

分布式版本控制Git（一）

自动化工具系列

大纲

- **Git是什么**
- **Git为何与众不同**
- **Git下载安装和基本概念理解**

版本控制

版本控制是指对软件开发过程中各种程序代码、配置文件及说明文档等文件变更的管理。

1 追踪文件的变更。什么时候、什么人更改了文件的什么内容等信息忠实地记录下来。每一次文件的改变，文件的版本号都将增加。

2 并行开发。在多人协同作业场景中，版本控制可以有效地解决版本的同步以及不同开发者之间的开发通信问题，提高协同开发的效率。并行开发中最常见的不同版本软件的Bug修正问题也可以通过版本控制中分支与合并的方法有效地解决。

所有的版本控制系统，其实只能跟踪文本文件的改动。

如TXT文件，网页，所有的程序代码等等，Git也不例外。版本控制系统可以告诉你每次的改动，比如在第5行加了一个单词“hello”，在第6行删了一个单词“world”。

而图片、视频这些二进制文件，虽然也能由版本控制系统管理，但没法跟踪文件的变化，只能把二进制文件每次改动串起来，也就是只知道图片从100KB改成了120KB，但到底改了啥，版本控制系统不知道，也没法知道。



分布式版本控制工具Git

Git：分布式版本控制系统。



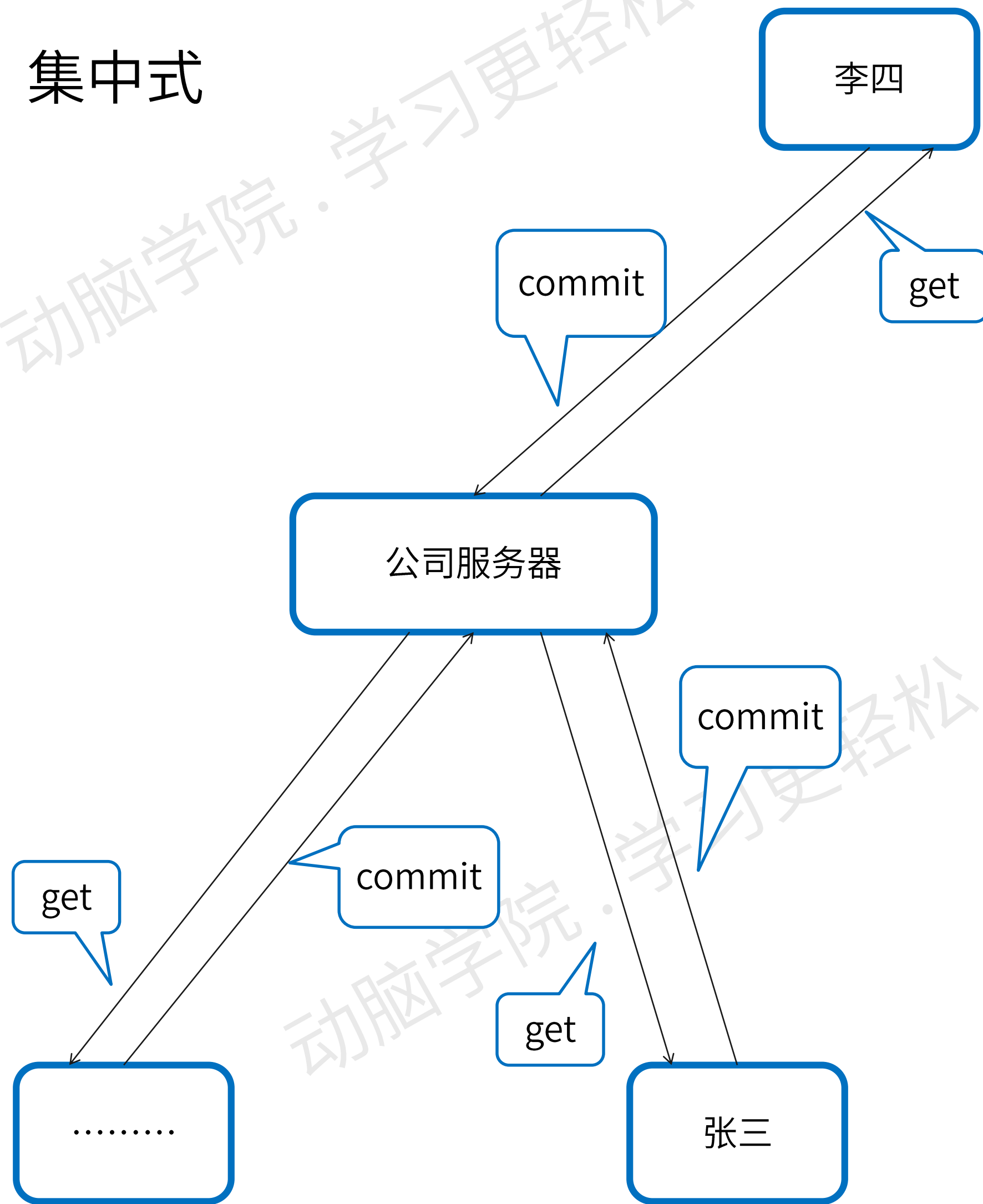
2005年，Linus花了两周时间自己用C写了一个分布式版本控制系统，一个月之内，Linux系统的源码已经由Git管理了。

