HW4 Bonus: HTTP Performance Analysis

Dowland Aiello

1 About

My bonus extension to the CSE 333 hw4 lab implements, and presents the findings, of a testing utility written in Rust that determines the maximum throughput in *requests per second*, *bytes per second*, and *latency per connection* as a function of the length of a query submitted to the http333d web server. The tool uses **httperf** and the **plotters** library written in Rust to automate the process of running these tests. The source code used to generate these tests is available **here**.

2 Findings

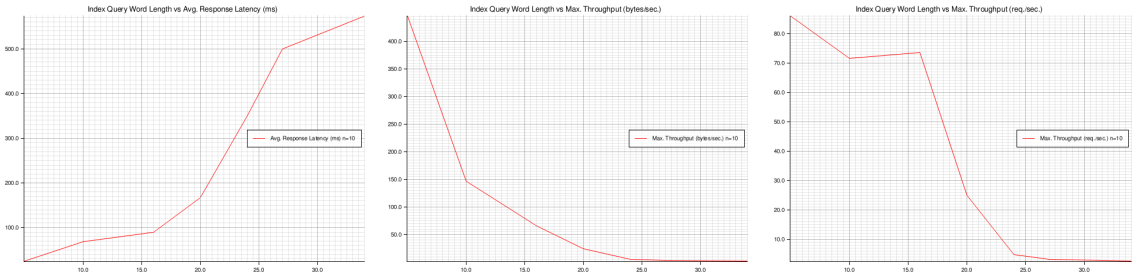

Figure 1: Request latency, and throughput (req./s, KB/s) as a function of query length

As is demonstrated in the 3 images above, the performance of the http333d server decreases exponentially as the length of a requested query increases. These results suggest that a compensating mechanism must be implemented in the http333d server to allow long, but frequent queries to place less of a bottleneck on the performance of the server. One such mechanism could be secondary caching in a least-recently-used manner, prioritized by query length, where such queries are persisted to memory, and evicted once they are no longer common. However, an alternative approach where the most frequently used branches along nested query rankings are cached instead (e.g., words like, "a") would be better suited for easing this bottleneck.

The efficacy of this alternative approach is suggested by the following results from the same testing program, with a dictionary of increasingly common words:

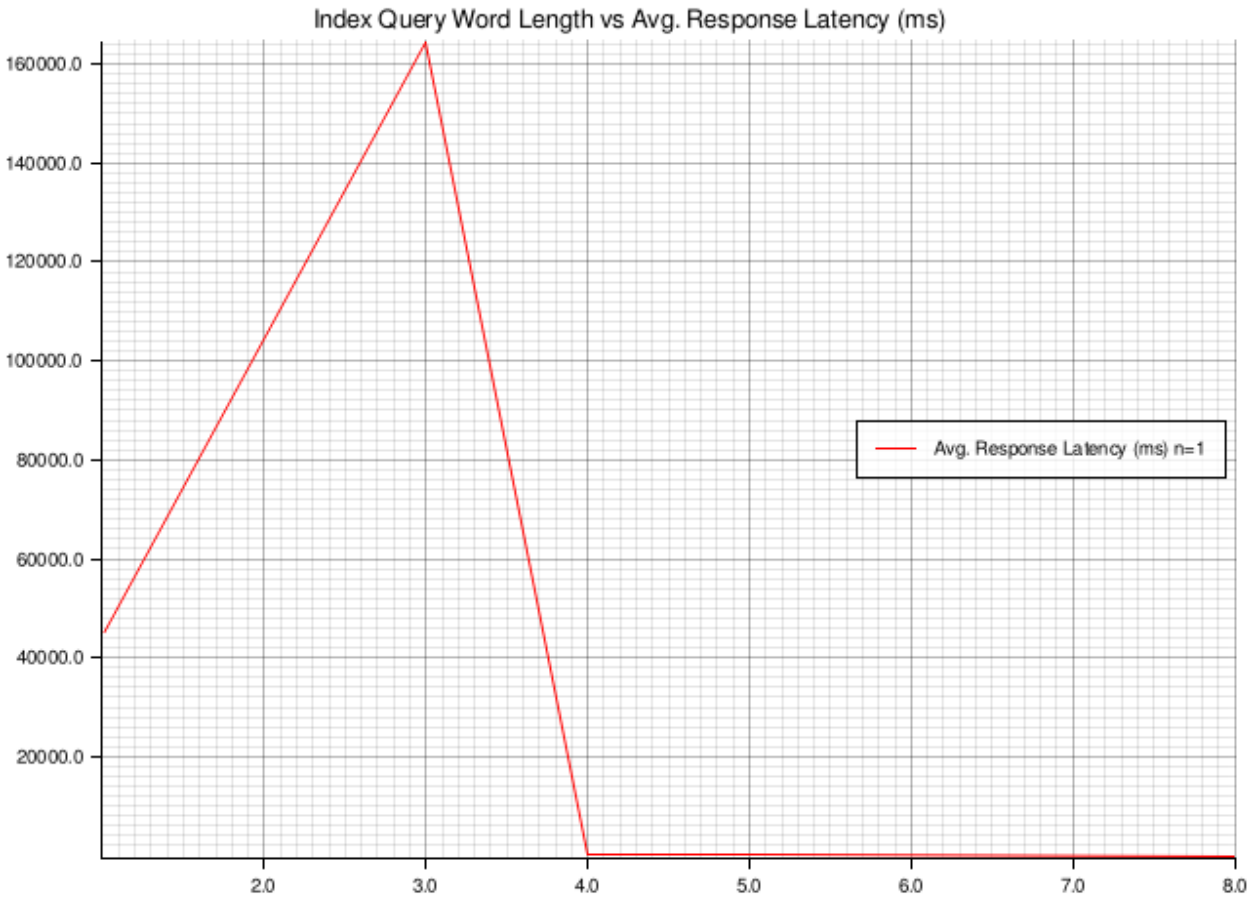

Figure 2: Shorter queries were correlated with exponentially larger response latency times than queries that used longer words.