Frontiers in Computing and Intelligent Systems
ISSN: 2832-6024 | Vol. 5, No. 3, 2023

Using IGMP Protocol to Improve the Latency of Cloud Computing

Jing Zhong *

University of Waterloo, Waterloo N2L 3G1, Canada
*** Corresponding author Email:** j9zhong@uwaterloo.ca

Abstract: To solve the problem of network latency of cloud computing, organizations usually use the edge computing, which means shorter physical distance from the client, or the parallel computing method, which means separate the task to multi cloud servers. However, these two major solutions do not effectively solve the problem of network latency caused by multiple clients accessing the same resources. In this paper, a new strategy is proposed based on the operation mode of Internet Group Management Protocol (IGMP) to solve the networks latency and waste of network resources caused by multiple clients' access. This paper would perform the comparison tasks by using Amazon Web Services (AWS). To show the differences, there would be a simulated test of 1000 clients who are trying to access cloud resources from one cloud server. By comparing the total time of 1000 clients receiving their resources, the original group takes 5309 seconds for the cloud server to process the tasks. The test group takes 5034 seconds for the cloud server to process the tasks, which is about 5.68% improvement. Through the research, the conclusion is that if cloud resources are partition properly, the grouping strategy could effectively alleviate the networks latency problem of multiple clients.

Keywords: Network Latency; Multi Clients; Cloud Server; CPU Utilization.

1. Introduction

As technology advances, an increasing number of organizations are embracing cloud computing as a solution to address inefficiencies. This technological paradigm shift has consequently led to the integration of cloud computing into organizational management strategies. However, while cloud computing offers substantial advantages, it also brings forth certain limitations that organizations must consider.

A notable drawback stems from the inherent dependency of cloud computing on network connections and the cloud infrastructure itself. This reliance engenders a critical concern known as network latency, a factor that can substantially impact job performance [1]. The genesis of network latency issues can be attributed to resource overload experienced by cloud servers. In essence, each client necessitates establishing an independent connection to the cloud server to cater to their specific business requirements. This decentralized approach results in the cloud server being burdened with managing multiple connections and addressing the distinct needs of various clients. Paradoxically, a significant portion of this work might be repetitive, with only the origin of the requests distinguishing them.

There are mainly two ways to reduce the network latency. Firstly, move dynamically cloud services to the edge of the client's network, so there is less transfer time between the cloud servers and the client[1-3]. Secondly, using virtual network functions to increase one cloud services computing capacity, which means a cloud service could separate the tasks to multiple virtual cloud services[4]. Also, there are some extensions based on these two ways. For example, cloud server using genetic-based strategy to help the clients addressing the tasks to multi-cloud service. Another strategy is using Dynamic Workflow Scheduling (DWS) to allocate tasks to cloud based on cost, load-balancing, etc[5]. In summary, existing methods to reduce network latency include either reducing the physical distance or using strategy to allocate tasks properly to multi cloud service, no matter the cloud service is virtual or real.

Considering these challenges, the paper delves into the concept of leveraging the IGMP operation mode to ameliorate the network latency predicament intrinsic to cloud computing. IGMP is the frame group management protocol for internet service. It is a protocol responsible for the management of IP multicast members in the TCP/IP protocol cluster. In a multicast network, the IGMP runs between the last hop router and the multicast receiver. it is mainly used for multicast membership relationships. If no multicast group members express interest in the multicast stream, the router will not forward any multicast traffic to the network segment. Only when a user expresses interest in this multicast traffic and joins the multicast group, the router will forward and push the traffic[6]. Since the IGMP is for internet service, and it only run between the last hop router and users, only the operation mode of IGMP would be used. Cloud server could create their own "network segment" based on the resources, and cloud server only need to push the required resource to the group only if there is any interesting client. As the result, there is no need to create and maintain the individual connection between the cloud server and the client. Also, for multi-task that perform the same operation to same resources, the cloud service only needs to calculate it once, and push the result to multi clients. The number of connections is reduced, and the number of tasks need to perform is reduced too.

Aiming at the problem of the network latency of cloud service, this paper puts forward the problem of using IGMP operation mode to reduce the network latency. Although the existing methods do solve part of the network delay problem, when the number of clients is increase, the network delay problem still exists. Also, it is waste of resources when there are multiple clients are performing same tasks, which means the server may be doing the same task again and again. Based on IGMP, some of the concepts are applied to the cloud. The cloud server performs a reasonable partitioning of resources.

The cloud server does nothing, when there are no clients are interest in the resource. Otherwise, the cloud server pushes the resources within corresponding partition. The previous initiator was the clients. In the new approach, the active agent is the cloud server.

2. Material and Methods

The cloud service platform which was used in the task is AWS. The comparison experiment is to simulate 1000 clients try to upload a same image picture to the cloud storage, and the cloud server would need to process the image. The client would need to reload the processed image. The conclusion was reached by comparing the total time spent in the two methods, and the maximum time spent by a client.

