Samuel Ingram

ingram23

CS 422 Project Proposal

Project Definition

I intend to create a server load balancer that will properly distribute requests to various compute servers depending on the current workload. This load balancer will act as the single point of entry for many requests that will then be distributed to different servers based on a multitude of different algorithms. These algorithms will then be analyzed under multiple scenarios to determine which is best for different applications and which is most practical in the real world. Primarily, these algorithms will be analyzed by timing them in terms of total time to complete a batch of requests and the average time per request, however, other factors such as data structure upkeep and overhead will be taken into account.

Project Motivation

Distributed computing and wide networks of servers are becoming quite commonplace in the modern world, so it is ever so pertinent that we understand how to handle such networks. The project motivation consists of two main components: my own self education on this topic, and attempting to discover important details of how load balancers should be implemented. If we can get a better grasp on the best implementation for such load balancers, we can more quickly fulfill requests to people on the internet.

Proposed Work

To implement this, I need two parts, the load balancer and the servers. After implementing these, I will need to test them. Below I have detailed some work items.

Load Balancer

- Program a single proxy server that then routes requests to actual servers
- Develop and implement various balancing algorithms
    - Static algorithms (no load info from servers)
    - Dynamic algorithms (load info sent to balancer)

Servers

- Will support simple HTTP requests, but will use a timer to simulate computation time
- Should support the sending of the current load data so that the balancer can use it

Analysis

- Gather time data for the algorithms under different scenarios
- Compile data and create charts

Related Work

When doing research, I found a number of good resources describing load balancers and algorithms on websites like jscape, ngix, and serverfault, but none of these websites showed any benchmark numbers for how fast or slow different implementations were. Some would describe when to use one approach over the other, but I never really got an idea for how much better one approach was. I hope that my numbers and findings can help answer that.

Weekly Schedule (starting 4/6/2020)

1. Develop simple load balancer that hands a single request off to a simple server
2. Create complete test servers and implement basic algorithms
3. Complete development on algorithms and gather data
4. Compile data and write report