GIT Training

Use GIT in Linux Workgroup

Barry Song <21cnbao@gmail.com>

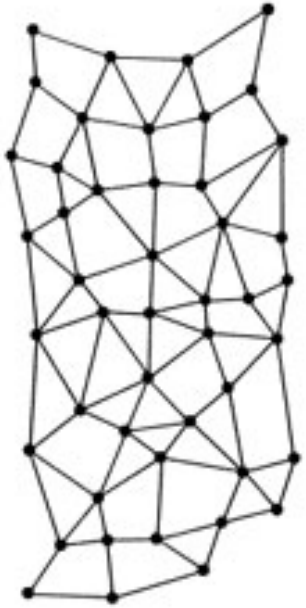
June 2011

宋宝华 Barry.Song@lianlab.org

Distributed?

Most Version Control Systems

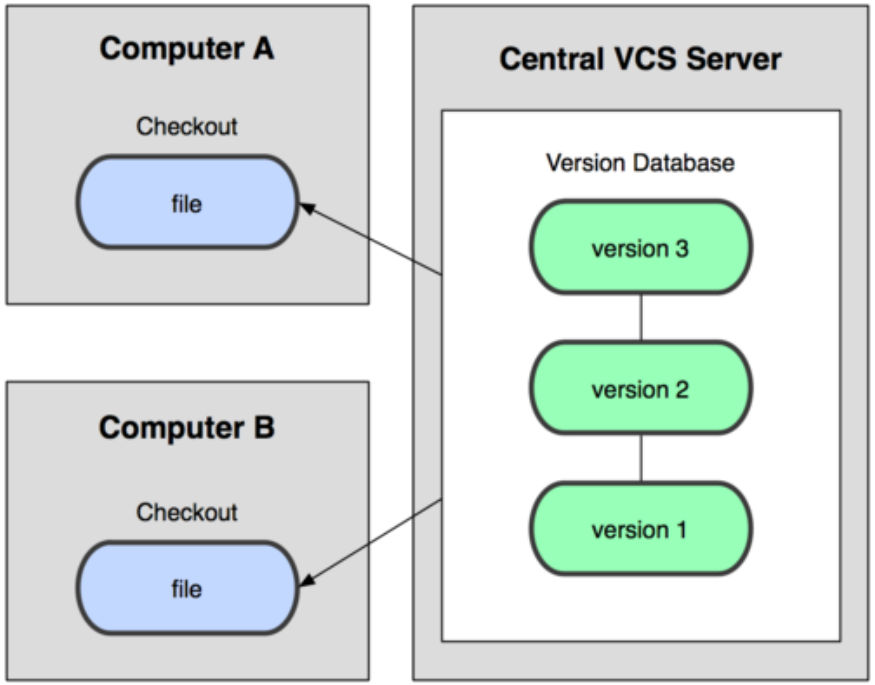
Linus’ Vision of Git

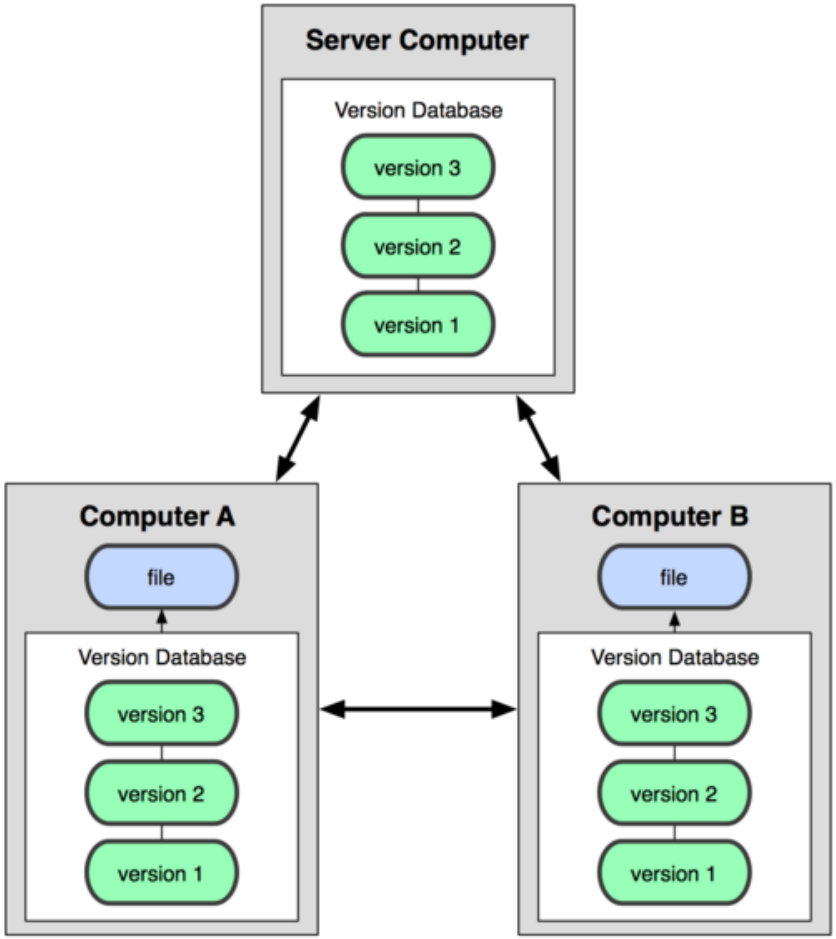


Reality of Distributed Systems

宋宝华 Barry.Song@lianlab.org

SVN/P4 vs GIT





宋宝华 Barry.Song@lianlab.org

A Brief History of Git

- Linus uses BitKeeper to manage Linux code
- Ran into BitKeeper licensing issue
  - Liked functionality
  - Looked at CVS as how not to do things
- April 5, 2005 - Linus sends out email showing first version
- June 15, 2005 - Git used for Linux version control

