



Basic git for everyday work

Amerigo Wang
Associate Software Engineer
xiyou.wangcong@gmail.com

What is git?

- **GIT is short for Global Information Tracker.**
- **GIT is a VCS written by Linus, in 2005, to replace BitKeeper.**
- **Distributed development, compare to CVS.**
- **Strong support for non-linear development.**
- **Support SSH, HTTP/HTTPS, git protocols.**
- **Many open source projects switch to GIT, e.g. Linux kernel.**

Getting started

- **Work on existed project: git clone**
- **Work on a new project: git init; git add**
- **Configure your git: git config user.name “foo”; git config user.email “a@b.c”**
- **Check status: git status**
- **Check changes: git diff**
- **Commit changes: git commit**
- **View history log: git log**
- **View commit: git show**

Basic concepts

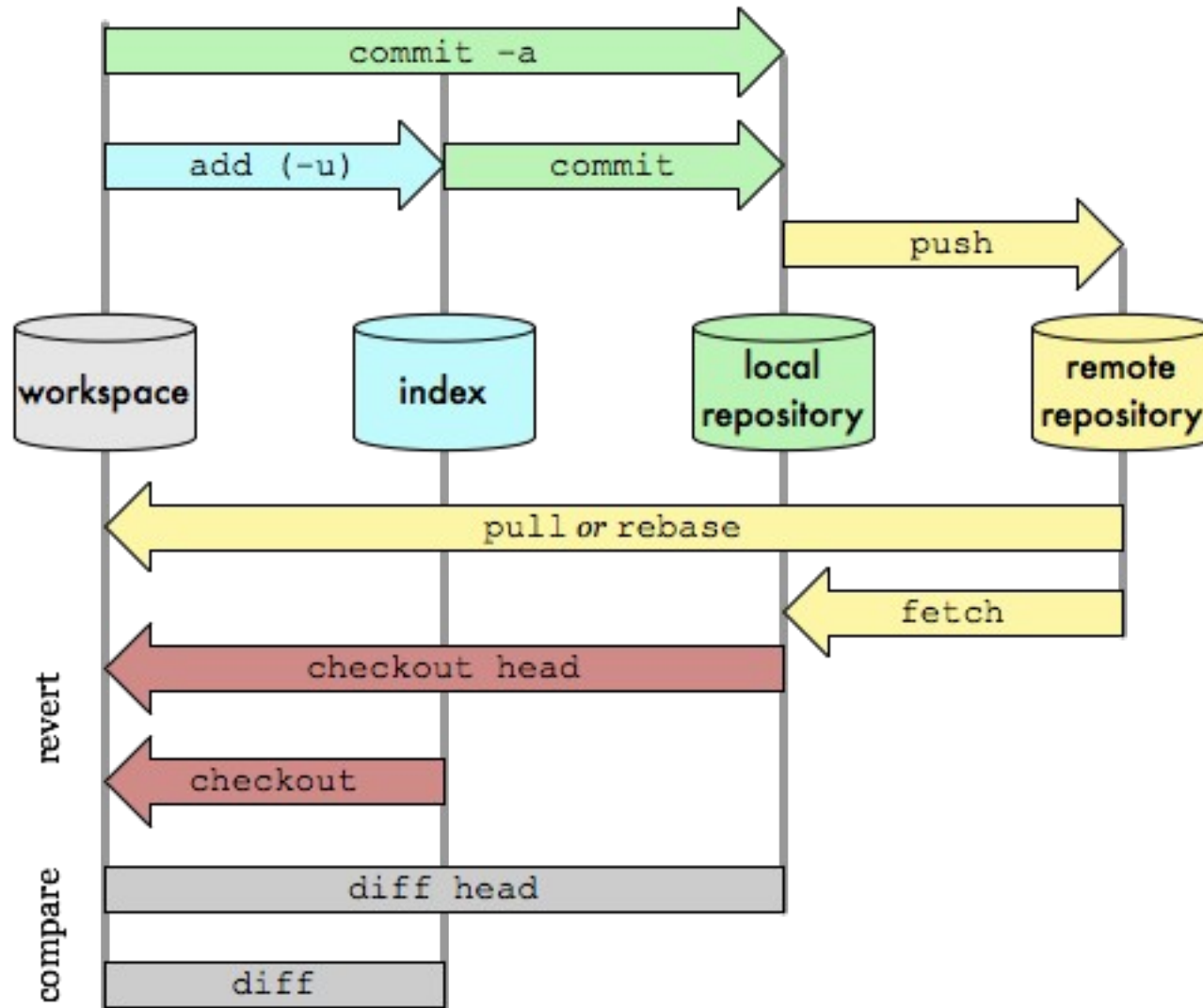
- Remote repository and local repository
- A repository has some branches, a branch has some commits.
- *origin* is the default remote repository name assigned to a repository that you create a clone from.
- *HEAD* is a keyword that refers to the most recent commit to the branch you're in.
- *master* is the name of the default branch in Git.
- *index* is a buffer between your working tree and what is stored in the repository.

