Synchronizing Multiple Datacenters Around the Globe Hosted by Cloud Service Providers

Shalini G N
M.Tech in CS&E
EWIT, Bengaluru, Karnataka

ArunBiradar, PhD
Professor & Head of the CS&E
EWIT, Bengaluru, Karnataka

ABSTRACT
As the urgency level of multiple datacenters across the globe has become common nowadays. There exists a request for inter-datacenter data transfers in bulk volumes. Where there is demand for inter-datacenter data transfer in bulk volume, their arises a challenge as how to schedule the large data at different urgency levels that uses available bandwidth. The solution to this challenge is to use controller called Software Defined Networking (SDN). And also should synchronize the files to the systems.

General Terms
Synchronization of the datacenters, algorithms as FIFO, Priority and bandwidth, Scheduling.

Keywords
Synchronize, Bulk data transfers, inter-datacenters, Softer Defined Networking (SDN).

1. INTRODUCTION
The cloud computing is an on-demand provisioning of server resources to applications with minimal efforts. The cloud datacenters systems are extended across geographical locations became common nowadays, which aimed to bring services closer to the users and make low power cost. There are many cloud service providers like Google, Amazon, Microsoft have invested significantly in constructing large-scale datacenters across the world, to host their services.

As in a geo-distributed datacenter system, there exists a basic demand to transfer bulk volumes of data from one datacenter to another datacenter. The main challenge was how to handle and schedule the bulk data transfers at different urgency levels to utilize available inter-datacenter bandwidth.

So to handle this challenge controller was developed such as Software Defined Networking [SDN] which decouples the control plane from the data paths. And here there exists issues in Datacenter such as synchronizing the datacenters and to transfer the data the datacenter should be selected dynamically.

Software-Defined Networking (SDN) is an emerging architecture that is dynamic, manageable, cost-effective, and adaptable, making it ideal for the high-bandwidth, dynamic nature of today's applications. This architecture decouples the network control and forwarding functions enabling the network control to become directly programmable and the underlying infrastructure to be abstracted for applications and network services.

The goal of SDN is to allow network engineers and administrators to respond quickly to changing business requirements. In a software-defined network, a network administrator can shape traffic from a centralized control console without having to touch individual switches, and can deliver services to wherever they are needed in the network, without regard to what specific devices a server or other device is connected to. The key technologies are functional separation, network virtualization and automation through programmability.

In this paper, firstly they formulated the problem of transferring bulk data to the datacenters. Here there is no deadline for the acceptance of the job in case if the job is rejected, the job is resubmitted by utilizing the available bandwidth.

Secondly, the algorithms are discussed to overcome the acceptance of the jobs and the priority to be given to the jobs. This task is done by the controller. The algorithms used are FIFO, Priority and the Bandwidth.

Third, the synchronization is the main task to be done, synchronization of datacenters is important because if the client is in need of data, but the particular datacenter is busy. In this case the other datacenter can provide the data at different urgency levels.

2. RELATED WORK
Federation of geo-distributed cloud services is a trend in cloud computing which, by spanning multiple datacenters at different geographical locations, can provide a cloud platform with much larger capacities. Such a geo-distributed cloud is ideal for supporting large-scale social media streaming applications with dynamic contents and demands, owing to its abundant on-demand bandwidth capacities and geographical proximity to different groups of users.

In recent advances the server virtualization technologies that allows for live migration of datacenter services within a local area network environment. In virtual server it retains the same network address as before, and any ongoing network level interactions are not disrupted. As in a LAN environment, storage requirements are normally met via either network attached storage or through a storage area network which allow for storage access.

The storage services has become cheaper because of low cost storage services, everyone wants to store each of the applications such as text files, audio, video etc. The kind of data over the network cannot be moved through traditional LAN technologies. Hence for the efficient utilization of the network, the Cloud Service Provider [CSP], uses WAN which is designed to achieve the high utilization of network with low latency. It uses WAN optimization techniques.

The main challenge is to capture data, where data should be available for processing. Now the Cloud Service Providers are working to provide the services having low latency and high throughput. Many datacenters are deployed across globe, where datacenters are connected to each other with high bandwidth links. The service providers redirect the user

request to the one of the nearest data center hosting the application. This achieves to provide for users a fast service.

As they transfer bulk volumes of data to the datacenter, if the datacenter is failed to access by the users then it causes problem to the user to retrieve the data. So to resolve this problem there should be synchronization of the datacenters where the back up or an copy of the data to be stored in an intermediate datacenter, hence users can retrieve the data from another datacenter.

When they aggregate all the data at one data center, then the size of that data is not small, hence it will be associated with many challenges. The one is with bandwidth which is not uniform across all the links between the datacenters. The other challenge is storage cost, which varies from one datacenter to another datacenter. The other challenge is the processing power and processing cost of datacenters. Hence, we require an efficient algorithm to move the data from all geographically distributed sites to an aggregation site.