Git迅速成为最流行的分布式版本控制系统。

2008年，GitHub网站上线了，它为开源项目免费提供Git存储，无数开源项目开始迁移至GitHub，包括jQuery，PHP，Ruby等等。

集中式VS分布式

集中式

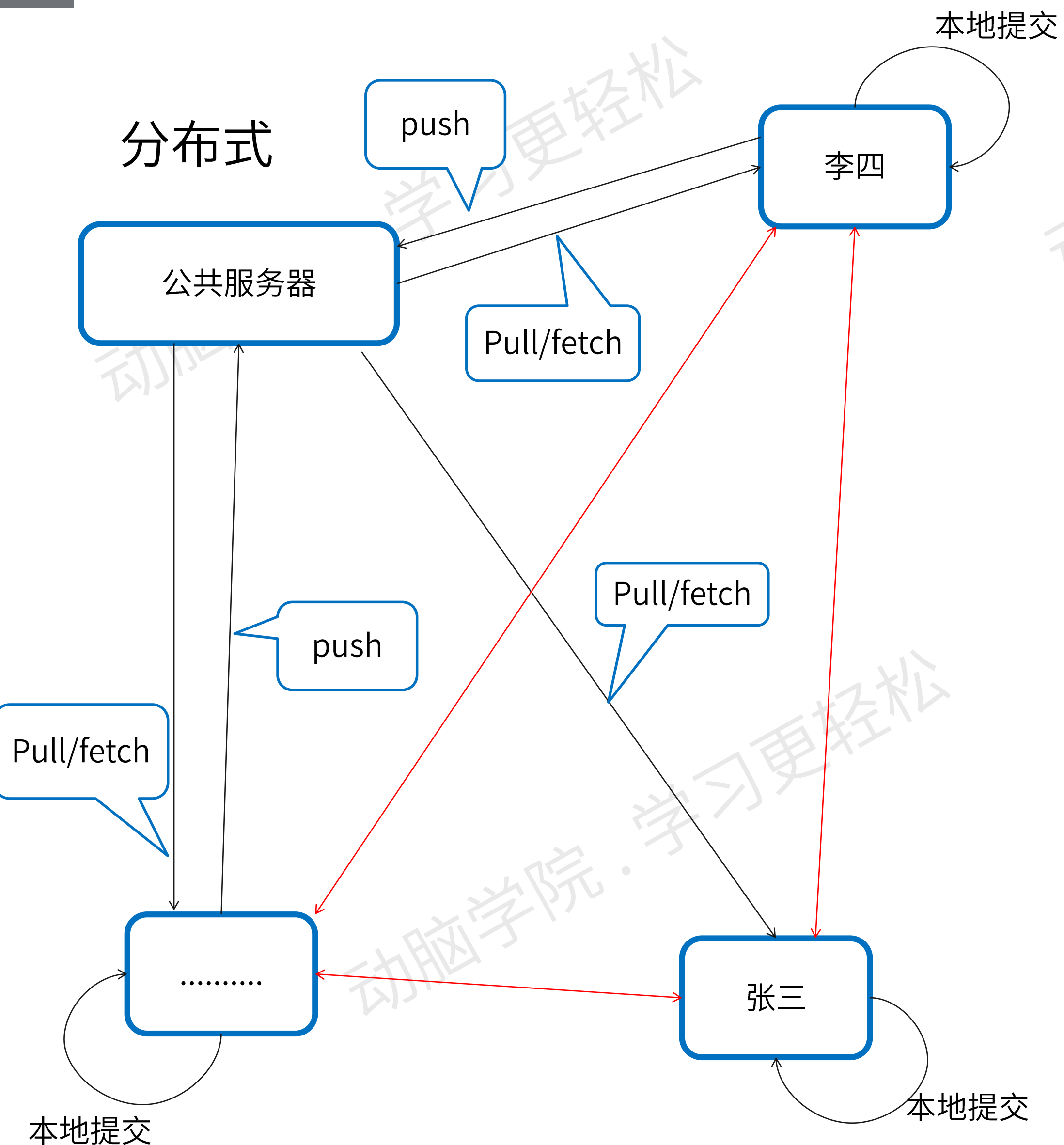


1、从服务器下载最新代码。

2、进入自己的分支，进行工作，每隔一个小时向服务器自己的分支提交一次代码（对代码改来改去，想看前一个小时自己修改了哪些代码，就需要这样做）。

3、从服务器下载最新代码，并解决服务器版本库与本地代码的冲突，将自己的分支合并到服务器主分支上。

集中式VS分布式



1 从服务器上克隆完整的Git仓库（包括代码和版本信息）到本级上。

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、合并分支，提交到本地版本库。

4 把服务器上最新版的代码pull/fetch下来，解决服务器版本库与本地代码的冲突,然后跟自己的主分支合并。

5 如果没问题，push到服务器

集中式VS分布式

集中式：如果中央服务器出故障，SVN的历史库存在于中央仓库，每次对比与提交代码都必须连接到中央仓库才能进行；断网的情况下，其实也能是工作，但是由于没有版本控制的记录，当多人修改后就比较难以快速的合并。

