RHODA Topology Configuration Using Bayesian Optimization

Maotong Xu, Min Tian, Eytan Modiano, Suresh Subramaniam

RHODA Topology Configuration Using Bayesian Optimization*

Maotong Xu[1][0000−0003−4513−3205], Min Tian[1][0000−0002−3352−3145], Eytan Modiano[2][0000−0001−8238−8130], and Suresh Subramaniam[1][0000−0003−1501−5953]

[1] George Washington University, Washington, DC 20052, USA
{htfy8927, mtian39, suresh}@gwu.edu
[2] Massachusetts Institute of Technology, Cambridge, MA 02139, USA
modiano@mit.edu

Abstract. The rapid growth of data center traffic requires data center networks (DCNs) to be scalable, energy-efficient, and provide low latencies. Optical Wavelength Division Multiplexing (WDM) is a promising technique to build data centers comprising millions of servers. In [24], a WDM-based Reconfigurable Hierarchical Optical DCN Architecture (RHODA) was presented, which can accommodate up to 10+ million of servers and a variety of traffic patterns. RHODA also saves tremendous amounts of power and cost through its extensive use of passive optical devices, and minimal use of power-hungry and costly devices. RHODA achieves high throughput through reconfigurable clustering of racks of servers. In this paper, we focus on the design of the cluster topology (also called inter-cluster network). Given the pair-wise cluster traffic, our objective for the cluster topology is to minimize the average hop length. In [24], a simple variant of the Hungarian algorithm that maximizes the one-hop or direct traffic among clusters was used. In this paper, we leverage the Bayesian Optimization (BO) framework and propose a fast algorithm to minimize the average number of hops in the inter-cluster network of RHODA. To the best of our knowledge, this is the first paper that employs BO to optimize optical DCN performance. We present our design decisions and modifications to BO based on the network constraints. Results show that BO can achieve optimal or near-optimal results, and outperforms a well-known regular topology (Gemnet) and the Hungarian-based method by up to 13% and 58%, respectively.

Keywords: Bayesian Optimization · data center networks · inter-cluster network · RHODA.

1 Introduction

Data center (DC) traffic has experienced dramatic growth, increasing at an annual rate of 31%, and will reach 3.3 ZB per year [22]. To store and process

* Supported by NSF award CNS-1618487 and CNS-1617091

huge amounts of data, DCs will consist of millions of servers. For example, Microsoft owns over one million servers and a single DC alone contains over 250,000 servers [23]. On the other hand, interactive applications, e.g., web searches and social media, require low network latency. For example, the acceptable latency range for stock exchange transactions is 5-100 ms [20]. Traffic within a DC is usually not uniformly distributed. For instance, measurements on a 1500-server Microsoft production DC network (DCN) reveal that only a few ToRs (Top-of-the-Rack, used as an alternative for rack) are hot and most of a ToR's traffic goes to a few other ToRs [11]. Finally, power consumption of data centers will reach 140 billion kilowatt-hours annually by 2020, and it will cost $13 billion annually [16].

Conventional electrical DCNs (e.g., FatTree [1], Bcube [8], VL2 [7] and Flattened Butterfly [5]) are built using a multi-layer approach, with a large number of switches at the bottom level to connect with servers/racks and a few high-end switches located at the upper layers to aggregate and distribute the traffic. Those networks rely heavily on high-cost and power-hungry electrical switches, and operators are facing limited scalability, high latency and low energy efficiency problems.

