Files development based on a version control system

– An introduction to Git

Xin-Qiu Yao

11/16/2012

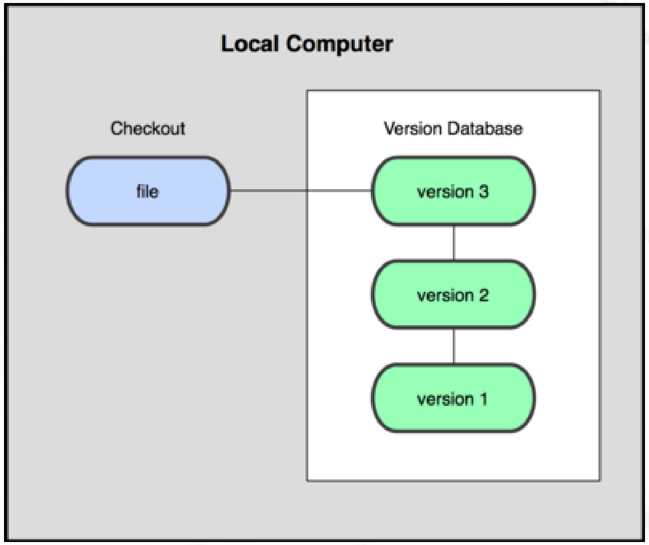
Overview

- What is version control
- Git and its advantages
- Basic commands of Git
- Work with remote servers – **Bitbucket**
- Other online resources

Git has much more function than introduced in these slides!

What is version control

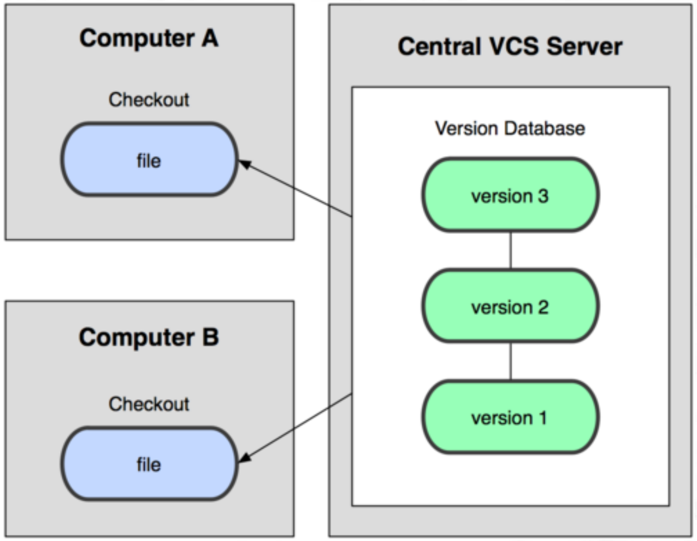
- Records changes to a file or set of files over time
  - Track changing history
  - Recall specific versions later
  - Recover lost files
  - Manage development efficiently (branches)
  - Collaboration



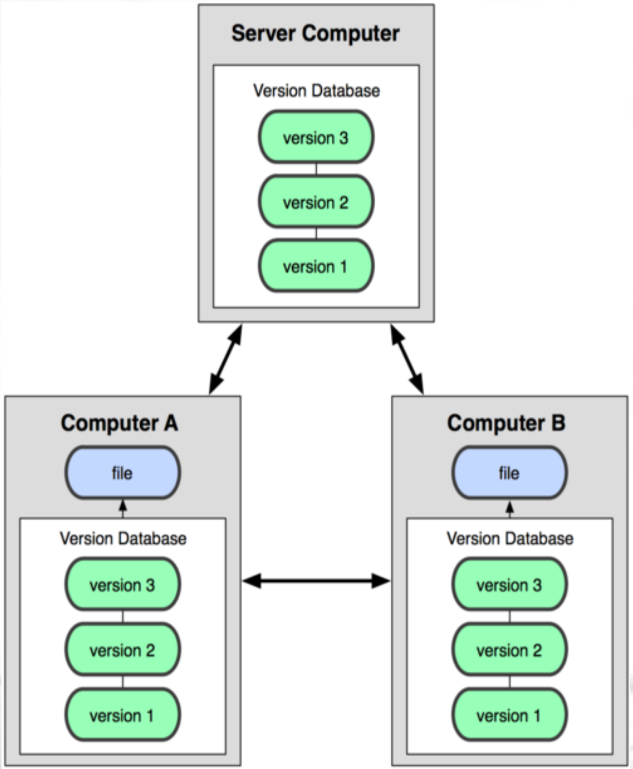
Git and its advantages

- Git is a distributed version control system
  - More robust
- Git is very flexible in making branches

Centralized (CVS, SVN)

