iJRASET
International Journal For Research in Applied Science and Engineering Technology

INTERNATIONAL JOURNAL FOR RESEARCH

IN APPLIED SCIENCE & ENGINEERING TECHNOLOGY

Volume: 9 **Issue:** XI **Month of publication:** November 2021

DOI: https://doi.org/10.22214/ijraset.2021.38748

www.ijraset.com

Call: 08813907089 | E-mail ID: ijraset@gmail.com

International Journal for Research in Applied Science & Engineering Technology (IJRASET)
ISSN: 2321-9653; IC Value: 45.98; SJ Impact Factor: 7.429
Volume 9 Issue XI Nov 2021- Available at www.ijraset.com

Client side and Server Side Load Balancing

Kunal Pimparkhede[1], Vinayak D. Shinde[2]
[1]*Research Scholar, Department of Computer Engineering*
[2]*Professor, Department of Computer Engineering Mumbai, India*

Abstract: In the microservice architecture it is vital to distribute loads across replicated instances of microservices. Load distribution such that no single instance is overloaded is called as load balancing. Often the instances of microservices are replicated across different racks, different data centers or even different geographies. Modern cloud based platforms offer deployment of microservices across different server instances which are geographically disperse. Having a system that will balance the load across service instances becomes a key success criteria for accurate functioning of distributed software architecture

Keywords: Load Balancing, Microservices, Distributed software system

I. INTRODUCTION

There are a different kinds of load balancing strategies that can be implemented to ensure that no single instance of a microservice is overloaded when compared to other instance [2]. Load balancing can be integrated with any load balancing algorithm. The strategy of load balancing only defines approach that can be adopted by front controllers to distribute load and does not define the actual algorithm. Algorithm can be plugged on either side of load balancing . There are two strategies in which load balancing can be done, on client side or on server side. The load balancing that happens on client side is called as client side load balancing and the load balancing that happens on server side is called as server side load balancing. In the microservice architecture it is vital to distribute loads across replicated instances of microservices. Load distribution such that no single instance is overloaded is called as load balancing. Often the instances of microservices are replicated across different racks, different data centers or even different geographies. Modern cloud based platforms offer deployment of microservices across different server instances which are geographically disperse. Having a system that will balance the load across service instances becomes a key success criteria for accurate functioning of distributed software architecture. There are a different kinds of load balancing strategies that can be implemented to ensure that no single instance of a microservice is overloaded when compared to other instance [2]. Load balancing can be integrated with any load balancing algorithm. The strategy of load balancing only defines approach that can be adopted by front controllers to distribute load and does not define the actual algorithm. Algorithm can be plugged on either side of load balancing . There are two strategies in which load balancing can be done , on client side or on server side. The load balancing that happens on client side is called as client side load balancing and the load balancing that happens on server side is called as server side load balancing

II. MATERIALS AND METHODS

Spring framework supports a Ribbon architecture, Ribbon is essentially a library used for client side load balancing. Client is a component that requests services from another component called as server. The problem statement is that there are a multiple instances of server running. In client side load balancing the client will own the choice making logic as to on which server instance its request should go. The code for load balancing is invoked by the client to invoke the right server instance. Client may create a Load Balancing client instance and invoke a method like choose() which will choose the right server instance for the client.

There are a different kinds of load balancing strategies that can be implemented to ensure that no single instance of a microservice is overloaded when compared to other instance [2]. Load balancing can be integrated with any load balancing algorithm. The strategy of load balancing only defines approach that can be adopted by front controllers to distribute load and does not define the actual algorithm. Algorithm can be plugged on either side of load balancing . There are two strategies in which load balancing can be done, on client side or on server side. The load balancing that happens on client side is called as client side load balancing and the load balancing that happens on server side is called as server side load balancing. In the microservice architecture it is vital to distribute loads across replicated instances of microservices. Load distribution such that no single instance is overloaded is called as load balancing. Often the instances of microservices are replicated across different racks, different data centers or even different geographies. Modern cloud based platforms offer deployment of microservices across different server instances which are geographically disperse. Having a system that will balance the load across service instances becomes a key success criteria for accurate functioning of distributed software architecture