Git config

- *~/.gitconfig*
  - *git config --global user.name "Barry Song"*
  - *git config --global user.email 21cnbao@gmail.com*
  - git config --global core.editor vim
  - git config --global merge.tool vimdiff
  - git config –l

    color.ui=auto

    user.name=Barry Song

    user.email=21cnbao@gmail.com

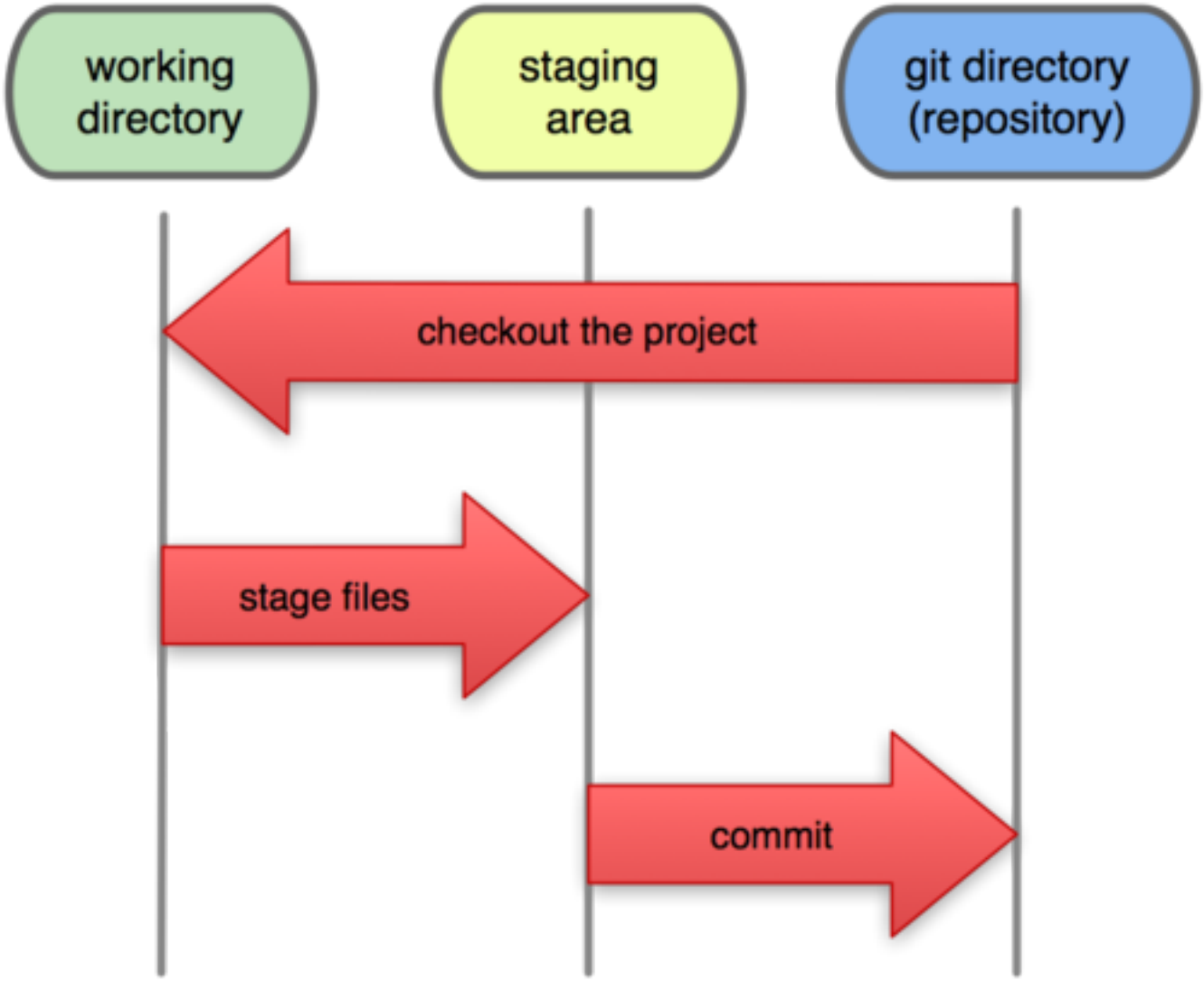
    merge.tool=vimdiff

    core.editor=vim
- *.git/config*
- .gitignore
  - Stored in directory for ignoring

git workflow

- Fetch or clone (create a copy of the remote repository)
- Modify the files in the local branch
- Stage the files
- Commit the files locally
- Push changes to remote repository

Local Operations



宋宝华 Barry.Song@lianlab.org

Working With Git

- `echo "I love Git" >> hello.txt`
- `git diff`
  - Shows changes we have made
- `git status`
  - Shows list of modified files
- `git add hello.txt`
- `git diff`
  - No changes shown as diff compares to the index
- `git diff HEAD`
  - Now can see the changes in working version
- `git status`
- `git commit -m 'Second commit'`

Our First Git Repository

- `mkdir first-git-repo && cd first-git-repo`
- `git init`
  - Creates the basic artifacts in the .git directory
- `echo "Hello World" > hello.txt`
- `git add .`
  - Adds content to the index
  - Index reflects the working version
  - Must be run prior to a commit
- `git commit -a -m 'Check in number one'`

The index – a recap

宋宝华 Barry.Song@lianlab.org

Comparing working dir, index, and branch

1 Have you forgotten to stage something?

`$ git diff` Compare your working directory with the index.

2 What will be in my next commit?

`$ git diff --cached` Compare your index with the latest commit on your branch.

3 What have I changed since the last commit?

`$ git diff HEAD` Compare your working directory *and* index with the latest commit on your branch.

