CS262 Project: Distributed File System

Lauren Cooke, Patrick Thornton, Andrew Holmes

Idea

OneDrive

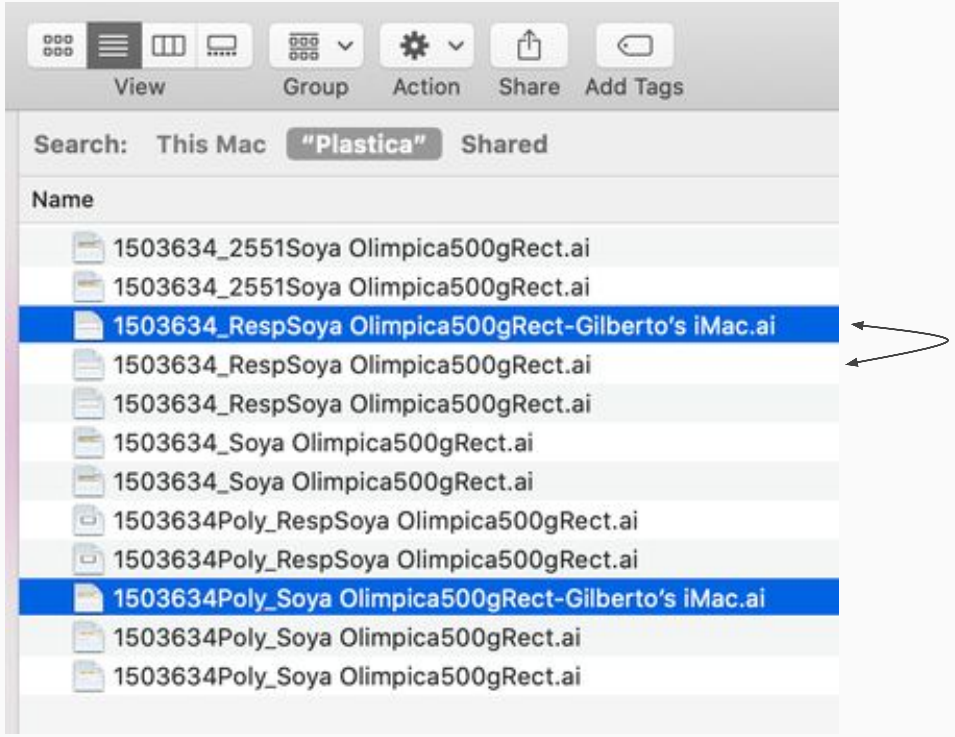
Core Features

- Whenever a file is saved locally, new version sent to server
- On login, latest versions of documents downloaded immediately
    - ‘Latest version’ determined via logical clocks, logging
- **Persistent**: external database
- ***n*-Fault tolerant**: *n* backup servers, take over upon failure

Data Race Problem

- If two or more users attempt to edit a file at once, we could…
  - **Lock down files**; only one editor at a time
  - OneDrive-style solution where **duplicate files** are created
  - Google Drive-style solution with live **concurrent editing**
- OneDrive-style solution chosen as best compromise

