ISSN(Online): 2320-9801
ISSN (Print): 2320-9798

International Journal of Innovative Research in Computer and Communication Engineering

(An ISO 3297: 2007 Certified Organization)

Website: www.ijircce.com

Vol. 5, Issue 7, July 2017

Measurability, Adaptability and Efficiency in Data Center Network using Space Shuffling

Vanishree M L

Assistant Professor, Department of CSE, Global Academy of Technology, Bengaluru, Karnataka, India

ABSTRACT: The increasing need of cloud and big data applications requires data center networks to be scalable and bandwidth-rich. Rigid topologies to increase network bandwidth are often used by data centre network architectures. Incremental network growth is a major limitation. Network architecture and protocols achieve high throughput, routing and forwarding scalability and flexibility for incremental growth. Unfortunately, existing data center network architectures focus on one or two of the above properties and pay little attention to the others. Here Space Shuffling in data centre applies greedy routing to achieve high throughput, routing and forwarding scalability and flexibility for incremental growth. The proposed greedy routing protocol of S2 effectively exploits the path diversity of densely connected topologies and enables key-based routing. Experimental studies show that S2 provides high bisectional bandwidth and throughput near-optimal routing path lengths, extremely small forwarding state, fairness among concurrent data flows and resiliency to network failures.

KEYWORDS: Data center networks, routing protocols, cloud computing, greedy routing, key-based routing.

I. INTRODUCTION

Data center networks, being an important computing and communication component for cloud services and big data processing, require high inter-server communication bandwidth and scalability [1]. Network topology and the corresponding routing protocol are determinate factors. Many applications of current data center networks are data-intensive and require substantial intra-network communication, such as MapReduce [2], Hadoop [3], and Dryad [4]. The densely connected topologies provide high bisection bandwidth and multiple parallel paths between any pair of servers. However Routing protocols that can effectively exploit the network bandwidth and path diversity are essential. simply providing high-bandwidth is not enough. The routing protocol should achieve high throughput on the network topology. In addition, the routing protocol should result in short paths to achieve low latency.

II. RELATED WORK

To improve data center performance such as bisection bandwidth, flexibility and failure, studies have proposed a number of new network topologies. Alfare set.al. [9] proposed a multi-rooted tree structure called fattree that provides multiple equal paths between any pair of servers and can be built with commodity switches. vl2 [10] is a data center network that uses flat addresses and provides layer-2 semantics. It's topology is a close network which is also a multi-rooted tree. To achieve high bisection bandwidth, including dcell , bcube , camcube and smallworld in data centers network designs use direct server-to-server connection in regular topologies [7]. However, none of these designs have considered the requirement of incremental growth of data centers to provide network flexibility and support. Jellyfish [5] is a recently proposed data center network architecture that applies random connections to allow arbitrary network size and incremental growth. It can be built with any number of switches and servers and can incorporate additional devices by slightly changing the current network using k-shortest path routing. Compared to fattree, jellyfish achieves higher network throughput [9] and supports more servers by using the same number of switches. Implementation of k-shortest path as suggested in the mpls, the expected number of forwarding entries per switch is proportional to k_n log n, where n is the number of switches in the network. In addition, k-shortest path algorithm is extremely time consuming its complexity is o(k_n(m + n log n)) for a single source (m is the number of links). This may result in slow convergence under network dynamics. Hence, jellyfish may suffer from both data plane and control plane scalability problems. It provides another multi-path solution for jellyfish, but the throughput of jellyfish may be degraded, study shows near-optimal-throughput topology design for both homogeneous and heterogeneous networks. However, it does not provide routing protocols that achieves the throughput in practice. As a scalable solution, greedy routing has been applied to enterprise and data center networks [7] [8]. Camcube employs greedy routing on a 3d torus topology to satisfy specific requirements. It provides an API for applications to implement their own routing protocols, called symbiotic routing. The small-world data centers (swdcs) are built with directly connected servers in three types: ring, 2d torus, and 3d hex. Torus[8] provides layer-2 semantics for all three network architectures [7] [8]. Swdc, jellyfish, and S2 all employ randomness to build physical topologies. However, their different logical organizations and routing protocols they demonstrate substantially have different performance. Jellyfish provides higher throughput using k-shortest path routing, but it sacrifices forwarding table scalability. S2 gets the best of both worlds. It uses greedy routing on randomly assigned coordinates in multiple spaces to achieve both high-throughput routing and small forwarding state.

