Jupiter Rising: A Decade of Clos Topologies and Centralized Control in Google's Datacenter Network

Authors: Arjun Singh, Joon Ong, Amit Agarwal, Glen Anderson, Ashby Armistead, Roy Bannon, Seb Boving, Gaurav Desai, Bob Felderman, Paulie Germano, Anand Kanagala, Jeff Provost, Jason Simmons, Eiichi Tanda, Jim Wanderer, Urs Hölzle, Stephen Stuart, and Amin Vahdat

Published: *Proc. of ACM SIGCOMM '15*, 45(4):183-197, Oct. 2015.

Reviewers: Allison McDonald and Buting Ma

I. Summary

This paper is written by a team of researchers at Google. They discuss the evolution of the Google data center over the last decade, describing five iterations of architecture and extracting common and persistent themes from the different designs. The authors identify three themes as being the most significant in the evolution of the data center topology, which has needed to adapt to an exponential increase in bandwidth requirements over the last 10 years as the Internet has exploded and Google has become an Internet giant. First, they focus on the choice of using Clos topologies for their building-level networks. Importantly, they note that commodity silicon is surprisingly sufficient for their needs and much more cost-effective. Second, because the network remains relatively static and the topology is known to them, they choose to implement their own centralized control mechanism rather than using more typical decentralized routing and management protocols, which are designed for managing many autonomous networks. Third, they determine that their design choices have facilitated dynamic growth of their own data centers locally and globally and interaction with the wide-area network.

II. Strengths

1. Clos Topology

This paper is a good example of real application of Clos topology, which is a multi-stage network topology. The reason for applying this topology in the datacenter is that it has good scalability. As the authors said in the second section of the paper, "we start with the key insight that we could scale fabrics to near arbitrary size by leveraging Clos topology". Because of its scalability, they were able to increase the bandwidth by adding more switches. As the traffic coming in and going out of the datacenter is increasing rapidly (nearly exponential according to Figure 1), such scalability is critical to handle the ever-increasing demand of bandwidth.

The second advantage brought by Clos topology is that it provides redundancy. In each stage, such as spine block and edge aggregation block, there are many nodes (switches) operating the same function at the same level. This feature provides robustness for the system, because even if some of the nodes fails, the others can take over their work and keep the whole system running, for example, the re-election of Firepath master. The failure of switches can be handled by the routing protocol in several hundreds of milliseconds as described in Section 5.

The third advantage of Clos topology is that it can provide safer external connectivity. By building a separate aggregation block for external connection, fewer nodes in the system will be exposed to the external network. As the authors described in section 5, it "limits the blast radius from an external facing configuration change" and provides a "limited place where we have to integrate out in-house IGP with external routing protocol".

These are all valuable experience, especially the third one, on applying Clos topology in real production environment, which does not usually appear in a theoretical paper.

2. *Centralized routing*

One thing that makes us interested while reading the paper is the routing protocol they used in their datacenter. Unlike the traditional distributed routing system, they applied a centralized one (Firepath). They do this to better support Clos topology, which is multipath and equal-cost. One of the spine blocks is elected as master, and is responsible for topology distribution. Other switches, called clients, will transmit their connection states to the master. The master constructs the topology and update it to the clients. Clients will calculate the routing based on the topology information from the master. To provide robustness to this centralize algorithm, the system has a subset of spine blocks being the master candidate. When the master fails, the system will detect the failure and elect the spine block who has the latest topology information to be a new master.

This routing system is much different from what we see in the distributed version. Unfortunately, the paper didn't show much detail about it. However, the centralized routing protocol should be a powerful variation that can benefit specific topologies such as Clos topology.

3. *Advantages of real-world implementation study*

There are several advantages to a paper being written by researchers at a large organization such as Google. The authors, as employees, have an intimate knowledge and access to the data centers they are describing and studying. They themselves are the people who have influenced the decisions described in the paper over the last decade, and have seen the design choices go from concepts to implementation in the data centers in production.

The authors are also able to describe the evolution of the Google data center over a large period of time. Most studies are necessarily limited in time and scope. By writing a descriptive paper from the vantage point of the organization, the authors were able to produce a longitudinal study that is not possible from an external academic position.

Furthermore, the authors can use this paper to shape or direct the cutting edge of network research. With the insight that production-level network access gives, the authors are in a position to identify the real problems that the academic and research community could be working on that would address problems that exist in-the-wild — even if we must acknowledge that they do this for their own benefit most of all. For example, in this paper, the authors discuss how Google has moved away from using open-source distributed routing protocols because their network requirements are different than those of other networks. This is counter to the academic community's emphasis on research on distributed and decentralized routing, which is obviously beneficial in wide-area network routing in the more chaotic and unstable Internet. This may have engendered research questions in the direction of centralized control mechanisms.

III. Improvements and Extensions

1. *Dearth of quantitative analysis*

This paper is a qualitative discussion of the design choices and developments that Google has gone through in the last decade. Because Google was only iterating on its own data center architecture, the paper has only minimal quantitative analysis of Google's design in comparison with other industry standards. Without this direct and measurable comparison, the benefits of the choices made by Google are less useful to the rest of the community. The paper would be strengthened with a more rigorous performance analysis.

2. *Description of protocol design*

This paper is on software defined network in google's datacenter, so the most important and interesting thing should be the "software", for example, their centralized routing protocol. However, in this paper, the authors talk much more about hardware, like how they connect switches together. They only talk about what they do, but almost nothing about how they do it.

Hence, a possible extension of this paper could be a more detailed introduction of their protocol design and a quantitative analysis on how it behaves on specific topology. The reason why the authors didn't do this even on their oldest implementation, which was already over 10 years old when they publish this paper, is because their current system could still be a refined version of the old one, hence there are still some critical parts that they want to keep proprietary.

3. *Style and purpose of the paper*

As we discussed in an earlier section, descriptive papers from industry can lend valuable insight into how well academic and theoretical design choices work in practice, and which problems arise that were perhaps not envisioned from a sterile or simulated environment. However, I believe their advantage is limited. First, because Google is an entity with corporate competitors, it could be argued that its contributions to the academic body of research are made with a strategic and financial calculus. It is highly unlikely that the most recent data center generation described in the paper — Jupiter, deployed in 2012 — was still the most up-to-date design in 2015, when the paper was published. This puts into question the utility of sharing design choices that do not reach up to the state-of-the-art. Additionally, as was mentioned in the previous section, many specific details about the routing protocols were left out of the paper, presumably for the purpose of protecting proprietary software. If the information being shared here is only part of the complete picture, one could question how useful the analysis is. Second, very few organizations have similar data center requirements as Google. A description of its internal network decisions, while interesting, really only benefits other large, centrally organized entities, most of whom will be other corporations — and even only few of them are operating at the scale that Google does.