BlinkDB - Benchmarking

Generated by Doxygen 1.13.2

Chapter 1

Class Index

1.1 Class List

Here are the classes, structs, unions and interfaces with brief descriptions:

Chapter 2

Class Documentation

2.1 BlinkDB Class Reference

A simple in-memory key-value store with basic persistence.

Public Member Functions

- BlinkDB ()

  Constructor that clears any existing AOF file.
- ∼**BlinkDB** ()

  Destructor that removes the AOF file upon shutdown.
- void set (const string &key, const string &value)

  Stores a key-value pair in the database.
- string get (const string &key)

  Retrieves the value associated with a key.
- void del (const string &key)

  Deletes a key-value pair from the database.

2.1.1 Detailed Description

A simple in-memory key-value store with basic persistence.

BlinkDB provides thread-safe operations for setting, getting, and deleting keys. It writes all operations to an append-only file (AOF) for durability.

2.1.2 Constructor & Destructor Documentation

2.1.2.1 BlinkDB()

```
BlinkDB::BlinkDB ()  [inline]
```

Constructor that clears any existing AOF file.

Removes the AOF file at startup to simulate a fresh database instance.

2.1.3 Member Function Documentation

2.1.3.1 del()

```
void BlinkDB::del (
            const string & key)  [inline]
```

Deletes a key-value pair from the database.

Parameters

key	The key to delete.

If the key exists in the in-memory store, it is removed and the deletion command ("DEL") is appended to the AOF file for persistence.

2.1.3.2 get()

```
string BlinkDB::get (
             const string & key)  [inline]
```

Retrieves the value associated with a key.

Parameters

key	The key to retrieve.

Returns

The value associated with the key, or an empty string if the key is not found.

This function is thread-safe and looks up the value in the in-memory store.

2.1.3.3 set()

```
void BlinkDB::set (
             const string & key,
             const string & value)  [inline]
```

Stores a key-value pair in the database.

Parameters

key	The key to store.
value	The value associated with the key.

This operation is thread-safe. It updates the in-memory store and appends the corresponding "SET" command to the AOF file for persistence.

The documentation for this class was generated from the following file:

- main.cpp

2.2 BlinkServer Class Reference

Implements a TCP server that handles multiple client connections using kqueue.

Public Member Functions

- **BlinkServer** ()

  Default constructor initializes server file descriptor and kqueue descriptor.
- void run ()

  Starts the server.

2.2.1 Detailed Description

Implements a TCP server that handles multiple client connections using kqueue.

BlinkServer listens on a specified port and accepts incoming client connections. It processes commands encoded in the Redis RESP-2 protocol (SET, GET, DEL) by interacting with an internal BlinkDB instance. The server uses kqueue for asynchronous event handling.

2.2.2 Member Function Documentation

2.2.2.1 run()

```
void BlinkServer::run ()  [inline]
```

Starts the server.

This function sets up the server socket, initializes kqueue, and enters the main event loop.

The documentation for this class was generated from the following file:

- main.cpp

2.3 ClientState Struct Reference

Maintains the read and write buffers for a connected client.

Public Attributes

- int **fd**

  File descriptor for the client's socket.
- string **read_buffer**

  Buffer to store data read from the client.
- string **write_buffer**

  Buffer to store data to be written to the client.

2.3.1 Detailed Description

Maintains the read and write buffers for a connected client.

This structure stores the state for a client connection including the socket file descriptor, a buffer for incoming data (read_buffer), and a buffer for outgoing data (write_buffer).

The documentation for this struct was generated from the following file:

- main.cpp

Index