©IJRASET: All Rights are Reserved

In the server side load balancing server owns the algorithm to decide which instance of the server to invoke against client request. This technique is often adopted by software systems when the design of client code is not in developers control rather its out of scope . This could be because of several reasons wherein the client code is an external code that calls the server to invoke its servers. Spring framework supports Zuul which is an API gateway that can be used for server side load balancing

There are a different kinds of load balancing strategies that can be implemented to ensure that no single instance of a microservice is overloaded when compared to other instance [2]. Load balancing can be integrated with any load balancing algorithm. The strategy of load balancing only defines approach that can be adopted by front controllers to distribute load and does not define the actual algorithm. Algorithm can be plugged on either side of load balancing . There are two strategies in which load balancing can be done , on client side or on server side. The load balancing that happens on client side is called as client side load balancing and the load balancing that happens on server side is called as server side load balancing

In the microservice architecture it is vital to distribute loads across replicated instances of microservices. Load distribution such that no single instance is overloaded is called as load balancing. Often the instances of microservices are replicated across different racks, different data centers or even different geographies[1]. Modern cloud based platforms offer deployment of microservices across different server instances which are geographically disperse. Having a system that will balance the load across service instances becomes a key success criteria for accurate functioning of distributed software architecture

III. PROPOSED METHODOLOGY

In the server side load balancing server owns the algorithm to decide which instance of the server to invoke against client request. This technique is often adopted by software systems when the design of client code is not in developers control rather its out of scope . This could be because of several reasons wherein the client code is an external code that calls the server to invoke its servers. Spring framework supports Zuul which is an API gateway that can be used for server side load balancing[5]

Spring framework supports a Ribbon architecture, Ribbon is essentially a library used for client side load balancing. Client is a component that requests services from another component called as server. The problem statement is that there are a multiple instances of server running[3]. In client side load balancing the client will own the choice making logic as to on which server instance its request should go. The code for load balancing is invoked by the client to invoke the right server instance. Client may create a Load Balancing client instance and invoke a method like choose() which will choose the right server instance for the client.

IV. CONCLUSION AND FUTURE WORK

Spring framework supports a Ribbon architecture, Ribbon is essentially a library used for client side load balancing. Client is a component that requests services from another component called as server. The problem statement is that there are a multiple instances of server running[4]. In client side load balancing the client will own the choice making logic as to on which server instance its request should go. The code for load balancing is invoked by the client to invoke the right server instance. Client may create a Load Balancing client instance and invoke a method like choose() which will choose the right server instance for the client.

V. ACKNOWLEDGEMENT

The authors recognize with gratitude to thank the efforts and direction given by our respective institute instructors and guide, Prof. Vinayak Shinde for their esteemed suggestions and remarks to improvise the manuscript.