Branches

- **Create a new branch: git branch**
- **Checkout a branch: git checkout**
- **Push a branch: git push**
- **Fetch a remote repository: git fetch**
- **Merge a branch: git merge**
- **Fetch and merge: git pull**
- **Rebase a branch: git rebase**

Git Data Transport Commands

<http://osteele.com>



Misc

- **git add**
- **git rm**
- **git mv**
- **git am**
- **git format-patch**
- **git send-email**
- **git grep**



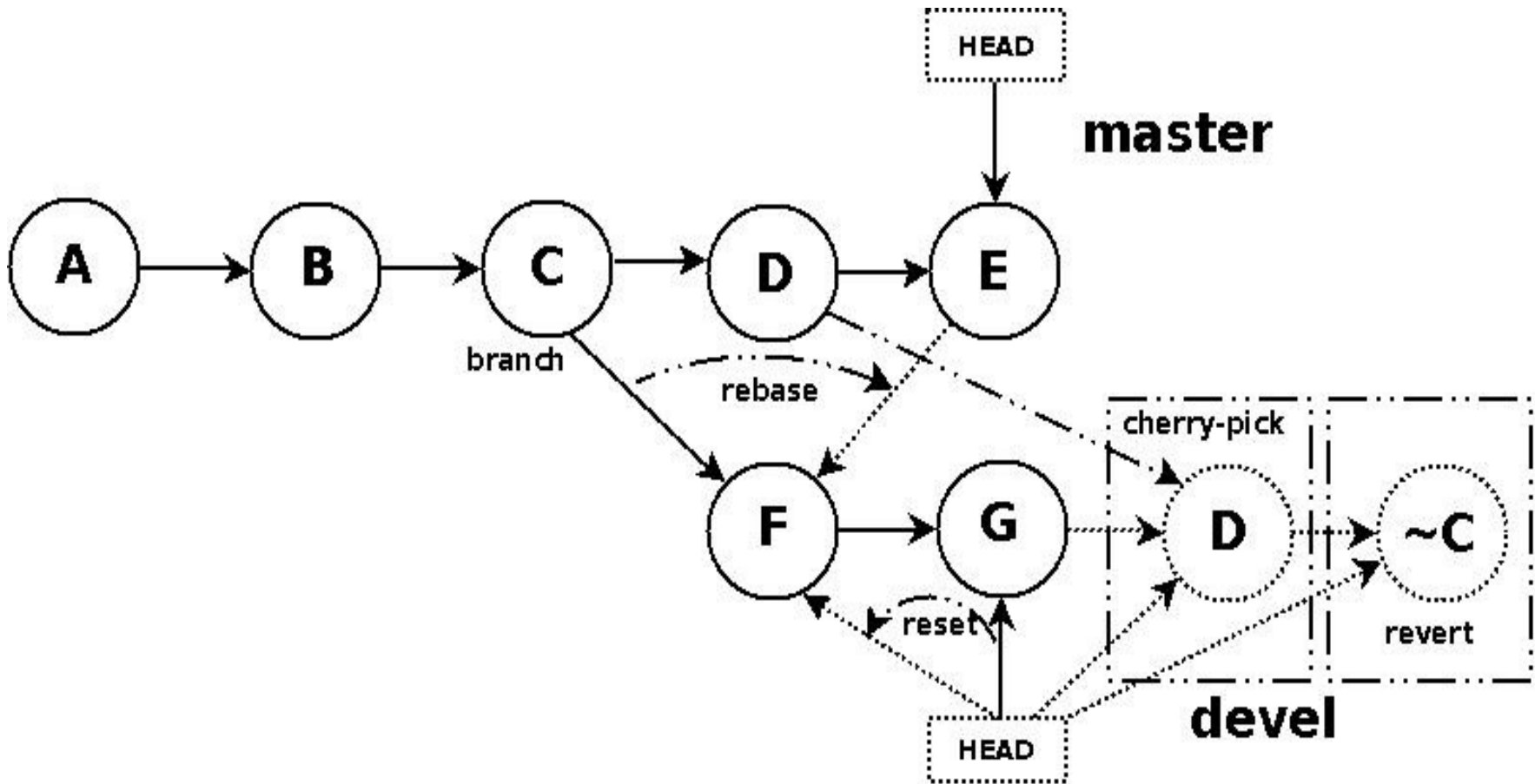
Git in the real world

Listing changes

- `git diff v2.6.33 v2.6.34-rc1`
- `git diff v2.6.33 v2.6.34-rc1 net/`
- `git log --pretty=oneline`
- `git log -M/--diff-filter`
- `git whatchanged`
- `git blame -L (replace git annotate)`

