Datacenter Networks
Project Progress Report

Krishangee Bora
Swapnil Pandey
Wasfi Momen

Introduction

Datacenter Networks (DCN) is one of the fastest-growing research topics due to the arrival of cloud computing and the future progress of edge computing. Previously, DCNs were restricted by hardware, geographical location, and strict function domains. With these aspects now unlocked, research for DCNs deal with managing the queue of several, multipurpose flows across wide geographical distances.

From the IEEE Infocom journal, we consider two conference papers on the topic of DCN: “Scheduling Jobs across Geo-Distributed Datacenters with Max-Min Fairness” by Chen et al. and “Multi-Tenant Multi-Objective Bandwidth Allocation in Datacenters Using Stacked Congestion Control” by Tian et al. The former takes a theoretical approach in solving a minimization problem in order to “optimize all the job completion times” and an example network scheduler to handle the queuing of all tasks in a geo-distributed network. The latter tries to solve the bandwidth allocation problem by improving congestion control for use cases within an Application Driven Network (ADN) [7]. In the following sections, we explain the problem that exists with DCNs and the way these papers attempt to solve their respective problems.

Detailed Problem Statement

DCNs wish to provide services for multiple organizations and users with both hard constraints on bandwidth and soft constraints on service guarantees. Research of the general problem on how to compact large data sets, store them in hardware, and virtualize delivery is already integrated to solutions like Apache Spark, Google’s MapReduce [3], and structures like Hadoop Distributed File System which make up cloud computing. The problem is that the demand for large-scale applications now both have the means to get near real-time computability, but DCNs need to optimize for application-specific goals. Applications such as dense-data sensor networks, media content delivery, and social media “oversubscribe” to a DCN model and make it troubling to allocate bandwidth on a wide scale with Quality of Service (QoS).

Part of the solution is the actual selection of tasks in the queue to complete across a distributed network. In a datacenter network, there is much more intra-network communication than inter-communication. Compiled with the fact that datacenters now are globally distributed, this intra-communication cost needs to be optimized. By taking the approach of a linear programming problem, a max-min fairness can allocate a finite share of resources, isolated from other network tasks, and reach an optimal completion time for all network tasks.

Another part of the solution relies in the unique congestion control of each application within a DCN in order to provide some form of service guarantee. TCP only considers a two-party fairness for congestion control but does not consider the congestion occurring in other parts of the network. A data-link layer protocol can be used to limit congestion within and between DCN clusters for latency or deadline goals. A solution can also consider the costs of the edge network between a DCN and a user to optimize congestion even further for multiple flows both interior and exterior parts of a DCN.

Motivation

As mentioned before, DCNs represent a highly evolutionary subset of general network theory. DCNs have since moved from oversubscribed network architecture of Three-Tier DCN and now to more cube, cell, or pod like structures where network broadcasts and objectives can be separated and isolated to work on certain tasks. The issue now comes with supporting the edge user with certain service guarantees such that it doesn't have an overall disruption to the DCN.

For example, Facebook's [1] datacenters ran into the issue of oversubscription in a Three-Tier DCN where clusters had huge communication costs compared to the cost of sending the information to the user. While this problem was alleviated with a different physical network structure, a software solution should be able to consider the flows of inter-communication between DCN clusters with compatibility of post-cloud infrastructure improvements.

Since cloud networks (and by extension DCN) are multipurpose, the ability to separate work into queues and manage their flows becomes a critical issue to model and implement. Research of machine learning or other statistical approaches will be relevant in the near future but dealing with the nature of DCNs will prove a foundation in which other solutions can base on, just like how previous concepts like MapReduce defined the cloud era.

Related Work

Due to the ubiquity of DCNs, large scale technology companies research and implement their own versions of DCN services. Facebook, Microsoft, Google, and Amazon have all contributed through various open-source technologies and frameworks presented. Google especially has published many known edge network use cases. For example, Google develops its machine learning platform via hosted datacenters [4].