Optical Wavelength Division Multiplexing (WDM) is a promising technology for meeting the traffic demand of data centers. It can support hundreds of wavelengths per fiber and 100 Gbps transmission rate per wavelength. Moreover, large-scale optical switches consume less power per bit/s, making the network architecture scalable and energy-efficient. RHODA [24] is a WDM-based reconfigurable hierarchical optical DCN architecture. The architecture can scale up to 10+ million servers and support various traffic patterns. RHODA achieves high throughput through reconfigurable clustering of racks of servers. Racks with large amounts of mutual traffic can be grouped into clusters, and a high-bandwidth intra-cluster network connects racks within a cluster. The clusters are connected through an inter-cluster network which is also reconfigurable based on the traffic demands among clusters. The inter-cluster network uses wavelength selective switches (WSSs) and optical space switches to achieve topology reconfigurability. The resulting inter-cluster network topology is degree-constrained because of port count limitation of WSSs. RHODA saves large amount of power and cost by extensively using passive devices (couplers, Arrayed Waveguide Grating Routers (AWGRs), and mux/demuxes), and minimally using power-hungry and expensive devices (e.g., WSSs).

Once clusters are defined and the intra- and inter-cluster topologies have been established, flows are routed within and between clusters over those topologies, possibly using multiple hops, where each hop corresponds to one optical transmission and reception. As packets have to be converted from optical to electrical to optical form for each hop, and queued up for transmission at intermediate nodes (ToRs), the network latency is largely determined by the number of hops. The focus of this paper is on the design of the inter-cluster network topology to minimize the average number of hops (weighted by traffic demands). In [24], the inter-cluster topology is constructed using a Hungarian-based method [12]. The

method first builds a circle/ring among clusters to first make the inter-cluster topology connected, and then the Hungarian algorithm is iteratively used to find a perfect bipartite matching to maximize the total traffic over the directly connected clusters (i.e., single-hop traffic). General regular topologies, e.g., Hypercube [6] and Gemnet [10], are attractive for uniform traffic, but they do not minimize the average hop distance for skewed traffic. In this paper, we propose an algorithm for inter-cluster topology design based on Bayesian Optimization (BO), and demonstrate that it produces optimal or near-optimal results very quickly.

Bayesian optimization (BO) [14] is a powerful tool to find optimal or near-optimal solutions for black-box problems, i.e., it is suitable in situations where a closed-form expression for the objective function is unknown. BO first uses a statistical model, e.g., Gaussian Process, to fit the objective function. Then, a pre-defined acquisition function is used to locate the next point to sample. The acquisition function can trade off between exploration and exploitation. This means that BO does not get stuck in local optima and can often find the global optimal solution.

In this paper, we use BO to find an inter-cluster topology for RHODA to minimize the average traffic-weighted hop distance. We describe our design decisions – the choice of prior model, acquisition function and optimizing algorithm. Further, we describe several modifications to BO based on topology constraints. Finally, we compare BO with the optimal solution obtained through solving an integer-linear program (ILP), and with the hop distances in the Gemnet topology and using Hungarian-based method. Our results show that BO can achieve optimal or near-optimal results, and outperforms Gemnet and Hungarian-based method by up to 13% and 58%, respectively.

The rest of this paper is organized as follows: Section 2 briefly describes the RHODA architecture. Section 3 presents the ILP model and our design decisions and modifications of BO based on network requirements. Section 4 presents performance evaluation results, and Section 5 concludes the paper.

2 RHODA Architecture

In this section, we briefly describe the RHODA architecture. The reader is referred to [24] for more design details of RHODA.

As Fig. 1 illustrates, RHODA is a DCN architecture consisting of N servers grouped into M racks, so that there are $m = N/M$ servers per rack. Each rack has a ToR switch used for both electronic switching of the packets within a rack and for communication with other racks. Each ToR connects to all of the m servers within the rack. For communication to and from other ToRs, each ToR has $T^{\text{intra}} + T^{\text{inter}}$ tunable transceivers (TRXs). RHODA consists of five parts, namely, cluster membership network (CMN), intra- and inter- signals demultiplexing part, intra-cluster network, inter-cluster network, and intra- and inter- signals multiplexing part.