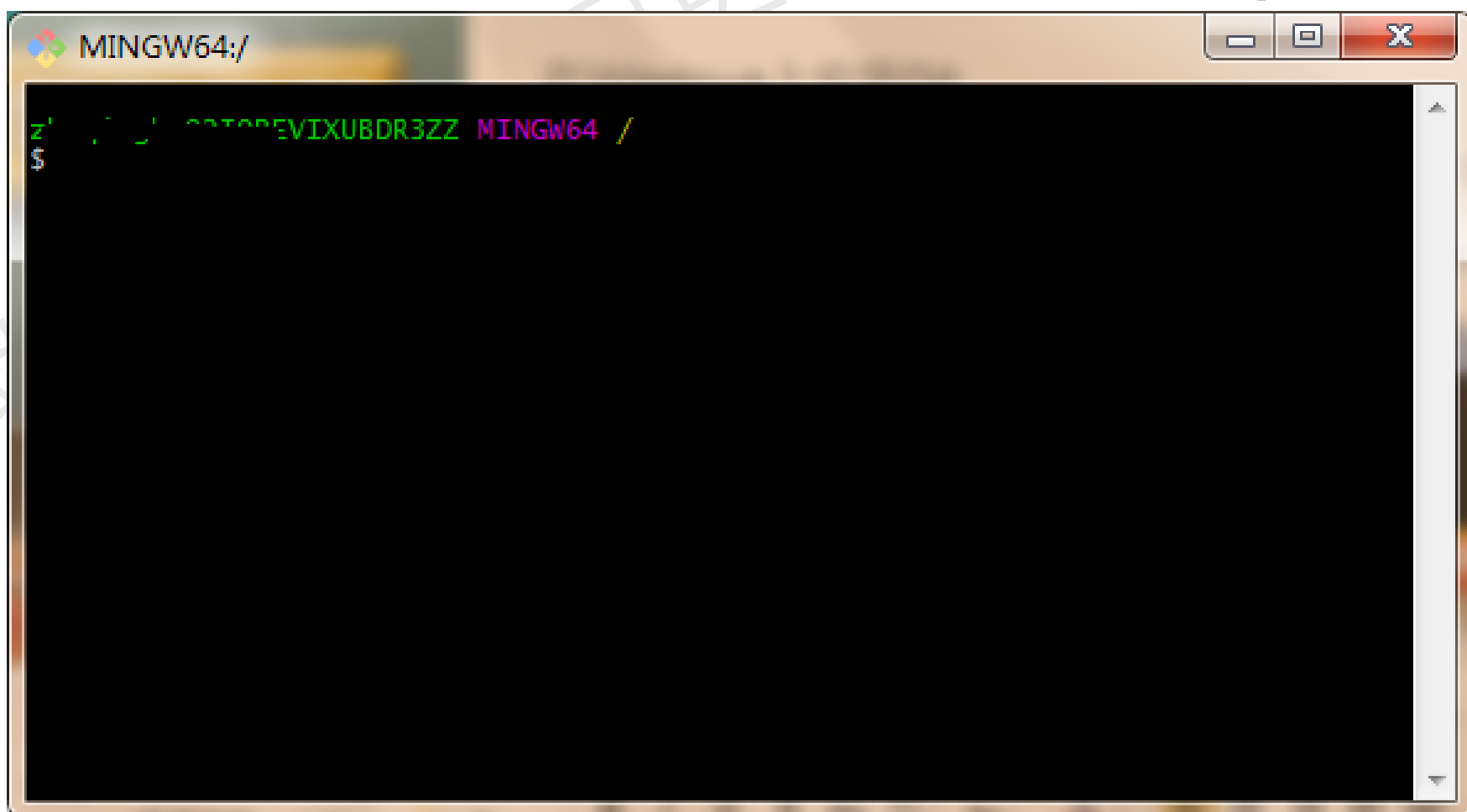
分布式：如果中央服务器故障，Git本地仓库包含代码库还有历史库，不需要网络，自己可以在脱机环境查看开发的版本历史，在本地的环境开发也可以进行比对，回退等操作。Git都在本地保存了版本记录，所以大家合并起来就容易多了，等到服务器好了，直接提交就行了；或者可以随时创建一个新的中央库，随便从其他人那里复制一个然后同步就立刻恢复了中央库。

svn中央服务器挂了，那我一样可以重新搭建一个服务器呢，然后将本地的项目上传就能恢复？

下载安装

1 从Git官网直接 <https://git-scm.com/downloads> 下载安装程序

2 安装完成后，在开始菜单找到“git” -> “git bash”，弹出下面界面表示Git安装成功。



3 设置用户信息，用户名称与邮件地址，每一个Git的提交都会使用这些信息，并且它会写入到你的每一次提交中，不可更改

```
$ git config --global user.name "Your Name"
```

```
$ git config --global user.email "email@example.com"
```

注意git config命令的--global参数，用了这个参数，表示你这台机器上所有的Git仓库都会使用这个配置，当然也可以对某个仓库指定不同的用户名和Email地址。

