ISSN (Online): 2456-0448
International Journal Of Innovative Research In Management, Engineering And Technology
Vol. 3, Issue 4, April 2018

Arranging Bulk Datatransfers Acrossgeo-Distibuted Data Centre

[1] I.Mary Linda, [2] Kilari. Laxmi Sudha, [3] S.Keshava Priya
[1] Assistant Professors, Dept. Of C.S.E., Bharath University, Chennai, Tamil Nadu, India
[2] [3] Student, Dept. Of C.S.E., Bharath University, Chennai, Tamil Nadu, India

Abstract: ***A challenge arises on how to schedule the bulk data transfers at different urgency levels, in order to fully utilize the available inter-datacenter bandwidth. The Software Defined Networking (SDN) paradigm has emerged recently which decouples the control plane from the data paths, enabling potential global optimization of data routing in a network. This paper aims to design a dynamic, highly efficient bulk data transfer service in a geo-distributed datacenter system, and engineer its design and solution algorithms closely within an SDN architecture. We model data transfer demands as delay tolerant migration requests with different finishing deadlines. Thanks to the flexibility provided by SDN, we enable dynamic, optimal routing of distinct chunks within each bulk data transfer (instead of treating each transfer as an infinite flow), which can be temporarily stored at intermediate datacenters to mitigate bandwidth contention with more urgent transfers. An optimal chunk routing optimization model is formulated to solve for the best chunk transfer schedules over time. To derive the optimal schedules in an online fashion, three algorithms are discussed, namely a bandwidth-reserving algorithm, a dynamically-adjusting algorithm, and a future-demand-friendly algorithm, targeting at different levels of optimality and scalability. We build an SDN system based on the Beacon platform and Open Flow APIs, and carefully engineer our bulk data transfer algorithms in the system. Extensive real-world experiments are carried out to compare the three algorithms as well as those from the existing literature, in terms of routoptimality, computational delay and overhead.***

I. INTRODUCTION

This paper proposes a novel optimization model for dynamic, highly efficient scheduling of bulk data transfers in a geo-distributed datacenter system, and engineers its design and solution algorithms practically within OpenFlow based SDN architecture. We model data transfer requests as delay tolerant data migration tasks with different finishing deadlines. Thanks to the flexibility of transmission scheduling provided by SDN, we enable dynamic, optimal routing of distinct chunks within each bulk data transfer which can be temporarily stored intermediate datacenters and transmitted only at carefully scheduled times, to mitigate bandwidth contention among tasks of different urgency levels. Our contributions are summarized as follows. First, we formulate the bulk data transfer problem into a novel, optimal chunk routing problem, which maximizes the aggregate utility gain due to timely transfer completions before the specified deadlines. Such an optimization model enables flexible, dynamic adjustment of chunk transfer schedules in a system with dynamically-arriving data transfer requests, which is impossible with a popularly-modeled flow-based optimal routing model.

II. LITERATURE SURVEY:

Jeffrey Dean and Sanjay Ghemawat they proposed MapReduce is a programming model and an associated implementation for processing and generating large data sets. Users specify a map function that processes a key/value pair to generate a set of intermediate key/value pairs, and a reduce function that merges all intermediate values associated with the same intermediate key. Many realworld tasks are expressible in this model, as shown in the paper. Programs written in this functional style are automatically parallelized and executed on a large cluster of commodity machines. The run-time system takes care of the details of partitioning the input data, scheduling the program' section a cross a set of machines, handling machine failures, and managing the required inter-machine communication

Maria Alejandra Rodriguez and Rajkumar Buyya they proposed Cloud computing is the latest distributed computing paradigm and it offers tremendous opportunities to solve large-scale scientific problems. However, it presents various challenges that need to be addressed inorder to be efficiently utilized for workflow applications. Although the workflow

scheduling problem has been widely studied there are very few initiatives tailored for cloud environments. Furthermore, the existing works fail to either meet the user's quality of service (QoS) requirements or to incorporate some basic principles of cloud computing such as the elasticity and heterogeneity of the computing resources. This paper proposes a resource provisioning and scheduling strategy for scientific workflows on Infrastructure as a Service (IaaS) clouds. We present an algorithm based on the meta-heuristic optimization technique, particle swarm

Hong Liu, Cedric F. Lam, and Chris Johnson they proposed We review the growing need for optical interconnect bandwidth in datacenter networks, and the opportunities and challenges for wavelength division multiplexing (WDM) to sustain the "last 2km" bandwidth growth inside datacenter networks.