III. PROPOSED ALGORITHM

Algorithm 1: Greediest Routing on Switch S

Input: Coordinates of all neighbours of switch S, destination addresses $\{X_t, \vec{ID}\}$.

```
if $\vec{X_s} = \vec{X_t}$
Then h ← the server connected to s with identifier ID
Forward the packet to h
return
Compute MCDL $(\vec{X_v}, \vec{X_t})$ for all s's neighbour switch v
Find v0 such that MCDL $(\vec{X_{v0}}, \vec{X_t})$ is the smallest
Forward the packet to v0 Greediest routing on S2 topologies provides delivery guarantee and loop-freedom.
```

Algorithm 2: Multi-Path Routing on Switch S

Input: Coordinates of all neighbors, destination addresses $\{X_t, \vec{Id}\}$

```
if $\vec{X_s} = X_t$
then h ← the server connected to s with identifier ID
Forward the packet to h
return
if the packet is not from a server connected to s
then perform greediest routing
return
$V \leftarrow \emptyset$
for each neighbour v of S
do if $MCDL(\vec{X_v}, \vec{X_t}) < MCDL(\vec{X_s}, \vec{X_t})$
then $V \leftarrow V \cup \{v\}$
Select v0 from V by hashing the source and destination addresses and ports
Forward the packet to v0
```

IV. COMPARISON

We evaluate the following performance criteria of S2 based on:-

A. *Bisection Bandwidth:*

We compare the minimum bisection bandwidth of S2, Jelly- fish, SWDC, and FatTree. The ratio of server number to switch number in above two configurations are 4.8:1 and 12.8:1 respectively. In Figure 1, we show the bisection bandwidth of S2, FatTree, and Jellyfish, in the two server to-switch ratios.

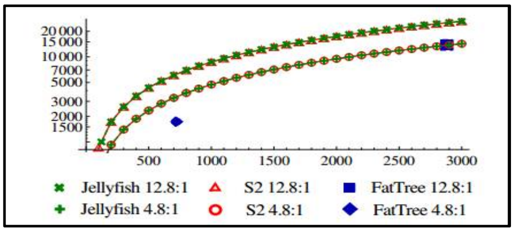

Fig. 1: Bisection bandwidth of S2, FatTree, and Jellyfish for two sever-to-switch ratios (12.8:1 and 4.8:1)

B. *Ideal Throughput:*

We model the computation of ideal throughput as a maximum multi-commodity network flow problem: each flow is a commodity without hard demand. We need to find a flow assignment that maximizes network throughput while satisfying capacity constraints on all links and flow conservation. We show the throughput versus the number of servers of a typical 10-port 125-switch network in Figure 2.

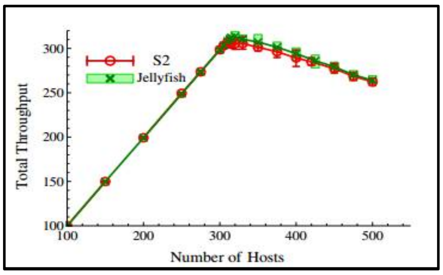

Fig. 2: Ideal throughput of S2 and Jellyfish for a 125-switch network

C. *Scalability:*

We consider each coordinate as an entry and compare the number of entries in forwarding tables. In practice, a coordinate requires much less space than a forwarding entry. Even though we give such a disadvantage to S2, S2 still shows huge lead in data plane scalability. Figure 3 shows the average and maximum forwarding table sizes of S2 and Jellyfish in networks with 10 inter-switch ports. The number of entries of S2 is less than 500 and does not increase when the network grows.

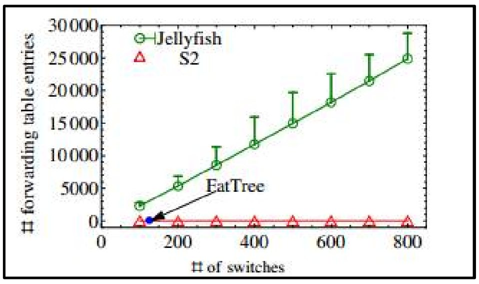

Fig. 3: Forwarding state of S2, Jellyfish, and FatTree

D. *Practical Throughput:*

It conducts experiments to measure the practical throughput of S2, SWDC, FatTree and Jellyfish for both single-path and multi-path routing. In Figure 4, we show the throughput of S2 and FatTree on networks in the FatTree's configuration of number of switches and switch-server ratio. We measure the throughput on a S2 network with the same number of switches and servers. The throughput is normalized to 100 with respect to the bisection bandwidth of the network. S2 shows slightly higher throughput than that of FatTree in all topologies.

