

Instructions to run the code

1. Install the jsoncpp library for reading .json files in C++ using the command `sudo apt-get install libjsoncpp-dev` in ubuntu.
2. Edit the configurations.json file.

Following are the parameters in the configurations.json:

- Number_of_runs : Total runs in the simulation
 - Number_of_cores : Total cores in the web server
 - Number_of_threads : Threads count in the web server
 - Buffer_length : Maximum length of the request buffer
 - Scheduling_type: Scheduling algorithm type 1 - FCFS, 2- Round robin algorithm.
 - Quantum_time: Time slice in round robin scheduling algorithm
 - Context_switch_overhead: Overhead time for doing context switch from one process to another process
 - Number of users: Total users in the request-response loop
 - Total_delays: Total number of requests processed to stop the simulation in each run
 - Mean_service_time: Average service time of a request
 - Service_time_distribution: 1 - constant, 2 - uniform, 3 - exponential
 - Thinktime: Time a user waits to send a request after receiving a response or timeout
 - Thinktime_distribution: 1 - constant, 2 - uniform, 3 - exponential
 - Timeout_time: variable component of the timeout
 - Timeout_minimum: Minimum time out for each request
 - Timeout_distribution: 1 - constant, 2 - uniform, 3 - exponential
3. To compile the C++ program use the command `g++ -o main main.cpp -ljsoncpp`
 4. To run the code use the command `./main`

Links of google sheets for Response time confidence intervals calculation

- [!\[\]\(467d80e979964f7f8c752fb22248b5b7_img.jpg\) ResponsetimeM10](#)
- [!\[\]\(b71552d33dbf62adf5e5199a70ee02bf_img.jpg\) ResponsetimeM20](#)
- [!\[\]\(03134b765d1473836ff001925b1b0550_img.jpg\) ResponsetimeM30](#)
- [!\[\]\(aed6947356668967079310026052edc0_img.jpg\) ResponseTimeM40](#)
- [!\[\]\(e61aeb0d9066d5d9e54d9b655f50da3d_img.jpg\) ResponsetimeM50](#)
- [!\[\]\(f7af41ce0777e13bda91fa715111c02a_img.jpg\) ResponsetimeM60](#)
- [!\[\]\(476ddb2354d4ad1cb23a2236b1e49873_img.jpg\) ResponsetimeM70](#)
- [!\[\]\(1d505a46c82c5cefa23b88c2eee900ce_img.jpg\) ResponsetimeM80](#)
- [!\[\]\(3a98690f11ee4baf67262bd776464219_img.jpg\) ResponsetimeM90](#)
- [!\[\]\(35522fe6386206890679adb7b63391b6_img.jpg\) ResponsetimeM100](#)
- [!\[\]\(d28d4a3445dac344f03b5cebc14c5170_img.jpg\) ResponsetimeM110](#)
- [!\[\]\(3e37ae08976ee7fa41b108254fcb66a7_img.jpg\) ResponsetimeM120](#)
- [!\[\]\(7b30e10e474a15019e378034a5556dd2_img.jpg\) ResponsetimeM130](#)
- [!\[\]\(be2bdf77bab097eb6ddf17878ba7ec4d_img.jpg\) ResponsetimeM140](#)
- [!\[\]\(a3b6961c19ef9a7399ba4d220fbe1b94_img.jpg\) ResponsetimeM150](#)
- [!\[\]\(f8936a35f239803f29013161729262d8_img.jpg\) ResponsetimeM160](#)
- [!\[\]\(1450f2e803fc906eeaaad04363880ce9_img.jpg\) ResponsetimeM170](#)
- [!\[\]\(e1133a641f38be314cd75f50ed4924a6_img.jpg\) ResponsetimeM180](#)
- [!\[\]\(64b8231e633e9e03e4b21734881f56cf_img.jpg\) ResponsetimeM190](#)
- [!\[\]\(aee5e761ba1b99a3cd9435e499b75672_img.jpg\) ResponsetimeM200](#)