In the datacenters, TCP congestion control and FIFO flow scheduling are currently used for data flow transport, which are not aware of deadlines. Previously they used end-to-end congestion control for the transportation of bulk volumes of data among datacenters in a geo-distributed cloud. Instead of end-to-end congestion control we propose store-and-forward in intermediate datacenters, such that when the data is not processed within the given time slot, the data is stored into an intermediate datacenter, once the previous data is processed, this data will again be resubmitted to the datacenter for processing.

Datacenter workloads impose unique requirements on the transport fabric .Interactive soft real-time workloads such as the ones seen in search, social networking, and retail generate a large number of small requests and responses across the datacenterthat are stitched together to perform a user-requested computation. These applications demand low latency for each of the short request or response flows, since user perceived performance is dictated by how quickly responses to all the requests are collected and delivered back to the user.
The study relies on flows with stringent deadlines, which will not assume any traffic patterns. Here the algorithms used are optimization algorithms to dynamically adjust flow transfer schedules under any traffic patterns. To minimize the cost problem for inter-datacenter traffic scheduling which was based on classic time expanded graphwhich was first used in NetStitcher.

As there is bulk data transfer in a geo-distributed cloud it includes task admission control where once data transfer task is admitted, it ensures its timely completion within the specified deadline, but once the task is not completed it gets rejected but we focus where the task is resubmitted by using infinite system lifespan. It uses data routing and store-and-forward.

3. METHODOLOGIES

3.1 Architecture

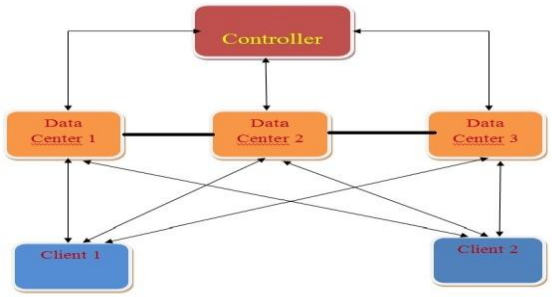

Fig 1: it shows the architecture of the system

Here, the controller is the one which admits the jobs to the datacenters according to the priorities given to the jobs. There can be n number of datacenters and the m number of clients. When client does few tasks such as get the files from the datacenters, upload the files to the datacenters, or delete the files. The client's sends the request to the datacenters which are the servers, then the datacenter admits to the controller, the controller sends the job to the queue, where the jobs will be waiting until the controller schedules. The controller first sends the job which has come first i.e FIFO. If that previous job is not yet completed, the controller sees the priority of the jobs and later it sends the job to the datacenters. If the arrived job is to be completed earlier than the previous job within the given time, the previous job is kept pending and the current job is completed. The previous job is not rejected rather than it is resubmitted according to the availability of the bandwidth.

The synchronization acts as, when the client does few operations such as get the file, or set the file or delete the file. The changes should be synched to the other datacenters. Imagine there are four clients and five datacenters, if the clients are continuously doing work and all the servers are busy, then client need not wait for the particular datacenter to respond, the datacenters which are free will respond the clients request. The datacenters are selected dynamically.

3.2 Mathematical Model

$$\text{Max} = \sum_{i=1}^{N} Wi * Fi$$

N are the number of jobs submitted with in time duration T,

M are the jobs Competed,

N-M jobs are pending in the Job Q with in time duration T,

Wi is the benefit (weight) associated with job i.

Fi is the flag indicates if job is completed or not.

The main goal is to maximize the number of jobs to be submitted within in the given time. If the jobs does not complete the task within the given time, the job is kept pending until the available bandwidth. Later, once the bandwidth is available the pending job is resubmitted. The

time taken for the clients is saved as they need not wait for the one datacenter to retrieve the desired data.

3.3 FIFO

The FIFO algorithm is used in this paper, as to submit the jobs to the datacenter. The controller when it admits the jobs from the datacenter, it maintains a queue where the jobs are kept in queue. The controller now decides, as which of the jobs to be sent to the datacenter. It uses the algorithm FIFO, where the first job is taken that will be firstly sent to the datacenter.

3.4 Priority

Here the priority scheduling is done dynamically, where the jobs are taken according to the availability of the bandwidth and the time taken for each job to be completed. Here the deadline is infinity so that no jobs will be rejected. The priority is given as such the jobs are scheduled according to the controller.

