Lab 4 Design

Corey Quon - 803 786 588

Jonathan Nguy - 603 799 761

Current Implementation

- get request
- check tracker for peers and create a list
- go through list and connect
- open disk file
- read from peer to disk

New Implementation (Edits)

- after we create the list,
  - divide the task based on peer number
    - go through list and connect to peer for segment
- after opening disk for result
  - read file into task buffer from peers
    - each peer responsible for certain section

Splitting Work

- divide each peer's work evenly depending on size
  - 512 bytes, 8 peers = 64 bytes/peer
- use pthreads to split work

Going Through List

- open sockets to each peer for certain segment
  - change original code to open multiple sockets
    - thus, allowing simultaneous download

Reading File

- each thread:
  - is able to read into the task buffer
  - knows what to contribute to downloaded file

Problems

- performance issues based on download strategies
  - peers should be favored if they are faster
    - could calculate this from previous tasks
    - use info to determine which peer should send more

Summary

- began the design, but not completely implemented
- data integrity would be a challenge
- parallel is usually better