宋宝华 Barry.Song@lianlab.org

Git status

- On branch master
- Changes to be committed
  - new file
  - modified
- Changed but not updated
  - deleted
  - modified
- Untracked files

Git commit

- git commit (edit message by vim)
- git commit –m “what is done” (simple message)
- git commit –a (skip stage)
- git commit –s (add Signed-off-by)
- git commit --amend (modify last commit)

  git commit -m 'initial commit’

  git add forgotten_file

  git commit --amend

Removing and renaming files

- git rm --cached newfile.c (after git add useless file)
- git rm

```
$ git rm myremovedfile.c
$ git commit
```

- Files are renamed with `git mv`. A rename is equivalent to removing the file and adding it again under a new name.

```
$ git mv oldfile.c newfile.c
$ git commit
```

```
$ mv oldfile.c newfile.c
$ git rm oldfile.c
rm 'oldfile.c'
$ git add newfile.c
$ git commit
```

宋宝华 Barry.Song@lianlab.org

Resetting your working directory and index

- Sometimes you just mess up your working directory and/or index and need to start over again.
- Unstage all files but keep the state of the working directory:

```
$ git reset                    Reset all files
$ git reset -- myfile.c        Reset just one file
```

- Unstage all files from the index and remove all changes made to tracked files:

```
$ git reset --hard
```

- Scrap the changes in a tracked file in your working directory ("undo checkout"):

```
$ git checkout HEAD -- myfile.c
```

- Remove all untracked files in your working directory:

```
$ git clean -f
```

Revert commit

- git revert
  - Reverts a commit
  - Does not delete the commit object, just applies a patch
  - Reverts can themselves be reverted!

Git diff between commits/branches

- *`git diff HEAD^^`*

  Show what has changed in last two commits

- *git diff HEAD~10..HEAD~2*

  Show what changed between 10 commits ago and two commits ago

- git diff master branch1

宋宝华 Barry.Song@lianlab.org

Git log

- *git log*
- *git log --grep*
- *git log HEAD~3..HEAD~1*
- *git log –p*
- *git log –p -2*
- *git log --stat*
- git log --pretty=oneline
- git log --pretty=short
- git log --pretty=format:"%h - %an, %ar : %s"
- git log --pretty="%h:%s" --author=gitster --since="2008-10-01" --before="2008-11-01"

Tag

- Tags are just human readable shortcuts for hashes
- *git tag*

  - git tag -a v1.0 -m 'my version 1.0’

  - git show v1.0

  tag v1.0

  Tagger: Barry Song <21cnbao@gmail.com>

  Date: Sun Jun 19 10:18:49 2011 +0800

  - git tag -a v1.2 9fceb02

- *git push*
  - git push origin v1.5
  - git push origin --tags

Branches (1)

- Use `git branch` to create a new branch from the commit that you currently have checked out:

```
$ git branch mybranch
```

- Use the same command to show you the local branches you have created:

```
$ git branch
* master
  mybranch
```

- The star before `master` tells you that this branch is currently checked out.
- Use `git checkout` to switch between branches:

```
$ git checkout mybranch
Switched to branch "mybranch”
```

Branches (2)

- `git checkout -b branch`
- `git checkout -b devel/branch`
- We can now make changes in one branch and propagate change using
  - git merge
  - git cherry-pick

Branches (3)

- *`git merge conflict`*

  - git merge branch1

  Auto-merging new.c

  CONFLICT (content): Merge conflict in new.c

  Automatic merge failed; fix conflicts and then commit the result.

  - git mergetool
  - git commit

Branches (4)