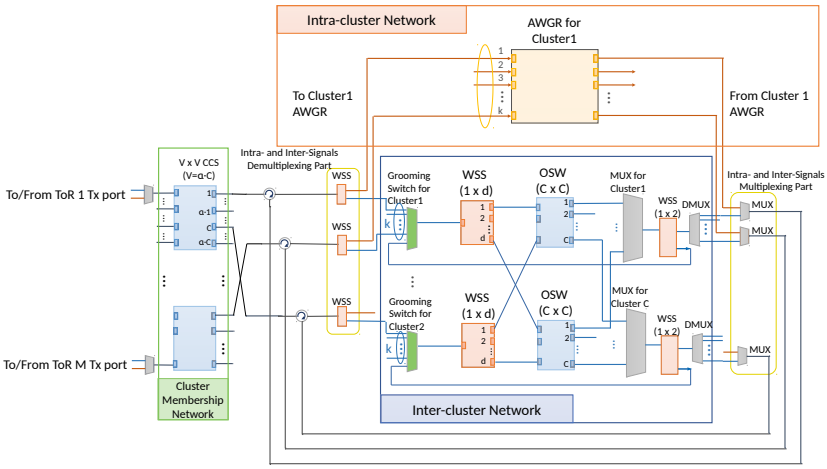


Fig. 1: The RHODA architecture. M: number of racks; k: number of racks per cluster; d: egress degree of a cluster; C: number of clusters.

RHODA lets cluster membership be reconfigurable. This is achieved by the CMN using a set of k/α $V \times V$ cluster configuration switches (CCS), where $V = \alpha \cdot C$ and α is a positive integer parameter that is chosen as a trade-off between the cost and complexity of the CMN and the reconfiguration flexibility.

Since most of the heavy communication in a data center is carried over small subsets of ToRs [17] (and these ToRs would ideally be configured to the same cluster), RHODA equips each cluster with a $k \times k$ AWGR to support large amounts of intra-cluster traffic. In the inter-cluster network, each cluster can be considered as the smallest communication element. Flows from racks are merged (using optical-to-electrical-to-optical conversion) by grooming switches (GSs) to reduce the number of wavelengths needed. The communication graph between clusters (i.e., the cluster topology) is then determined by C $1 \times d$ WSSs and d $C \times C$ optical space switches (OSWs). In particular, each cluster can send signals to up to d other clusters. Demultiplexers (DMUXs) split signals carried by different wavelengths. A signal carried on wavelength w is forwarded to the $\lceil \frac{w}{k} \rceil$th port of the DMUX.

3 Topology Configuration Algorithm

Our objective is to minimize the traffic-weighted average number of hops in the inter-cluster network of RHODA. We first present an ILP formulation, and then present the BO framework adapted to solve our topology design problem.

3.1 ILP Formulation

The network has a set of nodes (which are the clusters) $\mathcal{C}$, and a traffic matrix T. The number of nodes/clusters, $C = |\mathcal{C}|$. Each node has both an in-degree and

out-degree constraint of d. The following ILP constructs the degree-constrained and directed topology, and minimizes the traffic-weighted average hop distance over all node-pairs, assuming that multi-hop traffic is routed over the shortest path in the topology.

$$\min \frac{\sum_{i,j\in\mathcal{C}} T_{ij} \cdot H_{ij}}{\sum_{i,j\in\mathcal{C}} T_{ij}} \tag{1}$$

$$\text{s.t.} \quad L_{ij} \in \{0,1\}, \quad \forall i,j \in \mathcal{C} \tag{2}$$

$$\sum_{j\in\mathcal{C}, j\neq i} L_{ij} \leq d, \quad \forall i \in \mathcal{C} \tag{3}$$

$$\sum_{j\in\mathcal{C}, j\neq i} L_{ji} \leq d, \quad \forall i \in \mathcal{C} \tag{4}$$

$$L_{ii} = 0, \quad \forall i \in \mathcal{C} \tag{5}$$

$$H_{ii} = 0, \quad \forall i \in \mathcal{C} \tag{6}$$

$$0 < H_{ij} < C, \quad \forall i,j \in \mathcal{C}, i \neq j \tag{7}$$

$$H_{ij} = 1, \quad \textit{for } L_{ij} = 1, \; \forall i,j \in \mathcal{C}, i \neq j \tag{8}$$

$$H_{ij} = \min(H_{ik} + ((L_{kj} - 1) * (-a)) + 1)), \quad \textit{for } L_{ij} \neq 1, \; \forall i,j,k \in \mathcal{C}, \; i \neq j, j \neq k, i \neq k \tag{9}$$

