

Git usage in general, gerrit, github interaction, bare git commands

Miklos Vajna

Software Engineer



Collabora
Productivity

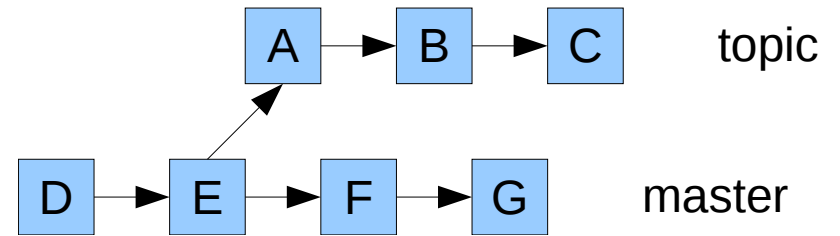
vmiklos@collabora.com

2024-07-12

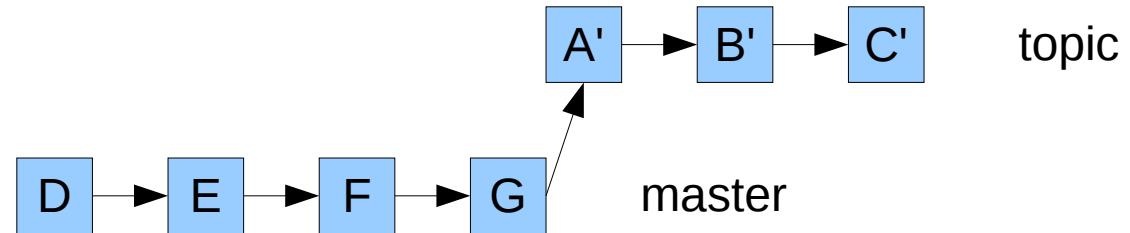
Rebase vs merge

If there is only one thing you would understand from this talk, then should be it

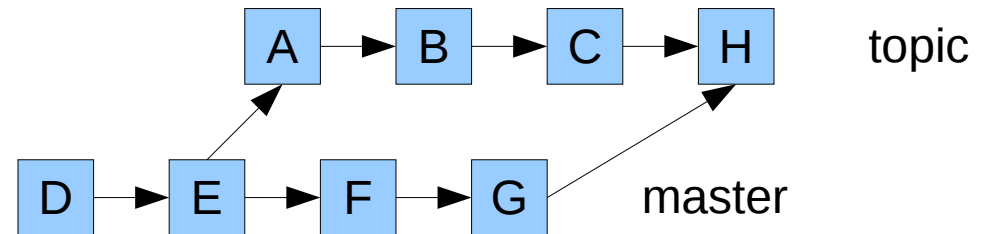
- At the beginning we have:



- Rebase:



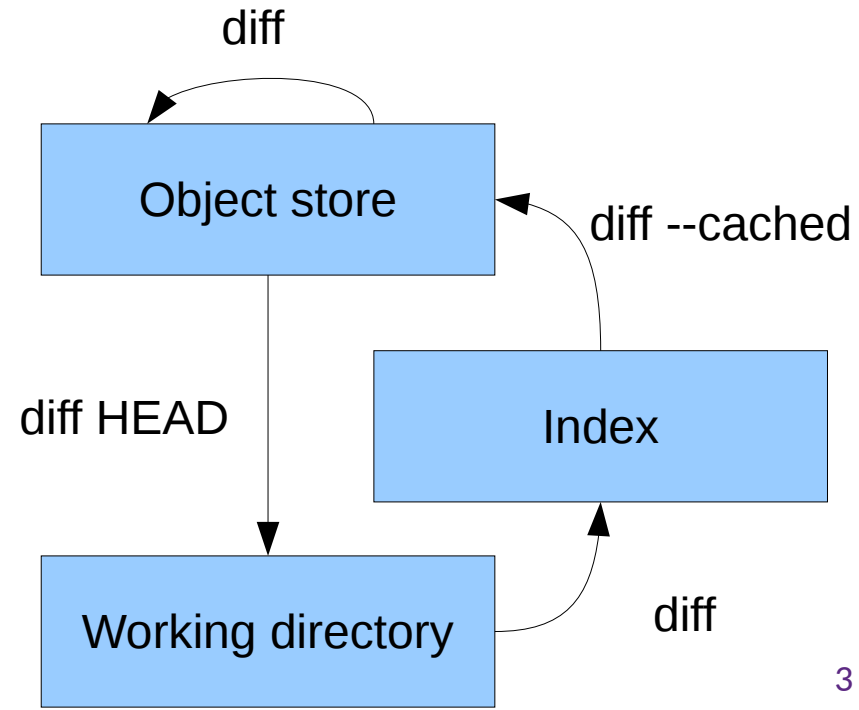
- Merge:



