DECO

Alberto Munoz & Matthew Ga

Introduction to the Project

- The project's structure is based off of the p2pchat lab, where users in a distributed network can send images to the network rather than messages.
- Users can use filters on the images with a set of pre-provided functions such as

frying and oil-painting.

- The messaging function of the original p2pchat lab has been entirely supplanted by images.
- The project was originally not a chat, but just a way to distribute photo editing among different computers to reduce the workload. But this was more fun.

demo

collection

THE ZUCC (NFT)

Limited Edition.
Please do not right click or screenshot.

This is a unique token in the blockchain worth thousands of dollars created by ~~con~~ artists making an honest living.

Mark Zuckerberg's Public Facebook Profile, 28 October 2021,
https://www.facebook.com/zuck/posts/10114027736704991

Moraine Lake

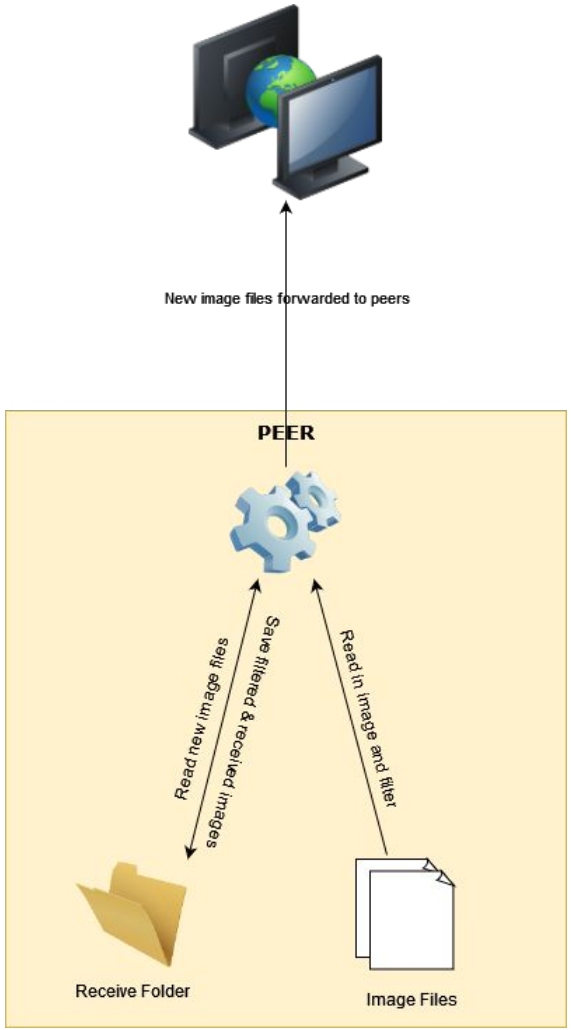
Oil Paint on Canvas, 2021

"Moraine lake, Alberta, Canada, sunrise 2019 banff lake louise," Chensiyuan, 18 June 2019, https://en.wikipedia.org/wiki/Moraine_Lake

what concepts did we use?