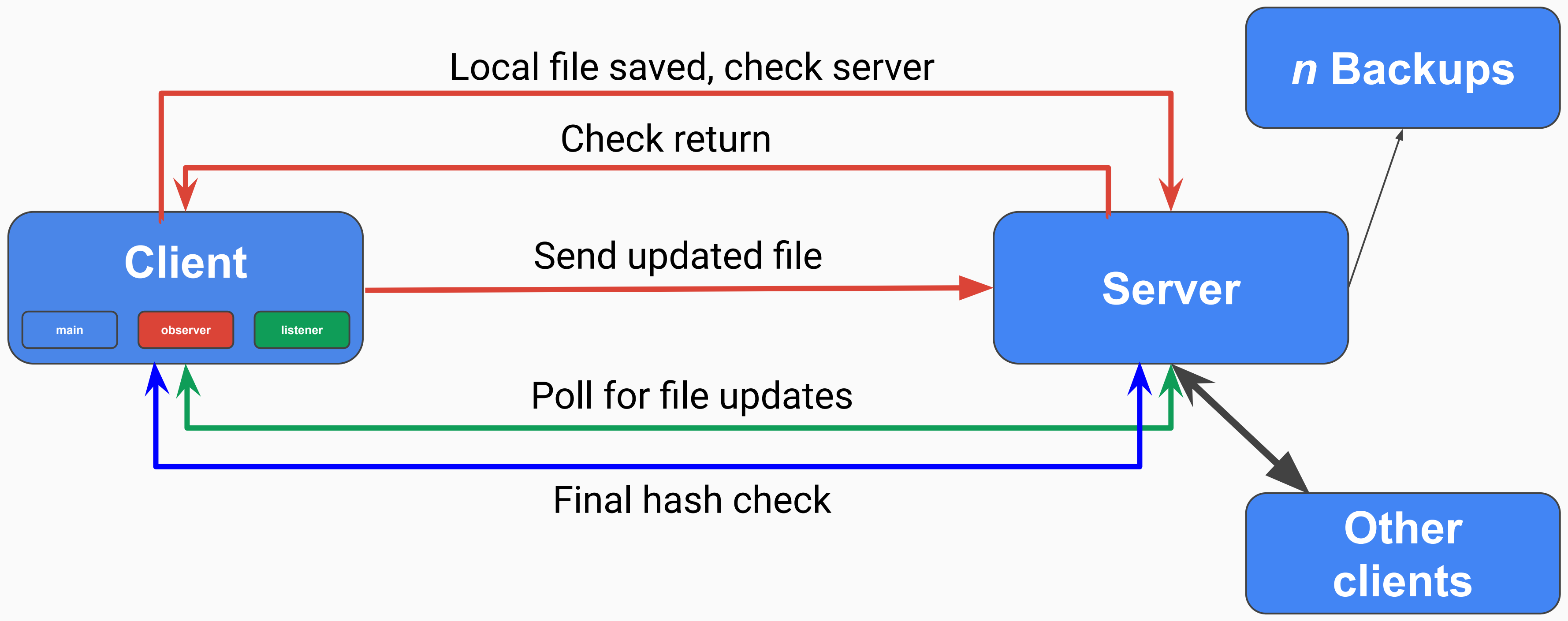
Data Race Problem



Search: This Mac "Plastica" Shared

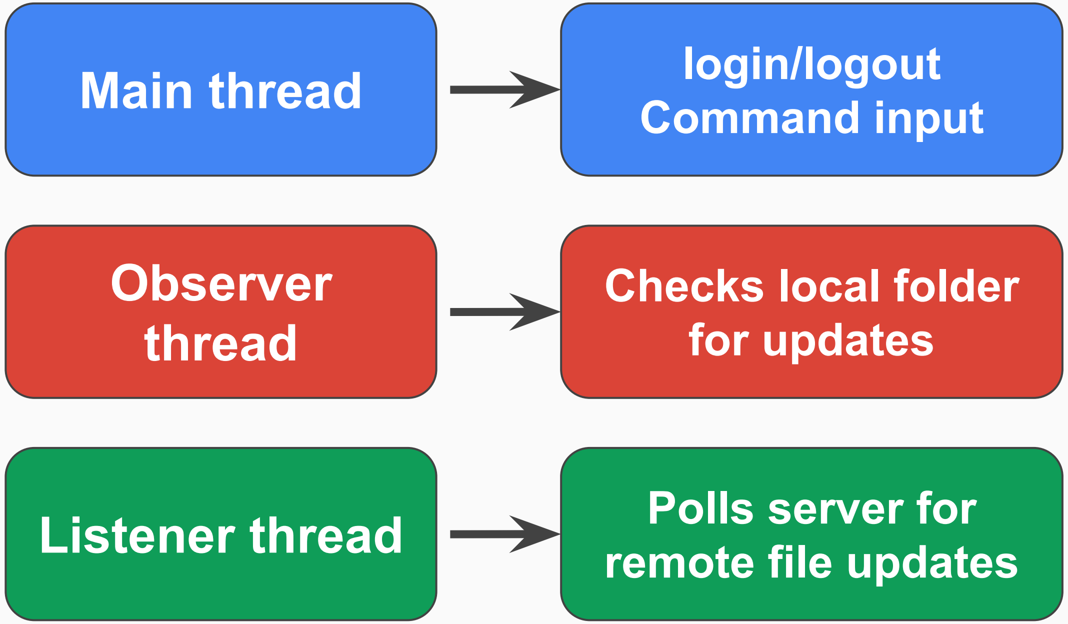
Name
1503634_2551Soya Olimpica500gRect.ai
1503634_2551Soya Olimpica500gRect.ai
1503634_RespSoya Olimpica500gRect-Gilberto's iMac.ai
1503634_RespSoya Olimpica500gRect.ai
1503634_RespSoya Olimpica500gRect.ai
1503634_Soya Olimpica500gRect.ai
1503634_Soya Olimpica500gRect.ai
1503634Poly_RespSoya Olimpica500gRect.ai
1503634Poly_RespSoya Olimpica500gRect.ai
1503634Poly_Soya Olimpica500gRect-Gilberto's iMac.ai
1503634Poly_Soya Olimpica500gRect.ai
1503634Poly_Soya Olimpica500gRect.ai