- *`git branch --merged`*

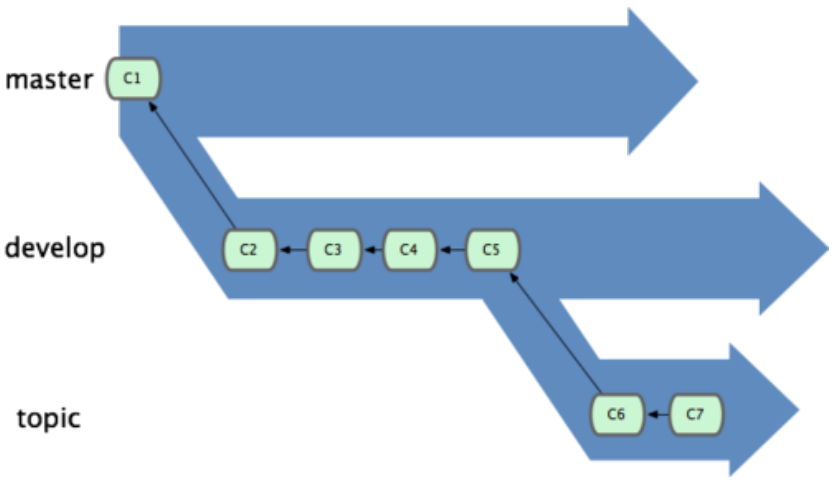
  `git branch –d merged_branch`

- *git branch --no-merged*

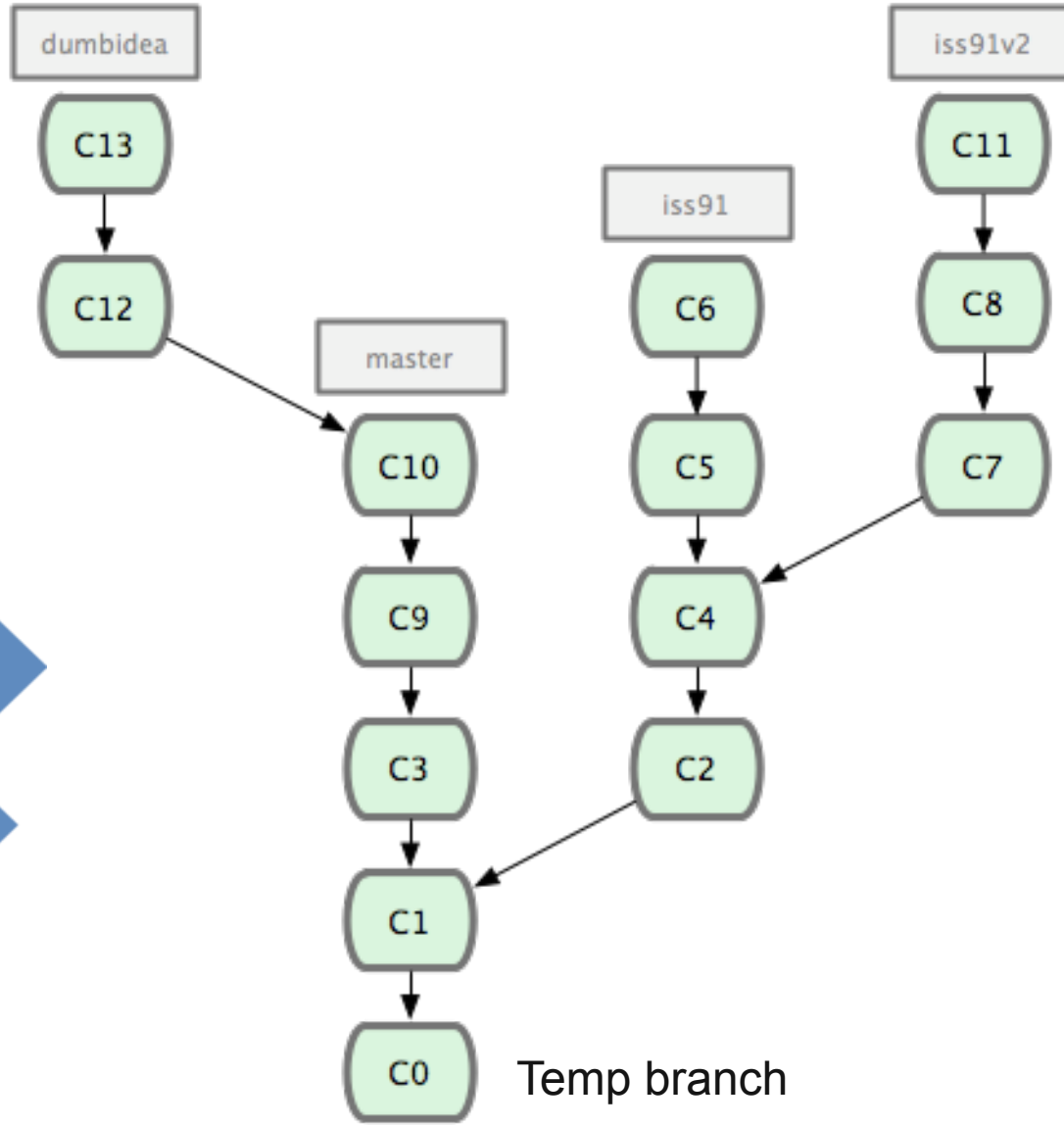
  git branch -d testing

  error: The branch 'testing' is not an ancestor of your current HEAD.

