Google

Mercurial at Google

Also known as Fig

Martin von Zweigbergk

2023-04-06

Background

About me

Martin von Zweigbergk

Ericsson

2004-2011

SIP telephony & IPTV
Drove migration to Git

Git contributor

2010-2013

Cleaned up `git rebase` code

Joined Google

2011-2014

Full-stack developer for compensation app

SWE on internal VCS

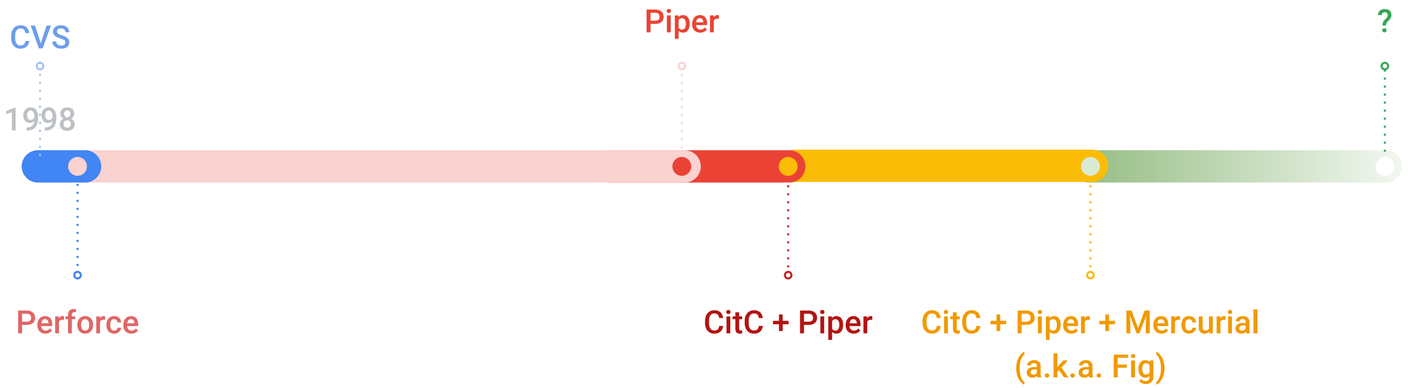
2014-now

Fig: Mercurial as a client for Google's internal monorepo

Background: VCS at Google

- Started with CVS, then Perforce
- Repo too large for Perforce ⇒ Piper was born
- Working copy too large for local disk ⇒ CitC was born
- Users wanted DVCS workflows (stacked commits) ⇒ Fig was born

For more info about the extreme size of Google's monorepo, see [Rachel Potvin's talk from @Scale]()



Workflow

Overview

A slice of the monorepo

- Users clone a slice of Piper (the monorepo) into a Mercurial repo
  - Sliced by both files and history
  - Files and commits outside the slice are not visible in the repo
- We use the `evolve` extension for better UX
- The evolve state is local (not exchanged)
  - But we create new obsmarkers locally when pulling a commit for a review that has been merged

Working copy

Restricted by narrow

- On local disk:
  - The user manages the set of tracked paths by narrow
- On CitC:
  - Automatically managed by Fig
  - Fig asks the file system which paths the user has touched

Code review

Review done in Piper

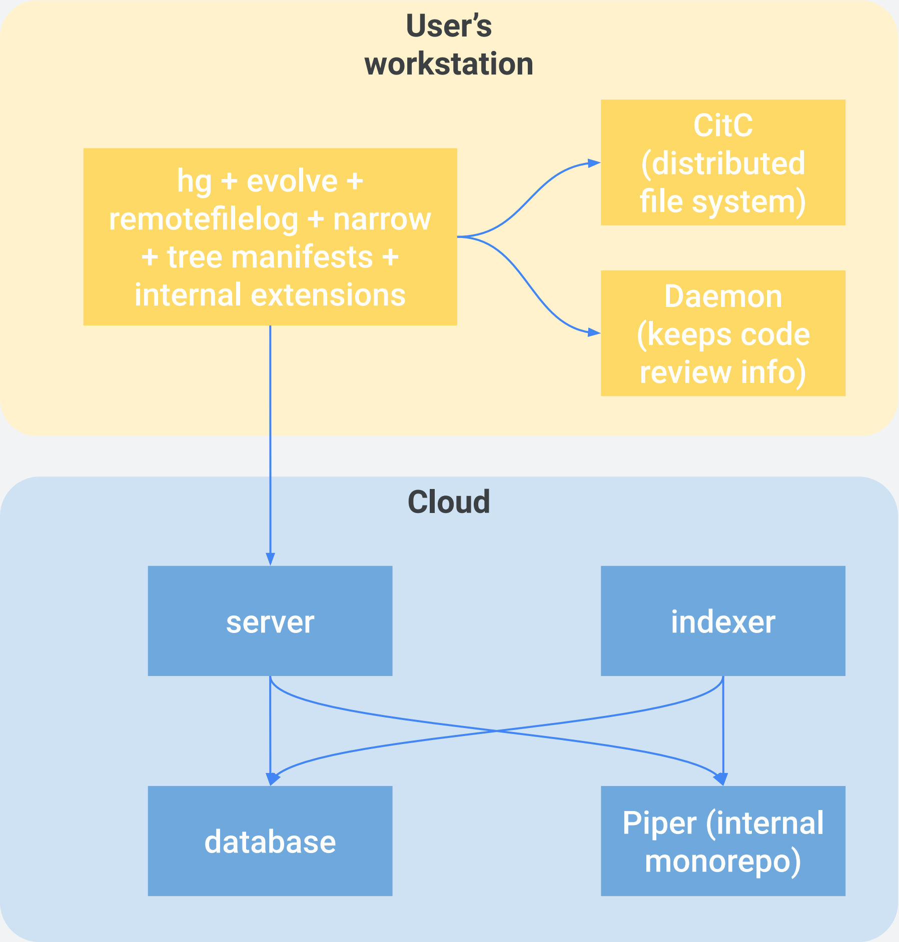
- Commits are uploaded to Piper as “changelists” for review
  - Once approved, the changelist gets “submitted” directly in Piper (“merged” in Heptapod-/GitHub-speak)
- Custom commands e.g. to:
  - Upload/export commits for code review
  - Fetch pending review as commit
- We have a custom `topics`-like extension for associating commits to changelists (N:1)