In the above formulation, the decision variables are the L_{ij}'s, where $L_{ij} = 1$ implies the establishment of a link between node i and j, and no link is established if $L_{ij} = 0$. H_{ij} represents the number of hops on the shortest path from i to j. In this ILP formulation, (1) is our objective, to optimize the traffic-weighted average number of hops. Inequalities (3) and (4) ensure that the in-degree and out-degree of each node is not more than D. The values of H_{ij}'s are determined by (6), (7), and (8). If a link from node i to j exists, H_{ij} is 1; on the other hand, if there is no link from i to j, H_{ij} is defined as the minimum value of $H_{ik} + ((L_{kj} - 1) * (-a) + 1)$, where a is defined as a large positive integer, so that if L_{kj} does not exist, $H_{ik} + ((L_{kj} - 1) \cdot (-a) + 1)$ will be an integer far larger than $C - 1$.

3.2 Bayesian Optimization

The ILP is too time-consuming for large instances (e.g., when there are more than 10 clusters and the degree is 4 or higher) and cannot be used in an online setting where the traffic dynamically changes and the topology needs to be configured quickly so that packets are not blocked for extended periods while the configuration takes place. We therefore seek a fast heuristic algorithm to minimize average hop distance. In [24], the topology is constructed by adapting the

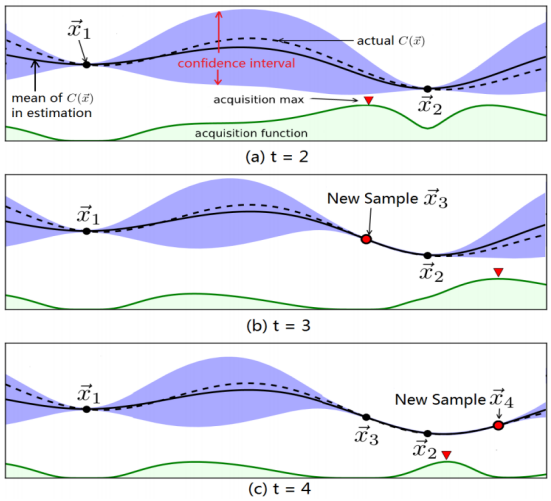


Fig. 2: An example of BO's working process [2].

well-known Hungarian assignment method [12] for finding the maximum-weight matching in a bipartite graph. Once a circle is formed among the clusters to ensure that the topology is connected, our approach in [12] iteratively applied the Hungarian algorithm until the degree constraint is violated. The resulting topology, however, may turn out to have a large *average* hop-distance. Grid Search and Random Search [3] can be used for our problem, but time consumption would be too large.

In this paper, we turn to a widely used framework to solve optimization problems, namely, Bayesian optimization (BO). For instance, it is useful for solving the following problem (shown in one dimension here):

$$\min_{x \in A} f(x),$$

where structure/concavity of objective function $f(x)$ is unknown but can be observed through evaluations. Further, BO aims to minimize the number of evaluations to save optimization cost/time.

Figure 2 shows an example of running BO on a 1D problem. The blue area is an area in which the objective function is expected to lie in with, say, 95% probability. The optimization starts with two points, i.e., $\boldsymbol{x}_1$ and $\boldsymbol{x}_2$. At each iteration, BO decides that the next point is sampled at the argmax of a predefined acquisition function. As shown in Fig. 2, $\boldsymbol{x}_3$ and $\boldsymbol{x}_4$ are sampled in the next two iterations.

BO consists of two essential components, i.e., a statistical model for modeling the objective function, and an acquisition function for deciding the next point to sample. First, BO evaluates the objective function based on multiple randomly chosen initial points (e.g., 5). Then, BO iteratively uses all available data to update the posterior probability on the objective function, use the current posterior probability to compute the acquisition function, and argmax the

acquisition function to find the next point at which to evaluate the objective function. The pseudocode is shown in Alg. 1.

