

## DESIGN DOCUMENT - PART B

### SERVER

- My Server uses **kqueue()** function call for efficiently handling multiple clients using I/O multiplexing.
- Created separate parser for handling serialization and deserialization of Redis RESP-2 protocol messages.

### CONFIGURATION

- Server uses port **9002**.
- Maximum number of events that can be retrieved in one call to kevent() is **1024**.
- Client request buffer size is set to **1024** characters.

### BENCHMARK CHOICES

- *redis-benchmark* utility provides keys as **key:<random\_integer>** and same value for all **set()** requests.
- Hence I created a custom benchmark data creator which generates random random strings of **length 16** for keys and values.

- Random keys and values are created and stored before the benchmarking starts and are auto-populated to each client get() and set() request on a random choice.
- This helps the benchmark test more realistic to the real-world scenarios.

## BASH SCRIPTS

1. *benchmark\_script.sh* - Performs set() and get() requests serially for each of the 9 benchmark tests.
2. *benchmark\_script\_v2.sh* - Performs set() requests for all the 9 benchmark tests followed by all the get() requests.