The normal way for a client to use cloud server to do the task is shown as Fig 1. Firstly, clients need to upload the original image to S3, which is cloud storage provide by AWS. Secondly, clients need to notify the cloud server by using SQS, which is a notification pipe. Then the server would need to download the image from S3, after it received the notification. After the cloud server finish the task, the cloud server needs to upload the processed image to S3 and notify the clients that the task was done. As the result the client could download the image. Each step would happen twice in opposite direction.

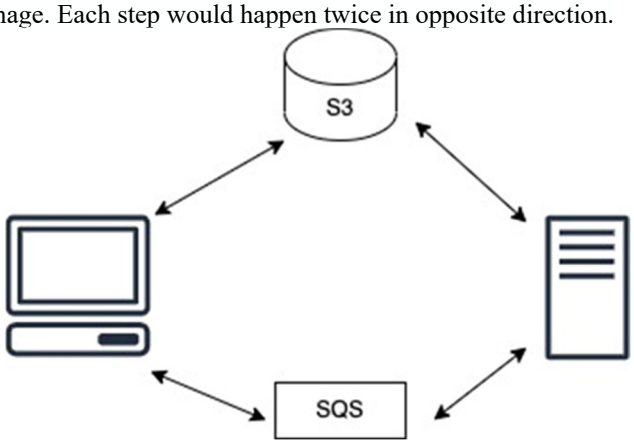

Fig 1. Normal way for a client uses cloud server to do task

As shown in Fig 2, on the client side, each client still must upload the original image to S3, because each client must assume themselves as the first uploader of the resource. Then the client also needs to notify the cloud server through SQS. So far, the procedure is the same as the original way. In the new approach, the procedure of cloud server is different. For the first client, after the cloud server receive the notification form the client, the cloud server needs to create a corresponding notification list for the initial image that was upload by the first client. Also, the cloud server would need to process the image too. After the task has done, the cloud server would need to upload the modify image to the cloud storage too. For the first client, the processing steps are similar. After the upload step for the cloud server, the next step is the notification step. The cloud server would iterate the notification list and push the modify image to the client. But for non-first client, the cloud server would check if the upload image were the initial image that the cloud server has, and it is not the initial image the clouds server receives for the non-first client. As the result, the cloud serves only add the new client to the notification list and start the iteration procedure for notification list.

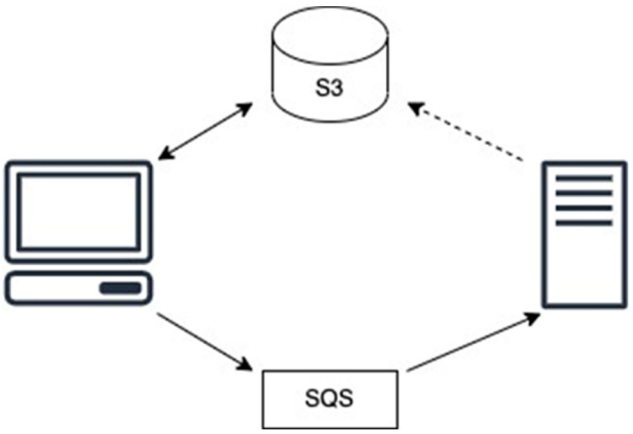

Fig 2. New approach

3. Result and Discussions

The total time cost for the original procedure to finish 1000 clients' tasks is 5309 seconds, and the average time is 5.3 seconds per client. The total time cost for the new approach to finish 1000 clients' tasks is 5034 seconds, and the average time is 5.0 seconds per client. The average length of time has decreased 5.66%, which also represents a real improvement in network latency.

Fig 3 is the original time cost for 1000 clients, and it is obviously some clients cost more than 10 seconds, which is almost double time cost for the average time cost, to finish the tasks. The maximum time cost for the original procedure is client 49, whose time cost is 25.7 seconds. In total, there are 4 clients cost more that 10 seconds, which means it is an obvious network delay phenomenon.

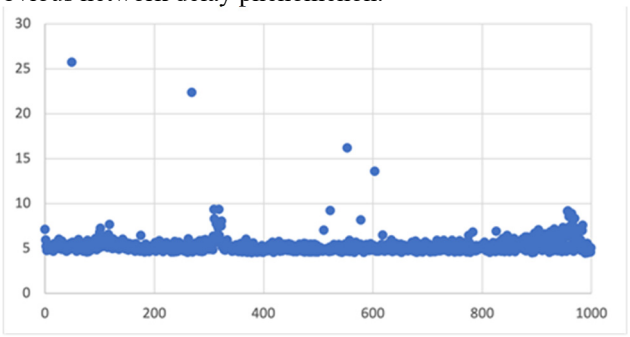

Fig 3. Original procedure

Fig 4 is the time cost for 1000 clients after using IGMP operation mode. The maximum time cost is client 0, and the time cost is 6.4 seconds, which is expected result, because only the first client requires the cloud server to perform the image task. Also, there are no clients that takes more than 10 seconds to finish their job. According to the result, all clients finish their tasks within plus minus 1 second of the average time costs.

