Computer Networks 68 (2014) 123–137

Contents lists available at ScienceDirect

Computer Networks

journal homepage: www.elsevier.com/locate/comnet

Multiple bulk data transfers scheduling among datacenters ☆

Yiwen Wang [a], Sen Su [a,*], Alex X. Liu [b], Zhongbao Zhang [a]

[a] State Key Laboratory of Networking and Switching Technology, Beijing University of Posts and Telecommunications, 10 Xi Tu Cheng Road, 10086 Beijing, China
[b] Department of Computer Science and Engineering, Michigan State University, 2132 Engineering Building, East Lansing, MI 48824-1226, United States

ARTICLE INFO

Article history:
Received 16 May 2013
Received in revised form 7 January 2014
Accepted 18 February 2014
Available online 27 February 2014

Keywords:
Traffic engineering
Bulk data transfers scheduling
Inter-datacenter traffic
Optimization models and methods

ABSTRACT

Bulk data migration between datacenters is often a critical step in deploying new services, improving reliability under failures, or implementing various cost reduction strategies for cloud companies. These bulk amounts of transferring data consume massive bandwidth, and further cause severe network congestion. Leveraging the *temporal* and *spacial* characteristics of inter-datacenter bulk data traffic, in this paper, we investigate the Multiple Bulk Data Transfers Scheduling (MBDTS) problem to reduce the network congestion. Temporally, we apply the store-and-forward transfer mode to reduce the peak traffic load on the link. Spatially, we propose to lexicographically minimize the congestion of all links among datacenters. To solve the MBDTS problem, we first model it as an optimization problem, and then propose a novel *Elastic Time-Expanded Network* technique to represent the time-varying network status as a static one with a reasonable expansion cost. Using this transformation, we reformulate the problem as a Linear Programming (LP) model, and obtain the optimal solution through iteratively solving the LP model. We have conducted extensive simulations on a real network topology. The results prove that our algorithm can significantly reduce the network congestion as well as balance the entire network traffic with practical computational costs.

1. Introduction

1.1. Background and motivation

Bulk data transfers across geographically distributed datacenters play a critical role in running data backup applications, delivering business services, and implementing various cost reduction strategies. To improve the fault tolerance, terabytes to petabytes of data are regularly replicated among three or more datacenters [1]. To provide a highly available and scalable service, cloud service providers always migrate large amounts of application content to datacenters closer to users, *e.g.*, Amazon CloudFront [2]. Besides, considering the spatial and temporal variability of each datacenter on operational costs, service providers may migrate data to datacenters located in the areas with low electricity costs [3], or carbon emission [4].

The volume of transferring bulk data always ranges from terabytes to petabytes [5]. Due to the peta-scale amount of data, the bulk data transfers impose a heavy load to the links between datacenters, and dominate the inter-datacenter aggregate traffic [6]. Especially when there exist multiple bulk data transfers, an inevitable problem arises with scheduling these bulk data transfers. A schedule without careful design may lead to highly

☆ The work was supported in part by the following funding agencies of China: National Natural Science Foundation under Grant 61170274, National Basic Research Program (973 Program) under Grant 2011CB302704, Innovative Research Groups of the National Natural Science Foundation under Grant 61121061, Important National Science & Technology Specific Projects: Next-Generation Broadband Wireless Mobile Communications Network (2012ZX03002008-002-03), and the Fundamental Research Funds for the Central Universities (2013RC1102).
* Corresponding author. Tel.: +86 1061198129; fax: +86 1062285223.
E-mail address: susen@bupt.edu.cn (S. Su).

http://dx.doi.org/10.1016/j.comnet.2014.02.017
1389-1286/

unbalanced use of the link capability in both spatial and temporal dimensions. This unbalanced bulk traffic always saturates links between datacenters and further induces severe network congestion [7,8] which, not only deteriorates the performance of other coexisting interactive transfers [9], but also impedes the capability of satisfying the flash traffic that may occur in various parts of the network.

To alleviate the network congestion, we can lease more links from ISPs to enlarge the capacity of networks, or choose to transfer bulk data via shipping storage devices by companies such as Fedex [10]. However, both of these methods increase the operational cost of cloud services providers on data transfers. In order to co-ordinate multiple bulk data transfers with time constraints, a careful strategy is called for to schedule transfers, so that the network congestion can be minimized, and bulk traffic can be balanced across the inter-datacenter links as well.