Undoing commits

- **git reset –hard HEAD**
- **git reset –hard ORIG_HEAD (for merges, resets)**
- **git reset --soft/--mixed**
- **git revert (-n)**
- **git checkout – path/to/file.c**
- **git commit –amend**
- **git rebase -i/--continue**



Q: Rebase VS Merge?

How to handle conflicts

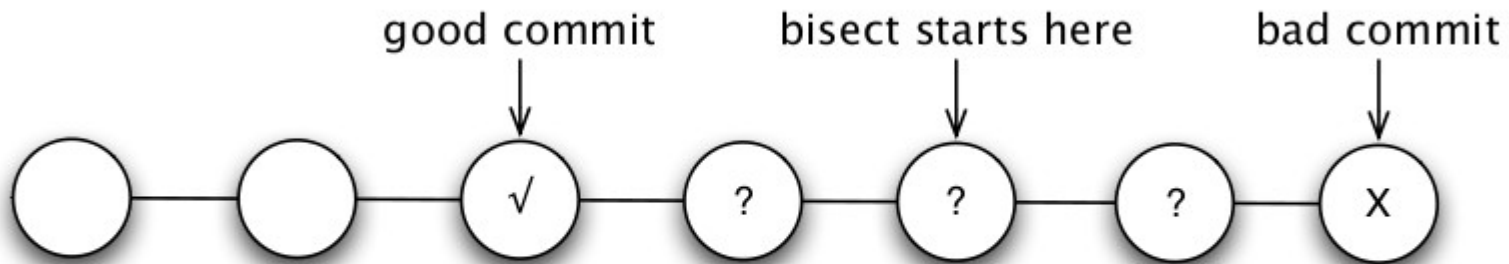
- `git diff`
- `git add`
- `git commit`
- `git log --merge --left-right -p`
- `git mergetool`

Using tags

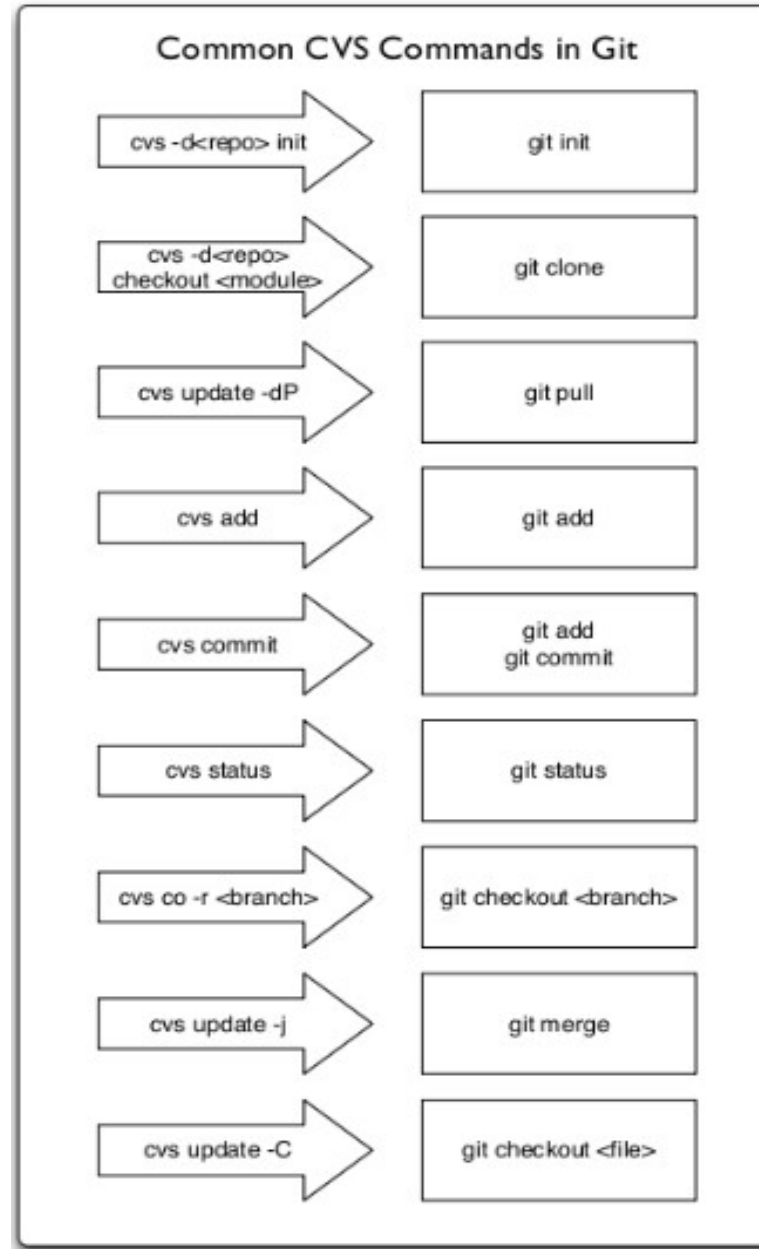
- `git tag -a v1.0 -m "GA version"`
- `git tag v1.0-beta beta`
- `git tag -l v2.6.31*`
- `git push origin --tags`
- Q: Tags VS branches?