Branch work model



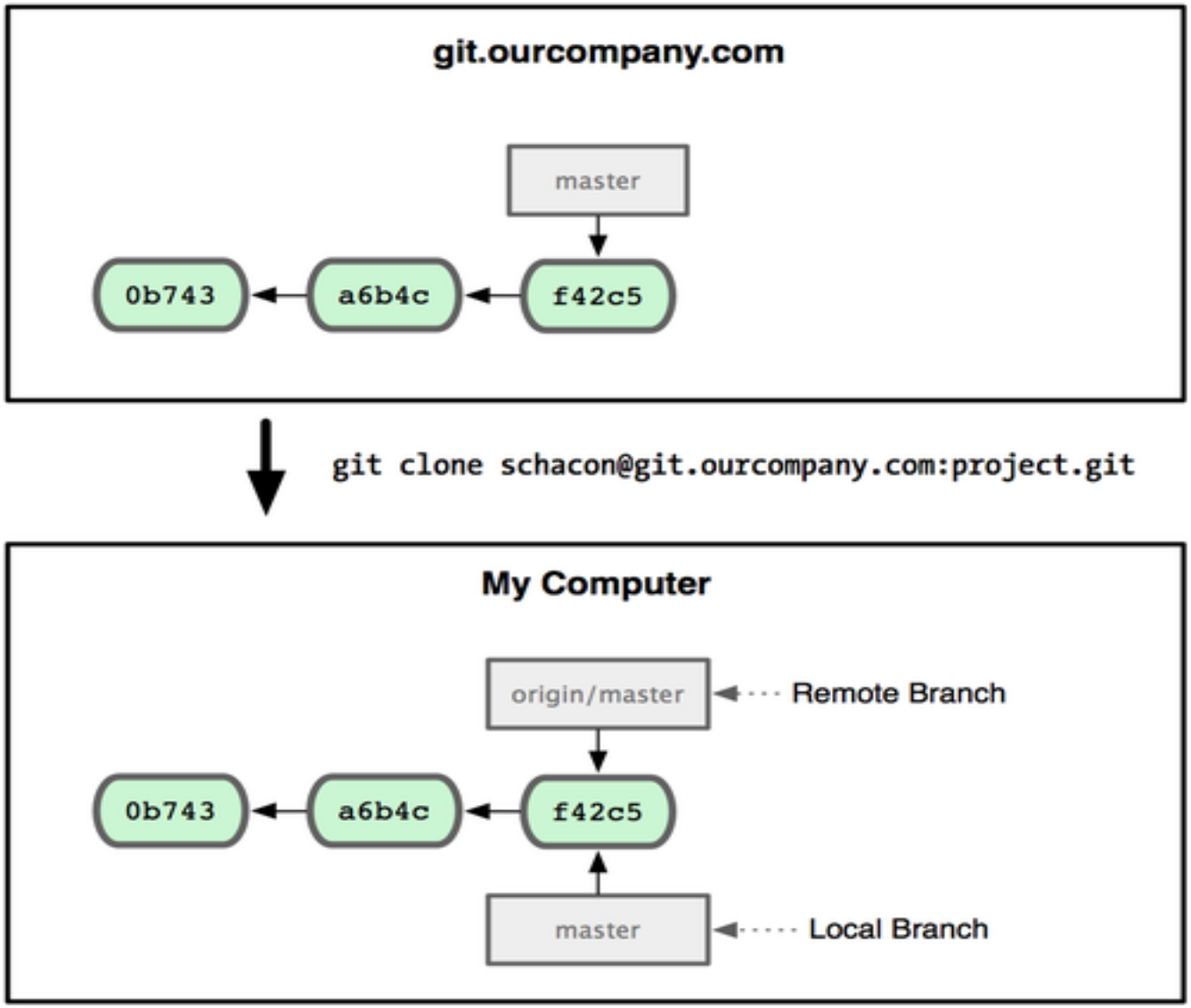
Long term branch



Temp branch

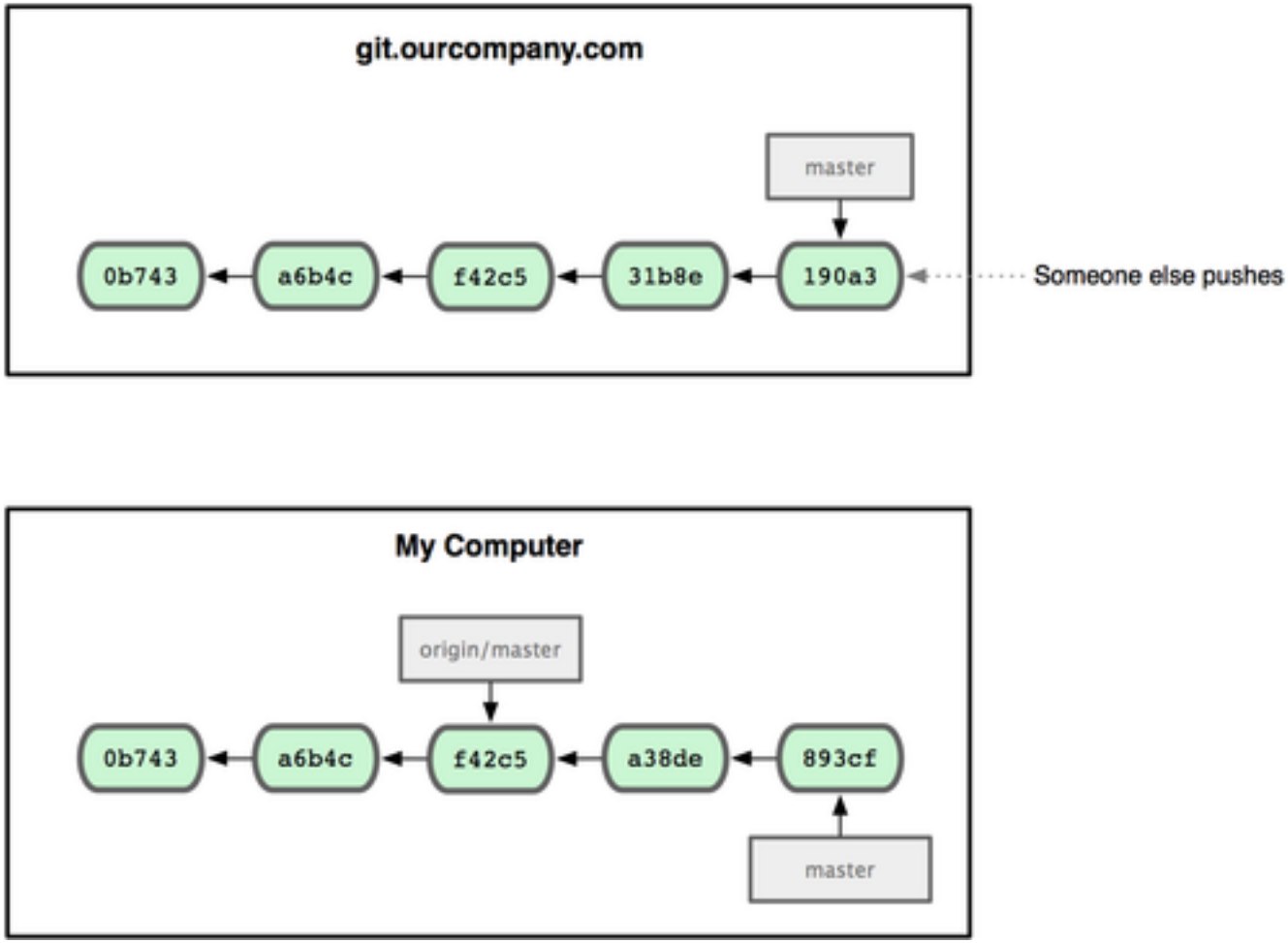
宋宝华 Barry.Song@lianlab.org

Git remote branch(1)