Diffing done properly

Problem: you need to understand what the index is

- Use-case: two changes to the same file, we want to commit only one of them
 - Solution: stage one file, not the other, commit (without -a)
- Use-case: conflict resolution, resolve conflicts one by one



Push tree

Problem: you want to push when your work is ready for review, but not pull

- A tree with no build in it that you can pull / re-base into to push
- This way you never have to pull & build in the middle of your productive hours
- <https://vmiklos.hu/blog/push-tree.html> has some details on this

Referenced clone, git worktree

Only clone core.git, online.git once

- Later: `git clone --reference core --origin logerrit --branch libreoffice-24-2 ssh://logerrit/core libreoffice-24-2`
- Or: `git clone --reference core --origin gitlab --branch distro/suse/libreoffice-7-0 git@gitlab.collabora.com:productivity/libreoffice/core.git suse-7.0`
- 'git worktree' can provide the same solution

Author, commit date

Author date is user input, not to be trusted too much

- If you like working by first creating a quick&dirty approach that works and then clean up using 'git commit --amend', then this will be the date of your first commit, a bit misleading
- Commit date: typically generated by Gerrit / GitHub
 - Guaranteed to be linear with our rebase workflow
 - This is closer to the “real date”, git log shows the author date by default
 - 'git config --global format.pretty fuller' to show both

core.git gerrit interaction

Manual vs helper on top of it

- Submit:
 - `git push origin HEAD:refs/for/distro/collabora/co-24.04`
 - Or 'git review' if you have it installed, works from .gitreview
 - 'git review -R' to avoid an automatic rebase
- Cherry-pick:
 - Fetch using the cmdline from the web interface, '...' → Download → cherry-pick gives: 'git fetch https://git.libreoffice.org/core refs/changes/71/170171/2'
 - Then checkout or cherry-pick FETCH_HEAD according to what you need
 - Or 'git review -x 170171'
 - Same, but checkout if you want to amend the change or create a new change on top of it without rebasing the change
- Best to avoid ./logerrit, only works in core.git, not in other gerrit repos

online.git GitHub interaction

Manual vs helper on top of it

- Submit:
 - `git push origin HEAD:private/nick/something`
 - Click on the printed URL to create the PR
 - Or `./g review` if you have `'gh'` installed
- Fetch:
 - If from a fork: `git remote add othernick git@github.com:othernick/online`
 - `git fetch othernick`
 - `git checkout -B othernick othernick/otherbranch`
 - can add commits and push back
 - can rewrite history (e.g. `git commit --amend`) and then `'git push --force-with-lease'`

Forked repo vs private branch

Only for the online.git / GitHub case

- Typical github style: fork the repo, create a branch there, submit PR
- core.git gerrit style was to not really work with explicit branches, just work with local master vs remote master (origin/master)
- Hack: private/<nick>/ prefix for the branch names: nothing to see there
 - It's assumed you'll create a PR once you're happy with the content
- That way no need to add lots of git remotes when fetching, just pull & cherry-pick
- Only works for committers

Gerrit review

The positive side

- 1 local commit gets turned into 1 remote change
 - Iterations of a change is a patch set
- No need to think about branch names before creating a PR
- Can review the commit message
- Can review each commit one by one
- Can cherry-pick on the web UI, if wanted

Cherry-pick between branches

Web vs local

- Benefit of web: if the change is trivial and the diff between branches is little, then this is fast
 - Only there on gerrit, not on github
- Benefit of local cherry-pick: can test / try out the change before sending out for review
 - Instead of hoping that automated tests are good enough to find if the result is broken

Staying up to date

How to learn what others did?

- Read the log on the web (poll):
 - <https://github.com/CollaboraOnline/online/commits/master>
 - <https://cgit.freedesktop.org/libreoffice/core/log/>
- Pull and then read 'git log' locally
- Email: subscribed to the libreoffice-commits@ mailing list (push)
 - <https://groups.google.com/g/collaboraonline-commits> would be the online side, though that's offline at the moment