UFR de **mathématique** et d'**informatique**
Université de Strasbourg

git-annex : a short introduction

Pierre-Antoine Senger

April 1, 2025

Overview

1. Context

2. Possible solutions

3. Installation and Usage

Context

Context

- **HPC** often requires large datasets
- **Git** is not designed for large files

Possible solutions

Keep the data locally

Works but not ideal, especially for:

- **Collaboration** (multiple users)
- **Reproducibility** (multiple runs)
- **Continuous integration** (CI)
- **Versioning** (multiple versions)
- **Backup** (multiple copies)

Avoidance

- **File generation** on demand or at execution time
- **Data reduction:** e.g. only store the most meaningful data
- **User specifies** which statistics/visualization to generate at each execution
- *etc.*

Cloud storage

Everyday user (e.g. Google Drive, Dropbox, OneDrive, etc.)
Targeted at scientific collaboration (e.g. Zenodo, Figshare, Dryad, OSF, etc.)
Problematics:

- **Requires lots of scripting** to integrate in the HPC workflow
- **Requires special security measures** to protect sensitive data
- **How to detect and manage errors?**
- **Compatibility issues**

Git LFS

Open source extension to Git

- Replaces large files with text pointers inside Git, while storing the file contents on a remote server
- No need for custom scripting
- **Consistent** between local and remote

Suboptimal usage of local storage:

- **Uses locally twice the space** (files are duplicated in `.git/lfs`)
- **Large files** are automatically downloaded when cloning a repo
- End users have nearly **no permission** on the remote server

Git-annex

Open source extension to Git

- Allows managing large files with Git without checking the file contents into Git
- Uses **symlinks** to optimize local storage
- **No duplication** of files
- No intrinsic limit on file size or bandwidth

More controls for the user:

- Can decide at anytime which files to keep locally
- Can use special command to download or drop files

Git-annex

Supports the download of large files content from either:

- Some other git-annex repository on another machine (provided there is a ssh connection possible)
- A cloud storage provider (e.g. Amazon S3, Google Drive, Dropbox, etc.)

Main drawback:

- **Non natively supported** by GitHub, GitLab, etc. (files needs to be managed with commandlines)
- **Learning curve**
- **Not as common** as Git LFS, support may be harder to find

Installation and Usage

Installation and Usage

- `sudo apt-get install git-annex` (Debian/Ubuntu)
- `sudo pacman -Syu git-annex` (Arch)
- `brew install git-annex` (MacOS)

Initialize and add a large file (single quotes are important):

```
git annex init 'PA laptop'
git annex addurl --file=large_file.zip download_url_link
git commit -m "Add large_file.zip"
git push origin main git-annex
git annex list
git annex whereis large_file.zip
```

Usage

Retrieve a file from another repository:

```
git annex init 'Alice laptop'
git annex get .
```

Annexing a new version:

```
git annex drop large_file.zip
git rm large_file.zip
git annex addurl --file=large_file.zip download_url_link
git commit -m "Update large_file.zip"
git annex sync
```

Usage

Retrieve the newer version:

```
git annex sync
git annex get .
```

The end

Thank you for your attention!

🎉

Any questions?

Contact Information:
pierre.antoine.senger@gmail.com
github.com/PA-Senger