一、svn介绍

概念：svn(subversion)是一个跨平台的版本管理系统，类似一个文件服务器，但是可以记录每个文件的每一次修改更新记录，这样就可以回退到到任意时刻的旧的版本，可以用来管理程序源码、其他文件类型（文本、视频、图片等等）；

  二、svn与git的区别

1、svn:

svn版本控制是集中式的数据管理，存在一个中央版本库，所有开发人员本地开发使用的代码都是来自这个版本库，在中央版本库中建立多个项目；

svn工作流程：

1、在中央版本库中创建一个项目，项目中包含主干和分支，分支是从主干复制的；

2、开发人员从中央版本库check out 下这个分支的代码；

3、增加自己的代码、修改或删除现存的代码；

4、commit 代码，如果修改期间其他人提交了代码，就会提示过期无法提交，就需要先up，之后再提交；如果up代码时出现冲突，就需要解决冲突之后在提交（开发人员协商解决冲突）

5、测试环境测试，解决问题之后合并到主干中，在之后更新生产环境代码；

缺点：

1、严重依赖网络环境，连接不到中央版本库时无法工作；

2、需要备份-------------------------->需要备份数据和修改的历史记录

2、git:

git是有Linus

开发的，所以和linux结合比较紧密，以至于windows需要使用cygwin才能完美工作；git是分布式版本控制系统，没有中央版本库的说法，和svn不同的是，开发者本地有一个完整的git仓库，但是实际使用中需要建立一个远程的git仓库，以方便共享代码和版本控制；

git工作流程：

1、本地创建一个git库，并将其add到远程git库中；

2、在本地添加、删除、修改文件，然后commit，此时commit 是提交到本地git仓库中；

3、将本地git库分支push到远程git库的分支，如果此时远程已经有别人push过，那远程git将不允许你push，必须先pull。并解决冲突之后才能在push到远程git库；

优点：

1、开发时不依赖网络，随时commit到本地，和随时查看修改历史，只有在合并代码时才需要连接远程git仓库；

2、svn还是主流，git正在发展，将来会成为主流，都掌握会更好；

三、svn之运维人员需要掌握的程度：

1、安装、部署、维护、排障；

2、简单使用，很多公司都是有开发来管理，包括建立仓库和添加删除账号。

3、对于版本控制系统------->运维相当于开发商，开发人员是业主，运维搭建的系统为开发人员服务，

四、svn之运行方式和访问方式：

运行方式：

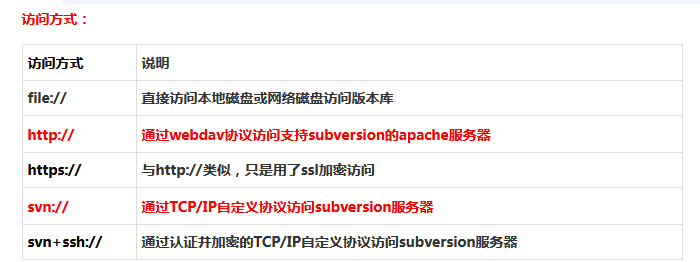
1、独立服务器访问：（svn://svn.etiantian.org/www/）

2、和apache等http服务结合：（http://svn.etiantian.org/www/）

a、单独安装apache+svn;

b、csvn（）是一个整合好的一体软件，带有web管理页面的svn软件;

3、本地直接访问：（file://application/data/www/）



五、数据存储

svn存储版本数据也有2种方式：BDB(一种事务安全型表类型)和FSFS(一种不需要数据库的存储系统)。因为BDB方式在服务器中断时，有可能锁住数据，所以还是FSFS方式更安全一点。

BDB：

伯克利DB（Berkeley DB），版本库可以使用的一种经过充分测试的后台数据库实现，不能再通过网络共享的文件系统上使用，伯克利DB是Subversion1.2版本以前的缺省版本库格式