Jim Gao, Google they proposed modern data center (DC) is a complex interaction of multiple mechanical, electrical and controls systems. The sheer number of possible operating configurations and nonlinear interdependencies make it difficult to understand and optimize energy efficiency. We develop a neural network framework that learns from actual operations data to model plant performance and predict PUE within a range of 0.004 +/0.005(mean absolute error +/1standard deviation), or 0.4% error for a PUE of 1.1. The model has been extensively tested and validated at Google DCs. The results demonstrate that machine learning an effective way of leveraging existing sensor data to model DC performance and improve energy efficiency.

SYSTEM REQUIREMENTS

SOFTWARE REQUIREMENTS

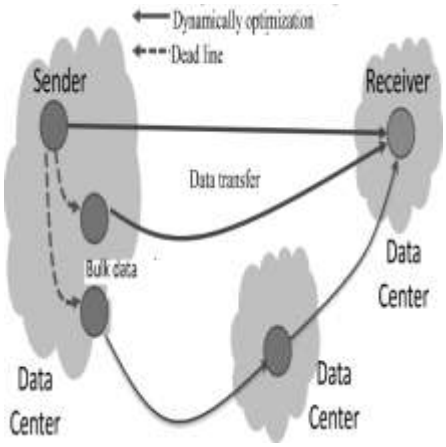
- Front End : Java
- Environment : Eclipse/Net Beans
- Back End : MySQL
- Operating System: Windows XP

HARDWARE REQUIREMENTS

- Processor : Pentium IV
- RAM : 512 MB
- Hard Disk : 80 GB

ISSN (Online): 2456-0448
International Journal Of Innovative Research In Management, Engineering And Technology
Vol. 3, Issue 4, April 2018

System Architecture

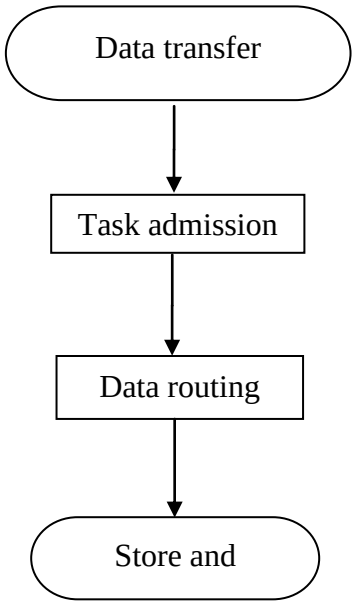


MODULE DESCRIPTION

1. SDN based architecture
2. The Optimal Chunk Routine
3. Dynamically optimization algorithm

SDN BASED ARCHITECTURE

We consider a cloud spanning multiple datacenters located in different geographic locations. Each datacenter is connected via a core switch to the other data centers. Data transfer requests may arise from each datacenter to move bulk volumes of data to another datacenter.



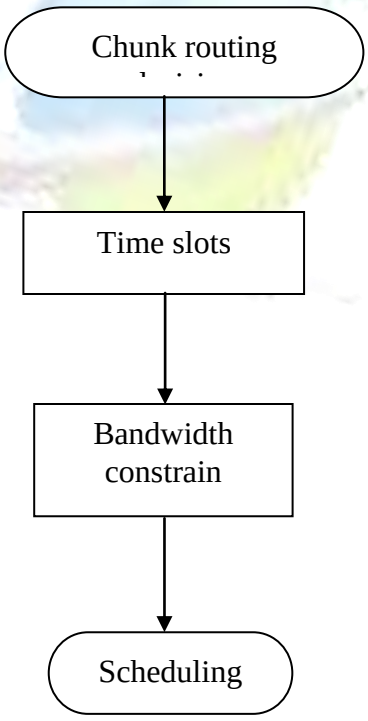
ISSN (Online): 2456-0448
International Journal Of Innovative Research In Management, Engineering And Technology
Vol. 3, Issue 4, April 2018

III. SDN BASED ARCHITECTURE

A gateway server is connected to the core switch in each datacenter; it also tracks network topology and bandwidth availability among the datacenters with the help of the switches. Combined closely with the SDN paradigm, a central controller is deployed to implement the optimal data transfer algorithms, dynamically configure the flow table on each switch, and instruct the gateway servers to store or to forward each data chunk.