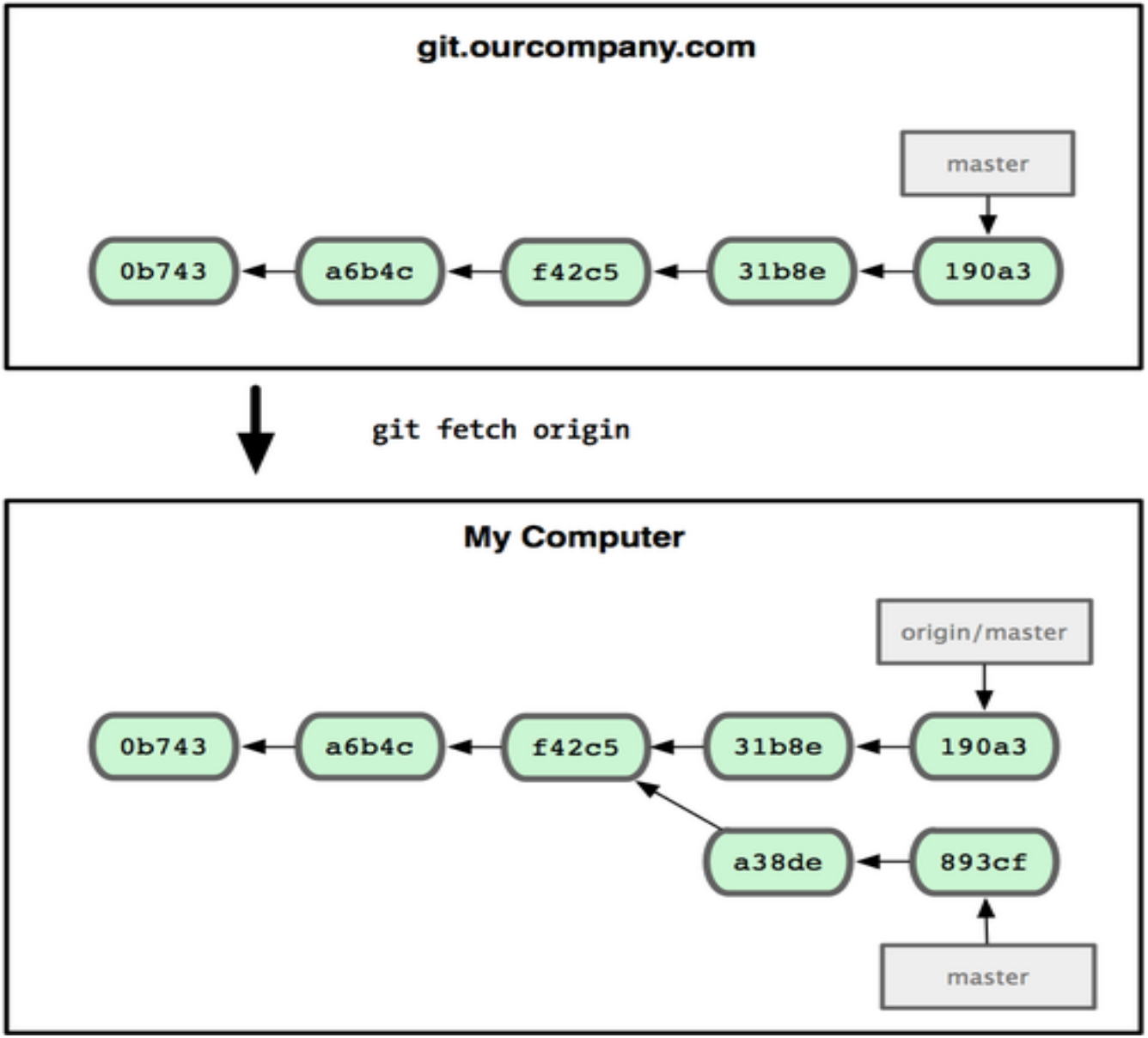
宋宝华 Barry.Song@lianlab.org

Git remote branch(2)



宋宝华 Barry.Song@lianlab.org

Git remote branch(3)



宋宝华 Barry.Song@lianlab.org

Push Branches

- User1: git push origin serverfix
  - git push origin serverfix:awesomebranch (remote branch with different name)
  - git push origin :serverfix (delete remote branch)
- User2: git fetch origin

  remote: Counting objects: 20, done.

  remote: Compressing objects: 100% (14/14), done.

  remote: Total 15 (delta 5), reused 0 (delta 0)

  Unpacking objects: 100% (15/15), done.

  From git@github.com:schacon/simplegit

  * [new branch] serverfix -> origin/serverfix
- git checkout -b serverfix origin/serverfix