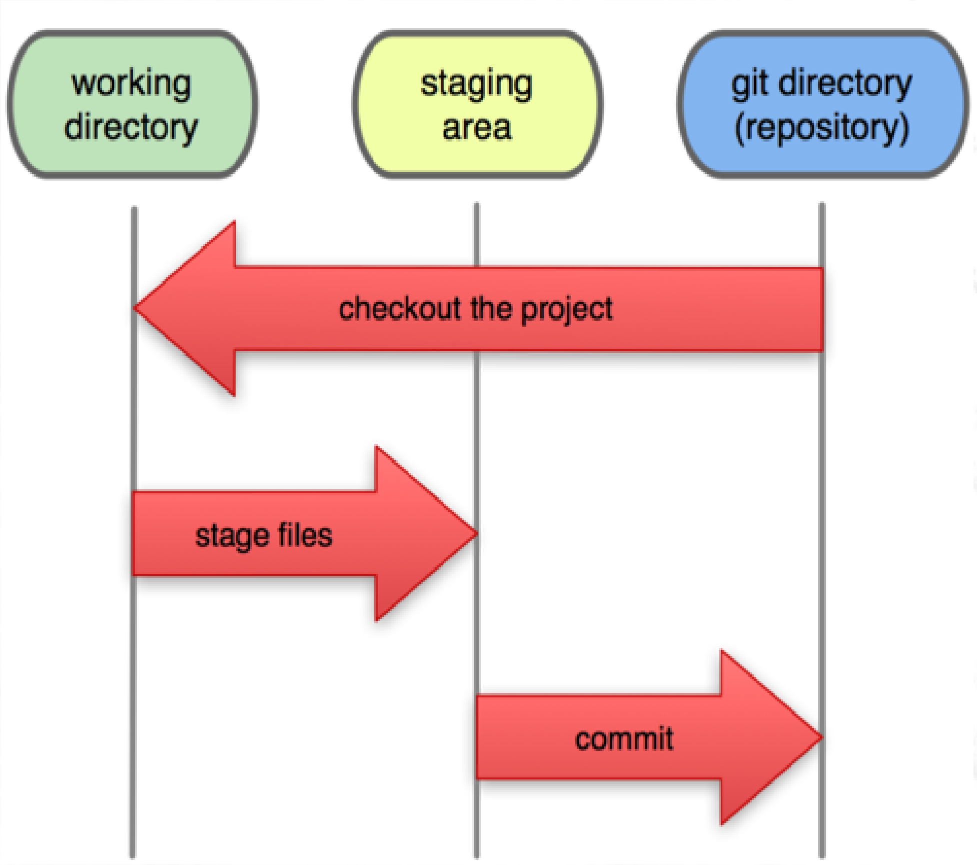


Distributed (Git)



Basic concepts of Git

- The three states
  - Working directory (what you see)
  - Staging area (ready to record)
  - Repository (recorded)



Install Git and setup a repository

- Install necessary libraries

```
$ yum install curl-devel expat-devel \
  gettext-devel openssl-devel zlib-devel
```

- Install Git and configure

```
$ yum install git-core
$ git config --global user.name “Your name”
$ git config --global user.email “Your Email”
```

- Go to your working directory

```
$ git init
$ git add .
$ git commit -m “Initial commit”
```

A simple cycle of development

- Modify files
- Check current status

```
$ git status
```

- Stage files

```
$ git add file
```

- Commit

```
$ git commit -m “A simple cycle”
```

- Check the log history

```
$ git log
```

More activities

- Modify files, generate new files, make new directories
- Check status and commit

```
$ git status
$ git add file.mod file.new dir
$ git commit -m “More activities I”
```

- Delete and rename files

```
$ git rm file
$ git mv file1 file2
$ git commit -m “More activities II”

$ git log
```

Recover deleted files

- Deleted but not staged

```
$ git checkout -- file
```

- Deleted, staged, but not committed

```
$ git reset HEAD file
$ git checkout -- file
```

- Deleted and committed
  - Recovery is possible but a bit more complicated
- Same commands for modified files mean discard changes