Algorithm 1: Bayesian Optimization Algorithm

```
1: Use a statistical model to model the objective function f
2: while Stopping condition is not satisfied do
3:     Update the posterior probability distribution
4:     Calculate the acquisition function using posterior distribution
5:     Argmax the acquisition function to locate next point
6:     Evaluate f with sample point
7:     Update dataset
8: end while
```

To leverage BO to find a good topology in our problem, we need to make several design decisions and modifications based on our requirements.

Choice of prior model We choose Gaussian Process as the statistical model for modeling the objective function. The Gaussian Process has desirable features, e.g., it is non-parametric and the model is approximately Gaussian (central limit theorem).

Random point We provide several random "points" for BO's first step. In our case, we provide several adjacency matrices (representing random degree-constrained topologies) as input to evaluate $f(x)$. We generate such a random matrix as follows. First build a circle among nodes to guarantee that all nodes are connected. We then randomly assign $d-1$ 1s (edges) on each row of the adjacency matrix. Then, we iterate over all columns and rows of the adjacency matrix in random order and adjust the entries to guarantee that each node has no more than d ingress/egress edges. The pseudocode is shown in Alg. 2.

Acquisition function There are three main acquisition functions used in BO, i.e., Probability of Improvement (PI) [13], Expected Improvement (EI) [4], and GP Upper Confidence Bound (GP-UCB) [21]. PI can get stuck in local optima and under-explored globally [19], and GP-UCB needs extra effort to tune its own parameter. So, in this paper, we use EI as the acquisition function.

Optimization algorithm As shown in Line 5 of Alg. 1, BO finds the argmax of the acquisition function to locate the next sample point. In our case, BO inputs a random point (i.e., adjacency matrix) for an optimizing algorithm (Opt-Alg), e.g., Limited-memory Broyden-Fletcher-Goldfarb-Shanno algorithm (L-BFGS) [15]. Then, L-BFGS uses the random point as an initial point and obtains derivatives of the acquisition function to identify the direction of steepest descent. However, the result (a matrix) from L-BFGS might not satisfy our topology degree constraints.

Algorithm 2: Random points Algorithm

Input: C, d
Output: Adjacency Matrix (AM)
Build a circle among clusters
for r in range(C) **do**
    Randomly assign $d-1$ elements to 1 in r^{th} row
end for
Find a random permutation of list range$(1, C)$ called (L_{col})
for c in L_{col} **do**
    $n\_edge = 0$
    Find a random permutation of list range$(1, C)$ called (L_{row})
    for r in L_{row} **do**
        if $AM[r, c] == 1$ **then**
            if $n\_edge \leq d-1$ **then**
                $n\_edge \mathrel{+}= 1$
            else
                $AM[r, c] = 0$
            end if
        end if
    end for
end for

We use two steps to make the result from L-BFGS be a valid (i.e., connected and degree-constrained) topology. First, we build a circle among clusters. Then, we sort the elements of the result matrix from L-BFGS in descending order and build edges between nodes one by one without violating degree constraints. The pseudocode is shown in Alg. 3.

Algorithm 3: Conversion Algorithm

Input: Result/Matrix from L-BFGS (RM)
Output: Adjacency matrix (AM)
Build a circle among nodes in AM
Sort elements of RM in descending order (L_{sort})
for e in L_{sort} **do**
    if Egress degree of c_i and ingress degree of c_j
    are both less than d **then**
        Build an edge from c_i to c_j
    end if
end for

Stopping condition BO needs a stopping condition, e.g., based on time and/or iterations, and the best solution achieved in that period is taken as the final result. The more iterations, the better the final result, but as we seek a solution quickly, we use a stopping condition of ϵ sec, e.g., 1 sec. After ϵ sec, BO outputs the best achievable topology.