4. SEQUENCE OF THE SYSTEM

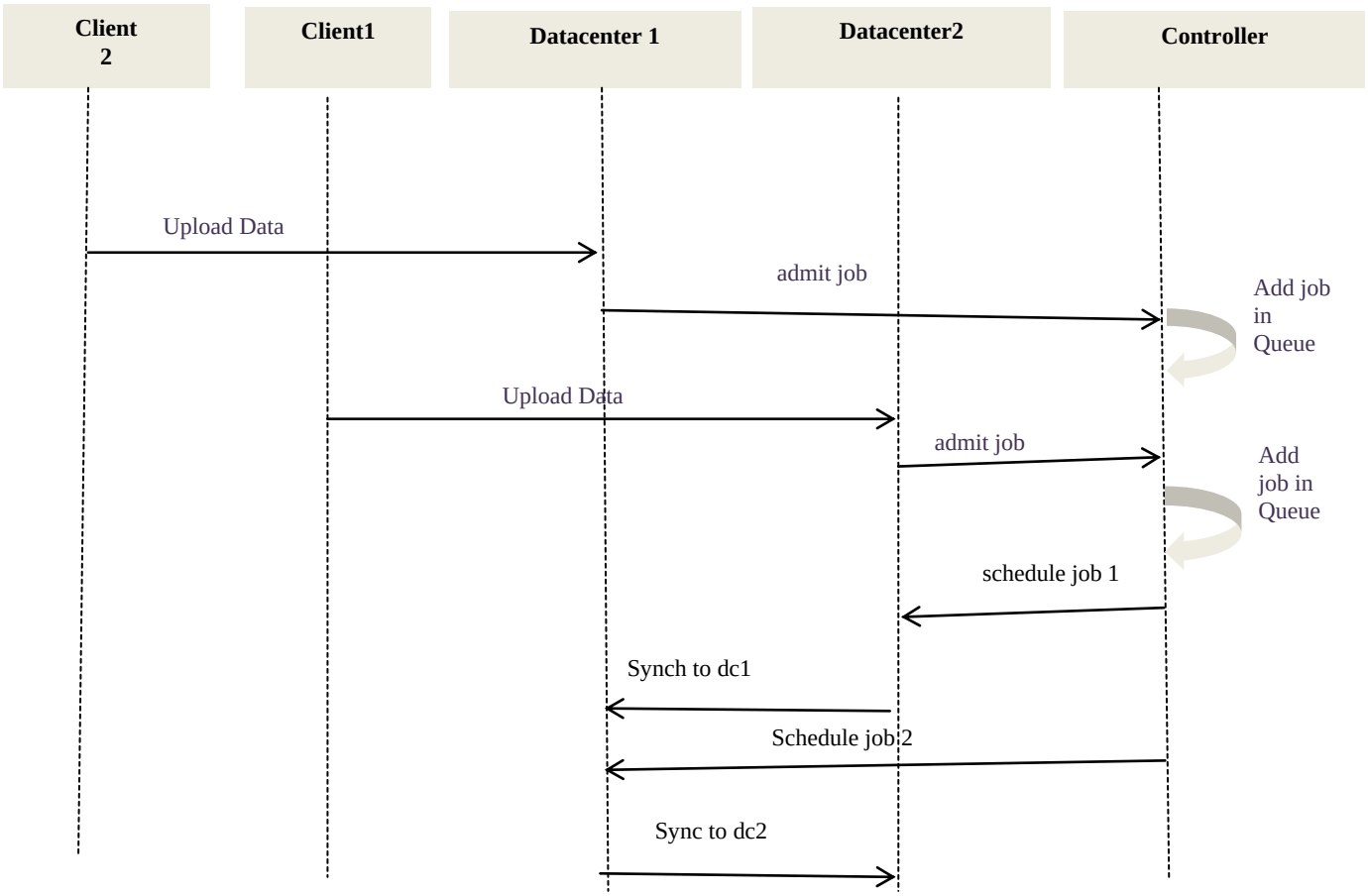

Fig 2: Sequence of the system

Here the sequence of the system is showed as, there are two clients and two datacenters and an controller, first one of the client uploads the data to one of the datacenter, then that datacenter will admit to the controller, then the controller will add to the queue, next the other client will upload the data to other datacenter, then the datacenter will admit the data to the controller, the controller will add that job to the queue. Now the job of the controller is to schedule the jobs and then these scheduled jobs are synchronized to the other datacenters.

5. CONCLUSION

Cloud Computing is an evolving area where enormous of data has been processed every day. This paper presents multiple datacenters deployed around the geographical locations, where the transfer of bulk data can be carried easily. There should be synchronization among the intermediate datacenters such that the data will be replicated in another datacenter so that the users can access the data. The optimal chunk routing problem has been resolved by using three dynamic algorithms. Finally it showed how to schedule the bulk data transfers across geo-distributed datacenters.

As the paper tells about how the bulk data transfers are carried around the globe, here the inter-datacenters are synchronized as it is very useful for the clients to access the data required without time consuming. This paper presents efficient usage of time for the transfer of data through datacenters at different urgency level by using the available bandwidth.

In future it can concentrate on the data which will be optimally chunked and the routing algorithm can be implemented for the better results and efficiently used by the clients.

6. ACKNOWLEDGMENTS

Our thanks to the respected guide who helped to carry out the work. And thanks to the experts who presented in different papers, which gave an idea to the work done.

7. REFERENCES

[1] K.K Ramakrishnan, P. Shenoy and J. van der Merwe 2007. Live data center mirgration across WANs.

[2] Y. Wu, C. Wu, B. Li, L. Zhang 2012. Scaling Social media applications in Geo-distributed clouds.

[3] C.works Wilson, H.Ballani, A. Rowtron 2011. Meeting deadlines in datacenters networks.

[4] B. Vamanan, J. Hasan, and T. Vijaykumar2012. Deadline aware datacenter TCP

[5] Yu, Wu, Zhizhong Zhang, Chuan Wu, Zongpeng Li 2015. Orchestrating Bulk data transfers across geo-distributed datacenters