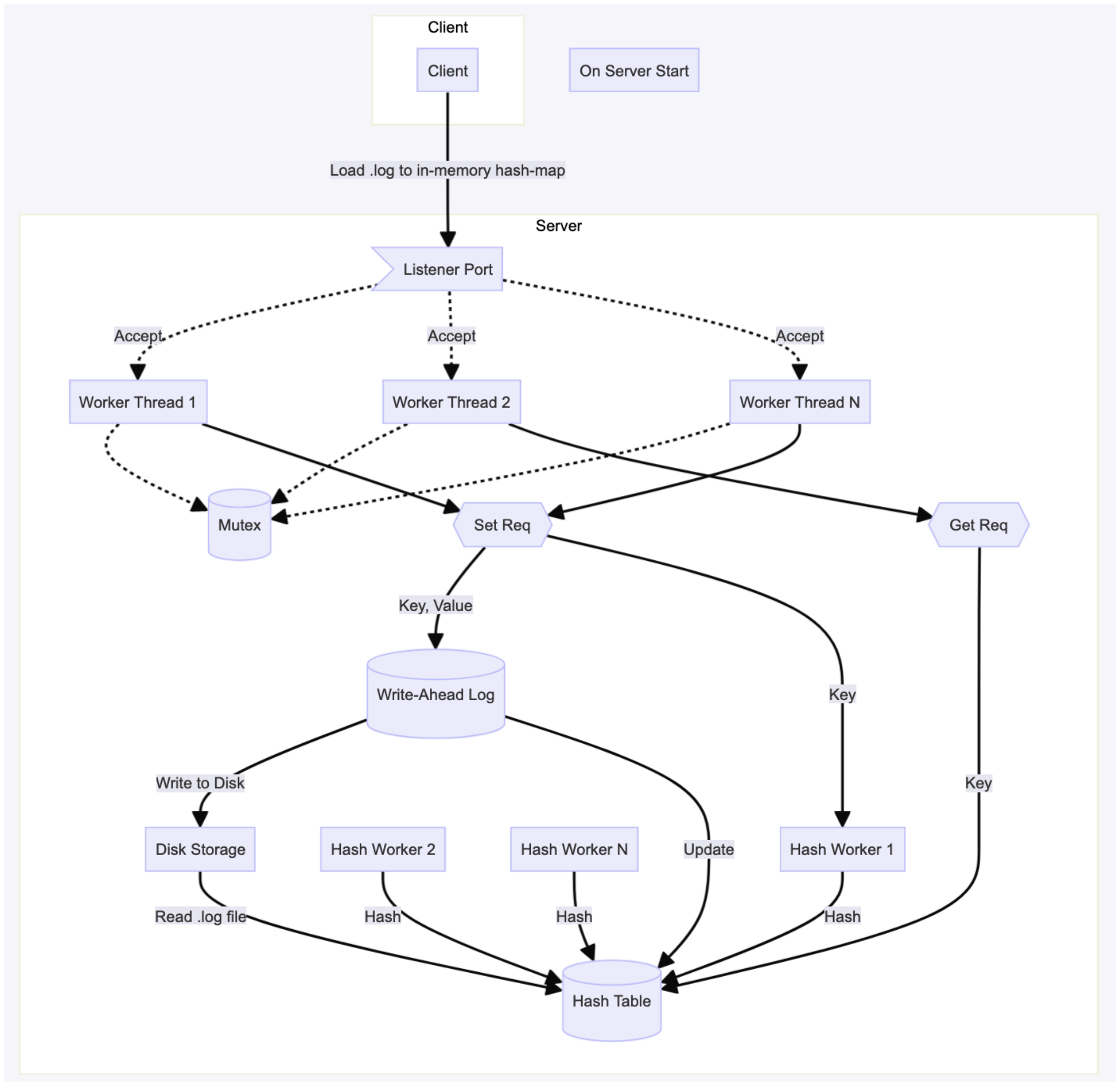
afaiz_hw1

February 5, 2024

0.1 Memcache-lite

0.1.1 Architecture



0.1.2 Files

1. server_mt.c : tcp concurrent multithreaded server for k-v store
2. client.c : client for connecting to a sock_stream(tcp) server

3. hashmap.c : implementation of hash map functions like get/set/delete/init
4. threadqueue.c: implementation of global shared queue for thread enqueue/dequeue

0.2 Multithreaded TCP Key-Value Server

0.2.1 Overview

This server implements a key-value store using a multithreaded architecture, in-memory hash table, and write-ahead logging for persistence. It supports set and get operations for key-value pairs.

0.2.2 Key Features

- Multithreading
  - Separate thread pools for handling client connections and hash table operations
  - Increased throughput and responsiveness under load
- Write-Ahead Logging (WAL):
  - Durability by logging set operations to a persistent file before updating the in-memory hash table
  - Enables recovery of data in case of server restart/crash
- In-Memory Hash Table:
  - Efficient storage and retrieval of key-value pairs
- Thread-Safe Data Structures:
  - Mutexes and condition variables protect shared data structures from race conditions

0.2.3 Data Structures

- Hash Table:
  - External chaining in case of collisions
  - Stores key-value pairs as kv_node_t structs
- Thread Queues:
  - Linked lists (threadqueue.h and hashmap.h) for client connections and hash table operations Used for thread communication and synchronization.
- Condition Variables:
  - Signal threads when new work is available on queues

0.2.4 Compilation

- Ensure required libraries are installed: pthread and readline.
- Compile with Server Side:

```
clang -c server_mt.c
clang -c hashmap.c
clang -c threadqueue.c
clang server_mt.o hashmap.o threadqueue.o
```

Client Side:

```
clang -c client.c
clang client.o
```

0.2.5 Execution

- Run the server with ./server
- Clients can connect on port 4096 (configurable in PORT)
  - Example: set A 10 abcde
  - Example: get A

0.2.6 Usage

- Set a key-value pair:
  - set key value_size value
- Get a value by key:
  - get key

0.2.7 Additional Notes

- Server logs set operations to server_run.log.
- Hash table is initialized with data from the log file on server start.
- Consider implementing features like deletion and expiration for a more complete key-value store.

0.2.8 Results

Results for a series of get/set requests are present in run_test1.txt and run_test2.txt

The individual test contains:

run_test1.sh

```
#!/bin/bash

./client set A 20 sdgjhgdgd
./client set A 20 854uV54wz50544354dg
./client set B 20 6554k54315444544weg
./client set C 20 4555445445540544egd
./client set D 20 4554544554445445dfdf
./client set E 20 5544554455445544v
./client set kk 40 sdgnkfjbgkn
./client get A
./client get P
./client get kk
./client get A
./client set C 20 dgsjdgnkjndg
./client set R 40 lsdghkjdsghboijfkngzdough
./client set S 40 lsdghkjdsghboijfkngzdough
```

run_test2.sh

```
#!/bin/bash

./client get A
./client get P
```

```
./client get kk
./client get A
./client get D
./client get kk
./client get E
```