System Design



Client logic

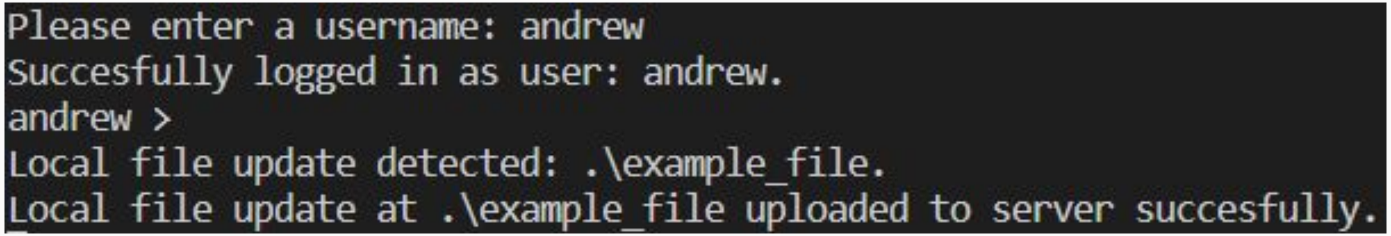
Client
main
observer
listener


Main thread
login/logout Command input
Observer thread
Checks local folder for updates
Listener thread
Polls server for remote file updates

Observer Thread

Observer thread

- Use library **watchdog** to monitor for file saves in certain folders
- On update:
  - Hash the file, check if hash is same as hash on server (i.e. no changes so don’t bother)
  - If different:
    - If previous editor MAC is same, save in
    - Else create branched version with modified filename



```
Please enter a username: andrew
Succesfully logged in as user: andrew.
andrew >
Local file update detected: .\example_file.
Local file update at .\example_file uploaded to server succesfully.
```

Listener Thread

Listener thread

- Poll server for file updates by checking file **hashes** and **clocks**
- If file with **different hash** on server, pull
- If branched version of local file, alert user to conflict

Technical Details

- gRPC implementation in Python
- Some files exceed gRPC's 4MB incoming message limit, so we made use of a **client- and server-streaming RPC**
- Files are **chunked** into **10KB** blocks and re-constructed on the other end
- Individual file size capped at **2GB**, allowing use of SQL **BLOB**s
- Use of **os** library to make directories, etc

Next Steps

- More work to be done on **persistence** and **fault tolerance**
- **Warning** the user when they enter a ‘data race’ or attempt to edit a file currently being edited
- **Additional client features**; listing available files, deleting files, sharing files

That is all.