Review changes

- Compare modified files with last commit

```
$ git diff [file]
```

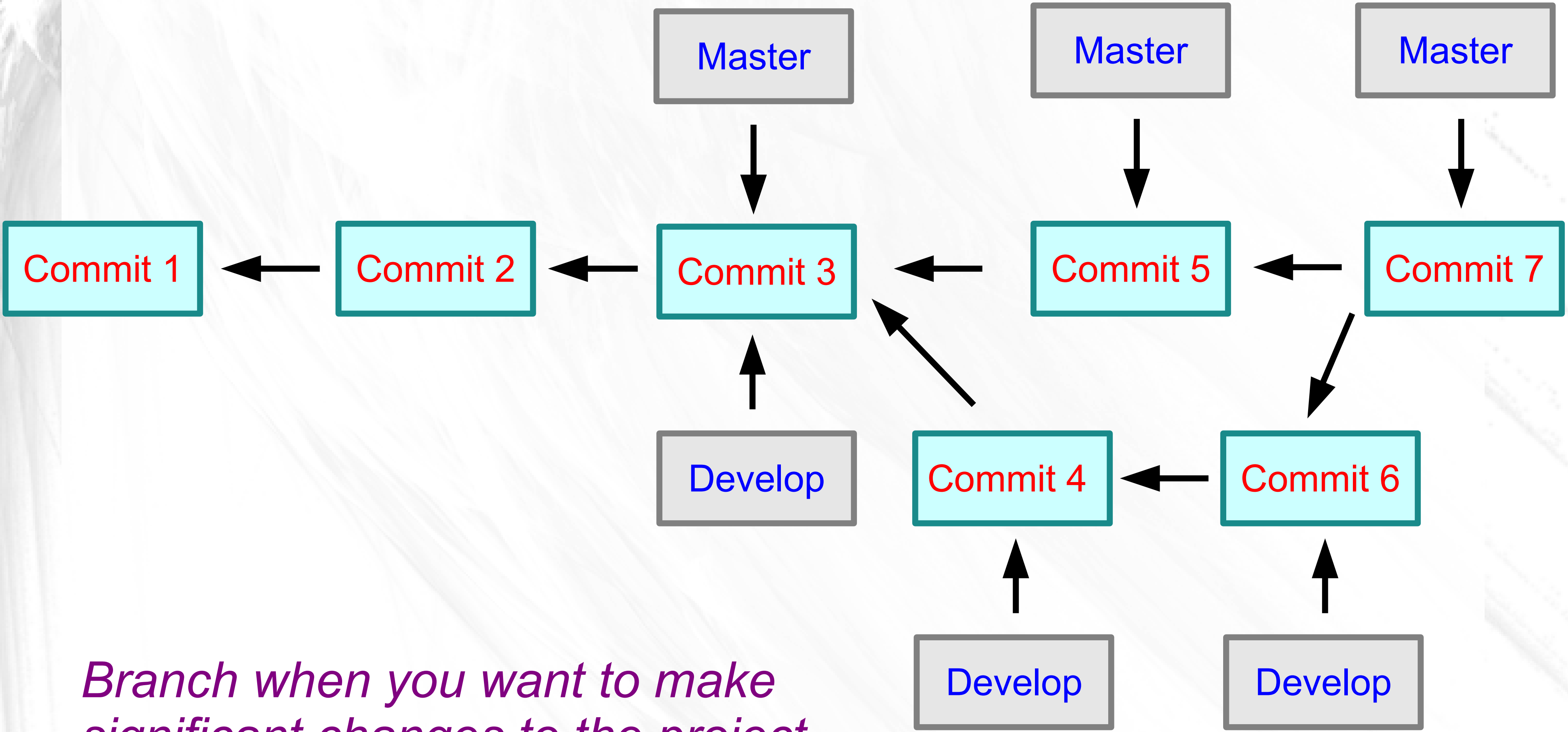
- Compare staged file with last commit

```
$ git diff --staged [file]
```

- Review changes of each commit

```
$ git log -p
```

Work with branches



Branch when you want to make significant changes to the project

How to make a branch

```
$ git branch new_branch

$ git checkout new_branch

$ git branch

(Modify files)

$ git commit -a -m “Commit on a new branch”


(Go back to master branch)
$ git checkout master

$ git branch

(Modify files)

$ git commit -a -m “Commit on master branch”
```

Merge branches

- Compare branches

```
$ git diff master new_branch
```

- Make sure you are on master branch

```
$ git branch
```

- Merge into master

```
$ git merge new_branch
```

- Resolve conflict and commit

```
$ git add files
$ git commit -m "Message"
```

- Delete a branch

```
$ git branch -d new_branch
```

Review history with branches

```
$ git log (show for only current branch)

$ git log --all (show for all branches)

$ git log --pretty=format:”%h - %ar, %an, %ae:
%s” --graph –all (show tree-like log history)


(Set above command as alias)
$ git config --global alias.logtree “--
pretty=format:\”%h - %ar, %an, %ae: %s\”
--graph –all”

$ git logtree
```

Work with remote servers
- Bitbucket (https://bitbucket.org)

- Export local repository onto Bitbucket
  - Add ssh public key to your account
  - Create an empty repository online
  - Go to local working directory and setup a repository
  - Add remote server

```
$ git remote add origin\
    git@bitbucket.org:user/project.git
$ git remote -v
$ git remote show origin
```

  - Upload

```
$ git push origin master
```

- Import Bitbucket repository into local machine

```
$ git clone git@bitbucket.org:user/project.git \
myproject

$ git remote -v
```

- Make modifications and upload

```
$ git fetch origin master

$ git status

$ git merge origin/master

$ git push origin master
```

Some tips

- Always Google
- Skip explicit staging

```
$ git commit -a -m “Message”
```

- Quickly make and checkout a new branch

```
$ git checkout -b new_branch
```

- Quickly download and merge remote repository

```
$ git pull origin master
```

- Set alias

```
$ git config --global alias.name command
```

- Get helps

```
$ git help command
```

- Visualization

```
$ gitk
```

Other online resources

- Download Git, http://git-scm.com/downloads
- Pro Git, http://git-scm.com/book
- SourceTree, http://www.sourcetreeapp.com/
- Github (latest version of Git), https://github.com/