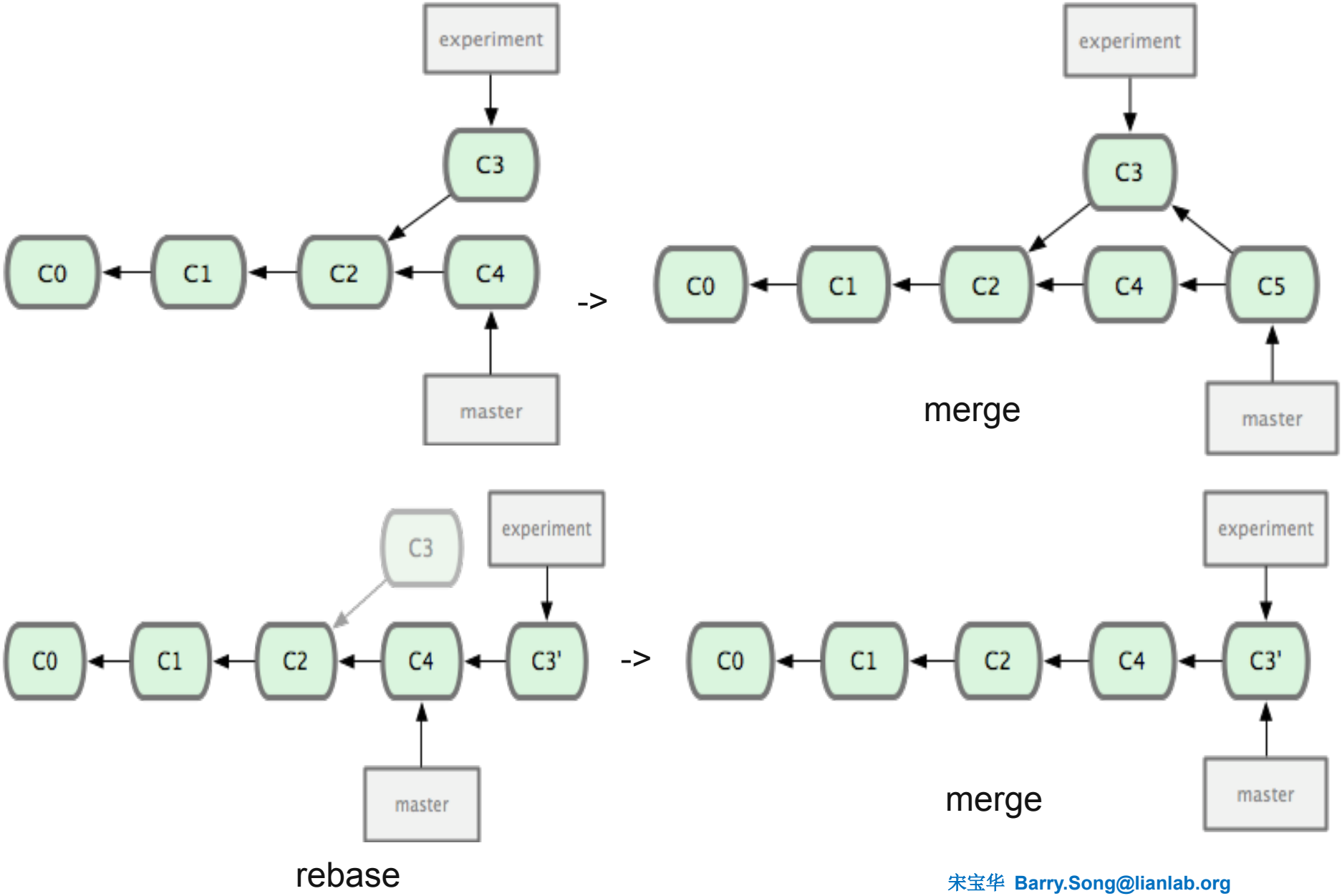
  git checkout --track origin/serverfix

  Branch serverfix set up to track remote branch refs/remotes/origin/serverfix.

  Switched to a new branch "serverfix"

宋宝华 Barry.Song@lianlab.org

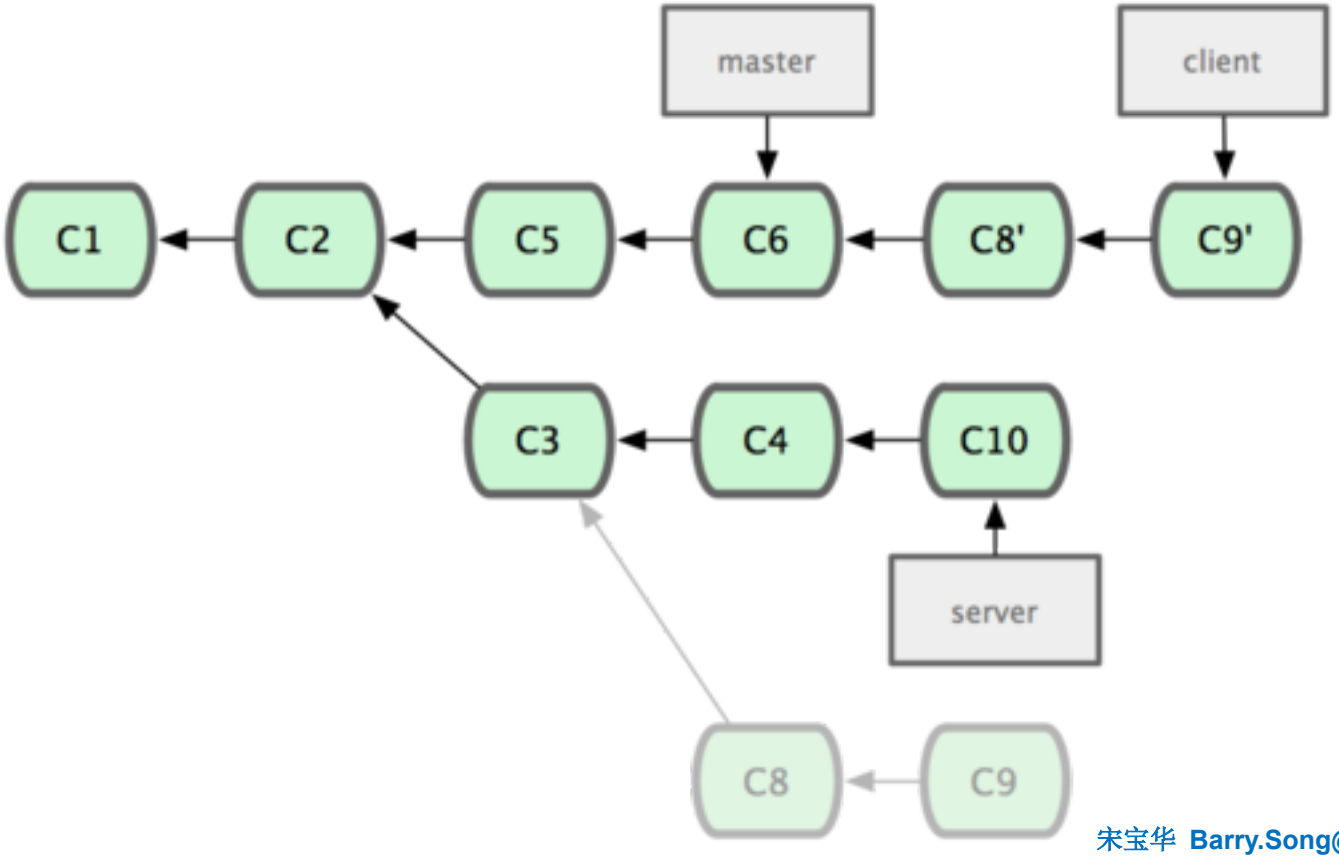
Merge vs Rebase



宋宝华 Barry.Song@lianlab.org

Rebase

- git checkout experiment
- git rebase master
- git rebase master server
- git rebase --onto master server client



