CS 7345 ADVANCED APPLICATIONS: LAB 4 NETWORKING

Overview: In this lab you will utilize Emscripten to convert a networked c++ library/application into a JS library using WebAssembly (WASM) and build a demo application. After conversion, evaluate the library performance.

Due: November 22, 2022 @ 11:59pm.

Code Requirements:

1. Identify code base to convert to a JS library .
    a. Any code base can be used that meets the requirements below
        i. Preferred code base should be related to your research or personal work, but that isn't a requirement, an example project will be provided to convert as a backup if nothing else can be found.
        ii. An existing c++ code base can be used that is publicly available and doesn't currently have an emscripten port library port
        iii. An existing custom library/program you have previously created (algorithms, search, math lib, etc) can be used, but all code must be included with lab submission.
2. The code base from Lab 3 can be (and probably should be) utilized, and make sure the library has a well defined interface.
    a. Interface should be well designed for usability and efficiency
3. Create a test application to demonstrate your JS library (demo application).
    a. Your application client can either run in NodeJS or in a browser.
        i. Most likely you will want your client to run in browser if you are utilizing the Emscripten c++ WebSockets API, but this is not an absolute requirement and Emscripten compatible networking approach can be used, node can be used for your library as long as it is tranpiled from c++.
    b. The application will require a server, which should not run in browser, but a c++ or node server can be used. The server does not have to be transpiled, but it can be.
    c. Application should test all features of your library and provide feedback to user on each feature and its use.
    d. Application should use clients as a way to process work, and could also be used as way to submit work, but that is not a requirement. Networked clients should be thought of as worker threads in Lab 3.
    e. The server should coordinate work being processed, similar to how the main thread controlled worker threads in Lab 3.

f. Application can be designed as either a technology demo or a product demo for your library (depends on selected code base). It should fully demonstrate and make all features understood to user from its operation.
g. Create ability for application to output performance data

4. Create comparison version for test application (comparison app).
    a. This application can be a native JS or C++ application.
        i. You can select which language will work better for your comparison.
        ii. You only need 1 language to be used for comparison, not both.
    b. It will be the comparative performance measure for the networked library
        i. You need to determine what features to compare for your project. Special attention should be paid on how you select and test your applications. More credit will be weighted in this assignment on your test plan, test data and how it is presented.
    c. Create ability for application to output performance data.
5. Extra Credit: Attempt to improve performance of WASM code, if possible, by modifying native and/or JavaScript code. Compiler optimizations are not enough, you must experiment with various strategies or techniques to see how they effect overall performance.

Report Requirements:

1. Document API and Networking features for library
    a. Provide developer documentation for the API explaining each endpoints use and functionality. Be sure to explain fully the input/output for method, expected outcomes as well as any assumptions that are made about its use and/or data requirements.
    b. Provide a second section that describes the design concepts of the networking architecture and execution model for library. Be sure to include explanations on any assumptions made in design with a pros/cons descriptions about those assumptions/features. Be sure to describe in detail any design patterns, class structure and provide explanations, pro/cons for design decisions and implementations.
    c. High-level class layout/UML should be included in writeup
2. Compare and Contrast the performance of the library code between demo application and comparison application
    a. Perform a timing analysis for each code base
        i. You need to determine what features to compare for your project. Special attention should be paid on how you select and test your applications. More credit will be weighted in this assignment on your test plan, test data and how it is presented.
        ii. Your analysis should be performed over multiple executions of the code base
            1. Do not compare off a single execution of each code base
        iii. Show confidence intervals for execution time at 95% confidence interval and discuss if the results are statistically significant

b. Extra Credit: perform same analysis above on any optimizations performed on code base.

Submission should reside in the Lab3 folder, with all code in the "Code" folder, raw data files generated in "Data" and your final report in "Report" folder in the git repo.