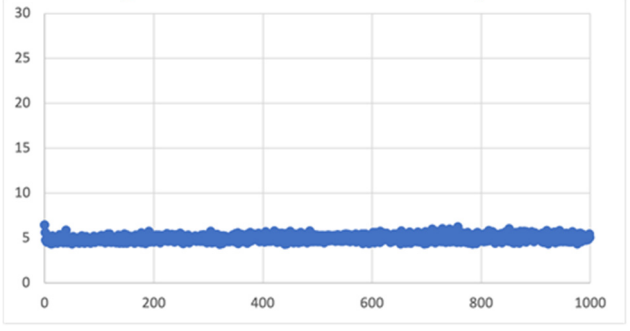

Fig 4. New approach

Fig 5 shows the CPU utilization of the cloud server when

it uses the original method for 1000 clients. There is specific time stamp that CPU utilization is extremely higher than normal, and the higher CPU utilization match the phenomenon in Fig 3, which is the clients around 300 take more time to receive the modify image. The average CPU utilization is 3.87%.

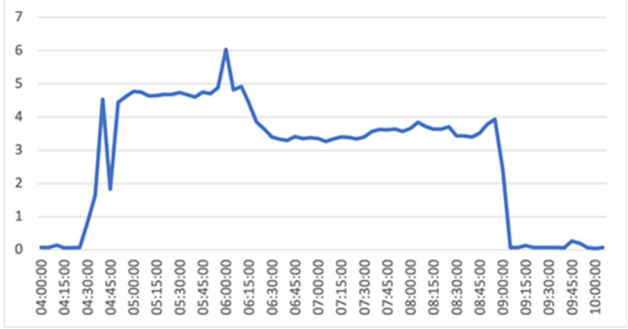

Fig 5. CPU utilization of original procedure

Fig 6 shows the CPU utilization of the cloud server when it uses the IGMP operation mode for 1000 clients. There is not extremely high CPU usage, and the CPU usage is similar for each time stamp. The average CPU utilization is 3.79% which is 2% less than the original one.

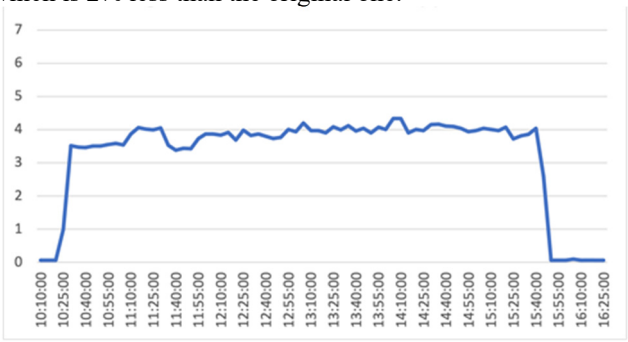

Fig 6. CPU utilization of new approach

Judge from the result, multi clients does cause network latency problems. Also, when the CPU usage is high, which means the usage of cloud resource is high, would cause the network latency. By using IGMP operation mode could help to reduce network latency problem. It reduces the average 5.68% less time cost compared to the original method, and 2% less usage of CPU. Even though the result shows IGMP operation mode reduce network latency, it is not suitable for those tasks, which are required by the client, could not be divided into modules. If the cloud server could not divide the tasks properly, then the cloud server would need to treat each client as initial client. The job for cloud servers of the initial client takes more extra step than the original ways.

4. Conclusion

This paper has presented using IGMP operation mode on cloud server could help to decrease the network latency problems that cause by the multi clients and the high usage of cloud server. It not only decreases the time cost, by also decrease the average CPU usage. There are some open problems that need to be fixed, for example, how to divide tasks properly.

References

[1] Charyyev, B., Arslan, E., Gunes, M. H., (2020) Latency Comparison of Cloud Datacenters and Edge Servers. In: GLOBECOM 2020 - 2020 IEEE Global Communications Conference, Taipei, pp. 1-6.

[2] Shi, T., Ma, H., Chen, G., (2019) A Genetic-Based Approach to Location-Aware Cloud Service Brokering in Multi-Cloud Environment. In: IEEE International Conference on Services Computing (SCC), Milan, pp. 146-153.

[3] Zhang, W., Hu, Y., Zhang, Y., Raychaudhuri, D., (2016) SEGUE: Quality of Service Aware Edge Cloud Service Migration. In: IEEE International Conference on Cloud Computing Technology and Science (CloudCom), Luxembourg, pp. 344-351.

[4] Cho, D., Taheri, J., Zomaya, A. Y., Bouvry, P., (2017) Real-Time Virtual Network Function (VNF) Migration toward Low Network Latency in Cloud Environments. In: IEEE 10th International Conference on Cloud Computing (CLOUD), Honolulu, pp. 798-801.

[5] Yu, Y., Shi, T., Ma, H., Chen, G., (2022) A Genetic Programming-Based Hyper-Heuristic Approach for Multi-Objective Dynamic Workflow Scheduling in Cloud Environment. In: IEEE Congress on Evolutionary Computation (CEC), Padua, pp. 1-8.

[6] RFC 2236. (1997) Internet Group Management Protocol. https://www.rfc-editor.org/info/rfc2236.