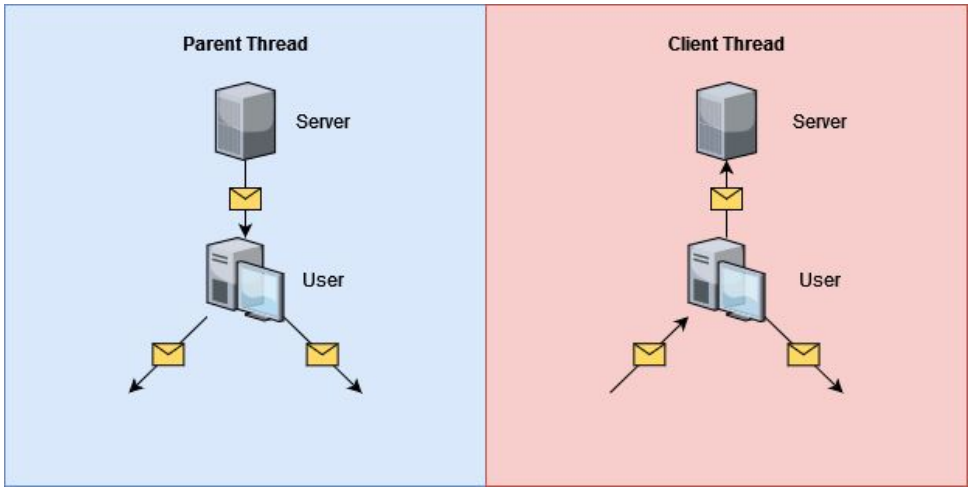
Files & Filesystems

- In order to send and receive files from many users, we wrote image files to a given directory.
- To receive files, there are 4 steps:
  1. Directories are changed to the folder meant to receive files.
  2. A file is read from a file descriptor corresponding to a socket using read.
  3. A new file is created & written and the data written to that file.
  4. The file mode is changed so that it could be read by the user using chmod.
- The resulting file is then forwarded to other peers.



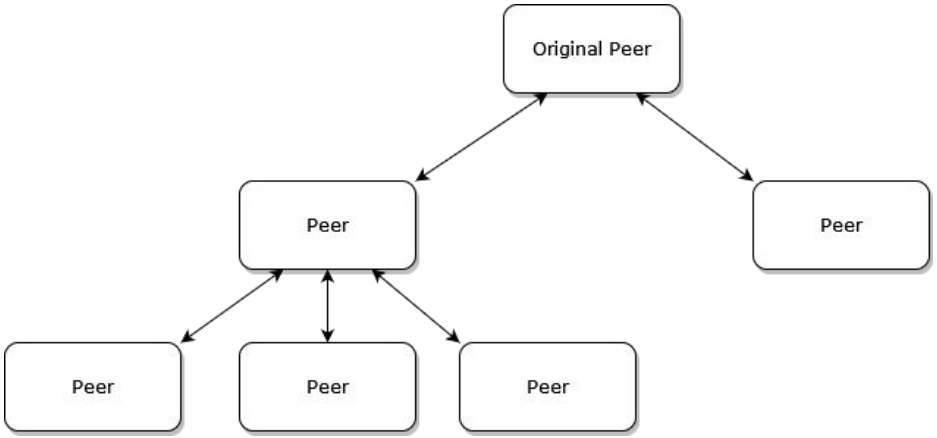
Threads & Synchronization

- The program utilizes multiple threads to handle connections & sending/receiving images.
- It utilizes locks to update the global client list to avoid serious bugs with connections since individual clients are handled by different threads.



Distributed Systems

- This project utilized major elements of p2pchat that included TCP socket connections
- With sockets, the network structure is identical to that of p2pchat
- Combining this with threads & synchronization and linked lists, we created a dynamic network capable of global file-sharing



How do files work?

- Rather than sending messages, you must type in a path to an existing file
- To run the program, specify a path to an existing directory that will receive images sent by other users in the network (preferably an empty directory specifically for this program)
- Typing in a path to a valid image (.png, .jpg, .jpeg) will write it to all other users in the network
- Use ‘f’, ‘s’, ‘o’, ‘i’ in any sequence after the file path to sequentially apply filters.
- If a filter is applied, a local copy is saved to your receiving folder as “xuser.png,” where x is the number of images forwarded by the user.
- Images received by others are marked as “clientx_y.png” where x is the client’s file descriptor and y is the number of images sent by that client.

live

demo

How to:

- The project files are online on GitHub: https://github.com/munozlun/deep_fry
- You can then download the executable ‘p2psnap’, which should work on the lab computers.
- Make sure to run the command ‘chmod 700 p2psnap’ to make the file executable.
- Finally, all you need to do is choose a directory to save images to and run it! (using ‘mkdir receive_folder’ in the same directory as p2psnap is helpful & convenient)
- Enjoy!

CREDITS

http://www.graphicsmagick.org/api/api.html

Charlie Curtsinger

https://www.makewordart.com