4 检查配置信息

```
git config --list
```

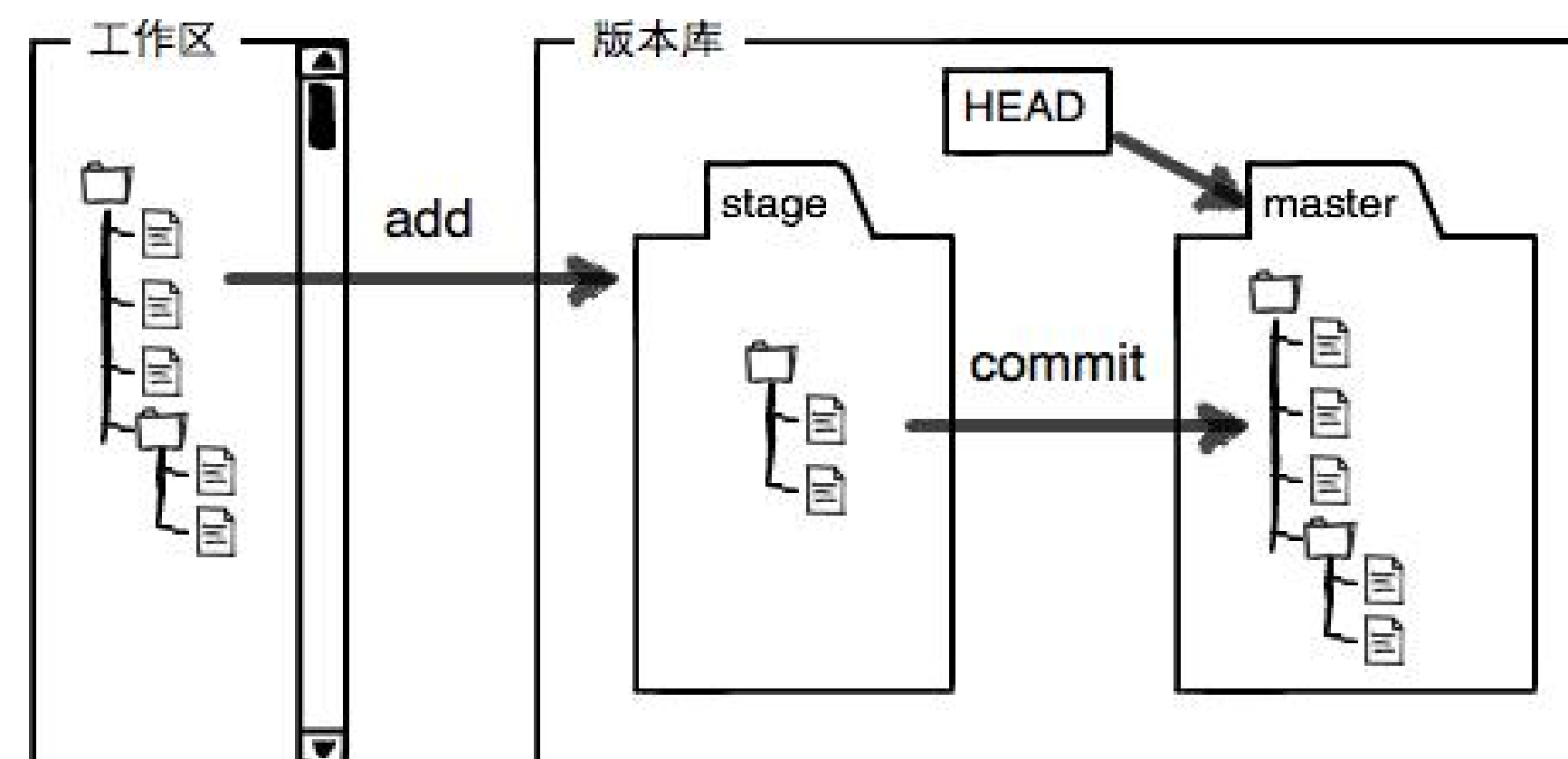
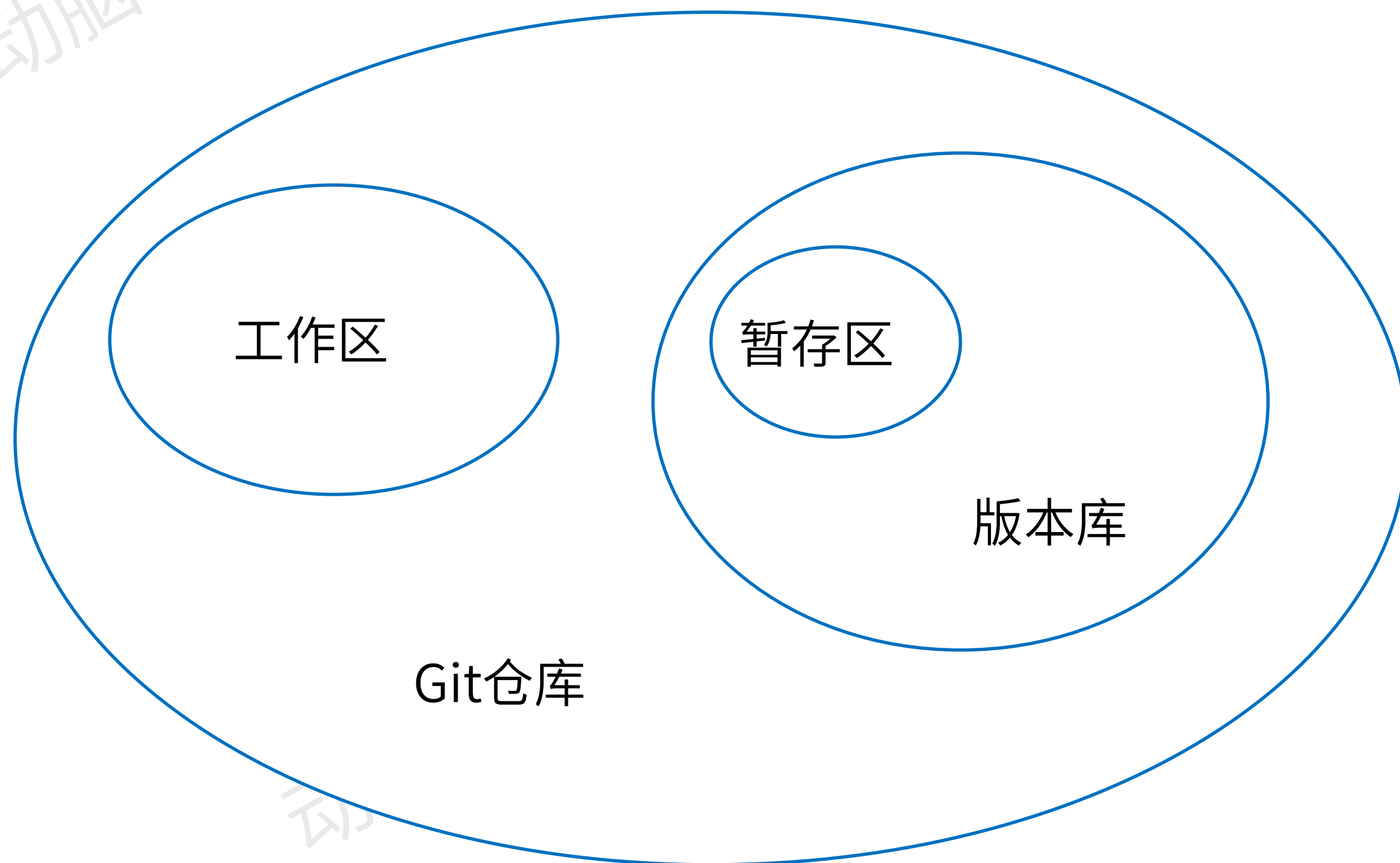

Git概念理解

Git工作区域种类:

工作区（工作目录）：电脑里能看到的目录

版本库（Git 仓库目录）：工作区有一个隐藏目录.git，就是Git的版本库。

暂存区域：一个文件，保存了下次将提交的文件列表信息，一般在 Git 仓库目录中。有时候也被称作`索引`，不过一般说法还是叫暂存区域。



tips: Git版本库中为我们自动创建的第一个分支master，以及指向master的一个指针叫HEAD。

Git概念理解

被管理文件的几种状态：

已修改 (modified)：表示修改了文件，但还没保存到本地版本库中

已暂存 (staged)：表示对一个已修改文件的当前版本做了标记，使之包含在下次提交的快照中

已提交 (committed)：表示数据已经安全的保存在本地版本库中

Git 工作流程：在工作目录中修改文件 → 暂存文件 → 将文件的快照放入暂存区域 → 提交更新，找到暂存区域的文件，将快照永久性存储到 Git 仓库目录。