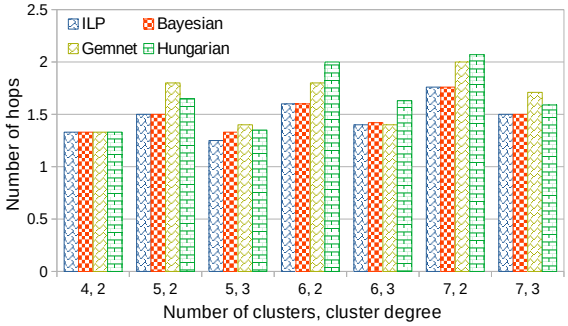


Fig. 3: Performance comparisons of Bayesian-based, Gemnet and Hungarian algorithms in cluster network.

4 Evaluation

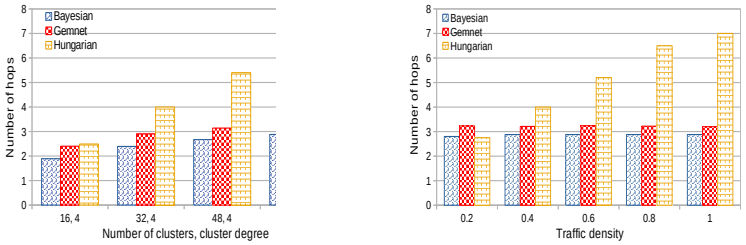
In this section, we conduct simulation results to compare the BO-based method with ILP, Hungarian-based method (HG), and Gemnet [10], in terms of the average number of hops. The ILP gives the optimal value, but it can only be obtained for small topologies in reasonable time. Gemnet provides a general method to build a topology with several desirable properties such as small average hop distance, diameter, etc. As opposed to ShuffleNet [9] and de Bruijn graph [18], Gemnet has no restriction on the number of nodes in its topology, and typically achieves smaller hop distances. Gemnet(M, K) arranges clusters in a cylinder of K columns and M clusters per column.

4.1 Cluster Topology

We first compare BO with ILP, Gemnet, and HG in terms of the average number of hops of the cluster topologies they produce. Flows are sent from each cluster to other clusters (uniform traffic), and the flow rate equals 1 Gbps. Figure 3 shows in terms of average number of hops, BO can achieve optimal results in some cases, and ILP outperforms BO by no more than 6% in the considered cases. Also, BO outperforms Gemnet and HG by up to 21% and 15%, respectively.

Then, we compare BO with Gemnet and HG with different traffic patterns in large networks and show comparison results in Figure 4. Figure 4(a) shows comparison results with different number of clusters under uniform traffic. Results show BO outperforms Gemnet and HG by up to 21% and 58%, respectively. Clearly, HG is not a good choice to configure topology under uniform traffic. The reason is that HG aims to find a matching to maximize single-hop traffic, and under uniform traffic, HG might perform the same as a random topology.

Figure 4(b) shows comparison results under different traffic densities in a network of 64 clusters with a cluster degree equal to 4. We define traffic density as the probability of a flow existing between clusters. The flow rate is randomly



(a) (b)

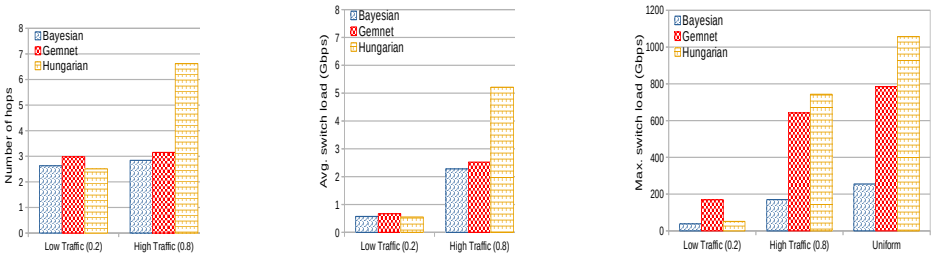
Fig. 4: Comparisons of Bayesian-based, Gemnet and Hungarian algorithms in cluster networks: (a) Comparison with different number of clusters and cluster degrees. (b) Comparison with different traffic densities.

chosen between 1 and 100 Gbps. Results show BO outperforms Gemnet and HG by up to 13% and 58%, respectively. Also, results show that HG outperforms Gemnet in a network with low traffic density. Under low traffic density, building direct links between clusters with large traffic benefits more in terms of the average number of hops.