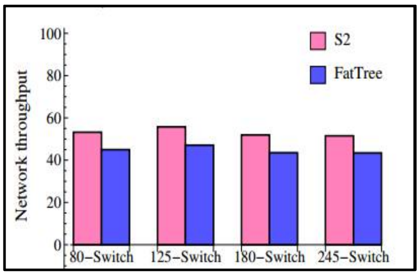

Fig. 4: S2 throughput on networks in FatTree configuration

Figure 5 shows that even under the S2 configuration, S2 provides higher throughput than all three types of SWDC especially when multi-pathing is used. We only show SWDC 2D in remaining results, as it is a middle course of all three types.

E. *Flow Completion Time:*

It evaluate both all-flow and per-flow completion time of data transmission. Figure 6 shows the time to complete transmitting all flows in the same set of experiments.

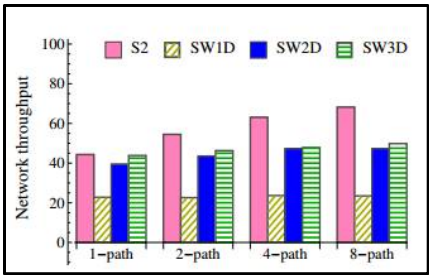

Fig. 5: Throughput of a 400-switch network in SWDC Configuration

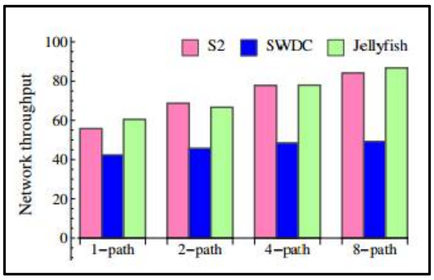

Fig 6: Throughput of a 250-switch 500- server network

V. RESULTS

We conduct extensive experiments to compare the proposed data center networks S2 with two other recently existing data center network, SWDC and Jellyfish. S2 provides shorter routing paths and higher throughput. Our results show that S2 is the best of both existing system. It demonstrates the importance of lead in scalability while provides likewise high throughput and bisectional bandwidth.

VI. CONCLUSION AND FUTURE WORK

The aim of this paper is to propose a novel data center network architecture called Space Shuffle (S2) which achieves high-bandwidth, flexibility, and routing scalability. The importance of this paper is in terms of the greediest routing protocol of S2, which is the first greedy routing protocol which achieves high-throughput multipath routing. In addition, S2 also supports the efficient key-based routing for various data center applications.

It conducts extensive experiments to compare S2 with two recently proposed data center networks, SWDC and Jellyfish. Our results show that S2 achieves the best of both worlds. S2 provides shorter routing paths and higher throughput, compared to SWDC.S2 demonstrates significant lead in scalability while provides likewise high throughput and bisectional bandwidth compared to Jellyfish, We expect greedy routing using multiple spaces may also be applied to other large-scale network environments such as peer-to-peer systems due to its scalability and efficiency.

REFERENCES

[1] A. Greenberg, J. Hamilton, D. A. Maltz, and P. Patel, “The cost of a cloud: Research problems in data center networks,” ACM Sigcomm CCR, 2008.
[2] J. Dean and S. Ghemawat, “Mapreduce: simplified data processing on large clusters,” Communications of the ACM, 2008.
[3] “Apache hadoop,” http://hadoop.apache.org/.
[4] M. Isard, M. Budiu, Y. Yu, A. Birrell, and D. Fetterly, “Dryad: distributed data-parallel programs from sequential building blocks,” in Proc. of ACM EuroSys, 2007.
[5] https://people.inf.ethz.ch/asingla/papers/jellyfish-nsdi12.pdf
[6] M. Yu, A. Fabrikant, and J. Rexford, “Buffalo: Bloom filter forwarding architecture for large organizations,” in Proc. of ACM CoNEXT, 2009.
[7] J.-Y. Shin, B. Wong, and E. G. Sirer, “Small-world datacenters,” in Proc. of ACM SOCC, 2011.
[8] C. Qian and S. Lam, “ROME: Routing On Metropolitan-scale Ethernet,” in Proc. of IEEE ICNP, 2012.
[9] M. Al-Fares, A. Loukissas, and A. Vahdat, “A scalable, commodity data center network architecture,” in Proc. of ACM SIGCOMM, 2008.
[10] A. Greenberg, J. R. Hamilton, N. Jain, S. Kandula, C. Kim, P. Lahiri, D. A. Maltz, P. Patel, and S. Sengupta, “Vl2: a scalable and flexible data center network,” in Proc. of ACM SIGCOMM, 2009.