Assignments for Distributed Computing CS G557: Scalable Service Implementation

SANTONU SARKAR (PI)

February 9, 2024

Instructions

The latest documentation for docker container is available here: `https://www.docker.com/` We advise that you spend time in learning docker usage, download the docker engine and create your own container where you can run your own application. Common Instructions:

1. Install Docker in your computer and learn the basics
2. Try running a simple application as a service, i.e. it should be awake all the time, waiting for a user-input. For instance, try implementing a simple calculator function that takes two numbers, adds them and returns it back. Then it waits for an user input.
3. Run such an application inside a container. Try calling the service (the addition service) from outside, from another “master” application. This application should call the containerized addition application by supplying two numbers, get the result and display/print it. This “master” application can be a simple terminal application or an web-based application, or a containerized application, it’s your choice.

1 Assignment: Compute $\pi(x)$ in a load-balanced manner

The Sieve of Eratosthenes(ca 240 BC) is an iterative and efficient way to find small primes less than 10,000,000. This technique helps in computing $\pi(x)$ which computes the number of primes not exceeding x. You need to implement $pi(x)$ service which will return for example $\pi(29,996,224,275,833) = 1,000,000,000,000$ (primes less than or equal to 29,996,224,275,833). Try this: `https://t5k.org/nthprime/`.

In this assignment, you should use a pre-computed π function supplied as a text file. Your application should read this text file to provide the answer $\pi(x)$, when x is supplied as an input. You are NOT supposed to use hash table to store the content of the text file. The text is very big, nearly 700MB. You have to perform a sequential search. Put this application (and the text file) in a container, as a service, and call the service from a master application from outside. The master application from outside, calls this service by supplying x and the containerized application provides the answer. In the actual assignment, you have to make a load-balanced version of this solution.

1.1 Actual Assignment

Implement a load balanced version of $\pi(x)$ in the following manner. Divide the text file into chunks of $2, 4, 8, 16$, and 32. Clone your application - create $2, 4, 8, 16$ and 32 instances of your docker application. Each instance will work on one chunk of the text file. Create a master application that takes x. Depending on the value of x, the master redirects the $\pi(x)$ computation request to the appropriate instance (that has the right chunk), and gets the result.

Note Actual task will be announced on the day of demonstration. You need to do this task in the lab. Everyone should do this independently (not a group activitiy). Meanwhile download the text file from Classroom google drive.