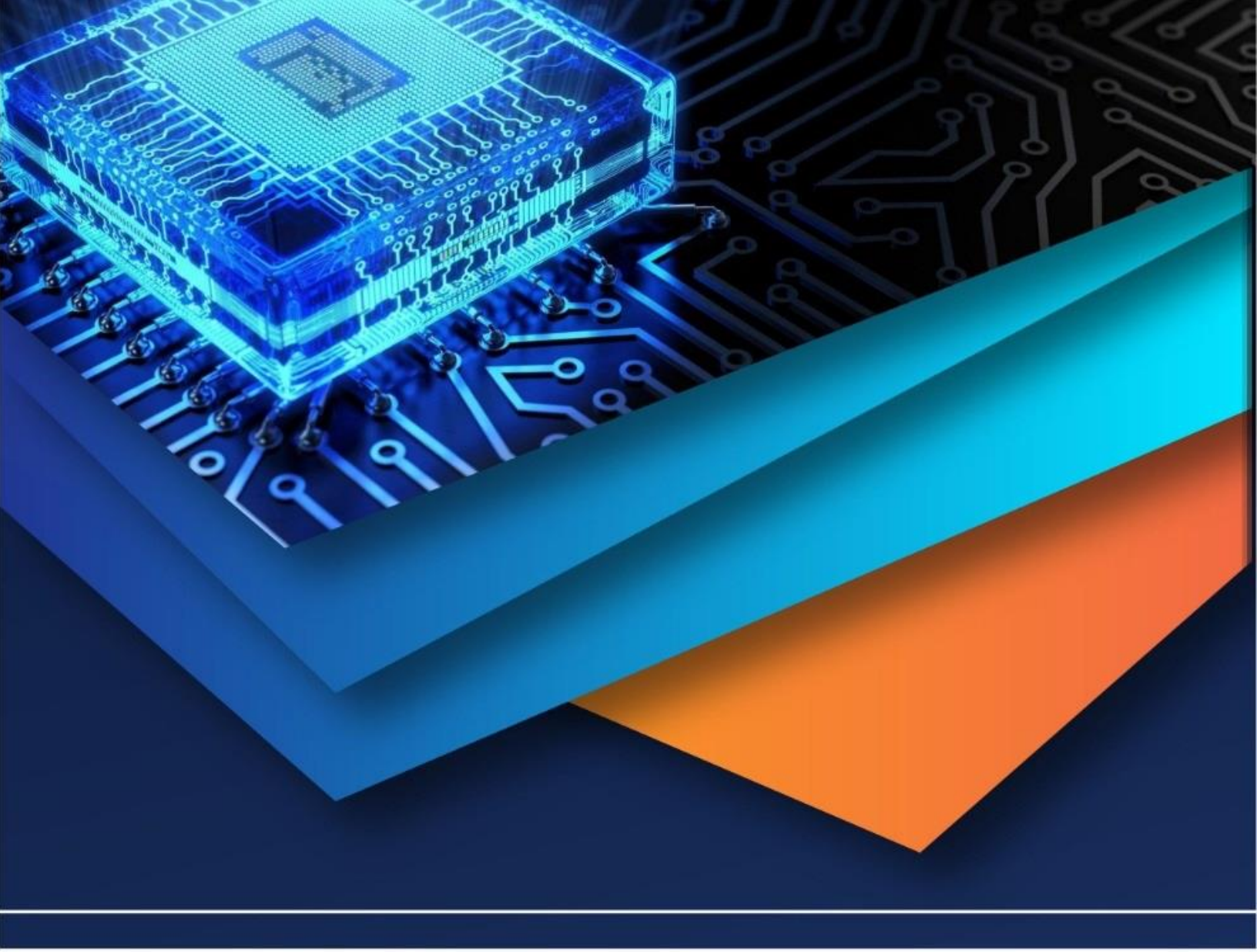
VI. CONFLICT OF INTEREST

The authors depict no conflict of interest.

REFERENCES

[1] Sindra, Aslam: Load Balancing Algorithm in Cloud computing. IEEE, 17-17 December 2015
[2] Mao Qilin: Load Balancing method based on SDN : In IEEE Explore. 13-14 June 2015
[3] Jiao Zhang: Load Balancing in Data center networks: A Survey : In IEEE 15 March 2018
[4] Sinha Sheikh Abdhullah : Review of recent load balancing techniques in Cloud Computing and BAT Algorithm variants IEEE ,16-18 March 2018
[5] M.C. Nkosi Multi-path load balancing for SDN plane, IEEE, 6-7 December 2018

©IJRASET: All Rights are Reserved



10.22214/IJRASET

45.98

IMPACT FACTOR: 7.129

IMPACT FACTOR: 7.429

INTERNATIONAL JOURNAL FOR RESEARCH

IN APPLIED SCIENCE & ENGINEERING TECHNOLOGY

Call : 08813907089 (24*7 Support on Whatsapp)