4.2 RHODA

We now evaluate the various algorithms based on how the generated inter-cluster topologies perform in the entire RHODA architecture. Recall that the overall hop distance in RHODA is based on both the intra- and inter-cluster hop distances, and we have only tried to optimize the inter-cluster topology in this paper. We assume the following numbers for the DCN: the number of clusters is 64, and each cluster has 64 racks ($k = 64$). The number of wavelengths on a fiber is 128, the bandwidth of a wavelength is 100 Gbps, and both T^{intra} and T^{inter} are equal to 2 (i.e., a total of 4 transceivers per rack). The degree of WSS is 4 (i.e., $d = 4$). We consider three traffic patterns, i.e., uniform, low density traffic (LT), and high density traffic (HT). We set the rack flow rate to be $1/k^2$ Gbps (the total flow rate from a cluster to another cluster equals 1 Gbps). Under uniform traffic pattern, a rack sends a flow to each of the other racks. Under LT and HT, the probability that a cluster sends flows to another cluster is 0.2 and 0.8, respectively. Given cluster c_i sends flows to cluster c_j, each rack in c_i sends a flow to each rack in c_j.

In Figure 5, we show comparison results of BO, Gemnet, and HG in RHODA with different metrics, i.e., the average number of hops, the average switch load, and the maximum switch load. Results show that RHODA outperforms Gemnet by up to 11%, 11%, and 77% in terms of average hop distance, average switch load, and maximum switch load, respectively. Also, RHODA outperforms HG by up to 58%, 58%, and 77% in terms of average hop distance, average switch load, and maximum switch load, respectively. Thus, the topology configured by BO not only achieves small hop distances, but also balances the traffic well among switches.



(a) (b) (c)

Fig. 5: Comparisons of Bayesian-based, Gemnet and Hungarian algorithms in RHODA: (a) Comparison in terms of number of hops. (b) Comparison in terms of average switch load. (c) Comparison in terms of maximum switch load.

5 Conclusion

A reconfigurable hierarchical optical DCN architecture (RHODA) was introduced in an earlier paper. RHODA groups racks into clusters and enables both clusters and the inter-cluster topology to be configurable. In this paper, we focus on optimizing the inter-cluster topology in terms of the weighted hop distance, and present an approach based on Bayesian Optimization (BO). We compare BO with ILP, Gemnet, and the Hungarian-based method. Results show that BO can achieve optimal or near-optimal results within a small amount of time, and it outperforms Gemnet and Hungarian-based method by up to 13% and 58%, respectively.

References

1. Al-Fares, M., Loukissas, A., Vahdat, A.: A scalable, commodity data center network architecture. In: ACM SIGCOMM Computer Communication Review. vol. 38, pp. 63–74. ACM (2008)
2. Alipourfard, O., Liu, H.H., Chen, J., Venkataraman, S., Yu, M., Zhang, M.: Cherrypick: Adaptively unearthing the best cloud configurations for big data analytics. In: NSDI. vol. 2, pp. 4–2 (2017)
3. Bergstra, J., Bengio, Y.: Random search for hyper-parameter optimization. Journal of Machine Learning Research **13**(Feb), 281–305 (2012)
4. Brochu, E., Cora, V.M., De Freitas, N.: A tutorial on bayesian optimization of expensive cost functions, with application to active user modeling and hierarchical reinforcement learning. arXiv preprint arXiv:1012.2599 (2010)
5. Csernai, M., Ciucu, F., Braun, R.P., Gulyás, A.: Towards 48-fold cabling complexity reduction in large flattened butterfly networks. In: 2015 IEEE Conference on Computer Communications (INFOCOM). pp. 109–117. IEEE (2015)
6. Dally, W.J., Towles, B.P.: Principles and practices of interconnection networks. Elsevier (2004)
7. Greenberg, A., Hamilton, J.R., Jain, N., Kandula, S., Kim, C., Lahiri, P., Maltz, D.A., Patel, P., Sengupta, S.: VL2: a scalable and flexible data center network.