Architecture

Server

Server speaks hg's wire protocol

- Our server acts (mostly) like a regular hg remote
- The indexer runs continuously, indexing submitted Piper changelists as public commits in the database
- On clone/pull, server returns bundle matching paths from client (`narrow` extension)
- Server can also include pending changelists (unmerged reviews), converted to commits on the fly
- Server writes draft changelists in Piper on push
- Commits uploaded for code review contain extra metadata about mapping to Piper

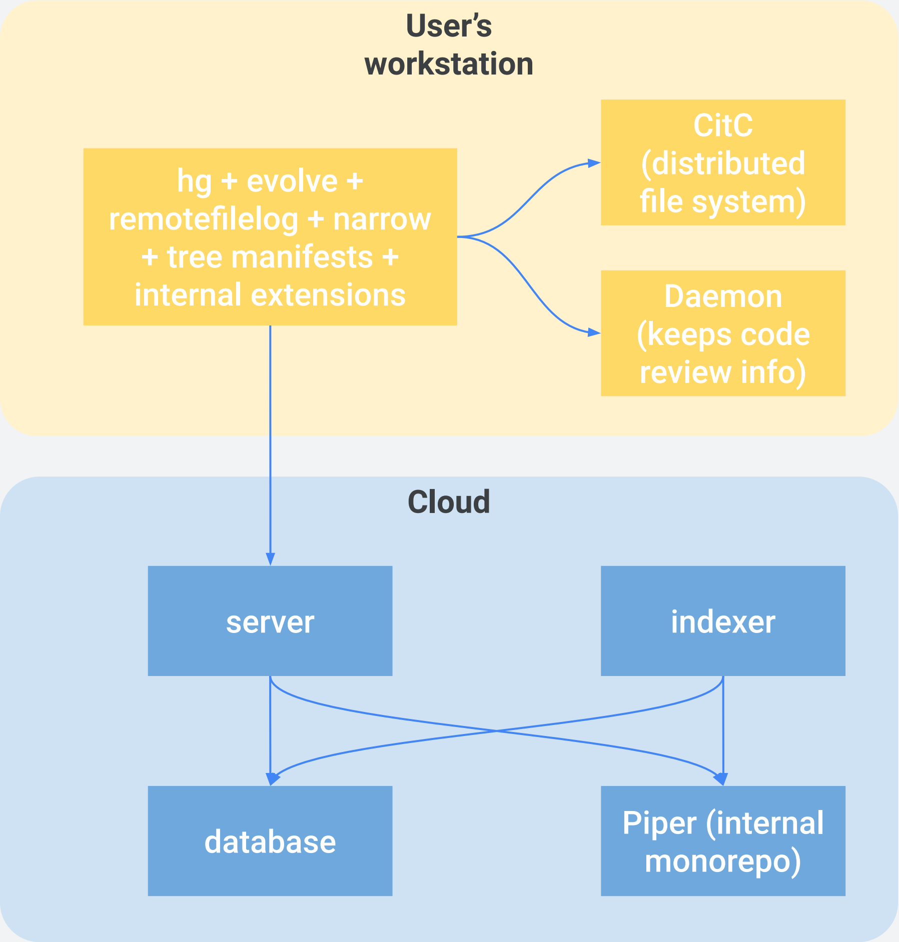

User's workstation
hg + evolve + remotefilelog + narrow + tree manifests + internal extensions
CitC (distributed file system)
Daemon (keeps code review info)
Cloud
server
indexer
database
Piper (internal monorepo)

Google

Client

Extensions for scalability

- Repos are stored in the file system
    - CitC (distributed) or local disk
- `remotefilelog` extension for fetching and storing file contents and for storing manifests
- `narrow` extension for slicing repo by paths and history
- Manifests stored using `treemanifests`
    - On CitC, manifests are created on the fly from the file system
- Custom extension for CitC integration


User's workstation
hg + evolve + remotefilelog + narrow + tree manifests + internal extensions
CitC (distributed file system)
Daemon (keeps code review info)
Cloud
server
indexer
database
Piper (internal monorepo)

Future plans

Current problems

- Performance (Python, eager/linear data structures do not scale to monorepo)
- Consistency (Mercurial is not designed for distributed storage)
- Integration (making APIs on top of a CLI)

Next steps

- Switching from hg to https://github.com/martinvonz/jj
- Moving repos from .hg/ to the cloud
  - Give illusion of having full repo locally

Thank you

Google

Email: martinvonz@google.com