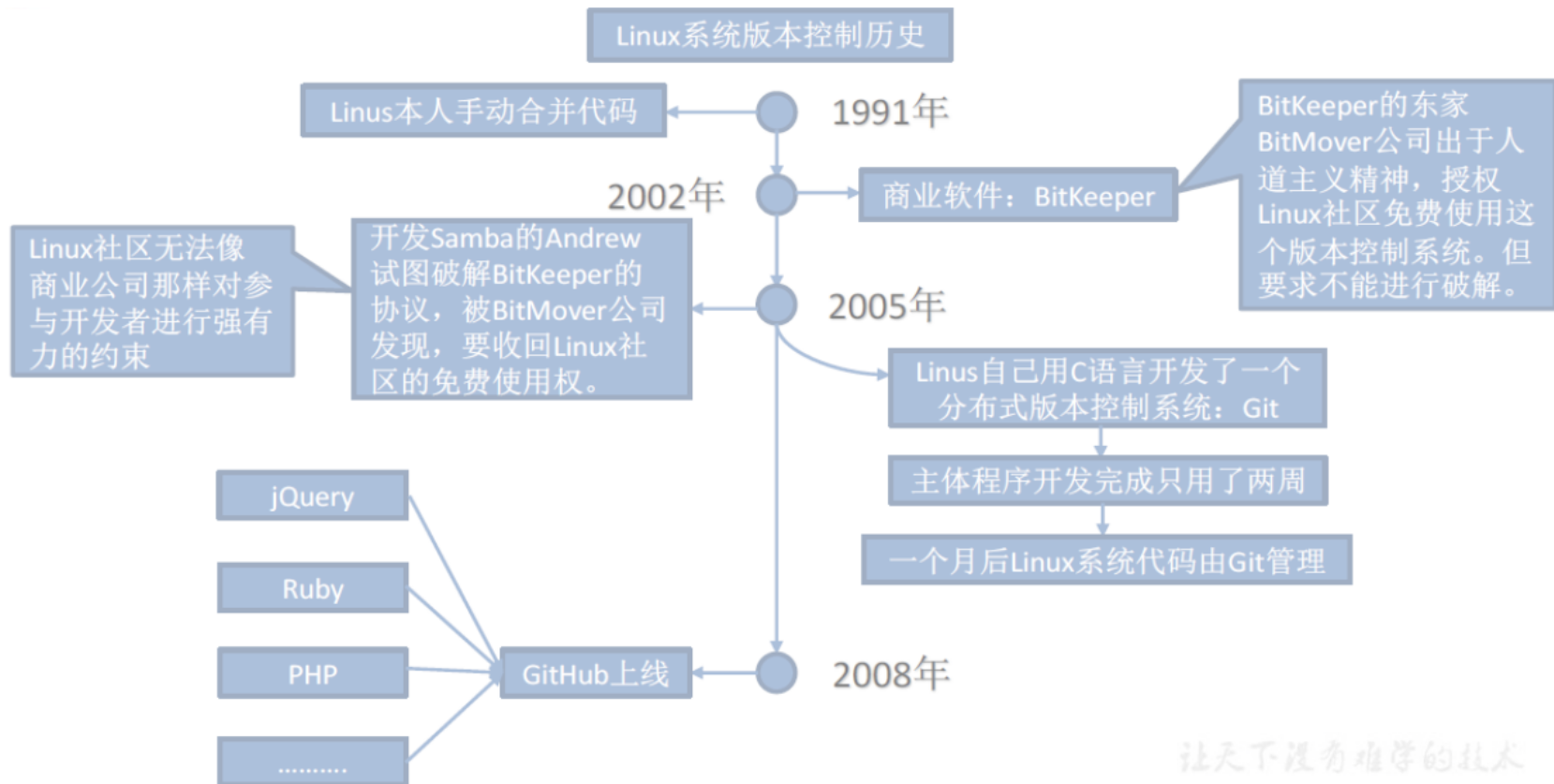


版本控制工具

- 集中式版本控制工具
 - CVS、SVN、VSS
 - 集中化的版本控制系统诸如 CVS、SVN 等，都有一个单一的集中管理的服务器，保存所有文件的修订版本，而协同工作的人们都通过客户端连到这台服务器，取出最新的文件或者提交更新。多年以来，这已成为版本控制系统的标准做法。
 - 这种做法带来了许多好处，每个人都可以在一定程度上看到项目中的其他人正在做些什么。而管理员也可以轻松掌控每个开发者的权限，并且管理一个集中化的版本控制系统，要远比在各个客户端上维护本地数据库来得轻松容易。
 - 事分两面，有好有坏。这么做显而易见的缺点是中央服务器的单点故障。如果服务器宕机一小时，那么在这一小时内，谁都无法提交更新，也就无法协同工作。



Git简史



让天下没有难学的技术

CC BY-SA 4.0 licensed presentation

Git工作机制

Git和代码托管中心

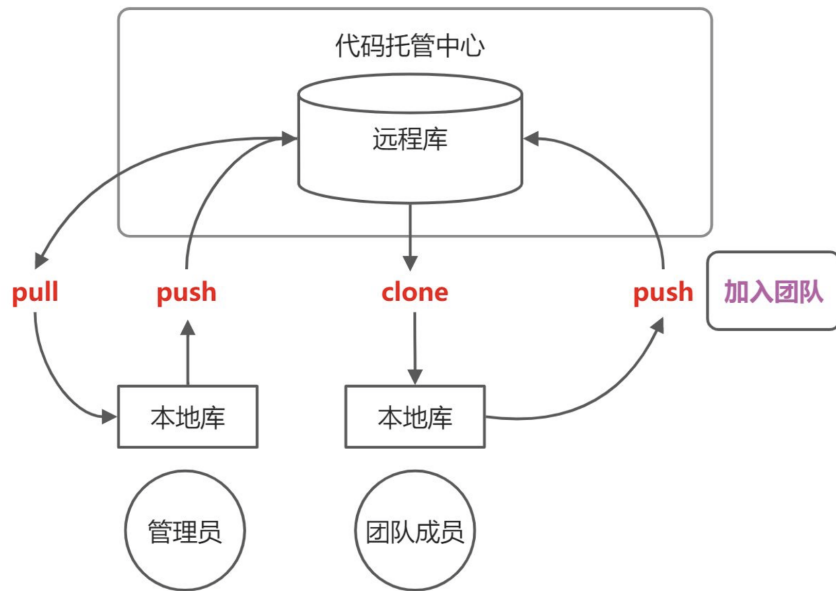
代码托管中心是基于网络服务器的远程代码仓库，一般我们简单称为远程库

局域网

- Gitlab

互联网

- Github
- Gitee



Git安装

代码托管中心是基于网络服务器的远程代码仓库，一般我们简单称为远程库

- 官网下载
- Mac terminal

Homebrew

```
$ brew install git
```

MacPorts

```
$ sudo port install git
```


Git常用命令

命令名称	作用
<code>git config --global user.name 用户名</code>	设置用户签名
<code>git config --global user.email 邮箱</code>	设置用户签名
<code>git init</code>	初始化本地库
<code>git status</code>	查看本地库状态
<code>git add 文件名</code>	添加到暂存区
<code>git commit m " 日志信息 " 文件名</code>	提交到本地库
<code>git reflog</code>	查看历史记录
<code>git reset hard 版本号</code>	版本穿梭