Commit rules

- All commits should be logically independent and as small as possible
- With Signed-off-by
- With subject and description

Making patch

- git format-patch
- git format-patch -M
- git format-patch -s --to 21cnbao@gmail.com --cc xxx@xxx.com
- git format-patch 19cd2 --cover-letter

Apply patch

- git apply --check
- git am
- git am conflict

  $ (fix the file)

  $ git add ticgit.gemspec

  $ git am --resolved

  Applying: seeing if this helps the gem
- git log contrib --not master
- git diff master...contrib

Git ls-files

- git ls-files
- git ls-fies -d | xargs git checkout --

Git send-email

- All commits should be logically independent and as small as possible
- With Signed-off-by and Cc
- With subject and description

Git stash

- git stash
- git stash apply

Git show

- git show 18bb2bdfbf8b726c1db0a4f68e65d5cfa18a3622

```
commit 18bb2bdfbf8b726c1db0a4f68e65d5cfa18a3622
Author: Barry Song <21cnbao@gmail.com>
Date:   Sun Jun 19 21:53:21 2011 +0800

    add f1

    Signed-off-by: Barry Song <21cnbao@gmail.com>

diff --git a/del.c b/del.c
index a6a1a7f..ad5672c 100644
--- a/del.c
+++ b/del.c
@@ -4,4 +4,4 @@ add()
  B1
  C1
  D1"
-}
+f1}
```

Git remote

- git remote –v

  origin git://git.kernel.org/pub/scm/linux/kernel/git/torvalds/linux-2.6.git (fetch)
  origin git://git.kernel.org/pub/scm/linux/kernel/git/torvalds/linux-2.6.git (push)

- git remove add

  git remote add linus git://git.kernel.org/pub/scm/linux/kernel/git/torvalds/linux-2.6

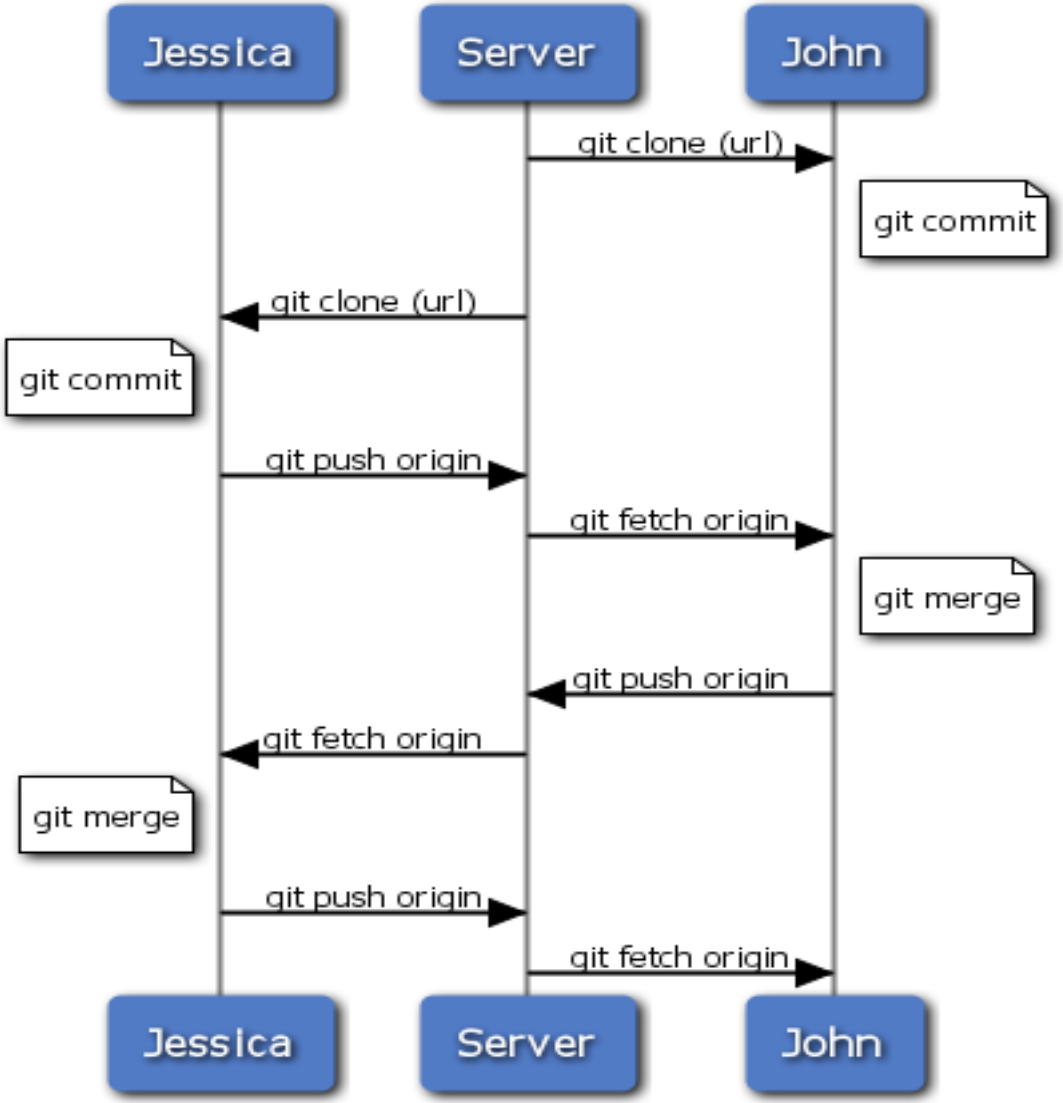
  git remote add rmk http://ftp.arm.linux.org.uk/pub/linux/arm/kernel/git-cur/linux-2.6-arm.git

- Git fetch

  git fetch linus
  git fetch rmk

- git remote show origin

Cooperation



宋宝华 Barry.Song@lianlab.org

宋宝华 Barry.Song@lianlab.org