Most of the known research lies in the IEEE SIGCOM AND InfoCom journals in which authors present a framework that uses both network switch configuration hardware and different algorithms in order to pursue a goal. The more recent papers above specify that the research focus on single-purpose DCNs and do not provide general approaches to tasks involved. The industry still suffers from oversubscription and future complex network configurations, so providing a software solution can alleviate the process of deciding multi-objective queuing in an edge network.

Plan of Attack

Our plan is to exploit vulnerabilities of the algorithms provided in the papers. The problems are dependent on each other since calculations must include communication costs at each step of the process.

One aspect to exploit could be the scalability aspect of the DCN. Both papers used available resources to model their respective problems and claim that with additional hardware resources their solutions would run faster with either a better CPU or with additional servers. Utilizing

GSU’s HPC cluster could be used as an experimental setup to see if either paper’s algorithm truly does stand when considering large-scale DCNs.

A better aspect is to compromise the solution within a particular use case and produce a non-optimal solution to the problem. For a multi-tenant and objective purpose DCN, injecting congestion at the different congestion control structures could produce a scenario in which they fight against each other for bandwidth or release too many resources so no jobs complete and fall out the queue.

Research Progress

The two papers mentioned constitute the two aspects of a general DCN that requires a service guarantee. Our current aim is to introduce a case which results in non-optimal solutions which degrade network performance. DCNs are very susceptible to huge loss if even one cluster fails to handle congestion or allocate resources accordingly as seen in the case of Facebook’s DCN [1].

Our current progress is involved into researching more about the current network configuration setups presented in DCNs such as fabric switches and isolated pod clusters, reading the references of each paper to understand prior work in the field, and attempting to create a small implementation of the exploited solutions.

For the technical implementation, the algorithms are provided within the papers create suitable pseudocode to implement. We’ve decided to use the python programming language combined with an existing network virtualization framework to simulate our exploit. Programs such as Mininet [6] or NS2 [7] seem to be very promising in modeling a large-scale network on single laptops where exploits can be compared with small samples and then generalized to the sample sizes that the papers provide.

Bibliography

[1] Alexey Andreyev, “Introducing data center fabric, the next-generation Facebook data center network”, 2014. [Online]. https://code.fb.com/production-engineering/introducing-data-center-fabric-the-next-generation-facebook-data-center-network/

[2] C. Tian *et al*., "Multi-tenant multi-objective bandwidth allocation in datacenters using stacked congestion control," *IEEE INFOCOM 2017 - IEEE Conference on Computer Communications*, Atlanta, GA, 2017, pp. 1-9. http://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=8056944&isnumber=8056940

[3] Jeffery Dean and Sanjay Ghemawat. “MapReduce: Simplified Data Processing on Large Clusters”. 2004. In *OSDI 2004*. https://static.googleusercontent.com/media/research.google.com/en//archive/mapreduce-osdi04.pdf

[4] Jeffrey Dean, “Build and train machine learning models on our new Google Cloud TPUs”, 2017. [Online] https://blog.google/products/google-cloud/google-cloud-offer-tpus-machine-learning/

[5] L. Chen, S. Liu, B. Li and B. Li, "Scheduling jobs across geo-distributed datacenters with max-min fairness," IEEE INFOCOM 2017 - IEEE Conference on Computer Communications, Atlanta, GA, 2017, pp. 1-9. http://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=8056949&isnumber=8056940

[6] Mininet: An Instant Virtual Network on your Laptop (or other PC). [Online]. http://mininet.org/

[7] The Network Simulator NS-2 [Online]. https://www.isi.edu/nsnam/ns/

[8] Yi Wang, Dong Lin, Changtai Li, Junping Zhang, Peng Liu, Chengchen Hu, and Gong Zhang. 2016. “Application Driven Network: providing On-Demand Services for Applications.” In *Proceedings of the 2016 ACM SIGCOMM Conference* (SIGCOMM '16). ACM, New York, NY, USA, 617-618. https://dl.acm.org/citation.cfm?id=2959075