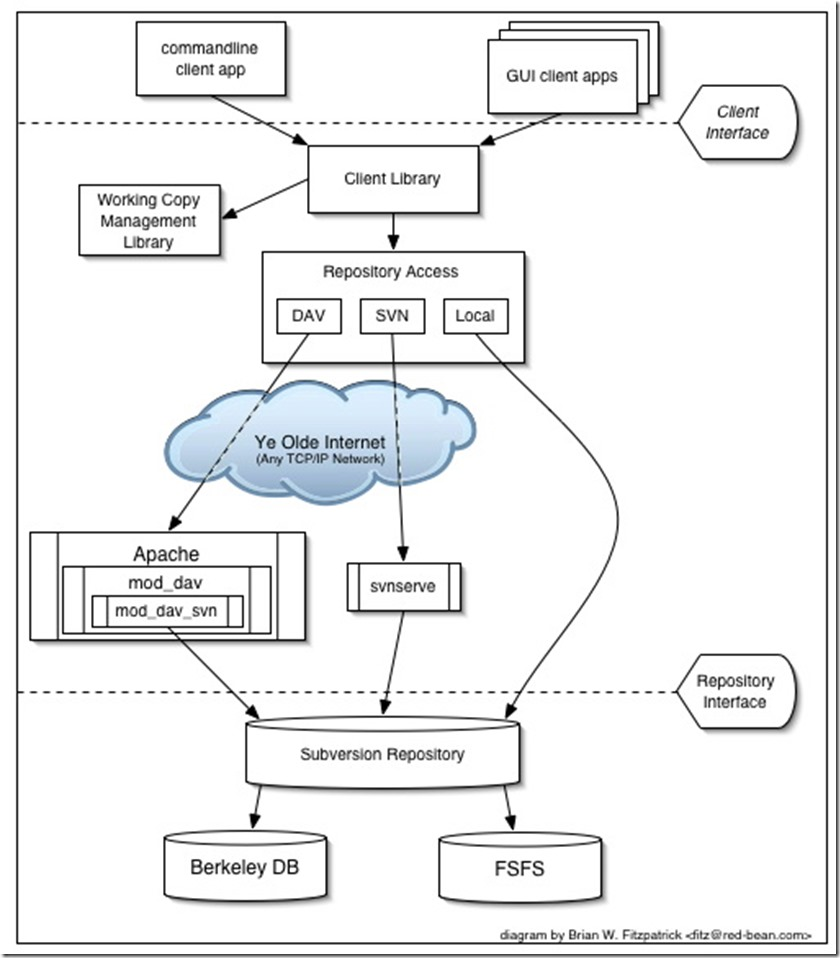
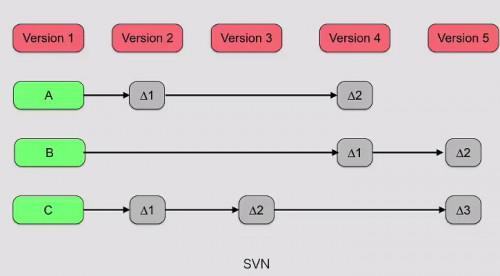
FSFS：

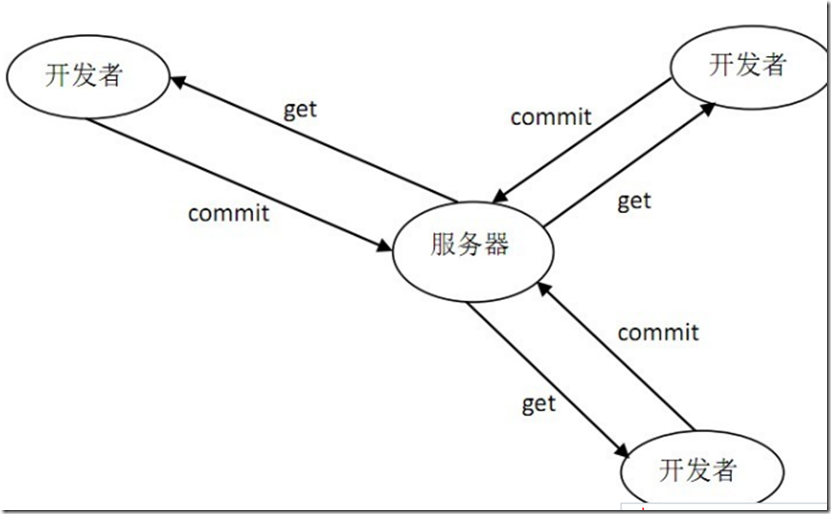
一个专用于Subversion版本库的文件系统后端，可以使用网络文件系统（例如：NFS或SMBFS）。是1.2版本及以后的缺省版本库格式。

svn是基于关系型数据库的BDB或一系列二进制文件的FSFS。这解决了许多问题（如：并行读写共享文件）以及添加了许多新功能（如：运行时的事物特性）。然而另一方面，数据存储由此变得不透明，不能像ftp,samba,nfs等能看到实体的文件。

六、原理

SVN是一个增量式的版本控制，它不会讲各个版本的副本都完整的保存下来，而只会记录下版本之间的差异，然后按照顺序更新或者恢复特定版本的数据。这使得服务端的存储量会非常低。





集中式代码管理的核心是SVN服务器，开发者在开始新一天工作之前必须现在本地update一下代码，然后开发、解决冲突、合并。所有的版本信息都放在SVN上面。