The Optimal Chunk Routing Problem

The objective function maximizes the overall weight of all the jobs to be accepted. A special case where implies the maximization of the total number of accepted jobs. Constraint (a) states that for each chunk w in each job J, it should be sent out from the source datacenter SJ at one time slot (i.e., the valid transmission interval of the job), if it is accepted for transfer at all in the system on the other hand, the chunk should arrive at the destination datacenter DJ via one of DJ's neighboring datacenters as well, as specified by Constraint (b). Constraint (c) enforces that at any intermediate datacenter n other than the source and destination of chunk w, if it receives the chunk at all in one time slot within the valid transmission interval of the job, it should send the chunk out as well within the interval. The optimization model in (1) is an offline optimization problem in nature. Given any job arrival pattern, it decides whether each job should be accepted for transfer under bandwidth constraints, and derives the best paths for chunks in accepted jobs, along which the chunks can reach their destinations within the respective deadlines



The Optimal Chunk Routing Problem

ISSN (Online): 2456-0448
International Journal Of Innovative Research In Management, Engineering And Technology
Vol. 3, Issue 4, April 2018

Dynamically optimization algorithm

We examine the computation complexity of different algorithms, in terms of the time the control spends on calculating the chunk routing schedules using each of the algorithms, referred to as the scheduling delay. Note that an important requirement is that the scheduling delay of an algorithm should be less than the length of a time slot. In our experimental setting, each transfer job has an average of 500 chunks (50 GB) to transmit, and thus the job rate can be greatly increased if the job size is reasonably constrained since the complexity of solving the integer problems mainly depends on the number of chunks and the lifetime of the jobs (specified by the deadline)

Dynamically optimization algorithm

IV. CONCLUSION

This paper presents our efforts to tackle an arising challenge in geo-distributed datacenters, i.e., deadline-aware bulk data transfers. Inspired by the emerging Software Defined Networking initiative that is well suited to deployment of an efficient scheduling algorithm with the global view of the network, we propose a reliable and efficient underlying bulk data transfer service in an inter data center network, featuring optimal routing for distinct chunks over time, which can be temporarily stored at intermediate datacenters and forwarded at carefully computed times. For practical application of the optimization framework, we derive three dynamic algorithms, targeting at different levels of optimality and scalability. We also present the design and implementation of our Bulk Data Transfer system, based on the Beacon platform and Open Flow AIPs. Experiments with realistic settings verify the practicality of the design and the efficiency of the three algorithms, based on extensive comparisons with schemes in the computer.

FUTURE ENHANCEMENT

In future work, we plan to improve network transaction to better handle configuration scenarios with large per-run cap times for each target algorithm run; specifically, we plan to integrate adaptive capping mechanism into cloud, which will require an extension of SDN models to handle the resulting partly censored data. While in this paper we aimed to find a single configuration with overall good performance, we also plan to use SDN models to determine good configurations on a per-instance basis. Finally, we plan to use these models to characterize the importance of individual parameters and their interactions, and to study interactions between parameters and instance features.

REFERENCES

[1]Data Center Map [Online]. Available: http://www.datacentermap. Com/datacenters.html, 2015.

[2] K. K. Ramakrishnan, P. Shenoy, and J. Van der Merwe, "Live data center migration across WANs: A robust cooperative context aware approach," in Proc. SIGCOMM Workshop Internet Net. Manage. 2007, pp. 262–267.

[3] Y. Wu, C. Wu, B. Li, L. Zhang, Z. Li, and F. C. M. Lau, "Scaling social media applications into geo-distributed clouds," in Proc. IEEE Conf. Compute. 2012, pp. 684–692.

[4] J. Dean and S. Ghemawat, "MapReduce: Simplified data processing on large clusters," Common. ACM, vol. 51, no. 1, pp. 107–113, Jan. 2008.

[5] A. Greenberg, G. Hjalmty's son, D. A. Maltz, A. Myers, J. Rexford, G. Xie, H. Yan, J. Zhan, and H. Zhang, "A clean slate 4D approach to network control and management," ACM SIGCOMM Compute. Common. Rev., vol. 35, no. 5, pp. 41–54, 2005.