Doing a bisect

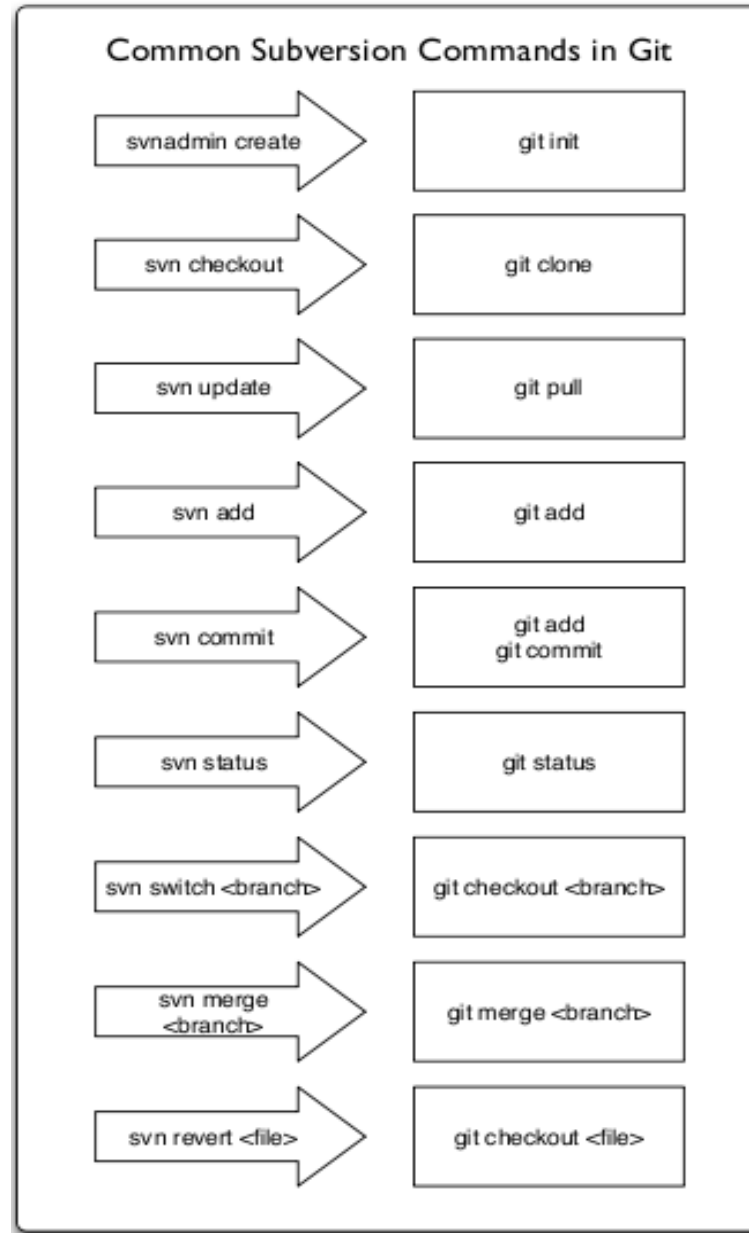
- `git bisect start`
- `git bisect bad`
- `git bisect good v2.6.31`
- `git bisect log/run/reset`



CVS to Git



SVN to Git



References

- *Pragmatic Version Control using Git*, Travis Swicegood, 2008.
- Git User's Manual:
<http://www.kernel.org/pub/software/scm/git/docs/user-manual.html>
- Git Cheat Sheet:
<http://ktown.kde.org/~zrusin/git/git-cheat-sheet.svg>
- Kernel Hackers' Guide to git: <http://linux.yyz.us/git-howto.html>

Questions?