1.2. Limitations of prior art

Most existing works [11–13] investigate bulk data transfer scheduling in the grid networks. Leveraging the delay-tolerance feature of bulk transfers, Chen and Primet [11] study the scheduling for deadline guaranteed multiple bulk data transfers in grid networks. They aim at minimizing the network congestion through providing a flexible scheduling algorithm. Rajah et al. [12] propose a multi-slice scheduling framework for multiple bulk transfers. They formulate the problem as a maximum concurrent multiple flows model and try to improve the throughput of transfers. Rajah et al. continue their work in [13] and propose a novel flexible admission control and periodic scheduling framework to lower the request rejection ratio. All of the works above study the bulk data transfer problem in static networks, and do not consider the practical spatial and temporal dynamics of network resources.

In [14], Laoutaris et al. propose to complete the delay-tolerant bulk transfers via the store-and-forward mode, such that the un-utilized bandwidth resource can be re-used, while saving the transfer cost. They further extend their work in [9], based on the predictable link capacity information, NetStitcher stitches together unutilized time-varying bandwidth across multiple datacenters for bulk transfers, so that the existing link capacity is maximally utilized. However, these works mainly focus on dealing with a single bulk data transfer, the more realistic problem of scheduling multiple bulk data transfers still remains unsolved.

1.3. Proposed approach

In this paper, we take the first step towards congestion avoidance in scheduling multiple bulk data transfers with time constraints. Our work is based on the observation of temporal and spatial patterns in the inter-datacenter traffic. *Temporally*, the inter-datacenter traffic exhibits a strong diurnal pattern [6]. This is because bandwidth resources of datacenters always over-provision to guarantee the peak demands. Suffering from the diurnal pattern of user demands [15], a substantial capacity of the different links is unused during the night. *Spatially*, the traffic distribution among datacenters is unbalanced across the network. This is because datacenters are geographically distributed in different time zones. When datacenters located in one time zone experience their peak traffic, the datacenters in other time zones may be idle [9]. The temporal and spatial characteristics of inter-datacenter traffic provide us more opportunities of network resources multiplexing.

Combining the delay-tolerant feature of bulk data transfers with the inter-datacenter traffic characteristics, we propose to reduce the link traffic load from both the temporal and spatial dimension. To reduce the peak traffic load over time, we propose to adopt the *store-and-forward* approach to complete bulk transfers. When the link load is high, we first store the bulk transferring data temporarily in intermediate datacenters, and forward them to the next node at a later time when more link capacities are available.

A toy example is given to illustrate our idea in Fig. 1. There are two transfers r_1 and r_2 that share the same link with fixed capacity of 3 Gbps. r_1 requires to transfer 400 Gb data within time window [100 s, 300 s], and r_2 requires to transfer 400 Gb data within time window [0 s, 400 s]. Fig. 1(a) shows the end-to-end based scheduling, where r_1 and r_2 obtain constant 2 Gbps and 1 Gbps bandwidth allocation in their respective time windows. We can observe that all the 3 Gbps link capacities are occupied during the time [100 s, 300 s] in such scheduling. If there comes a transfer r_3 with 200 Gb data within time window [100 s, 300 s], it has to be blocked. Fig. 1(b) shows the store-and-forward based scheduling, where r_1 obtains 2 Gbps bandwidth in [100 s, 300 s], and r_2 obtains 2 Gbps bandwidth in intervals [0 s, 100 s] and [300 s, 400 s]. We can observe that the total traffic load remains to be 2 Gbps during the time [0 s, 400 s], since we store the traffic of r_2 temporarily when the peak link load is high, and continue its transfer at a later time when link capacity is available. Benefiting from the store-and-forward based scheduling, the network capability of accommodating flash traffic is improved, and r_3 can be accepted as well.

To balance the bulk data traffic across links between inter-datacenters, we propose to split the bulk data into blocks and transfer them along multiple multi-hop paths. The existing works on balancing traffic [16,17,11] only focus on minimizing the maximally loaded links congestion. However, when scheduling multiple bulk transfers with different data amounts and time constraints, each transfer passes along different paths, leading the lower bounds of each link congestion to become different as well. Thus, only minimizing the maximally loaded links congestion will lead the traffic on other links to be unbalanced and their congestion to be unabated. To minimize all links congestion as much as possible, but not just the highest loaded links, we propose the *lexicographical minimization* as our objective. More specifically, the lexicographical minimization first tries to minimize the traffic of the maximally loaded link in the network. Then, it attempts to minimize the traffic of the second maximally loaded link in the network, *etc.*, until all links congestion are minimized.

Fig. 2 shows a toy example to illustrate the benefits of lexicographical minimization in balancing bulk traffic.