COMPARING THE PERFORMANCE OF CONCURRENCY MODELS OF LOAD BALANCER ARCHITECTURES

Ragavan Thiruchittampalam

189341P

Degree of Master of Science

Department of Computer Science and Engineering

University of Moratuwa

Sri Lanka

May 2021

COMPARING THE PERFORMANCE OF CONCURRENCY MODELS OF LOAD BALANCER ARCHITECTURES

Ragavan Thiruchittampalam

189341P

Thesis submitted in partial fulfillment of the requirements for the degree of Master of Science in Computer Science

Department of Computer Science and Engineering

University of Moratuwa

Sri Lanka

May 2021

DECLARATION

I declare that this is my own work and this thesis does not incorporate without acknowledgement any material previously submitted for a Degree or Diploma in any other University or institute of higher learning and to the best of my knowledge and belief it does not contain any material previously published or written by another person except where the acknowledgement is made in the text.

Also, I hereby grant to University of Moratuwa the non-exclusive right to reproduce and distribute my thesis, in whole or in part in print, electronic or other medium. I retain the right to use this content in whole or part in future works (such as articles or books).

Signature : **Date :**

The above candidate has carried out research for the Masters thesis under my supervision.

Name of the supervisor:

Signature of the supervisor : **Date :**

ABSTRACT

The performance of load balancers is increasingly essential to distribute application traffic across multiple instances efficiently while maintaining a large number of concurrent users and application reliability. There is a myriad of factors that influence the performance of a load balancer, and in this study, the impact of the concurrency model of server architectures in performance is investigated in detail. In this research, how different server architectures - thread-per-connection, reactor and disruptor - can be used to build load balancers was studied and the strength and weaknesses of their concurrency model under a heavy concurrent workload was analysed. Two different reference implementations for thread-per-connection, with and without thread pool, were created to understand the impact of a thread pool. The reactor architecture was implemented utilizing the HTTPCore-NIO library and the disruptor architecture was developed using Netty transport and LMAX java library.

Besides, each implementation was extensively tuned and the performance of the best performing load balancer in each server architecture was compared. Each chosen architecture of load balancer has a distinct set of properties that control the performance, therefore, tuning each implementation was treated as a separate effort. Through the benchmarking tool, JMeter, extensive experiments will be conducted, and response time, throughput, CPU and memory usage were measured to analyse the impact of server architecture on performance. This study produced a comprehensive survey on several concurrency models of load balancer architectures, an experimental illustration and a detailed analysis of load balancers, in terms of performance, under high concurrent load. The results show with proper tuning, the peak throughput of each load balancer is increased by more than 15% compared to its baseline configuration.

Reactor based architecture with the configuration, 4 threads per reactor and 512 worker threads produce the peak throughput of 4362 requests per second which is the highest throughput produced in the experiments. In terms of throughout, the peak throughput produced by reactor-based architecture is 19% better than the maximum throughput generated by disruptor based architecture. The average response time increases exponentially for all load balancer architectures but at different rates. Reactor based architecture produces the best response time for all concurrent connections used in the experiments. A long-tail problem is completely visible in thread-per-connection architecture under high concurrency, where it has a long and thick tail in the response time distribution. The key factors affecting the performance of these architectures are the non-blocking nature of I/O operations, CPU usage, handling contention for shared resources, memory footprint and supporting a high number of concurrent connections.

Keywords: Load Balancer, Performance, Architecture, Concurrency

ACKNOWLEDGEMENT

It is my pleasure to thank everyone who has supported to do this research successfully. I especially wish to thank my research supervisor, Dr Indika Perera, for his valuable advice, support and guidance throughout this research.

My sincere thanks to the entire panel of lecturers of the Master's Program. Without their guidance, I would not have gained this vast knowledge. It is important to mention here that their guidance from the first day brought me to continue my studies.

Finally, I extend my thanks to my family for encouraging and supporting me to follow this program.

Contents

List of Figures

List of Tables