In: ACM SIGCOMM computer communication review. vol. 39, pp. 51–62. ACM (2009)
8. Guo, C., Lu, G., Li, D., Wu, H., Zhang, X., Shi, Y., Tian, C., Zhang, Y., Lu, S.: BCube: a high performance, server-centric network architecture for modular data centers. ACM SIGCOMM Computer Communication Review **39**(4), 63–74 (2009)
9. Hluchyj, M.G., Karol, M.J.: Shuffle net: An application of generalized perfect shuffles to multihop lightwave networks. Journal of Lightwave Technology **9**(10), 1386–1397 (1991)
10. Iness, J., Banerjee, S., Mukherjee, B.: Gemnet: A generalized, shuffle-exchange-based, regular, scalable, modular, multihop, wdm lightwave network. IEEE/ACM Transactions on Networking (TON) **3**(4), 470–476 (1995)
11. Kandula, S., Padhye, J., Bahl, P.: Flyways to de-congest data center networks (2009)
12. Kuhn, H.W.: The hungarian method for the assignment problem. Naval research logistics quarterly **2**(1-2), 83–97 (1955)
13. Kushner, H.J.: A new method of locating the maximum point of an arbitrary multipeak curve in the presence of noise. Journal of Basic Engineering **86**(1), 97–106 (1964)
14. Mockus, J.: Bayesian heuristic approach to global optimization and examples. Journal of Global Optimization **22**(1-4), 191–203 (2002)
15. Mokhtari, A., Ribeiro, A.: Global convergence of online limited memory bfgs. The Journal of Machine Learning Research **16**(1), 3151–3181 (2015)
16. Pierre Delforge: Available online: "`http://www.nrdc.org/energy/data-center-efficiency-assessment.asp`" (2015)
17. Roy, A., Zeng, H., Bagga, J., Porter, G., Snoeren, A.C.: Inside the social network's (datacenter) network. In: ACM SIGCOMM Computer Communication Review. vol. 45, pp. 123–137. ACM (2015)
18. Sivarajan, K., Ramaswami, R.: Multihop lightwave networks based on de bruijn graphs. In: IEEE INFCOM'91. The conference on Computer Communications. Tenth Annual Joint Comference of the IEEE Computer and Communications Societies Proceedings. pp. 1001–1011. IEEE (1991)
19. Snoek, J., Larochelle, H., Adams, R.P.: Practical bayesian optimization of machine learning algorithms. In: Advances in neural information processing systems. pp. 2951–2959 (2012)
20. Some interesting bits about latency: Available online: "`https://www.citycloud.com/city-cloud/some-interesting-bits-about-latency`"
21. Srinivas, N., Krause, A., Kakade, S.M., Seeger, M.: Gaussian process optimization in the bandit setting: No regret and experimental design. arXiv preprint arXiv:0912.3995 (2009)
22. Wang, L., Wang, X., Tornatore, M., Kim, K.J., Kim, S.M., Kim, D.U., Han, K.E., Mukherjee, B.: Scheduling with machine-learning-based flow detection for packet-switched optical data center networks. Journal of Optical Communications and Networking **10**(4), 365–375 (2018)
23. Who Has the Most Web Servers: Available online: "`http://www.datacenterknowledge.com/archives/2009/05/14/whos-got-the-most-web-servers`"
24. Xu, M., Diakonikolas, J., Modiano, E., Subramaniam, S.: A hierarchical wdm-based scalable data center network architecture. In: Communications (ICC), 2019 IEEE International Conference on. IEEE (2019 Available in arXiv preprint arXiv:190106450)